Machine Vision and Applications (2011) 22:349–358
DOI 10.1007/s00138-009-0245-4

ORIGINAL PAPER

Embedded omni-vision navigator based on multi-object tracking

Huazhu Fu · Zuoliang Cao · Xiaochun Cao

Received: 21 February 2009 / Accepted: 28 December 2009 / Published online: 27 January 2010

Abstract This paper presents an embedded omni-vision navigation system which involves landmark recognition, multi-object tracking, and vehicle localization. A new tracking algorithm, the feature matching embedded particle filter, is proposed. Landmark recognition is used to provide the front-end targets. A global localization method for omni-vision based on coordinate transformation is also proposed. The digital signal processor (DSP) provides a hardware platform for on-board tracker. Dynamic navigator employs DSP tracker to follow the landmarks in real time during the arbitrary movement of the vehicle and computes the position for localization based on time sequence images analysis. Experimental results demonstrated that the navigator can efficiently offer the vehicle guidance.

Keywords Omni-vision · Particle filter · Embedded · Multi-object tracker

This paper contains the results of research of the international science and technology collaboration project of China and Finland (2006DFA12410) supported by the Ministry of Science and Technology of the People's Republic of China. Additionally, the research involves part of the "863" High-Tech Program (2007AA04Z229) supported by the Ministry of Science and Technology of the People's Republic of China.

H. Fu (✉) · Z. Cao
School of Mechanical Engineering, Tianjin University of Technology, Tianjin, People's Republic of China
e-mail: fuhuazhu@163.com; huazhufu@gmail.com

Z. Cao
e-mail: zlcao@126.com

X. Cao
School of Computer Science and Technology, Tianjin University of Technology, Tianjin, People's Republic of China
e-mail: xcao@tju.edu.cn

1 Introduction

Autonomous navigation is of primary importance in applications involving the usage of autonomous guided vehicles (AGV). Vision based mobile robot navigation has been the source of countless research contributions, from the domains of both vision and control. Vision is becoming more and more common in applications such as localization, automatic map construction, autonomous navigation, path following, inspection, monitoring, and risky situation detection [2]. Vision based navigation systems provide an interesting option for both indoor and outdoor navigation as they can be used in environments without an external supporting infrastructure for the navigation, which is unlike GPS, for example. However, the environment has to contain some natural or artificial features that can be observed by the vision system, and these features have to have some relationship to spatial locations in the navigation environment [1].

More recently, omni-directional vision has been used for navigation thanks to its advantage of acquiring a hemispherical view with one look. Extremely wide-angle optical image equipment called fisheye lens is one of the most efficient ways to establish omni-directional vision system. It eliminates the need for a mechanical scan, which introduces the time latency of the system. The structure of fisheye lens is relatively dense and well-knit, while the structure of reflector lenses consists of two parts and is fragile. It is easier for a fisheye lens to find and track features since they stay longer in its 180º field of view of the environment [3–6].

The efficient tracking with fisheye lens for AGV in complex environments is a challenging task for the vision community. Particle filter, which is based on the theory of Bayesian correlation, maintains multiple hypotheses at the same time and uses a probabilistic motion model to predict the position of the moving objects. Particle filter based methods provide a promising approach to vision-based navigation, as they are computationally efficient and can be used to combine (or fuse) information from various sensors and sensor features [7–11]. In this paper, we employed particle filter as the probabilistic framework and designed a new algorithm, the feature-matching embedded particle filter (FMEPF), to integrate advantages of particle filter and feature matching.

In order to track a moving object in real time without delay or loss of the image data, a processor with the ability of effective computation and low energy cost is required. The digital signal processor (DSP) fits our demands, which is well known for powerful operation capability and parallel operation of instruction. It has been widely used in complicated algorithm calculation such as video/imaging processing, audio signal analysis, and intelligent control. In our AGV platform, we transplanted the particle filter algorithm into DSP Processor, as a compatible on-board multi-object tracker [12].

This paper is organized as follows. In Sect. 2, the method to recognize landmarks is explained. We describe the new tracking algorithm of feature matching embedded particle filter in Sect. 3, and the unique localization method in Sect. 4. The hardware and software of the whole navigation system are described in Sect. 5. Experimental results and our observations are reported in Sect. 6. Section 7 concludes this paper.

2 Landmark recognition by region of feature shift

For localization, the landmarks must be detected from the previous frame. Although the fisheye lens take the advantage of an extremely wide angle of view, there is an inherent distortion which makes it difficult to find the marks accurately. However, there are some properties of the landmark in the fisheye image:

1. The feature of landmarks (as colors, intensity, saliency, and template) could be set different from natural environment. The confidence score (weight) of feature could be computed by histogram likelihood.
2. The feature weight of the region around the landmarks is higher than other interference regions. It means the center of feature weight is closer to the targets when feature search is done in full image or a large range.
3. The available landmarks are usually near the center region of the fisheye image, so we could search the center in preference to other areas. It could remove the interference of edge area and also avoid inefficiency by searching the full image.

According to these feature, we designed a new method to recognize targets based on region of feature shift. The steps are shown as Fig. 1.

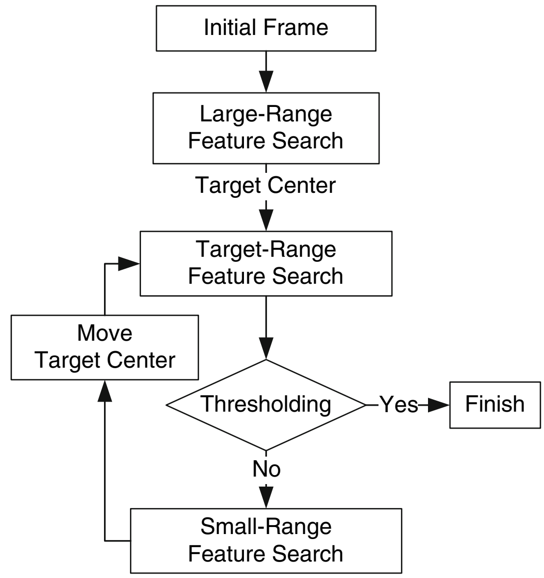


Fig. 1 The flowchart of landmark recognition

STEP 1: Search feature area in a large region near the image center, and tag the center of this result as a feature center.

