Sensors and Materials, Vol. 27, No. 8 (2015) 755–761
MYU Tokyo

S & M 1112

Using Laser Range Finder and Multitarget Tracking-Learning-Detection Algorithm for Intelligent Mobile Robot

Jr-Hung Guo* and Kuo-Lan Su[1]

Graduate School of Engineering Science and Technology,
National Yunlin University of Science & Technology,
123 University Road, Section 3, Douliou, Yunlin 64002, Taiwan R.O.C.
[1]Department of Electrical Engineering, National Yunlin University of Science & Technology,
123 University Road, Section 3, Douliou, Yunlin 64002, Taiwan R.O.C.

(Received July 2, 2014; accepted February 10, 2015)

Key words: laser ranger finder, multitarget tracking, tracking-learning-detection (TLD), intelligent mobile robot

Self-positioning and obstacle avoidance for an intelligent mobile robot plays an important role in the technologies of the mobile robot. Because of its capabilities of fast response and accurate measurement, the laser range finder (LRF) is a relatively good choice as a sensor for this purpose. However, there are some drawbacks of the LRF. One is that the LRF usually senses in the horizontal plane, and thus the obstacle may not be detected if it is not in the detection plane. Obstacles with high reflectivity will also cause the LRFs to have incorrect detections. Both will lead the mobile robot to a dangerous situation. In this paper, a modified tracking-learning-detection (TLD) image recognition system, which can detect multiple targets simultaneously, is used to assist LRF in positioning and obstacle avoidance.

1. Introduction

Self-positioning and obstacle avoidance for an intelligent mobile robot has been an important topic of mobile robot technology for a long time. In the studies of Guo *et al.*,(1–3) ultrasonic sensors, infrared sensors, and laser range finders (LRFs) were used for positioning and obstacle avoidance. These sensors are currently commonly used for obstacle avoidance and positioning for intelligent mobile robots. The LRF is the most popular one among these because of its characteristics of fast response and high precision.

However, there are some drawbacks of the LRF. One is that the LRF usually senses in the horizontal plane, thus the obstacle may not be detected if it is not in the detection plane. Furthermore, for some obstacles with a high reflective surface or coating, the laser beam will be reflected more than usual, and this will cause incorrect detection by

*Corresponding author: e-mail: g9710801@yuntech.edu.tw the LRF. Both will lead the mobile robot to a dangerous situation. To overcome the shortcomings of the laser range finder and for a more accurate detection, one possible way is using multiple LRFs to scan in *x*-, *y*-, and *z*-directions. Another possible way is making the LRF rotatable using a mechanical structure to obtain 3D information, *e.g.*, Google's Street View Car.(4)

The control and stability in the moving direction and posture of an intelligent mobile robot, and also the vibration generated during movement, are also very important for the intelligent mobile robot. A general way for these jobs is by adding an electronic compass, a gyroscope, a vibration detection sensor or the like. However, they will cause the structure of the robot to change and the cost to increase. In this paper, we propose a new method of detecting the moving direction, the posture, and the vibration situation by tracking multiple targets simultaneously using image tracking technology. The trajectories of all tracked objects will be calculated and compared with the moving trajectory of the intelligent mobile robot, then the current moving direction, the posture, and the vibration state of the intelligent mobile robot can be determined. This can not only assist the LRF in obstacle avoidance and self-positioning, but also reduce the number of sensors. The image information can also be used to determine the sizes of the tracked obstacles. Hence, the intelligent mobile robot will be more intelligent.

The image-tracking algorithm, tracking-learning-detection (TLD),(5) used in this work is an algorithm for long-term tracking of unknown objects developed by Zdenek Kalal, Surrey University. It combines the traditional tracking algorithm and detection algorithm to solve the deformation and occlusion problems during the tracking process. It constantly updates the significant feature points of the tracking template and the target model and related parameters of the detection module by learning mechanisms, allowing a more stable and reliable tracking performance. The TLD tracking architecture is shown in Fig. 1.

In this paper, the single-target tracking is extended to multitarget tracking based on the TLD algorithm. The trajectory of the tracked object will be recorded to provide the information of the motion state, as well as the direction deviation or posture change of

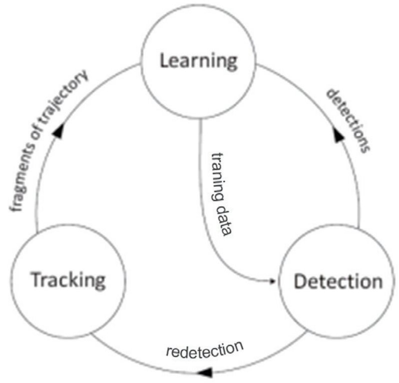


Fig. 1. TLD tracking architecture.(8)

the intelligent mobile robot. From the pixels of the image and the focal length of the camera, the size of the target and the distance to the target can be determined and used as auxiliary information for robot positioning. The above results can assist LRF for preventing detection errors and misjudgment. Using the trajectory vector of the tracked targets, the intelligent mobile robot can adjust its moving direction and posture.

