Furniture board segmentation based on multiple sensors and integer programming model*

GUO Jian-shuang (郭建双)**, **ZHANG Jian-hua** (张剑华), **ZHANG Shao-bo** (张少波), **CHEN Jia-wei** (陈嘉伟), and **LIU Sheng** (刘盛)

College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China

(Received 13 September 2018; Revised 22 October 2018)

In the current furniture production line, the level of automation in the stage of loading and unloading is not high enough. In order to improve its automation, a novel integer programming based method for automatically segmenting board is proposed and a multi-sensor configuration is given. In such a configuration, we include multiple cameras and Lidar sensors. The cameras attached on each board are used to collect quick response (QR) code information, while Lidar sensors can obtain each board's contours information. We then formulate each board's segmentation as the integer programming problem. The experimental results show that our method can achieve a very high segmentation accuracy of 95% on average.

Document code: A **Article ID:** 1673-1905(2019)02-0151-5

DOI https://doi.org/10.1007/s11801-019-8147-9

In the current furniture production line, the boards loading and unloading is the most difficult one to realize automation. Although a structure for automatic loading and unloading of furniture boards was completed based on a visual sensor[1], it didn't implement the visual module core algorithm nor a board segmentation method and can't be utilized practically. A monocular vision bar loading and unloading system was designed and built up[2-4]. In fact, most of the boards loading and unloading systems are manually transported. Therefore, segmentation algorithms for the furniture boards are very rare.

Considering the actual process of the boards loading and unloading, a quick response (QR) code is attached on each board, containing information about its length and width. The QR code is randomly placed on the surface of each board. The color and texture of the boards are almost identical and the boards can be placed and stacked at will. Due to the similarity between the boards color and texture, traditional segmentation methods[5-9] can't be used in our case. Refs.[10-12] are not applicable to this case as well.

Based on a multi-sensor configuration, the main purpose of this paper is to study how to segment and locate boards accurately by using the contours information of each board when multiple boards are stacked randomly. The main sensors used in this study include multiple Lidar sensors and several common color cameras. We use the cameras to collect the data from the QR code attached on each board. The method in Ref.[13] is used to detect the QR codes. And the libdmtx library[14] is used to read the content of the QR codes, where the length and width information of the corresponding board is contained. Then, several Lidar sensors are used to obtain contours information of different edges from the boards. The location information of the QR codes, the length, width and contours information of the boards are input into an integer programming model[15-17]. Thus we can treat the segmentation problem as a board arrangement problem, where some constraints must be fulfilled. In this sense, it's just like placing an appropriate number of the boards with appropriate sizes into a constraint space, which can naturally be solved by integer programming. Finally, the board segmentation is carried out by solving our integer programming model. Experimental results show that our method can accurately segment arbitrary stacked boards. Consequently, the proposed method can provide support for boards loading and unloading.

In our segmentation process, the number of the boards, the size of each board and the outer contours of the boards are obtained in advance. The goal is to segment and locate each board based on the above information. Our segmentation method steps are as follows. With the existing edge contours information as a constraint, our method iteratively searches for a certain number and size of the boards arrangements. Our method is completed when it finds a solution that satisfies the above conditions.

The design of our boards segmentation based on integer programming model is as follows. First, the model describes the input information of the QR codes set as:

$$\boldsymbol{Q}_M=\{\boldsymbol{q}_1,\ldots,\boldsymbol{q}_i,\ldots,\boldsymbol{q}_M\}. \quad (1)$$

* This work has been supported by the National Natural Science Foundation of China (No.61876167).

** E-mail:534951043@qq.com

The set of $\boldsymbol{q}_i$ is described as:

$$\boldsymbol{q}_i=\{p_i,\ w_i,\ h_i\}, \tag{2}$$

where p_i denotes position information of the i-th QR code, w_i and h_i denote the length and width information of the boards contained in the i-th QR code, and M is the number of the QR codes. The contours information containing the lines set is described as:

$$\boldsymbol{L}_N=\{l_1,\ldots,l_j,\ldots,l_N\}, \tag{3}$$

where N is the size of the lines set. According to the nearest distance from the line to the QR code, the lines set is clustered by QR codes. We describe lines set as:

$$\boldsymbol{L}_{\text{nearest}M}=\{\boldsymbol{L}_{\text{nearest}1},\ldots,\boldsymbol{L}_{\text{nearest}i},\ldots,\boldsymbol{L}_{\text{nearest}M}\}, \tag{4}$$

$$\boldsymbol{L}_{\text{nearest}i}=\{l_0,\ldots,l_k\},\ k\in[0,4]. \tag{5}$$

The segmentation result of the algorithm is described as:

$$\boldsymbol{R}_M=\{\boldsymbol{r}_1,\ldots,\boldsymbol{r}_i,\ldots,\boldsymbol{r}_M\}, \tag{6}$$

$$\boldsymbol{r}_i=\{\boldsymbol{n}_{i1},\ \boldsymbol{n}_{i2},\ p_i\}, \tag{7}$$

where $\boldsymbol{n}_{i1}$ and $\boldsymbol{n}_{i2}$ respectively represent vectors in both directions of $\boldsymbol{r}_i$. And p_i denotes the intersection of $\boldsymbol{n}_{i1}$ and $\boldsymbol{n}_{i2}$. It is also the common starting point of the vectors $\boldsymbol{n}_{i1}$ and $\boldsymbol{n}_{i2}$. The two vectors $\boldsymbol{n}_{i1}$ and $\boldsymbol{n}_{i2}$ are perpendicular to each other, and $\boldsymbol{r}_i$ describes a rectangle.

The model's objective function based on integer programming is designed as follows:

$$E=\sum_{i=1}^{M}\{S(\lambda*(r_{in1},r_{in2}\in \boldsymbol{L}_{\text{nearest}i})+(1-\lambda)*r_{in1},r_{in2}\in\boldsymbol{L}'_{\text{nearest}i}))-q_{iw}*q_{ih}\},\quad (\lambda=1,\ 0). \tag{8}$$

Constraints are as follows:

$$\sum_{i=1}^{M}\text{Area}(\boldsymbol{r}_i)\cap\sum_{j=1}^{M}\text{Area}(\boldsymbol{r}_j)=0\ , \tag{9}$$

$$\text{Area}(\boldsymbol{r}_i)\subset\text{Area}((x-q_{ipx})^2+(y-q_{ipy})^2=D_i)\ ,$$

$$D_i=q^2_{iw}+q^2_{ih},\quad (\forall i\in M, x\in Z, y\in Z). \tag{10}$$

In Eq.(8), when the k value of $\boldsymbol{L}_{\text{nearest}i}$ is greater than or equal to 2, λ is equal to 1 otherwise 0. If λ is 0, $\boldsymbol{L}_{\text{nearest}i}$ clusters the line sets again based on the nearest distance from lines to the QR codes. The clustered result is regarded as $\boldsymbol{L}'_{\text{nearest}i}$. In the process of clustering again, the predicted result is then taken as input. S is the area calculated for the set $\boldsymbol{r}_i$. In Eq.(9), the covered areas of any two elements $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ in $\boldsymbol{R}_M$ do not coincide. The area represents the region where $\boldsymbol{R}_M$ is located. In Eq.(10), the region of any element $\boldsymbol{r}_i$ in $\boldsymbol{R}_M$ is limited to the region where q_{ip} is the center and D_i is the diameter.

Based on the two constraints in Eqs.(9) and (10), the problem is solved by running the objective function iteratively. When a set of qualifying optimal solutions is found, the task of integer programming is considered as finish and the segmentation result is obtained.

Our multi-sensor system mainly includes ordinary cameras and single-line Lidar sensors, as shown in Fig.1. In order to combine the information, it's necessary to uniformly calibrate the coordinate system between different sensors and robots.

The detection of the QR codes is based on the SSD[13] model under the framework in Ref.[18]. The detection effect is shown in Fig.2(a). Because the difference between QR codes and boards is relatively large, the trained SSD model can basically detect QR codes with an accuracy of 99%. In addition, QR codes are identified by the libdmtx library[14]. The recognition effect is shown in Fig.2(b). By adjusting the ambient light, the recognition rate of the QR codes exceeds 99%. Therefore, the above recognition efficiency provides a guarantee for the accuracy of the method.

