Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Motion Planning for Human-Robot Interaction Based on Stereo Vision and SIFT

Hong Liu
Key Laboratory of Machine Perception and Intelligence,
Key Laboratory of Integrated Microsystem,
Shenzhen Graduate School, Peking University, China
E-mail: hongliu@pku.edu.cn

Jie Zhou
Key Laboratory of Integrated Microsystem,
Key Laboratory of Machine Perception and Intelligence,
Shenzhen Graduate School, Peking University, China
E-mail: zhouj06278@szcie.pku.edu.cn

Abstract—**It is very important for a robot to obverse its environment in real-time and walk without collision in a crowd. This paper presents a motion planning method, based on visual feedback, for safe Human-Robot Interaction (HRI) in dynamic environments. Firstly, in order to improve accuracy of features marching, Scale Invariant Feature Transform (SIFT) is merged into binocular stereo vision, which is used to detect motion of people. Secondly, by improving Lazy PRM, a robot can find the shortest safe path and move to predetermined destination along the path. Experimental results show that position of people can be detected in real-time in environments with several people walking inside, and the accuracy can reach 96%. Therefore, a robot can arrive at the goal configuration node without collision with people much faster than Lazy PRM.**

Keywords—**visual feedback, stereo vision, Lazy PRM, SIFT, path planning**

I. Introduction

Motion planning, an essential part in a system of intelligent autonomous mobile robots, enables robots to plan a path based on the roadmap and move to a predetermined destination through the path without guidance of human operator. It has become a very increasing interest field to guide a mobile robot to move to its destination by vision navigation.

In recent years, great progress has been achieved in the research of motion planning and many methods have been put forward. The classic PRM preceded in two phases: A learning phase and a query phase [1], turned out to be easy to be implemented and applicable to many different types of motion planning problems [2]. However its bottleneck is that difficult regions can't be covered by random sampling [3]. Recently some improved approaches have been proposed and perform well in various problems. OBPRM [4] can largely increase the connectivity of roadmap. However its implementation does require complex computation. Gaussian Sampler [3], which only samples node around obstacles, can reduce the number of configuration node in C_{free} to improve the efficient of PRM. However not all nodes near obstacle are as useful as nodes in the narrow passage to the connectivity of roadmap. Visibility based PRM [5] can quickly builds map with a small numbers of node. Whereas, it fail to find a path in an environment where narrow passage existed. Bridge Test Strategy [6] can efficiently solve the problem of narrow passage for PRM planning, but it needs to combine with other methods such as uniform sampler. Lazy PRM [7] assumes that all nodes and edges in roadmap are valid, and collision checks of nodes and edges are executed when a path has been found. Therefore, it can reduce the number of collision check, and minimize the running time of the planner accordingly. This method can be efficiently used in dynamic environments. However, the positions of obstacles change quickly, and it is useless to check collision before a path is found. So it is unwise to give up an invalid path whose front part is collision-free.

The approaches mentioned above present their advantages in different environments. However, most of those researches are implemented in simulated environments. They don't need to consider how to detect the static or dynamic obstacles. In practical applications, before motion planning, a robot has to know the roadmap of its environment, indicating the position of obstacles. In [8], the authors designed a system of object tracking for mobile robots by using monocular vision. Nevertheless, monocular vision can't be widely applied to the projects of distance measuring because of its limitations in small range of observation and few information of distance. Another system based on omni-directional vision to detect human motion for mobile robots was presented in [9]. Though omni-directional vision has the advantage of wide range of view, it has a difficulty in distance measuring. Stereo vision is widely used for its precision and the abundant information. J. Miura and Y. Shirai [10] formulated a vision-motion planning for mobile robots under uncertainty of visual observations. However, their solution still requires too much computation when it is used in a real-time application.

This paper aims to present a HRI system for mobile robots walking in dynamic indoor environments with several people walking inside, but without any other static obstacles such as desks and chairs. In such a situation, we only need to consider how to detect the motion of people in real-time and to guide the robot to move from its start configuration to end configuration without colliding with the moving people. In our method, binocular stereo vision is utilized for its advantages of precision and the abundant information it contains, to detect the position of people. And SIFT [11, 12], a very robust features detecting and matching method, is merged into stereo vision to increase the accuracy of feature marching. Finally, an improved Lazy PRM is used to plan a safe path for a mobile robot. That makes the robot can arrive at its destination more quickly.