2. Algorithms

2.1 *TLD algorithm*

The TLD Tracker uses the Lucas-Kanada[6,7] optical flow method, which is a frame-to-frame tracking method, for target motion estimation. Because the accuracy of the optical flow tracking is poor, the tracking may fail and the detector will start to detect. The detector records the presented position and appearance information; such recorded information is called learning. When the target is lost, it repositions with the recorded learning information of the target.

PN learning[8,9] is very important in the tracking algorithm of the TLD method. In PN learning, P is Positive Constraint, also known as P-expert or growing event, and N is Negative Constraint, also known as N-expert or pruning event.

PN learning consists of four parts, as follows.

(1) A classifier for learning
(2) Training sample set: some labeled samples
(3) Supervised learning: training method for classifier from the training sample set
(4) PN experts: generating positive and negative samples in the learning process

In the detector, the random forest method is used to update and forecast the target detection classifier in real time. In the TLD algorithm, it uses 2-bit binary patterns (2bitBPs) for feature detection. These features measure gradient orientation within a certain area. Moreover, these features will be quantized and coded for output. The detected features form a tree and finally a forest for learning and identification. 2bitBPs are similar to a harr-type feature and include the type and value of the feature. The random forest method can reveal information about a specific area of the gradient direction and the value of that information into a coded output.

To judge if a patch model is the target of interest, the type of feature has to be known. Take a rectangle frame at (x, y) of the image patch. The tuple (x, y, *width*, *height*) is the respective type of feature. Divide the selected rectangle frame into left and right parts of equal size, and compare the average intensity (grayness) of the two parts. The result may be the left part being grayer than the right part, or vice versa. Similarly, divide the selected rectangle frame into top and down parts of equal size. It is possible that the top part is grayer than the bottom part, or vice versa. Hence, there are four possible cases in total, which can be described to use 2 bits.

We use the nFern type to express the object. Each Fern is actually a quad-tree generated by the 2bitBP feature detector. The number of layers of a Fern is equal to the number of types of feature. Determine each layer of the patch on the basis of the relative type of feature, and calculate the value of the corresponding type of feature. By using 2bitBP features, there are four possible cases. By repeating this process, it finally will fall into a leaf node. The whole process can be divided into two processes.

(1) Training process: Record the number of positive examples (*NP*) and the number of negative examples (*NN*) on each leaf node. The posterior probability of positive examples on the leaf node is *NP*/(*NP* + *NN*).

(2) Evaluation process: The testing patch will ultimately fall into a leaf node. Because the posterior probability of positive examples that fell into each leaf node has been recorded in the training process, the final output is a positive probability example of the patch.

The TLD algorithm uses the Lucas-Kanada optical flow method to perform image processing to the frame. Although it can track a single target, some problems will occur for multitarget tracking. We cut a frame into nine areas and use parallel processing to perform multitarget tracking. The tracking is still updated in frame. The architecture of TLD multitarget tracking is shown in Fig. 2. After a video frame is obtained, it is divided into nine areas. The nine separate areas are processed by the TLD method in parallel. Finally, we can obtain the moving trajectory of the tracked object in the nine areas.

Thus, with a slight change in the algorithm, we can perform multitarget tracking, and the trajectories of the tracked objects can be determined. If the actual moving direction of an intelligent mobile robot is consistent with the command, the trajectories of all tracked objects will be the same. Hence, it can be used as the motion vector of the intelligent mobile robot. If these trajectories are different from the control command, that may be caused by the errors of the motion system, or uneven ground, or problems with the mechanism. Therefore, it is possible to use such information as reference for the posture or the status of vibration for an intelligent mobile robot, without the need to use other sensors for detection.