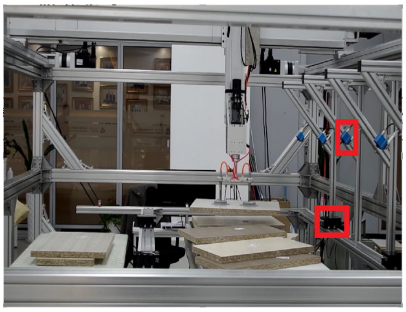


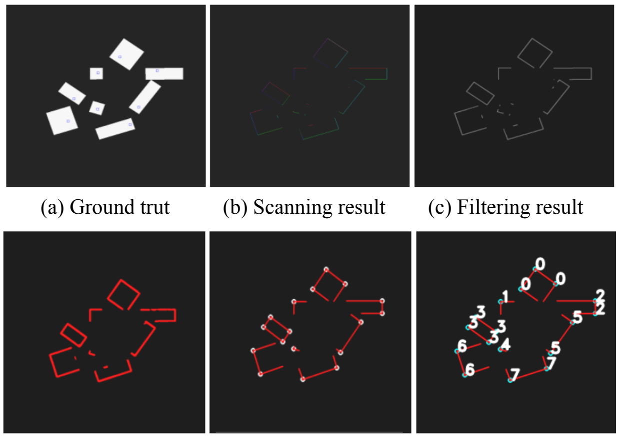
Fig.1 Working environment (Black boxes are Lidar sensors, blue cubes are cameras and the entire outer frame is a robot.)

Fig.2 (a) QR code detection and (b) QR code recognition

In the method, the detected QR codes need to be decoded. And the current position information is obtained from the QR codes according to the detection result. The length and width information of the boards is extracted from the decoded information.

The contours data of the boards was acquired using multiple Lidar sensors. The distance information measured by each Lidar sensor is converted into a unified coordinate system. The result of the conversion is shown in Fig.3(b). Fig.3(c) is generated by threshold processing of Fig.3(b). Fig.3(c) is detected by the Hough line detection method. The result of the lines set is shown in Fig.3(d). Then the set of the lines is fitted with respect to their length and position. We detect and extract right angles information from the lines set. The effect of the right angles detection is shown in Fig.3(e).

In fact, multiple right angles of the same board must have a common right angle edge. The method clusters right angles based on the common right angle sides. After the clustering is completed, the contours of the right angle set of the cluster are extracted. The clustered sets are sorted according to the contours centroid position. The obtained contours information is shown in Fig.3(f).



(a) Ground trut (b) Scanning result (c) Filtering result

(d) Lines set (e) Right angle sets (f) Contours information

Fig.3 Lidar data acquisition and processing

The segmentation method takes two main inputs. The first part is the information of the QR codes. The second part is the contours information of the boards. Combining them together, the segmentation method based on the above integer programming model is designed. Our segmentation method consists of the following steps.

Step 1: Sort the QR codes based on the position information. Based on the shortest Euclidean distance, the contours set is clustered by QR codes. $\boldsymbol{L}_{\mathrm{nearest}M}$ is the result of the clustered set and k is the size of $\boldsymbol{L}_{\mathrm{nearest}M}$.

Step 2: In the clustered set $\boldsymbol{L}_{\mathrm{nearest}M}$, each $\boldsymbol{L}_{\mathrm{nearest}i}$ is ordered based on the value of k. Each l_i of $\boldsymbol{L}_{\mathrm{nearest}i}$ is a candidate for $\boldsymbol{r}_i$ to predict ground truth. Therefore, the larger the value of k, the more accurate the prediction of $\boldsymbol{r}_i$. When the value is greater than or equal to 3, the predicted result of $\boldsymbol{r}_i$ is regarded as the ground truth. Therefore, the method prioritizes the $\boldsymbol{L}_{\mathrm{nearest}i}$ with a k value greater than or equal to 3.

Step 3: For $\boldsymbol{L}_{\mathrm{nearest}i}$ with the k equal to 2, if the QR code information $q_{iw}=q_{ih}$, the predicted result $\boldsymbol{r}_i$ is considered to be a ground truth. The remaining $\boldsymbol{L}_{\mathrm{nearest}i}$ in which k is less than or equal to 2 are clustered again into $\boldsymbol{L}'_{\mathrm{nearest}M}$. The condition of clustering is to remove the QR code information that has been predicted to be the ground truth $\boldsymbol{r}_i$ but to add its predicted lines information into the lines set.