STEP 2: In the target region surround the feature center, feature match again, and compute the feature weights. When the weight of feature is higher than threshold, the recognize process is finished and the feature center is the target.

STEP 3: If the weight is lower than threshold, enlarge the searching region (Scale is between 1.5 and 2, which floats according to the background noise level). Shift the feature center to the new position which is the center of the extended region of feature matching. And return to the STEP 2.

3 The feature-matching embedded particle filter algorithm

3.1 Particle filter

We use a fisheye lens with the view angle of 185° to build the omni-directional vision system. The extremely wide angle of view causes a lot of information redundancy making it difficult to detect the landmarks accurately only by recognize process, especially when the landmarks are obscured and the recognition may lead to localization error. On the other hand, the method only relying on the feature matching in each frame would increase computational cost. So we designed a multi-object real-time tracker as back-end of recognition.

We take particle filtering as a framework of the tracker. Particle filtering is a Monte Carlo sampling approach to

Bayesian filtering, where the probability density is represented by a set of weighted samples (called particles). These samples describe possible instantiations of the state of the system [13]. As a consequence, the distribution over the location of the tracking object is represented by the multiple discrete particles.

The dynamic system model is defined as

$$\begin{cases} x_k = f(x_{k-1}, v_{k-1}) \\ z_k = h(x_k, w_k) \end{cases} \tag{1}$$

where v_{k-1} is the system disturbance at time k , w_k is the observation noise, $x_k \in R^n$ is the system state specified by a magnitude and a direction as target velocity, position, and rotation angle. z_k is the observation, $f(\cdot)$ is state transition function, and $h(\cdot)$ is the system observation function.

In the Bayes filtering, the posterior distribution is iteratively updated over the current state X_k, given all observations $Z_k = \{Z_1, \ldots, Z_k\}$ up to time k, as follows:

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})\mathrm{d}x_{k-1} \tag{2}$$

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \tag{3}$$

where $p(Z_k|X_k)$ expresses the observation model which specifies the likelihood of an object being in a specific state and $p(X_k|X_{k-1})$ is the transition model which specifies how objects move between frames. In a particle filter, prior distribution $p(X_{k-1}|Z_{k-1})$ is approximated recursively as a set of N weighted samples, which is the weight for particle. Based on the Monte Carlo approximation of the integral, we can get

$$p(X_k|Z_k) \approx kp(Z_k|X_k)\sum_{i=1}^{N} w_{k-1}^{(i)} p(X_k|X_{k-1}^{(i)}) \tag{4}$$

3.2 Feature matching embedded particle filter

In experiments, the single particle filtering could meet the requirement of the tracker on the whole. But if the target moves with speed or haste, the track window might drift. Otherwise, the precision and stability are related with the particle number; the large number would involve massive calculation and reduce efficiency. So in this paper we proposed a new algorithm, the feature-matching embedded particle filter (FMEPF), to combine particle filter and feature matching for object tracking, integrating advantages of the two methods. The particle filter is used as the probabilistic framework to track a region of interest (ROI), and then the feature matching is applied in this ROI. Finally, the location of the tracking object is represented by the result of matching.

In FMEPF, more particles are not needed to track accurately and robustly, because in each frame, targets could be detected by the feature matching from the ROI. The size of the ROI could be set based on the velocity of the target.

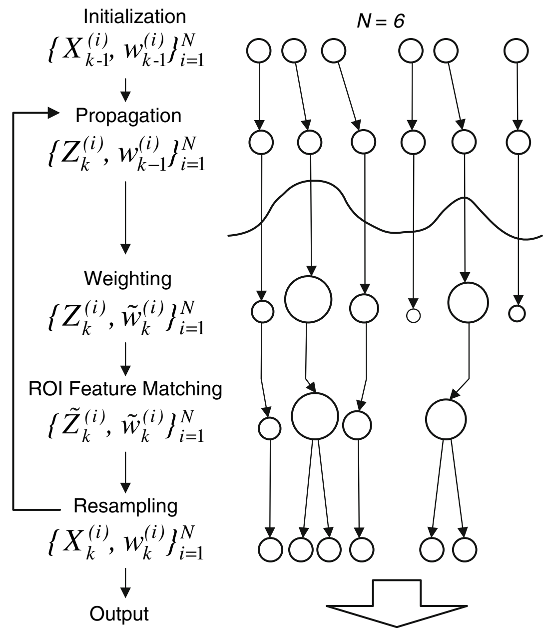


Fig. 2 The flowchart of FMEPF tracking

The general operation of the multi-object tracker based on FMEPF is illustrated in Fig. 2.

The principal steps in the FMEPF algorithm are

STEP 1 Initialization
Generate particles from the initial distribution $p(X_0)$ to $\{X_0^{(i)}, w_0^{(i)}\}_{i=1}^N$, and set $k = 1$.

STEP 2 Propagation
For $i = 1, \ldots, N.$, Sample $Z_k^{(i)}$ according to the transition model $p(Z_k^{(i)}|X_{0:k-1}^{(i)})$.

STEP 3 Weighting
Calculate the importance weight likelihood:

$$\tilde{w}_k^{(i)} = w_{k-1}^{(i)} p(Z_k|X_k^{(i)}) \qquad i = 1, \ldots, N. \tag{5}$$

Normalize the weights:

$$w_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{j=1}^{N} \tilde{w}_k^{(j)}} \qquad i = 1, \ldots, N. \tag{6}$$

Output a set of particles $\{\tilde{Z}_k^{(i)}, w_k^{(i)}\}_{i=1}^N$ that can be used to approximate the posterior distribution as

$$p(\tilde{Z}_k|Z_k) = \sum_{i=1}^{N} w_k^{(i)} \delta(X_k - X_k^{(i)}) \tag{7}$$

where $\delta(g)$ is the Dirac delta function.

