



# Appearance-based navigation and homing for autonomous mobile robot<sup>☆</sup>

Naoya Ohnishi <sup>a,1</sup>, Atsushi Imaiya <sup>b,\*</sup>

<sup>a</sup> Graduate School of Science and Technology, Chiba University, Japan

<sup>b</sup> Institute of Media and Information Technology, Chiba University, Japan

## ARTICLE INFO

### Article history:

Received 5 December 2011

Received in revised form 30 October 2012

Accepted 28 November 2012

### Keywords:

Visual navigation

Visual homing

Nonholonomic mobile robot

Visual potential

Optical flow

Dominant plane

## ABSTRACT

In this paper, we develop an algorithm for navigating a mobile robot using the visual potential. The visual potential is computed from an image sequence and optical flow computed from successive images captured by a camera mounted on the robot, that is, the visual potential for navigation is computed from appearances of the workspace observed as an image sequence. The direction to the destination is provided at the initial position of the robot. The robot dynamically selects a local pathway to the destination without collision with obstacles and without any knowledge of the robot workspace. Furthermore, the guidance algorithm to destination allows the mobile robot to return from the destination to the initial position. We present the experimental results of navigation and homing in synthetic and real environments.

© 2013 Elsevier B.V. All rights reserved.

## 1. Introduction

Recently, the control systems for autonomous mobile robots are being actively studied, since mobile robots are widely used in various environments, for example, in factories, offices, and homes. In such environments, the robot must move safely without collision with obstacles and walls. If the robot has an environmental map, the potential field method [1] is a valid method of path planning. In the real world, however, the robot might not have the environmental map, and the environment may dynamically change [2]. Furthermore, many of the mobile robots in the real environment are wheel-driven robots, and they move by a nonholonomic mechanism [3,4]. In this paper, we propose an algorithm for autonomous-mobile-robot navigation using the visual potential field. The visual potential field is computed from an image sequence captured by the camera mounted on the mobile robot, that is, the robot computes control force using appearances of the workspace which are observed as an image sequence by a camera mounted on the robot. Our algorithm enables the mobile robot to navigate without collision with obstacles. In a real environment, the payload of a mobile robot, for example, power supply, capacity of input devices and computing power, is restricted. Therefore, mobile robots are required to have simple mechanisms and devices [5,6]. We use an uncalibrated monocular camera as a sensor for obtaining information on the environment. This vision sensor is

a low-cost device that is easily mounted on mobile robots. Therefore, we use visual information for mobile robot navigation. On the other hand, the navigation problem of a mobile robot is to determine the robot motion at an arbitrary time [6]. This means that the robot computes its velocity using real-time data obtained in the current situation, such as a range data of obstacles, its position data in a workspace, and images captured by the camera mounted on the robot.

In a previous paper [6], we developed a featureless robot navigation method based on a planar area and an optical flow field [7,8] computed from a pair of successive images. A planar area in the world is called a dominant plane, and it corresponds to the largest part of an image. This method also yields local obstacle maps by detecting the dominant plane in images. We accepted the following four assumptions.

1. The ground plane is the planar area.
2. The camera mounted on the mobile robot is looking downward.
3. The robot observes the world using the camera mounted on itself for navigation.
4. The camera on the robot captures a sequence of images since the robot is moving.
5. The planar area occupies more than 1/2 of the image.

These assumptions are illustrated in Fig. 1. Therefore, if there are no obstacles around the robot, and since the robot does not touch the obstacles, the ground plane corresponds to the dominant plane in the image observed through the camera mounted on the mobile robot. From the dominant plane, we define the potential field on the image. Using the potential field, we proposed the algorithm for avoiding obstacles [9] and navigating a corridor environment [10] of the mobile robot.

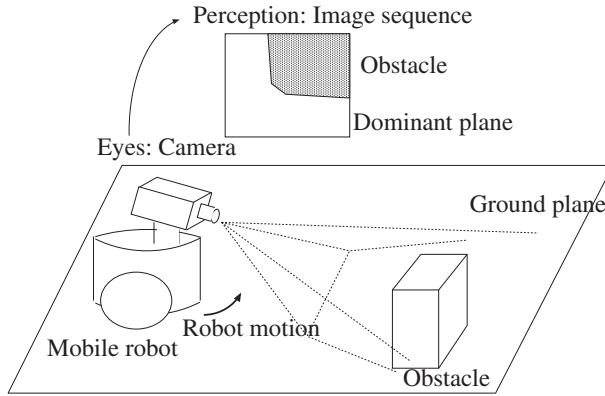
<sup>☆</sup> This paper has been recommended for acceptance by Matti Pietikainen.