2.2 *Object tracking vector analysis*

Since the multitarget TLD tracking algorithm can record the moving trajectories of the tracked objects, we can know the starting and end points of an object being tracked. The trajectory vector can be calculated as

$$\boldsymbol{AB} = (x_2 - x_1, y_2 - y_1), \tag{1}$$

$$r = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \tag{2}$$

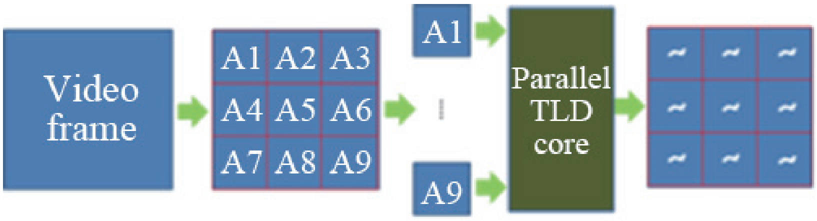


Fig. 2. (Color online) TLD multitarget tracking architecture.

where $(x_2 - x_1)$ is the component in the X-direction and $(y_2 - y_1)$ is the component in the Y-direction, i.e.,

$$X = (x_2 - x_1), \tag{3}$$

$$Y = (y_2 - y_1). \tag{4}$$

The angle of the vector is

$$\sin\theta = \frac{y}{r} \text{ or } \cos\theta = \frac{x}{r}. \tag{5}$$

The moving vector of the tracked object in the nine areas can be obtained using the above equations. The vectors in these nine areas should be the same under general conditions. For some cases of the fail tracking or tracking moving objects, it is necessary to isolate the wrong information and to classify the data in order to find the correct information. Here, we use the data classification approach. We compare the vectors area-by-area in the following manner:

$$\begin{cases} \left|\boldsymbol{v}_i - \boldsymbol{v}_j\right| \lesseqgtr \left|\boldsymbol{v}_i - \boldsymbol{v}_{j+1}\right|, \text{false} \\ \left|\boldsymbol{v}_i - \boldsymbol{v}_j\right| = \left|\boldsymbol{v}_i - \boldsymbol{v}_{j+1}\right|, \text{true} \end{cases} \tag{6}$$

where $\boldsymbol{v}_i$ is the reference vector, and $\boldsymbol{v}_j$ is the vector in another area. The evaluation function of the vectors in all areas is defined by

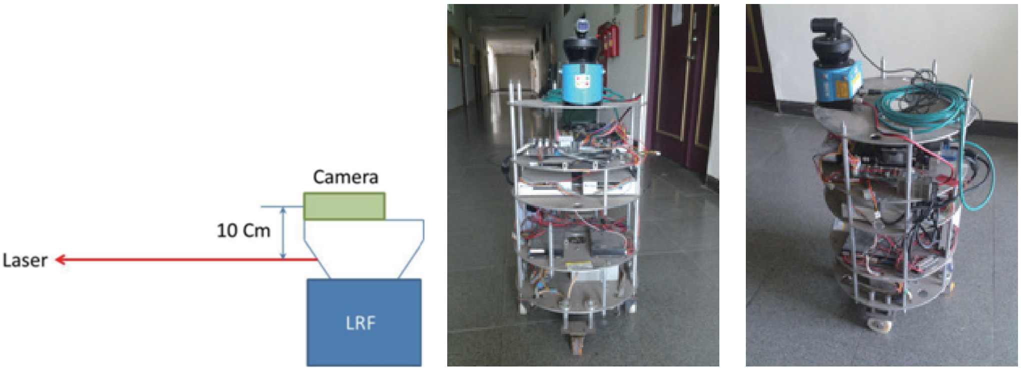
$$I = \sum_{i=1}^{K} f\left[\left|\boldsymbol{v}_i - \boldsymbol{v}_j\right| = \left|\boldsymbol{v}_i - \boldsymbol{v}_{j+1}\right|\right], j = 1, 2, \ldots, k. \tag{7}$$

The vectors in doubt can be eliminated using the above equation. Then, the motion vector of the tracked object can be determined by the majority vote method. Then, the data from the LRF can be modified using the obtained information.