STEP 4 ROI Feature Matching
Set ROI window according $\{\tilde{Z}_k^{(i)}, w_k^{(i)}\}_{i=1}^N$, apply feature matching in this region and output the result as $\{X_k^{(i)}, w_k^{(i)}\}_{i=1}^N$ the target distribution.

STEP 5 Resampling
Resample the new particles $X_k^{(i)}$ with probability $w_t^{(i)}$ to obtain N independent and identically distributed random particles $X_k^{(j)}$ approximately distributed according to $p(X_k|Z_k)$.

STEP 6 Set $k = k + 1$, and return to STEP 2.

4 Localization for vehicle based on fisheye lens

4.1 Equidistance projection model of the fisheye lens

As known, the lens projection model generally described as

$$r = f(w) \tag{8}$$

where the r is the distance from the distorted point to the lens' center, and ω means the angle of an incident ray as

$$\omega = \arctan \frac{\sqrt{x^2 + y^2}}{z} = \arctan \frac{r'}{z} \tag{9}$$

where $P(x, y, z)$ is the point in the world coordinate system, and r' is the distance from the P to the center in $Z = z$ plane [14].

In this paper, the equidistance projection is used in fisheye lens coordinate building. As shown in Fig. 3, the relation between the angle of view ω with the radial distance r between projection point and projection center is one-to-one correspondence. Based on this mapping, the image coordinate distance could be transformed to the space angle of the landmark.

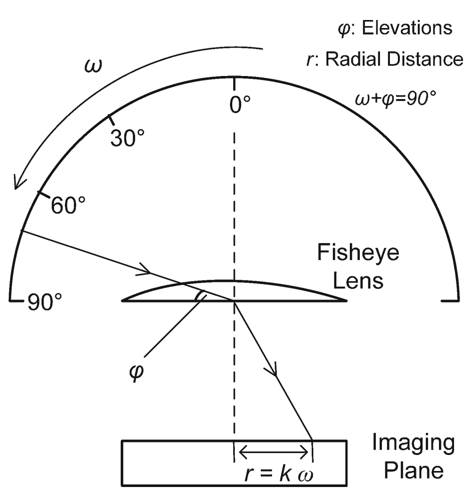


Fig. 3 The equidistance projection model

4.2 Localization for vehicle

Double landmarks are fixed on the edge of the AGV moving area. The height of two landmarks and the distance between them are measured as the known parameters. When the AGV is being navigated, the fisheye lens is plumbed and placed on top of the AGV and two landmarks are tracked by multi-object tracker to get the landmark positions in the image. In this section, we will discuss how to localize the AGV utilizing the space and image information of landmarks.

In a localization system, there are three coordinate systems: the space coordinate system (XOY), the vehicle system $(X'O'Y')$, and the image system $(X''O''Y'')$. The landmark's coordinates in fisheye image system are the only information the guidance can get from the environment. The main problem in the navigation system is how to plan the desired path with the beacon's information. So the relations among these coordinate systems should be built up first. As Figs. 4 and 5 show, utilizing the elevations obtained from image and demarcated parameters of landmarks, the physical space position of AGV is confirmed. The landmark A is

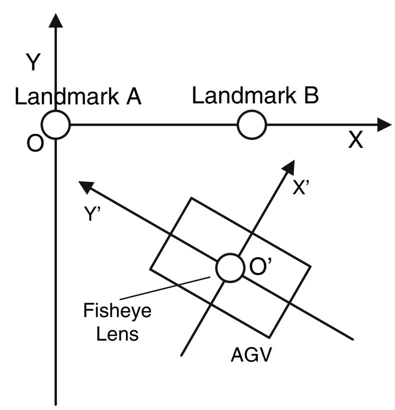


Fig. 4 The relations between the space system and the vehicle system

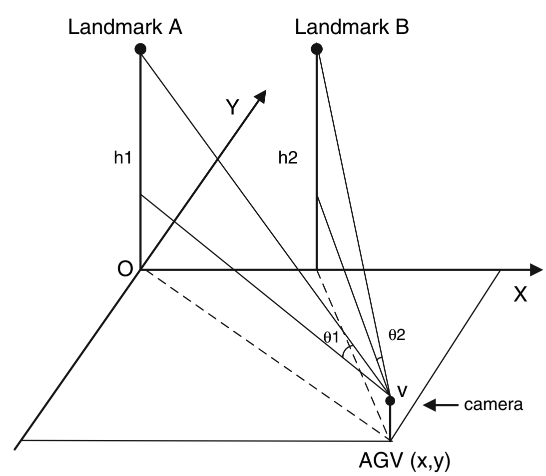


Fig. 5 The space structure for landmark localization

chosen as the origin, AB is set as axis X and the direction from A to B is the positive orientation of axis X. Axis Y is vertical to Axis X. The coordinates (X_i'', Y_i'') of landmarks in the fisheye image is the only input parameter. According to the equidistance projection of fisheye lens, $r = k\omega$, the landmark coordinates in the vehicle coordinate system (X_i', Y_i') are

$$\begin{cases} X_i' = \dfrac{(h_i - v)\, X_i''}{r_i \tan \varphi_i} \\ Y_i' = \dfrac{(h_i - v)\, Y_i''}{r_i \tan \varphi_i} \end{cases} \tag{10}$$

where the (X_i'', Y_i'') are landmark coordinates in the fisheye image, v is the height from ground to lens, as $i = 1$ or $i = 2$. On the basis of the principle of the coordinate conversion, we can get the coordinate relation between the space system and the vehicle system as the following equations:

$$\begin{pmatrix} X_0 \\ Y_0 \\ 1 \end{pmatrix} = \begin{pmatrix} -\cos\theta & \sin\theta & X_1 \\ -\sin\theta & -\cos\theta & Y_1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_1' \\ Y_1' \\ 1 \end{pmatrix} \tag{11}$$

where the (X_0, Y_0) is the AGV physical space coordinates. The θ which is the horizontal direction of AGV in space coordinate system could be obtained from the fisheye image coordinate system as shown in Fig. 6.