The rest of this paper is organized as follows. Method of features matching and distance detecting with stereo vision is described in detail in section II. The general framework of motion planning is introduced in section III. Section IV shows the experimental results of tests in indoor environments and the analysis is presented as well. Finally, this paper is concluded in section V.

II. VISUAL FEEDBACK

It is very important for a mobile robot to be capable of obtaining information of environments in real-time to know where the safe position is. The main task of this part is to extract the coordinates of moving people by binocular stereo vision. Then motion planning can be performed with this information. In this part, a method of feature extracting and matching, SIFT is merged into binocular stereo vision to improve the accuracy of feature matching.

The system space conversion is shown in Fig.1. Real workspace is converted to simulated workspace by stereo vision processing. And then the simulated workspace is converted into configuration workspace by mapping. Motion planning is based on configuration workspace, and results of motion planning are shown in a simulated workspace.

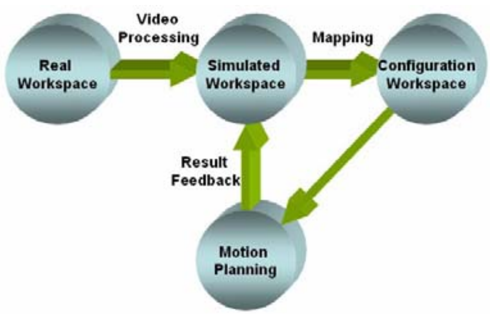


Figure 1. System space conversion

(a) (b)

Figure 2. A pair of stereo image of experiment. (a) Image extracted from left video. (b) Image extracted from right video.

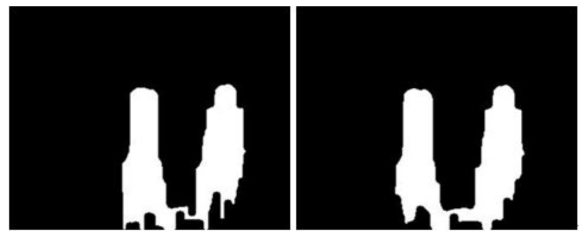
Firstly, gray images are obtained from images captured from the left and the right videos, which are shown in Fig.2. And then the outline of moving people can be extracted by temporal differencing [13]. The pixel difference function D_k can be defined as formula (1), and a motion image G_k can be extracted by a threshold using formula (2),

$$D_k = | I_k - I_{k-1} | , \quad (1)$$

$$G_k = \begin{cases} 1 \quad , & D_k \geq T \\ 0 \quad , & D_k \leq T \end{cases} . \quad (2)$$

Here k is the number of current frame and T is a threshold which is set according to different environments.

Fig.3 shows the area of moving people that have been detected from Fig.2 by temporal differencing and morphological processing. Then the areas of single people can be cut out for matching between left image and right image.



(a) (b)

Figure 3. Area of moving people detected. (a) Left. (b) Right.

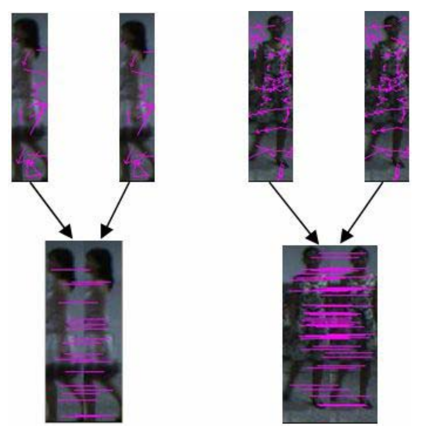


Figure 4. Keypoints selection and matching. (a) keypoints of the left person in two images and feature matching. (b) keypoints of the right person in two images and feature matching

SIFT [11, 12] is an algorithm for extracting local features from images and match two images. It is widely applied to objects recognition because of its robustness, rapidity and accuracy. The processing of SIFT is illustrated in Fig.4. The keypoints are shown as vectors, represented in pink arrows in above pictures, indicating scales, orientation, and location. Pairs of matching keypoints are connected by short lines in the pictures. Areas without people in them are not required to be processed, because those areas are irrelative to motion of people. That can largely reduce the number of keypoints, thus accelerate the processing of feature matching. That is very important in a real-time application. By now, the matching keypoints on moving people in the left image and the right image have been obtained. Then we can compute the position of people in real environments by binocular stereo vision.