\* Corresponding author at: Yaoi-cho 1-33, Inage-ku, Chiba 263-8522, Japan. Tel.: +81 43 290 35 40, +81 43 290 35 30; fax: +81 43 290 35 18.

E-mail address: [imiyai@faculty.chiba-u.jp](mailto:imiyai@faculty.chiba-u.jp) (A. Imaiya).

<sup>1</sup> Present address: Power and Industry Systems Research and Development Centers, Power System Company, Toshiba, Fuchu, Tokyo, Japan.





**Fig. 1.** Perception and cognition of motion and obstacles in workspace by an autonomous mobile robot. The mobile robot has a camera, which corresponds to eyes. The robot perceives an optical flow field from ego-motion.

Visual navigation of mobile robots is currently one of the most challenging problems in the fields of robot vision and computer vision [5,11–14]. The main task in this problem is to determine the control force of the mobile robot using images captured by the camera mounted on the mobile robot in a workspace. In this paper, we develop a navigation algorithm for an autonomous mobile robot using a visual potential field on images observed through the monocular camera mounted on the robot. The visual potential field on an image is an approximation of the projection of the potential field in the workspace to the image plane.

Our algorithm is summarised as follows:

1. Extract the safe area of the images, which corresponds to the area in which the robot can move without collision with obstacles.
2. Extract the obstacle region on the image as the complement of the safe area.
3. Compute the control force using the potential field computed from the obstacle area on the image.

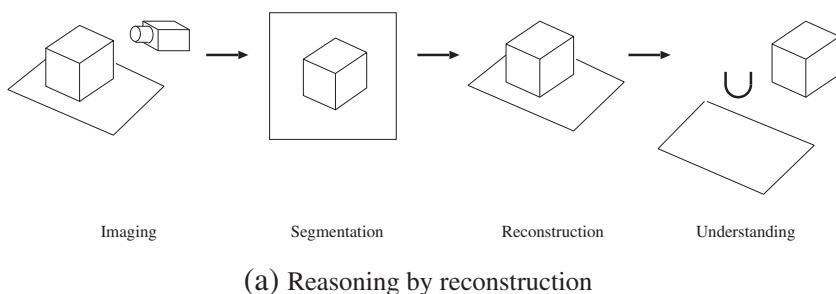
For the detection of the safe area for the robot to move, we detect the dominant plane from a sequence of images captured by the camera

mounted on the robot. The optical flow and homography of the ground plane allow us to detect the safe area and the obstacle area on the image [15–17]. Using the data in the image, we define the control force. This strategy was not adopted by Sobey [18] who also used optical flow and potential field. In this viewpoint, our method is image-based robot navigation. Furthermore, for the detection of the dominant plane on the image as a fundamental cue for the detection of the safe area in the space, we used full-resolution images observed by the camera mounted on the robot.

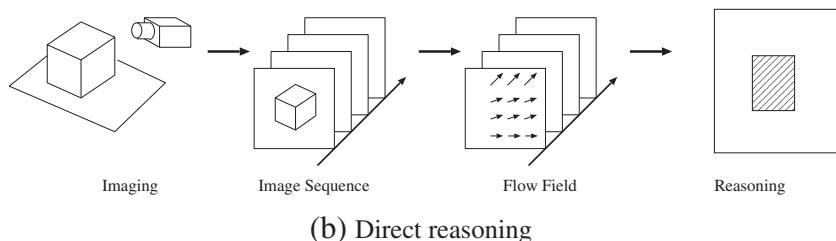
For the visual navigation, the robot is required to understand the three-dimensional configurations of the objects in the workspace using an image/image sequence captured by the camera mounted on the robot. To understand the three-dimensional configurations of geometrical objects and obstacles in the workspace, a typical method is the three-dimensional reconstruction of geometric objects from an image/image sequence. Since the mobile robot automatically captures an image sequence using the camera/cameras mounted on the robot, we develop an algorithm that directly achieves spatial reasoning to determine the navigation direction without any reconstruction processes from an image/image sequence. For direct reasoning, we use dominant planes on an image, since the dominant plane is a binary feature on an image. Figs. 2(a) and 2(b) show the typical steps for spatial reasoning for robot navigation with reconstruction, and our direct reasoning method using the dominant plane detected from an image sequence, respectively. For the spatial reasoning for two-dimensional geometrical information without reconstruction, we use the hierarchical property of the dominant planes [19].