The direction of the image Axis y is the positive moving orientation of AGV. As a result, the direction of AGV in physical coordinate system which is presented by θ is the included angle from the positive orientation of Axis y' to the AGV moving direction. A (X_1'', Y_1'') and B (X_2'', Y_2'') are the image coordinates of two landmarks. (X_0'', Y_0'') are the image coordinates of the projection center. We can get the AGV orientation θ as

$$\theta = \begin{cases} \pi, & Y_1'' \geq Y_2'' \text{ and } X_1'' = X_2'' \\ -\arctan\left(\dfrac{Y_1'' - Y_2''}{X_1'' - X_2''}\right), & Y_1'' \leq Y_2'' \\ -\pi, & Y_1'' \leq Y_2'' \text{ and } X_1'' = X_2'' \\ \pi - \arctan\left(\dfrac{Y_1'' - Y_2''}{X_1'' - X_2''}\right), & Y_1'' \geq Y_2'' \end{cases} \tag{12}$$

With the Eqs. 10, 11 and 12, we can calculate the physical space coordinates of the Vehicle (X_0, Y_0) as

$$\begin{cases} X_0 = -\frac{(h_1-v)(X_1''-X_0'')\cos\theta}{r_1\tan(\theta_1)} + \frac{(h_1-v)(Y_1''-Y_0'')\sin\theta}{r_1\tan(\theta_1)} + X_1 \\ Y_0 = -\frac{(h_1-v)(X_1''-X_0'')\sin\theta}{r_1\tan(\theta_1)} - \frac{(h_1-v)(Y_1''-Y_0'')\cos\theta}{r_1\tan(\theta_1)} + Y_1 \end{cases} \tag{13}$$

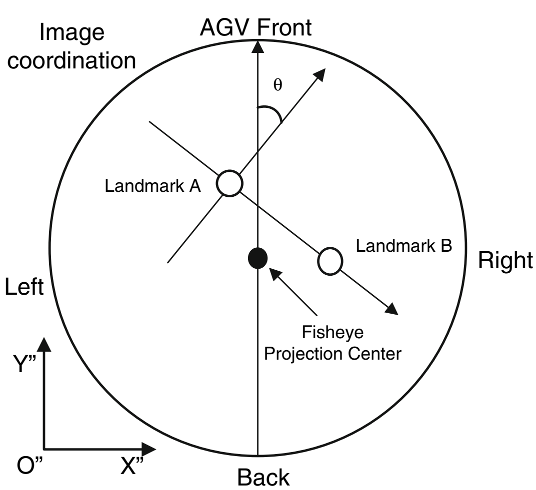


Fig. 6 The coordinates system of the fisheye image

5 The navigation system configuration

5.1 Multi-object DSP tracker

We used the FMEPF algorithm to achieve an ideal tracking effect. For localization, the navigator often needs more than two landmarks to track and the computation of Multi-object tracking is difficult with the common CPU based on x86 structure. On another side, the navigator is used for vehicle system which requires small volume and low power dissipation. So, we present a compatible embedded real-time image processing platform for the navigation system by utilizing DSP. We choose the TMS320C6437 DSP of Texas Instruments, which is the one of the TMS320C64x+ DSPs series.

The DM6437 device is the third-generation high-performance fixed-point DSP, which is good at digital media applications by VelociTI very-long-instruction-word (VLIW) architecture developed by Texas Instruments (TI). The device is upward code-compatible from previous devices that are part of the C6000 DSP platform. The DSP also supports added functionality and has an expanded instruction set from previous devices [15,16].

The DSP Developer's Kit includes 600 MHz DM6437 chip, 2 configurable video ports 128 MB DDR2 DRAM, UART and CAN I/O ports, 64 KB SRAM, 16 Mbytes Nor Flash, and 64 Mbytes NAND Flash. With these peripheral sets, the circuit is more suitable for image processing. And the TMS320DM6437 processor also takes full advantage of the DaVinci software and development infrastructure by allowing designers to focus on the application functionality. Developers can implement video, audio, voice, and speech technology through simple calls to the DaVinci application programming interface (API) that manages the implementation of codec engines and matching screen resolutions. The hardware configuration of the DM6347 circuit is shown in Fig. 7.

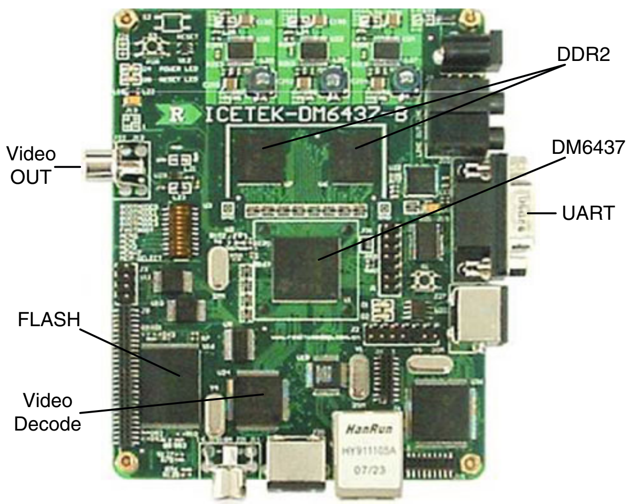


Fig. 7 Hardware profile of DSP tracker

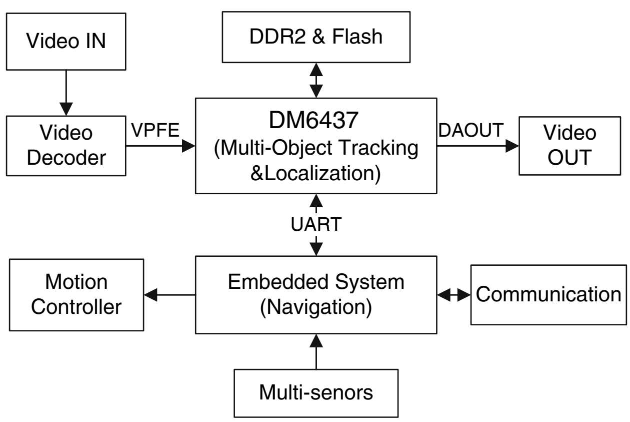


Fig. 8 Hardware configuration of navigator

5.2 Navigator hardware

Besides DSP tracker, the navigation system also employs an embedded platform which consists of multi-sensors, remote control, internet port, and motor servo system. Its functions are communication, obstacle avoidance, guide path plan, multi-sensor fusion, and motor driving. The whole navigation hardware structure is shown in Fig. 8.