Step 4: $\boldsymbol{L}'_{\mathrm{nearest}M}$ will replace $\boldsymbol{L}_{\mathrm{nearest}M}$ and repeat Step 2 and Step 3 until $\boldsymbol{L}'_{\mathrm{nearest}M}$ is no longer updated. If the prediction has been completed at this time, the result of the prediction is completed.

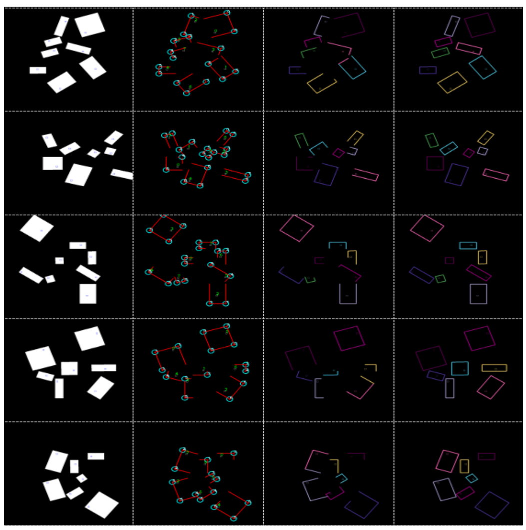
Step 5: If there is $\boldsymbol{r}_i$ that does not complete the prediction, it is assumed that a certain subset of the $\boldsymbol{L}'_{\mathrm{nearest}i}$ set is $\boldsymbol{r}'_i$. We determine whether $\boldsymbol{r}'_i$ meets the model's constraints. If all constraints are met, $\boldsymbol{r}'_i$ is considered to be a solution that conforms to the ground truth. If there is a situation that does not meet the constraint condition, $\boldsymbol{r}'_i$ is considered unacceptable as a solution and discarded. The subset is selected in $\boldsymbol{L}'_{\mathrm{nearest}i}$ and assumed as our new solution. Iterate this way until the algorithm finds a solution where all assumptions are consistent with the constraints.

In our study, the experiment is divided into two parts: simulation data experiment and real data experiment. Firstly, more data obtained through simulation experiments are used to test the reliability of the method. Secondly, the experiment in the real environment combined with hardware equipment is conducted to test the segmentation effect.

In the simulation, each board is replaced with a rectangle rotated at a random angle. The program is used to simulate the Lidar sensors to obtain information of the contours and the cameras to obtain information of the QR codes. To simulate field acquisition data, add some random noise to increase the complexity of the data. Finally, a certain number of the boards at random positions are generated to test our algorithm.

Ground truth is shown in Fig.4(a). The information of the contours and the QR codes are processed as shown in Fig.4(b). The nearest straight lines of the QR codes are clustered as shown in Fig.4(c). Iterative planning is then performed, as shown in Fig.4(d).

In the real experiment, our method uses five Lidar sensors and three cameras to acquire data. Due to the large working area of the boards loading and unloading, the pictures collected from multiple cameras are spliced. The stitched results are shown in Fig.5. Results for three different segmentation methods are shown in Fig.6. According to the data in Tab.1, we can draw conclusions that when the color and texture of the boards are almost identical, the segmentation results of SLIC[5] and BSR[8] are not ideal. Obviously, the results of our segmentation are better. Fig.7(b) shows the information processing results of the contours and the QR codes. Based on the above conditions, the result of the QR codes clustered by the closest distance to the lines set is shown in Fig.7(c). The method steps are then iterated until the final segmentation result is obtained as shown in Fig.7(d).