Visual coordinate system [14] is shown in Fig.5. In order to measure the distance between robot and people, it is necessary to set the workspace coordinate system (*x, y, z*), whose original point is the position of robot, *X-Y* plane denotes the ground, and axis *z* is vertical to the ground. The visual coordinate system is set above the workspace coordinate system with original point O' on the axis *z* of workspace coordinate system.

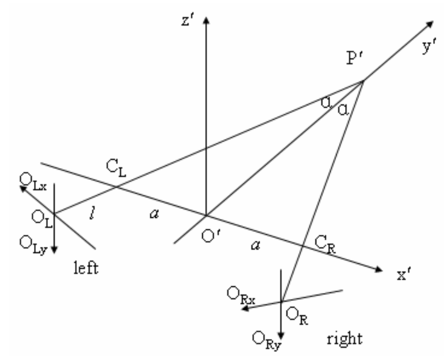


Figure 5. The visual coordinate system

Two cameras are put on both sides of axis *z*. We set the photo center to be C_L and C_R. A point P' can have an imaged point (O_{Lx}, O_{Ly}) in the left eye coordinate system and another point (O_{Rx}, O_{Ry}) in the right eye coordinate system respectively.

Therefore coordinate can be transformed from left eye and right eye coordinate systems to visual coordinate system by following formulas,

$$\begin{bmatrix} x_L' \\ y_L' \\ z_L' \end{bmatrix} = \begin{bmatrix} -\cos\alpha & 0 \\ \sin\beta & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} O_{Lx} \\ O_{Ly} \end{bmatrix} + \begin{bmatrix} -a - l\sin\alpha \\ -l\cos\alpha \\ 0 \end{bmatrix}, \tag{3}$$

$$\begin{bmatrix} x_R' \\ y_R' \\ z_R' \end{bmatrix} = \begin{bmatrix} -\cos\alpha & 0 \\ \sin\beta & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} O_{Rx} \\ O_{Ry} \end{bmatrix} + \begin{bmatrix} -a - l\sin\alpha \\ -l\cos\alpha \\ 0 \end{bmatrix}. \tag{4}$$

Here, (x_L' , y_L' , z_L') and (x_R' , y_R' , z_R') are the coordinates of the original point of the left and right eye coordinate systems separately. The axis lines of left eye and right eye are set to A_L and A_R, and their functions can be defined as (5) and (6),

$$A_L : \frac{x' + a}{x_L' + a} = \frac{y'}{y_L'}, z' = 0 , \tag{5}$$

$$A_R : \frac{x' + a}{x_R' + a} = \frac{y'}{y_R'}, z' = 0 \cdot \tag{6}$$

Here, x' , y' and z' are the position variables of visual coordinate system.

Therefore, once any point *P* on the target object is obtained, its coordinates in left image and right image are set to be (O_{Lxp} ,O_{LyP}) and (O_{RxP}, O_{RxP}), the position of this point in visual coordinate system can be computed by following formulas.

$$x' = \frac{al(O_{LxP} + O_{RxP})}{(l^2 + O_{Lxp} + O_{RxP})\sin 2\alpha + l(O_{LxP} - O_{RxP})\cos 2\alpha}, \tag{7}$$

$$y' = \frac{2a(l\cos\alpha + O_{LxP}\sin\alpha)(l\cos\alpha + O_{RxP}\sin\alpha)}{(l^2 + O_{Lxp} + O_{RxP})\sin 2\alpha + l(O_{LxP} - O_{RxP})\cos 2\alpha}, \tag{8}$$

$$z' = \frac{2a(l\cos\alpha - O_{LxP}\sin\alpha)O_{RxP}}{(l^2 + O_{Lxp} + O_{RxP})\sin 2\alpha + l(O_{LxP} - O_{RxP})\cos 2\alpha}. \tag{9}$$

Then, the coordinate of point *P* in workspace coordinate system can be computed by formula shown as follow,

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ h_c \end{bmatrix}. \tag{10}$$

Here Φ is the angle of elevation of camera, and h_c is the distance between the original points of workspace coordinate system and visual coordinate system.