5.3 Navigator software system

The multi-object tracker is the core of the navigator; the recognition provides the prior targets for tracking, and at the back-end are the navigation and driver models which involve positional estimation, surrounding perception and control AGV movement. The data flowchart is shown in Fig. 9.

The image captured by fisheye lens and CCD camera, through the A/D decoder, is transformed to digital sign and input to DSP tracker. Before the tracking, the navigator needs to recognize and initialize the landmarks as the prior targets to

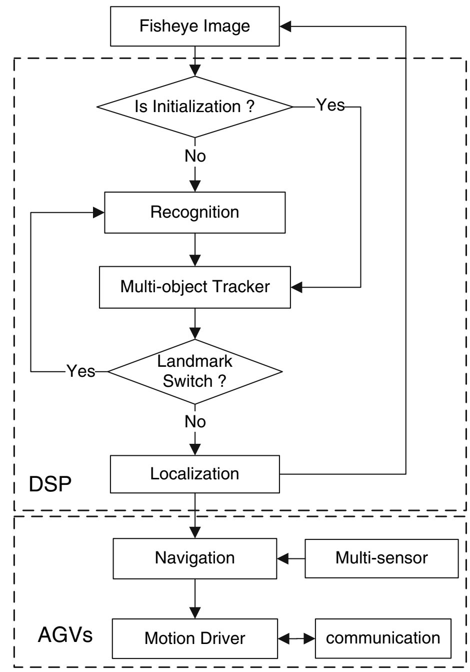


Fig. 9 Data flowchart of navigation system

tracking process. In tracker, the targets are real-time tracked by FMEPF algorithm and estimated whether they are in the available area. If they are invalid, the system goes to Recognition Model and switches new landmarks in next group. If the targets are valid, we localize the physical space coordinates and the horizontal direction of AGV according to the

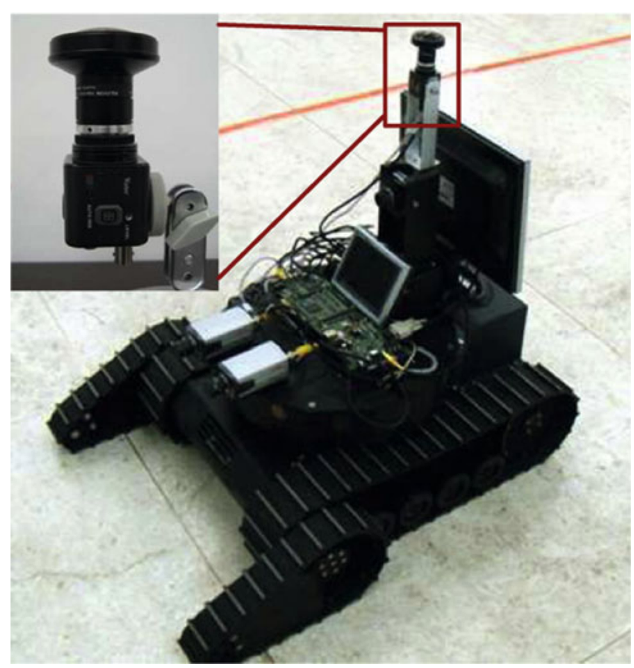


Fig. 10 The CCD camera with fisheye lens and automatic guided vehicles

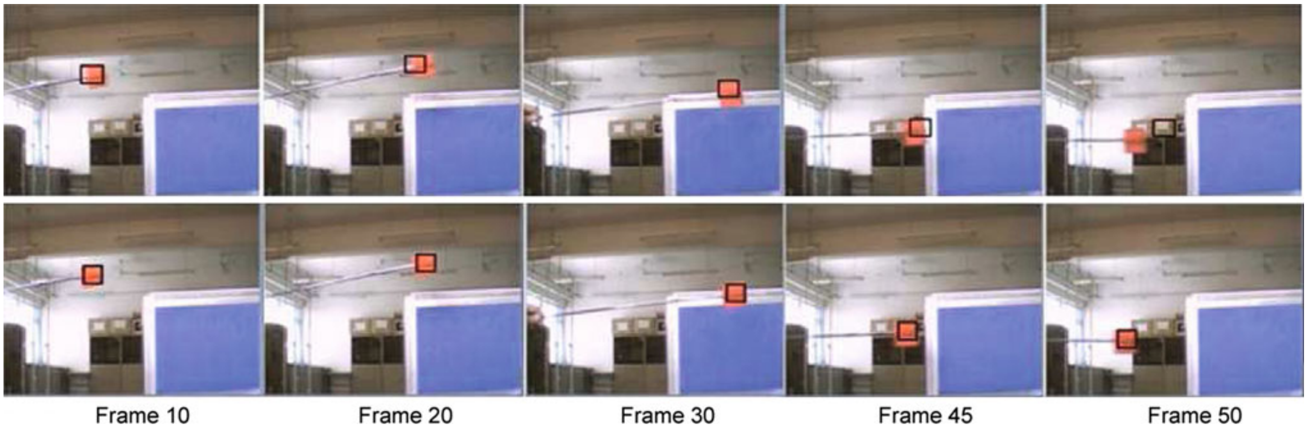


Fig. 11 Comparisons between the conventional particle filter (*upper row*) and our FMEPF (*bottom row*). Tracked landmarks are marked in *black squares*

coordinates of landmarks in fisheye image. Then DSP tracker outputs the AGV position to the Vehicle-embedded platform. With the information of AGV, the navigator integrates other sensor's information and finishes the path planning.