3. Experimental Results

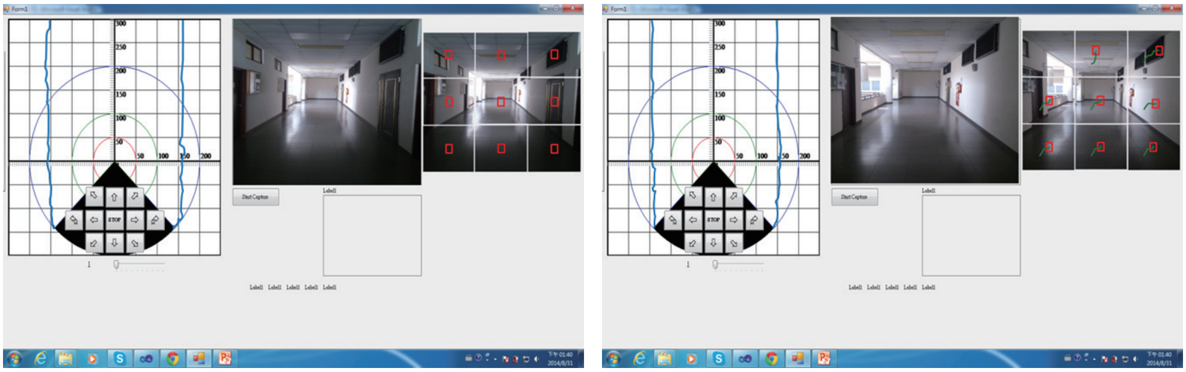
The laser range finder that we use is SICK's LMS-100. It is mounted on a two-wheel-driven mobile robot. We use a Microsoft LifeCam Cinema, which is a camera with a large aperture and a high resolution, for image capturing. This camera is mounted on top of the installed LRF, as shown in Fig. 3(a). Figures 3(b) and 3(c) show the front and side views of the mobile robot. The main processor is INTEL I7-2620 with 4 Gb of memory. The developing tools are EMGUCV 2.4 and Visual Studio 2010 (VB).

For the TLD image tracking, we cut the image area into nine subareas. After the system starts operating, it will begin tracking from the center of each subarea. Because the TLD recognition algorithm cannot identify the target very precisely, it will have a



(a) (b) (c)

Fig. 3. (Color online) Experimental platform: (a) camera and LRF, (b) robot front side, and (c) robot left side.



(a) (b)

Fig. 4. (Color online) Experimental results of the multitarget: (a) start to track and (b) tracking results.

tracking error or tracking failure if the object is moving fast or the image of the object has a large difference, as shown in Fig. 4. Figure 4(a) shows a screen shot when the system starts to track, and Fig. 4(b) shows the tracking status of each subarea after operation. In Fig. 4(b), we can see that the tracking cannot proceed in the upper left subarea. Moreover, there are different vectors in the right part of the area; hence, we need to use eqs. (6) and (7) to carry out the analysis of the trajectory vector of each subarea to find the correct trajectory vector.

When the correct motion vector is obtained, it can be used to adjust the posture of the intelligent mobile robot. If the correct motion vector cannot be obtained, the image tracking system has to retrack from the center of each subarea. Thus, the TLD system can provide the right information for the intelligent mobile robot.

4. Conclusions

We successfully built a platform for intelligent mobile robot self-positioning and tracking with a camera and a laser range finder. The TLD algorithm is an algorithm for tracking a target by learning the target's optical flow information. However, this algorithm can track only one target. Hence, we modified the TLD algorithm so that it can track nine targets simultaneously. By the technique of parallel processing, only 40% more computation time compared with that of the original single-target tracking is needed for tracking nine targets simultaneously. However, owing to the capability of precisely distinguishing the tracked objects, there is still a big step in the improvement of the rate of failure. In the future, we will use the Compute Unified Device Architecture (CUDA) to raise the overall efficiency. Moreover, we will continue to improve the relevant hardware and software for better image recognition, tracking efficiency, and more accurate self-positioning and tracking results.

References

1 J. H. Guo, K. L. Su, C. J. Wu and S. V. Shiau: Artif. Life Rob. **14** (2009) 257.

2 T. L. Chien, H. Guo, K. L. Su and S. V. Shiau: Proceedings of ICM2007 4th IEEE International Conference on Mechatronics (IEEE, 2007) p. 1.

3 K. L. Su: Proceedings of SMC'06, IEEE International Conference in Taipei (IEEE, 2006) Vol. 2, p. 966.

4 Street View Car: Home page of Google Street View Car, www.google.com/intl/en/maps/about/behind-the-scenes/streetview/ (accessed April 2014).

5 K. Zdenek, K. Mikolajczyk and J. Matas: IEEE Trans. **34** (2012) 1409.

6 B. D. Lucas: Generalized Image Matching by the Method of Differences (Carnegie Mellon University, Pittsburgh, 1995).

7 B. D. Lucas and T. Kanade: Proceedings of 7th Int. Joint Conf. on Artificial Intelligence (IJCAI) (DARPA, Vancouver, 1981) p. 674.

8 PN learning: Home page of ICIP2010 PN learning, person-al.ee.surrey.ac.uk/Personal/Z.Kalal/Publications/2010_icip_poster.pdf (accessed April 2014).

9 Z. Kalal, J. Matas and K. Mikolajczyk: Proceedings of Computer Vision and Pattern Recognition (CVPR) (IEEE, San Francisco, 2010) p. 49.