III. Motion Planning

In dynamic environments, it is unwise to check collision in learning phase. Because that the obstacles keep moving all the time. Though these nodes are free, they may be not free a few moments later. Therefore, it is better to check collision after a path is found. The original Lazy PRM [7] is designed for static environment. In the roadmap building phase, all nodes and edges are assumed to be collision-free. Then, the shortest path is searched for in the roadmap without considering collision. After that, it is checked for collision. If collision occurs, the corresponding nodes and edges are removed from the roadmap, and then another shortest path will be searched for. This algorithm can largely reduce the number of collision-checks, which is the most time consuming step. However, it can't adapt to dynamic environment sufficiently. Therefore, an improved Lazy PRM is introduced in this paper.

Firstly, it is not necessary to remove nodes and edges where collisions occur. As they may be free with the moving of obstacles at next time. Secondly, it is unwise to give up the path on which collisions occur. Sometimes collision only happens at the end of the path. After all, this path is the shortest one currently. Robot can keep walking along this path until a collision node is near by, and then searching for another shortest path between the current node and the goal node. At this time, robot is already near the goal a lot. In a better situation, the collision nodes and edges become free again when robot is close to them. Then robot can continue using this shortest path to reach destination.

The aim of our method is to make a robot arrive at its destination as fast as possible. After finding a shortest path in the roadmap, validity of nodes on this path is checked. Robot can go ahead along the path if some nodes are free in the front part of the path. Fig.6 shows the shortest path found from the current position of robot and goal node without collision-checks.

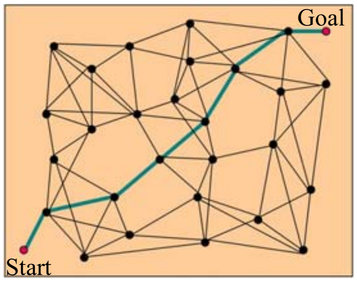


Figure 6. The green curve is the shortest path found in a roadmap. The two red nodes are the current position of robot and the goal node

The algorithm of motion planning is shown in algorithm 1. After building a roadmap with K-near connection, the shortest path can be found by the function of *SearchPath*. And then each node on the path is checked for collision or not by *CollisionCheck*. Then the robot can go to the safe nodes until it happens to meet a node not valid.

Algorithm 1: Motion Planning

Data: $G = (V, E)$ is a roadmap, V and E are sets of node and edge respectively. P is the shortest path. q is a node.

begin
  $V \leftarrow$ Sample();
  $G \leftarrow$ ConstructRoadMap(V);
  while $q_{current}$ is not q_{goal} **do**
    $P \leftarrow$ SearchPath(G);
    SearchTimes ++;
    for each node q on the path do
      if CollisionCheck(q) is true **then**
        mark q to be unfree;
        break;
      else
        $q_{current} = q$;
    end for
    if SearchTimes is 10 **then**
      resume all unfree nodes;
  end while
end

IV. EXPERIMENTS AND DISCUSSIONS

In order to evaluate our method, some experiments have been implemented in indoor environments with several people moving inside. Experiments were run on PC with a 2.0G Hz AMD processor and 512 MB RAM. Program has been implemented in C++ in Microsoft visual studio 2005 running under Windows XP.

A. Visual Feedback Tasks

Videos of people moving in indoor environment were obtained by binocular stereo vision system. The two cameras were set to be parallel with a distance of 18cm. The width and the length of room are both about 5 meters.

A simulated workspace was constructed according to the motion of people detected by binocular stereo vision. Human motion in video can be simulated in real-time by the green cylinders in simulated workspace entirely. In two indoor environments, 5 groups of videos with different number of people were tested. Each video contained more than 400 frames. The accuracy achieved 96%. Fig.7 displays some results of visual feedback in an environment with 2 people. When some one is occluded by another one, it is fail to detect position of people behind. The errors position detecting from videos are lower than 1 meter. But it won't influence the accuracy of collision-checks in motion planning, because the error only takes place in the position of people who is far from the robot. That can be accepted in a dynamic environment. Besides, we have set a safe distance of 1 meter. So that a robot will not collide with people even though existing 1 meter error.