6 Experimental result and discussion

6.1 Experimental environment

We divided experiments into three steps: tracking algorithm comparison experiment, experiment of multi-object DSP tracker, and AGV navigation experiment. The environment is as follows:

The PC of simulation test consisted of P4 3.06 GHz CPU, 512 MB memory, windows XP and Visual Studio 2005. The Camera is WATEC WAT-221s, with $1/2'$ CCD and output format is PAL. The fisheye lens is Fujinon FE185C046HA-1 fixed focus $1/2''$ fisheye lens, whose angle of view is $185° \times 185°$, and supports for up to 5 mega pixel. The lens is designed with equidistance projection $r = k\omega$, where the k is measured 3.2 as image is 720×576 pixels. In the vehicle-mounted experiment, the tracked mobile robot, shown in Fig. 10, is used as the vehicle.

6.2 Tracking algorithm comparison experiment

To illustrate the differences between the conventional particle filter and our FMEPF, we apply them to the same image sequences. These video sequences are captured at 5 frames per second in a usual office environment. The resolution of each image is 640×480 pixels and the tracked region is 40×40 pixels. The region feature matching used histogram matching. Experimental results demonstrate that our algorithm outperforms the other method and performs well with fewer samples. Some tracking results of the two trackers are shown in Fig. 11, the estimated trajectories for FMEPF tracking in experimental is shown in Fig. 12.

We applied at least 300 samples to the conventional particle filter. As it can be seen in Fig. 11 Upper Row, the algorithm tracked target at beginning, but caused by rapid movement the tracker loses the target (as frames 45 and 50 shown). Using the same observation model and dynamic model, thanks to the embedded region feature-matching analysis, our tracker can maintain target tracking with fewer samples; only 100

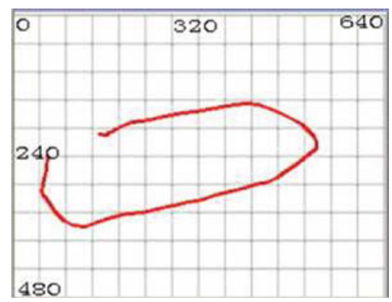


Fig. 12 The estimated trajectories in the FMEPF experiment (unit: pixel)

Table 1 The UV components of color

Color	Threshold of U component	Threshold of V component
Green	90–100	100–110
Yellow	60–70	125–135
Red	110–120	160–170
Purple	135–145	130–140
Blue	140–150	90–110
Orange	100–115	135–145

The values may be changed with different illumination

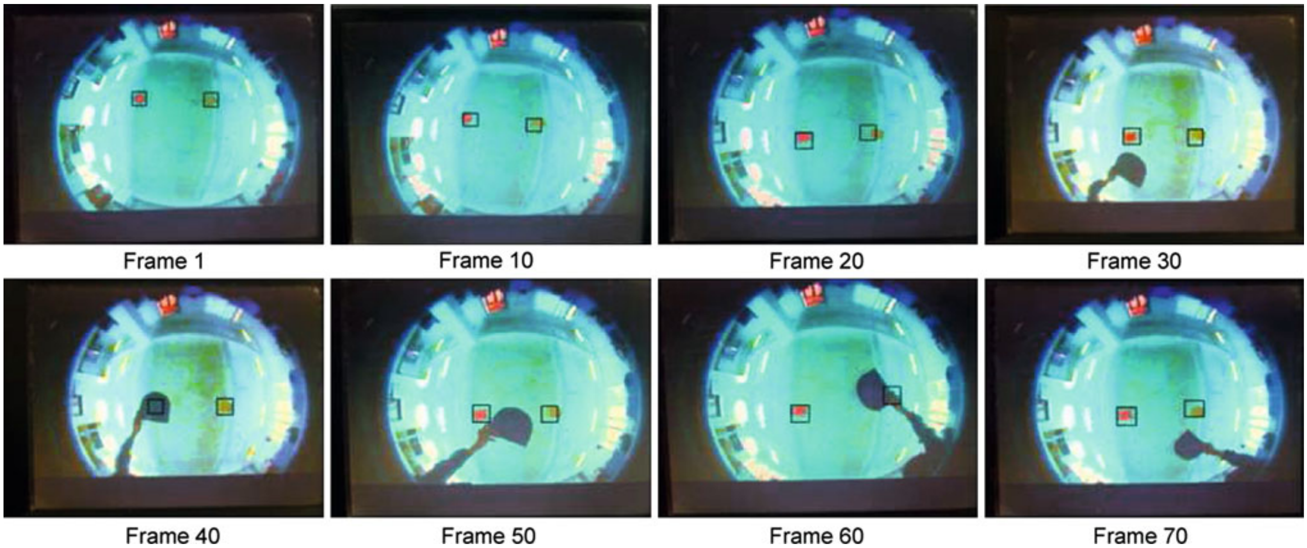


Fig. 13 The images of the multi-object DSP tracker's experiment. The *left* target is red landmark and the *right* is yellow landmark. The *black block* is artificial chaff

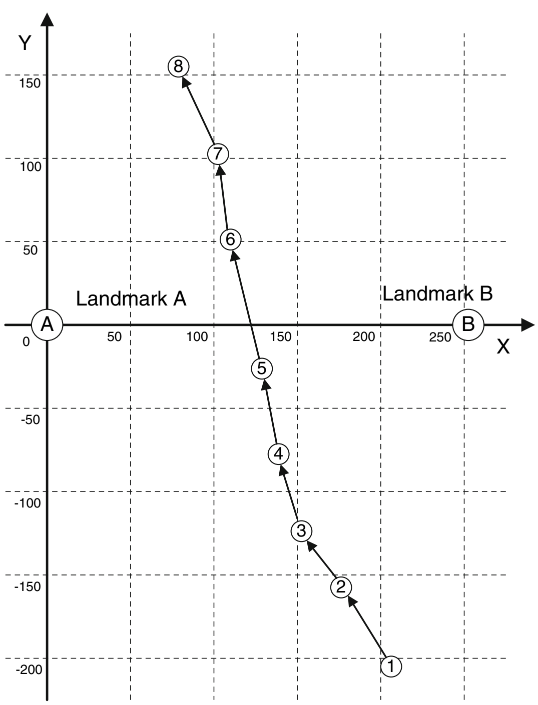


Fig. 14 AGV moving Path (unit: cm) and the labels are the sampling points

samples are needed to robustly track the target. Hence our algorithm is very efficient and real-time (Fig. 12).