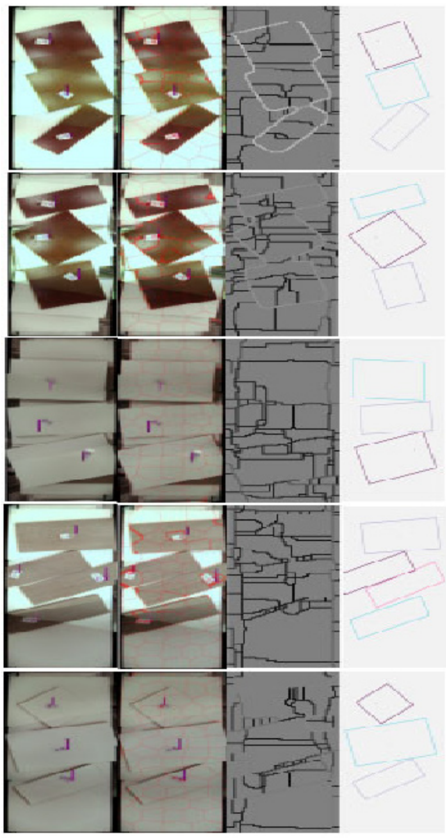
(a) Ground truth (b) Input (c) Clusters (d) Segmentation

Fig.4 Simulation results

(a) (b) (c) (d)

Fig.5 (a, b, c) Collected images from different cameras; (d) Stitched results

In our real experiment, the collected data are a little few due to the limitation of the working environment and actual conditions. Because of the stacking of the boards, we need to solve the problem that how to layer them. We control the Lidar sensors to scan the contours of the top boards. When the robot completes the grasping of the highest layer boards, the lifting machine automatically lifts the next layer boards to the scanning height. In this way, the layering of the boards is realized. The number of the boards per layer can only be limited to four or so. In the simulation, the number of the QR codes β is set to test the method. The β value is from 1 to 8. In the experiment, different β values are simulated and tested for 1 000 times.

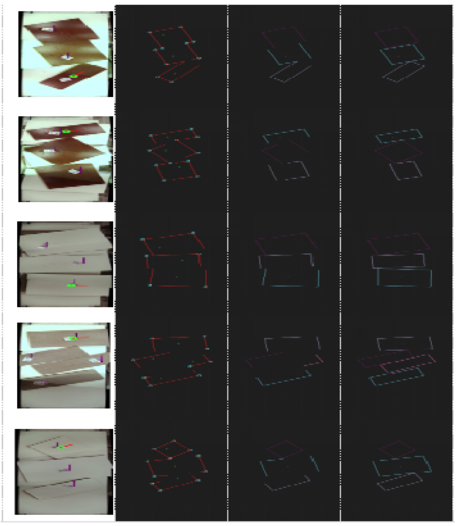


(a)Ground truth (b) SLIC[5] (c) BSR[8] (d) Ours

Fig.6 Segmentation results of three algorithms

Tab.1 Accuracy of three algorithms

Algorithm	SLIC	BSR	Ours
Accuracy	30%	35%	95%



(a) Ground truth (b) Input (c) Clusters (d) Segmentation

Fig.7 Experimental results

We treat the experimental results in four cases. In the first one, all segmentation results are exactly accurate. In the second one, there is one failure segment. In the third one, two failure segments are resulted. And in the last case, there are three or more failure segments. The proportions of their experiment times to the total number of experiments are expressed as t, t_1, t_2 and r, respectively. The st represents the proportion of all successfully predicted samples to all the experimental samples. The t shows the proportion of correct cases in all segmentation cases. As shown in Fig.8, when the number of the boards is less than 6, the accuracy of the segmentation method is high enough.

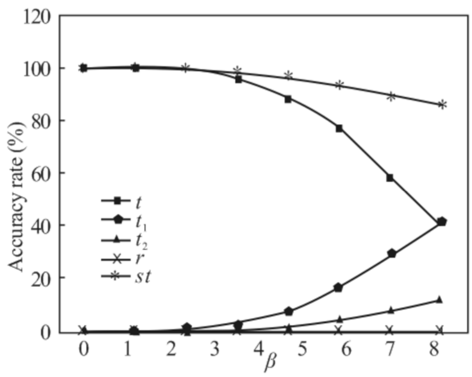


Fig.8 Simulation data analysis diagram

It can be seen from Tab.2 that when the number of the boards is less than or equal to 6, the method can predict the ground truth better and its average accuracy in this case can reach 98%. When the number of the boards is more than 6, there is a problem that boards are severely blocked. Even so, the average accuracy in this case can reach 88%. The segmentation performance of the method is relatively stable and the average accuracy is more than 95%.