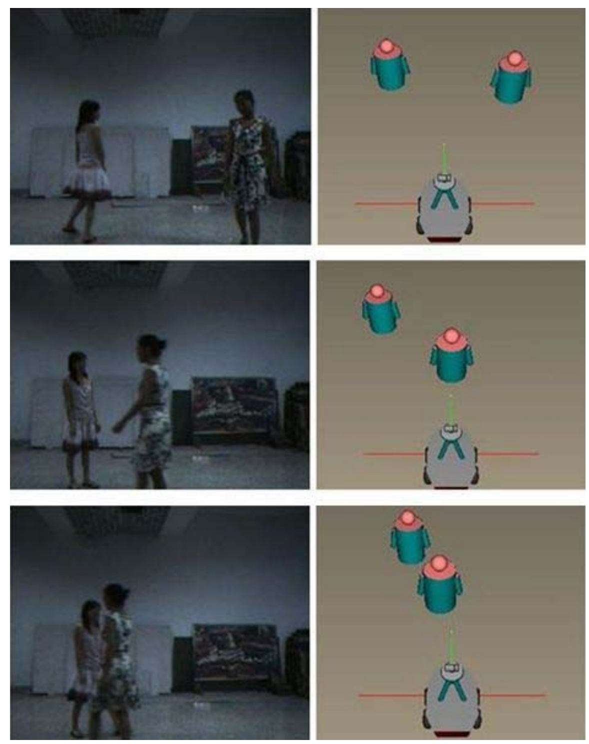


Figure 7. Motions of people in video are simulated. Left pictures are some frames extracted from left video, and right pictures are the simulation of motions of people from the left one respectively

In the case that two objects have the same color and brightness, there is no problem if there is no occlusion, because information of color and brightness isn't used in temporal differencing method. But the color of object can be used to identify covered objects when they become apart again. When occlusion happens, the foreground pictures are merged into one big block, the previous position of the person is kept to be current position since position of the person behind is lost. In the worst situation, two people keep walking together with one hiding behind another all the time. Thus the current position of the person will be wrong because it is still the previous position, while the person has gone far away.

TABLE I. COMPARISON WITH ORIGINAL LAZY PRM

Num of sample nodes	**Num of people**	**Number of nodes on path**						**Time(s)**					
		Original			Ours			Original			Ours		
		Max	Min	Ave	Max	Min	Ave	Max	Min	Ave	Max	Min	Ave
1000	2	27	24	**25.8**	32	23	**26.4**	18.50	13.91	**15.59**	19.03	12.33	**12.80**
	3	34	26	**27.8**	34	24	**26.9**	30.35	16.56	**20.67**	23.20	12.32	**16.28**
	4	34	26	**29.4**	36	27	**33.9**	44.84	26.00	**29.51**	32.41	18.43	**24.27**
	5	36	26	**30.3**	50	33	**41.4**	50.44	22.01	**34.57**	32.92	22.19	**26.96**
2000	2	40	28	**34.1**	41	27	**34.3**	38.92	19.30	**28.33**	36.5	18.56	**22.33**
	3	49	34	**38.7**	54	34	**41.0**	48.19	27.65	**35.66**	43.22	21.68	**28.03**
	4	53	31	**39.1**	61	35	**42.2**	53.13	36.31	**41.82**	46.39	23.06	**32.23**
	5	59	38	**43.9**	58	35	**42.1**	55.28	40.34	**46.34**	47.53	21.50	**33.55**
3000	2	49	36	**42.0**	48	37	**41.1**	35.94	23.50	**29.60**	38.06	18.50	**24.86**
	3	50	39	**44.0**	54	37	**44.8**	59.04	29.50	**39.26**	43.44	23.19	**31.61**
	4	55	39	**46.8**	61	42	**46.6**	55.28	37.62	**51.55**	53.35	22.97	**34.35**
	5	64	36	**47.7**	68	43	**50.9**	72.72	38.10	**55.14**	59.50	23.78	**36.43**

B. Motion Planning Tasks

Motion planning was tested in different number of sampled nodes, 1000, 2000 and 3000.We tested both our method and original Lazy PRM for 10 times in different environments with different number of people from 2 to 5. The comparisons are shown in table I. As is shown in the table, in an environment with a few numbers of people, there are not many differences in number of nodes on path between original Lazy PRM and our method. Sometimes, our method even has more number of nodes on path than the original Lazy PRM. However, our method performs great advantages in time robot arrived at its destination as shown in the last column of table I. Especially in environments with more people, our method is much faster than the original Lazy PRM.