6.3 Experiment of multi-object DSP tracker

We implement the tracker on DSP platform and test it with a real video sequence. The resolution of video is 720 × 576 pixels and the tracked region is 30 × 30 pixels. We use color histogram as the feature vector in the experiment. The format of image is YUV: 422, so the color threshold of UV parameter must be calibrated first. The detected UV values are shown in Table 1.

As shown in Table 1, the yellow and red colors are well segregated from background. So we set the yellow and red to be the main features of the landmark. Then we track the two landmarks by DSP tracker, and the tracked results are as shown in Fig. 13.

In the experiments, landmarks were recognized by region of feature shift from the first frame, and then the moving targets were tracked by FMEPF algorithm. The number of particles of each target is 100. Experimental results demonstrate that on the DSP platform the tracking algorithm successfully deals with two fast moving landmarks like frames 1–30. The DSP tracker is very efficient and meets the needs of real-time. When the target is lost or covered, the tracker restarts the recognition model to detect new targets. This provides the robustness and continuity for tracking.

6.4 AGV navigation experiment

Two beacons with different colors are hanged on the roof as landmarks. The height of Landmark A and B are 2.56 and 2.49 m. The distance between them is 2.65 m. The height of lens is 0.87 m. Then the AGV moves along the path, as shown in Fig. 14. We pick up eight sampling points, shown in Fig. 15. The localization results are shown in the Table 2.

As the experiments showed, the navigation system is accurate to a certain extent. The moving area has been limited by the landmarks. Although fisheye lens take the advantage

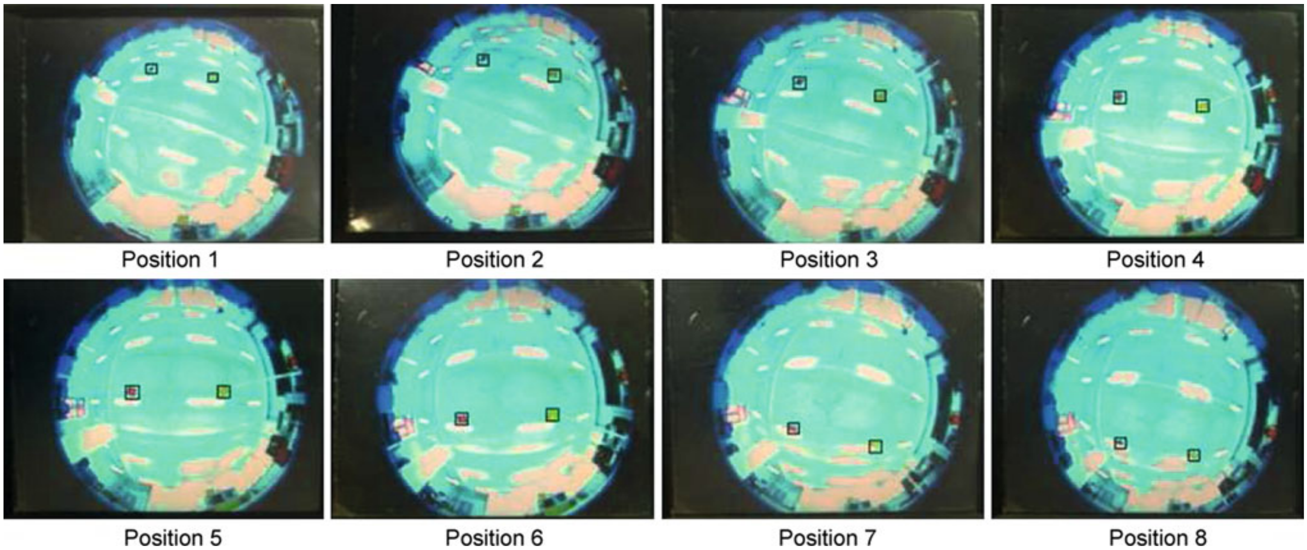


Fig. 15 The sampling point images of AGV navigation experiment. The images were captured from fisheye lens. In each image, the *left* tracking target is landmark A featured by *red color*, the *right* is landmark B featured by *yellow color*

Table 2 The sampling point localization results (cm)

Sampling point	Actual coordinates	Localization coordinates
Position 1	(210, −214) 14º	(199, −201) 17º
Position 2	(186, −160) 25º	(180, −155) 23º
Position 3	(159, −133) 17º	(155, −129) 15º
Position 4	(146, −80) 9º	(145, −77) 10º
Position 5	(133, −23) 6º	(130, −25) 5º
Position 6	(119, 53) 7º	(121, 56) 8º
Position 7	(106, 105) 15º	(111, 112) 13º
Position 8	(80, 159) 11º	(87, 168) 14º

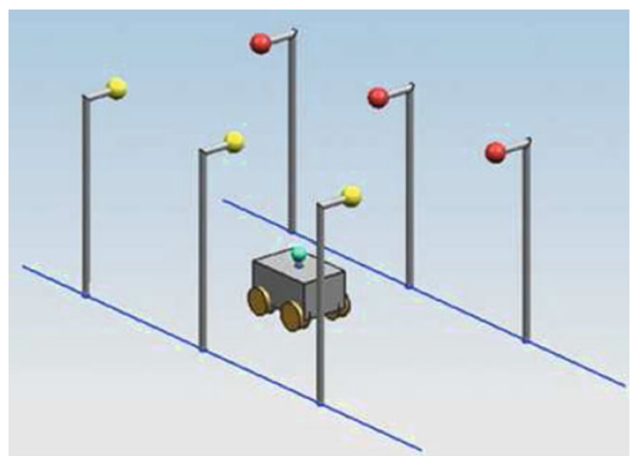


Fig. 16 The sequence landmarks of AGV navigation

of an extremely wide angle of view, there is a distortion in the fisheye image, especially near the edge where the distortion is much more serious. As AGV is away from the landmark, the projection coordinates of the marks will close the edge of the fisheye image, and several pixels tolerances caused by distortion would lead to a large mistake of localization.