The optical flow field [7,20–22] is the apparent motion in successive images captured by the moving camera. This is a fundamental feature for recognising environment information around the mobile robot. Additionally, the optical flow field is considered to be fundamental information for obstacle detection in the context of biological data processing [23]. Therefore, the use of optical flow is an appropriate method for the visual navigation of the mobile robot from the viewpoint of the affinity between robots and human beings.

In mobile robot navigation by the potential method, an artificial potential field defines the navigation force that guides the robot to the destination. Usually, an attraction to the destination and a repulsive from obstacles are generated to guide the robot to the destination

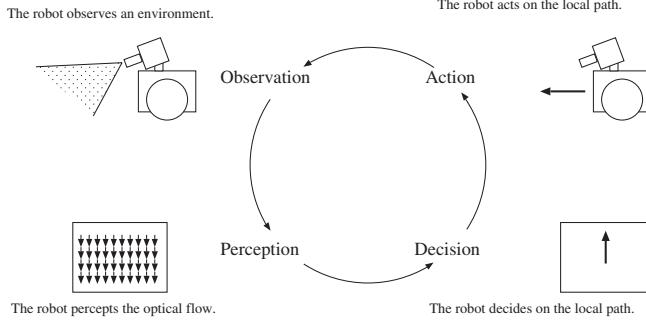


(a) Reasoning by reconstruction



(b) Direct reasoning

**Fig. 2.** Process for spatial reasoning. (a) shows typical steps for spatial reasoning for robot navigation with reconstruction. For spatial reasoning for navigation, the robot reconstructs spatial configurations of objects from an image/image sequence. (b) shows steps for spatial reasoning for robot navigation without reconstruction using the dominant plane detected from an image sequence. For the spatial reasoning for two-dimensional geometrical information without reconstruction, we use the hierarchical property of the dominant planes.



**Fig. 3.** Observation–perception–decision–action cycle for vision-based robot navigation. First, the mobile robot equipped with a camera observes an environment. Next, an optical flow field relative to the robot motion is computed from images obtained by the camera. The optical flow field is used to decide the local path. The robot moves in the direction of the computed local path.

without collision with the obstacles in the robot workspace [24,25]. Basically, these two guiding forces are generated from the terrain and the obstacle configuration in the workspace, which are usually pre-input to the robot. Then, we use the optical flow field as the guide field to the destination of the robot, since the flow vectors indicate the direction of robot motion to a local destination. Furthermore, using images captured by the camera mounted on the robot, the potential field for the repulsive force from obstacles is generated to avoid collision. The overview of our algorithm is shown in Fig. 3. The mobile robot dynamically computes the local path as the camera captures the images.

In Section 2, we review the related works. In Section 3, we overview our featureless method for detecting the dominant plane using optical flow fields. Furthermore, we introduce a pyramid-transform-based multiresolution-version of dominant plane detection algorithm. This multiresolution method guarantees robust separation of the dominant plane and the obstacle region on the image.

In Section 4, we present the algorithm for generating the visual potential field and the control force computed from the visual field. Furthermore, we define the guiding force to the destination for the