Original Lazy PRM takes a lot of time to find a totally safe path that may be much longer than the shortest one. Moreover, the path may become unsafe a few moments later. Then it is necessary to search for a new path again. Our method can take full advantage of the shortest path that has been found, until it comes across an unsafe node. Then another shortest path will be searched for from the current node to the goal node. Therefore, the robot can arrive at destination more quickly.

V. CONCLUSIONS

This paper presents a new motion planning method for a robot working in real dynamic environments. Binocular stereo vision is utilized to detect people's motion in video. A method of feature extracting and matching, SIFT is merged to increase the accuracy of detecting of distance between people and robot. Besides, Lazy PRM is improved by changing its steps of collision-checks and searching strategy. Collision-checks are finished when robot begins walking along the shortest path which has been found. Robots can walk along the path which is safe at front part until it meets obstacles. Experimental results have demonstrated the effectivity of our approach.

ACKNOWLEDGMENTS

This work is supported by National Natural Science Foundation of China (NSFC, No.60675025, and No.60975050) and National High Technology Research and Development Program of China (863 Program, No.2006AA04Z247), Shenzhen Bureau of Science Technology and Information.

REFERENCES

[1] L. E. Kavraki, P. ˇSvestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," IEEE Transaction on Robotics and Automation, vol. 12, no. 4, pp. 566–580, 1996.

[2] R. Geraerts and M. H. Overmars. "A comparative study of probabilistic roadmap Planners", The 5th International Workshop on the Algorithmic Foundations of Robotics (WAFR), pp. 249-264, 2002.

[3] V. Boor, M. H. Overmars, and A. F. van der Stappen, "The Gaussian sampling strategy for probabilistic roadmap planners", IEEE International Conf. Robotics and Automation, pp. 1018–1023, 1999.

[4] N. M. Amato, O. B. Bayazit, L. K. Dale, C. Jones, and D. Vallejo, "OBPRM: An obstacle-based PRM for 3D workspaces", The 3rd International Workshop on the Algorithmic Foundations of Robotics (WAFR), pp. 155–168, 1998.

[5] C. Nissoux, T. Siméon, and J.-P. Laumond, "Visibility-based probabilistic roadmaps," IEEE International Conf. Intelligent Robots and Systems, pp. 1316–1321, 1999.

[6] D. Hsu, T. Jiang, R. John, and Z. Sun, "The bridge test for sampling narrow passages with probabilistic roadmap planners", IEEE Conf. Robotics and Automation, pp. 4420-4426, 2003.

[7] R. Bohlin and L. E. Kavraki, "Path planning using Lazy PRM", IEEE International Conf. Robotics and Automation, pp.521-528, 2000.

[8] X. Guo, C. Wang, and Z. Qu, "Object Tracking for Autonomous Mobile Robot based on Feedback of Monocular-Vision", IEEE Conf. Industrial Electronics and Applications, pp. 467-470, 2007.

[9] H. Liu, N. Dong, H. Zha, "Omni-directional Vision based Human Motion Detection for Autonomous Mobile Robots", IEEE International Conf. Systems, Man and Cybernetics, pp. 2236-2241, 2005.

[10] J. Miura and Y. Shirai, "Vision and Motion Planning for a Mobile Robot under Uncertainty", International Journal of Robotics Research, Vol. 16, No. 6, pp. 806-825, 1997.

[11] D. G. lowe, "Object Recognition from Local Scale-Invariant Features", IEEE International Conf. Computer Vision, pp. 1150-1157, 1999.

[12] D. G. lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.

[13] A. J. Lipton, H. Fujiyoshi and R. S. Patil, "Moving Target Classification and Tracking from Real-time Vision", IEEE Workshop on Applications of Computer Vision, pp. 8-14, 1998.

[14] G. Gao, J. Luo, S. Xie, X. Li, "Intelligent Sensor and it's Application (in Chinese)", Chemical Industry Press, pp. 85-86, 2005.