To reduce this tolerances, we set a sequence of landmarks, as Fig. 16 shows. As the projection point moves out of the effective coverage, the tracker recognizes the next couple of beacons in the AGV front area and refreshes space coordinate insuring long-distance and a larger-area AGV movement.

7 Conclusion

In this paper, we built a navigation system based on omni-directional vision. The multi-object tracker using region feature matching embedded particle filter is employed as core of guidance system. We proposed particle filter as the probabilistic framework, embedded region feature matching to integrate advantages of the two methods. The tracker based on FMEPF algorithm could robustly and effectively track moving multiple targets, and the DSP platform has advantages of high efficiency, real time, low power, and small volume. We also designed an omni-directional vision localization method based on two beacons, with the extremely wide angle of view offered by fisheye lens, the coordinates, and orientation of AGV could be computed. The experiment shows that the navigation system could be applied to AGV guidance, vehicle-mounted system, robotic, security inspection, and other related areas.

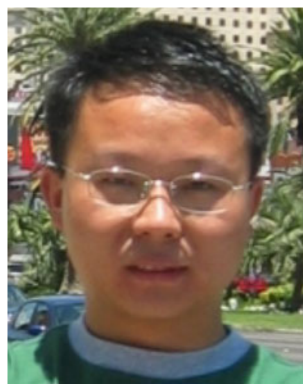
References

1. Cao, Z., Liu, S., Roning, J.: Omni-directional vision localization based on particle filter. Image and Graphics 2007, Chengdu China, Fourth International Conference, pp. 478–483 (2007)
2. Bonin-Font, F., Ortiz, A., Oliver, G.: Visual navigation for mobile robots: a survey. J. Intell. Robot. Syst. **53**(3), 263–296 (2008)
3. Winters, N., Gaspar, J., Lacey, G., Santos-Victor, J.: Omni-directional vision for robot navigation. In: Proceedings of IEEE Workshop on Omnidirectional Vision (2000)
4. Martinet, P., Thibaud, C.: Automatic guided vehicles: robust controller design in image space. Auton. Robot. **8**(1), 25–42 (2000)
5. Vanijja, V., Horiguchi, S.: Omni-directional stereoscopic images from one omni-directional camera. J. VLSI Signal Process. **42**(1), 91–101 (2006)
6. Wilcox, P.D.: Omni-directional guided wave transducer arrays for the rapid inspection of large areas of plate structures. IEEE Trans. Ultrason. Ferroelectr. Freq. Control. **50**(6), 699–709 (2003)
7. Kotecha, J.H., Djuric, P.M.: Gaussian particle filtering. IEEE Trans. Signal Process. **51**(10), 2592–2601 (2003)
8. Charif, H., McKenna, S.: Tracking the activity of participants in a meeting. Mach. Vis. Appl. **17**(2), 83–93 (2006)
9. Menegatti, E., Pretto, A., PageIIo, E.: Omnidirectional vision scan matching for robot localization in dynamic environments. IEEE Trans. Robot. Autom. **22**(3), 97–109 (2006)
10. Sankaranarayanan, A.C., Srivastava, A., Chellappa, R.: Algorithmic and architectural optimizations for computationally efficient particle filtering. IEEE Trans. Image Process. **17**(5), 737–748 (2008)
11. Kwok, N., Rad, A.: A modified particle filter for simultaneous localization and mapping. J. Intell. Robot. Syst. **46**(4), 365–382 (2006)
12. Cho, J., Jin, S., Pham, X., et al.: Multiple objects tracking circuit using particle filters with multiple features. IEEE Int. Conf. Robot. Autom. 4639–4644 (2007)
13. Shaohua Kevin, Z., Chellappa, R., Baback, M.: Visual tracking and recognition using appearance-adaptive models in particle filters. IEEE Trans. Image Process. **13**(11), 1491–1506 (2004)
14. Wang, Y.: Fisheye Lens Optics. China Science Press, China (2006)
15. Texas Instruments: TMS320DM6437 Digital Media Processor, Texas Instruments (2008)
16. Texas Instruments: TMS320C64x/C64x+ DSP CPU and Instruction Set Reference Guide, Texas Instruments (2008)

Author Biographies

Huazhu Fu received his B.S. degree in Mathematics Science from Nankai University in 2006, and the M.E. degree in Electromechanical Engineering from Tianjin University of Technology in 2010. His current research interests include omni-vision system, object recognition and tracking, embedded system, and their applications to computer vision and robotic navigation.

Zuoliang Cao received his B.S. (1967) degree from Harbin Engineering University, China. He, as an engineer, was employed at Behai Manufacturing Engineering Co. for 10 years. Then he joined the faculty groups of the Tianjin University of Technology. His research interests focus on Robotics, Computer Vision and Mechanical Control Engineering. He has received numerous awards including the Excellent Contribution Expert Award issued by Chinese Government and Machatronics Expert Award issued by Tianjin Scientific Committee. He spent 3 years at the University of Cincinnati as a visiting professor and worked on the cooperation project of mobile robots. He cooperated with Tennant Co. of USA to develop a Vision Guided AGV and even had a US patent award. He also built an advanced prototype of an Omni-vision Guided Vehicle sponsored by 863 China National High-Tech Development Program and got the Invention Award on Tianjin Science and Technology Progress. More than 200 scientific papers and books have been published.



Xiaochun Cao received his B.E. and M.E. degrees both in Computer Science from Beihang University (BUAA), Beijing, China in 1999 and 2002, respectively. He received his Ph.D. degree in Computer Science from University of Central Florida, Orlando, FL in 2006 with dissertation nominated for the university-level Award for the Outstanding Dissertation. After graduation, he spent about two and half years at ObjectVideo Inc. as a Research Scientist. Since August 2008, he has been with Tianjin University, where he is currently Professor of Computer Science and Adjunct Professor of Computer Software. He has authored and co-authored over 40 peer-reviewed journal and conference papers, and has been in the organizing and the technical committees of several international colloquia. In 2004, he was a recipient of the Pierro Zamperoni best paper award in the International Conf. on Pattern Recognition.