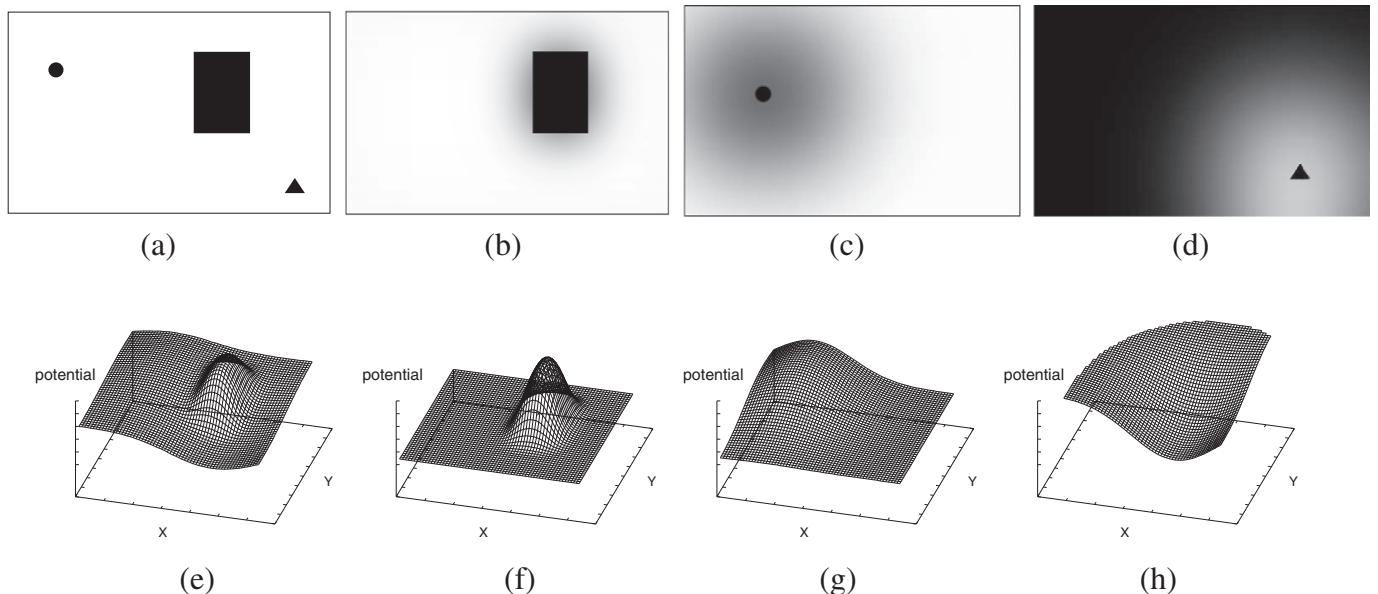
robot. In Section 5, we demonstrate some experimental results of robot navigation.

## 2. Related works

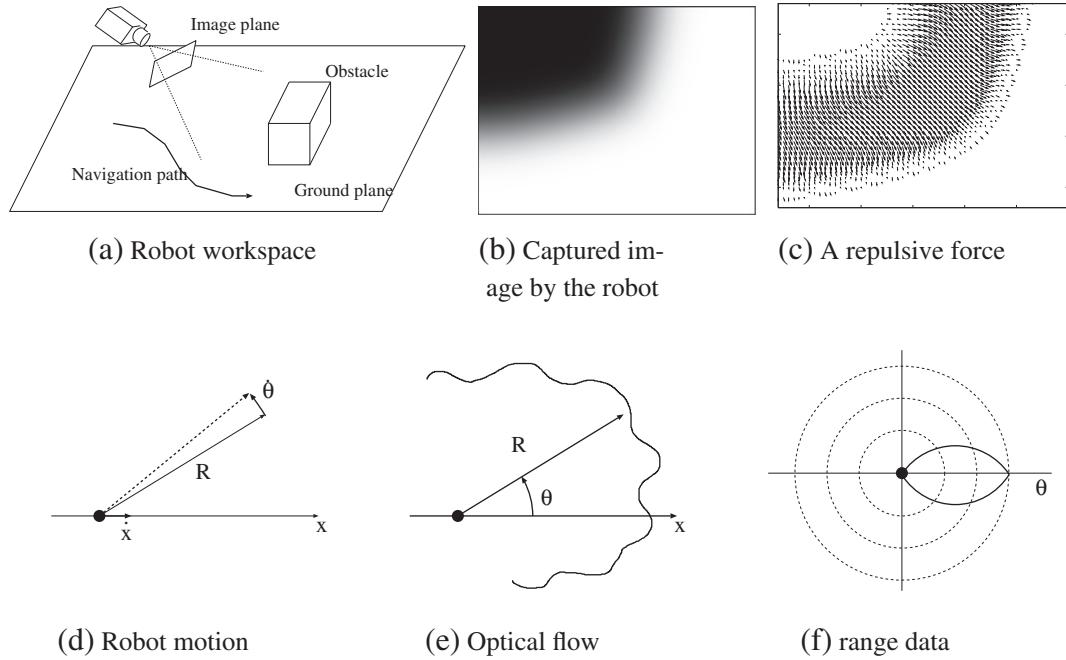
The navigation systems of nonholonomic mobile robots using vision systems [26–28] are being widely studied. A two-wheeled robot, which is addressed in our work, is a typical nonholonomic mobile system. In a previous work [25], visual information was used as the control parameters for tracking predetermined trajectories. Therefore, Tews et al. [25] implemented the visual servo system for the trajectory control of a nonholonomic mobile robot. In this paper, however, we use visual information for collision-free navigation by our robot without any predetermined robot trajectories.