Tab.2 Simulated segmentation results

β	8	7	6	5
St (accuracy rate)	0.858	0.893	0.936	0.963
β	4	3	2	1
St (accuracy rate)	0.985	0.997	0.999	1

In practical applications, the effect of prediction is more accurate as the number of boards is gradually reduced. At present, the segmentation accuracy of this method may not reach 100%. In order to ensure normal loading and unloading in practical applications, it is verified whether the prediction is completely successful. If the prediction is completely successful, the robot will grab them in turn. Otherwise, the robot grabs one of the accurately predicted boards. Then, we can use this method again after grabbing. Follow the above steps to grab in sequence until all the boards are grabbed.

This paper studies and explores boards segmentation technology. A segmentation method based on multiple sensors and integer programming models is proposed. This method enables boards segmentation in complex situations. Both theoretical analysis and experiments show that the segmentation method has a high success rate. It has certain application value in the automatic loading and unloading industry of furniture boards. Next step is to explore the optimization of the algorithm and to improve the robustness of the algorithm combined with other similar industries.

References

[1] L Ping, L. I., Chen Q. Y., Duan X. Y., Chen H., Deng Z. W. and Liao F., Mechanical & Electrical Engineering Technology **45**, 4 (2016). (in Chinese)

[2] Gao-De WU, Jun-Hua C., Jian-Min Z., Jian-Qiang MA, Nian-Fu L. and Fu Z., Transducer & Microsystem Technologies **36**, 157 (2017). (in Chinese)

[3] Bin LI and B. Luo, Transducer & Microsystem Technologies **35**, 150 (2016). (in Chinese)

[4] Lima J., F Simões, Uchiyama H., Teichrieb V. and Marchand E., Machine Vision & Applications **27**, 1 (2016).

[5] Achanta R., Shaji A., Smith K., Lucchi A., Fua P. and Süsstrunk S., SLIC superpixels, Switzerland: EPFL, 2010.

[6] Roy A. and Todorovic S., A Multi-scale CNN for Affordance Segmentation in RGB Images, European Conference on Computer Vision, Springer International Publishing,186 (2016).

[7] Wang J., Wang Z., Tao D., See S. and Wang G., Learning Common and Specific Features for RGB-D Semantic Segmentation with Deconvolutional Networks, European Conference on Computer Vision, Springer International Publishing, 664 (2016).

[8] Arbelaez Pablo, Maire Michael, Fowlkes Charless and Malik Jitendra, IEEE Transactions on Pattern Analysis and Machine Intelligence **33**, 898 (2011).

[9] Xiang Fang and Wang Hongfu, Computer Simulation **28**, 280 (2011). (in Chinese)

[10] WANG Jing, ZHENG Yong-guo, PAN Zhen-kuan, ZHANG Wei-zhong and WANG Guo-dong, Optoelectronics Letters **11**, 307 (2015).

[11] GENG Hui-qiang, ZHANG Hua, XUE Yan-bing, ZHOU Mian, XU Guang-ping and GAO Zan, Optoelectronics Letters **13**, 381 (2017).

[12] WANG Xiao-peng, ZHANG Wen and CUI Ying, Optoelectronics Letters **11**, 395 (2015).

[13] Liu W., Anguelov D., Erhan D., Szegedy C., Reed S. and Fu C.Y., SSD: Single Shot MultiBox Detector, 21 (2015).

[14] http://libdmtx.wikidot.com/general-instructions.

[15] Alves C., Clautiaux F., Carvalho J.V.D. and Rietz J., Linear and Integer Programming, Dual-Feasible Functions for Integer Programming and Combinatorial Optimization, Springer International Publishing, 1065 (2016).

[16] Basu A., Conforti M., Summa M.D. and Paat J., The Structure of the Infinite Models in Integer Programming, International Conference on Integer Programming and Combinatorial Optimization, Springer, Cham, 63 (2017).

[17] Sun Xiaoling and Li Duan, Integer Programming, Science Press, 2011. (in Chinese)

[18] Jia Y., Shelhamer E., Donahue J., Karayev S., Long J. and Girshick R., Caffe: Convolutional Architecture for Fast Feature Embedding, 675 (2014).