The path-planning problem of the mobile robot in the configuration space is to determine the trajectory of the mobile robot. The trajectory is determined as the path from the start point to the destination point without collision with obstacles in the configuration space. The potential field method [1] yields a path from the start point to the destination point using the gradient field computed from the potential field derived from the map of the configuration of the robot workspace [29–31]. For path planning by the potential method, a robot is required to store the map of its workspace. The potential field method determines a navigation path using the repulsive force from obstacles in a map, the attractive force from the start and the repulsive force to the destination, as shown in Figs. 4(e), (f), (g) and (h) show the potential functions of the images in Fig. 4(a), (b), (c), and (d), respectively.

Accepting the visual potential field, the robot navigates by referring to a sequence of images without using the spatial configuration of obstacles. We introduce an algorithm for generating the visual potential field from an image sequence observed from a camera mounted on the robot. The visual potential field is computed on an image plane. Using the visual potential field and optical flow, we define a control force for avoiding the obstacles and guiding the robot to its destination, as shown in Figs. 5(a)–5(c). Figs. 5(a), 5(b), and 5(c) show the configuration of robot workspace, an image captured by the camera mounted on the mobile robot, and the repulsive force from the obstacle, respectively. Furthermore, Figs. 5(d), 5(e), and 5(f) show geometrical properties of insect inspired range potential. An insect understands the range



**Fig. 4.** Global potential method. (a) Configuration of the obstacle and starting point and destination. The circle, the triangle and the rectangle are the starting point, the destination and an obstacle, respectively. (b) Repulsive force from the obstacle. (c) Attractive force from the start. (d) Repulsive force to the destination. (e) Potential in the environment. The sum of the repulsive force and the attractive force. (f) Repulsive force from the obstacle. (g) Attractive force from the start. (h) Repulsive force to the destination.



**Fig. 5.** Appearance-based visual potential and insect-inspired range potential. (a), (b) and (c) show configuration of robot workspace, an image captured by the camera mounted on the mobile robot, and the repulsive force from the obstacle, respectively. (d), (e), and (f) show the relation between motion and optical flow, the range data in the eye-centred coordinate, and the range-based potential for control to the local destination, respectively.

data in the eye-centred coordinates using the optical flow field observed while it is flying around the environment. This range data observed by the moving camera (eye) yields the range-based potential field around the insect for navigation to the local destination. (d), (e), and (f) show the relation between motion and optical flow, the range data in the eye-centred coordinate, and the range-based potential for control to the local destination, respectively. In these figures,  $\dot{x}$ ,  $\theta$ , and  $R$  are the velocity of the insect, the angular velocity of objects, and the distance to the object, respectively. Then, the relationship  $\dot{\theta} = \dot{x} \sin \frac{\theta}{R}$  is satisfied.

Appearance-based object recognition is a stable and robust method of volumetric shape recognition method from a series of images [32]. The method is later extended to motion tracking such as hand gesture tracking and recognition [33,34] and lip-reading from an image sequence. Furthermore, the method is applied to the localisation of the mobile robot [35,36], that is, the method achieves the localisation of the robot from images without three-dimensional reconstruction of the spatial positions of the objects from an image sequence. Appearance-based method allows to obtain spatial features without reconstructing the spatial positions of objects in the space.

Using a vision system inspired by insects, navigation algorithms for the autonomous robot are proposed [37–39,18,40]. The insect-inspired vision for robot control uses simple information observed by the vision system mounted on the robot. Optical flows are features computed by the moving camera. Therefore, optical flow is a fundamental simple cue for controlling autonomous vehicles. Sobey [18] introduced a control strategy for robot navigation using optical flow and the potential field. His idea is described below.

This situation of a monocular-camera robot is similar to that of a flying or walking insect. The aim of this paper is to explore the motion strategies adopted by animals to maximise the safety of navigating with a single visual sensor. Central to this paper is the idea that motion must always be in the direction of an obstacle at a known range, because this is a safe way to proceed.

Sobey used the relationship between optical flow and range, which is possibly used by insects for the detection of obstacles in

the space for motion control,<sup>2</sup> as shown in Figs. 5(a)–5(b). This range in the space detected by the optical flow observed by the camera is used for the generation of the potential, which is used for the computation of the control force to avoid collision with obstacles. Furthermore, Sobey used low-resolution images to overcome the inaccuracy of range detection from optical flow.

### 3. Dominant-plane detection from optical flow

In this section, we briefly describe an algorithm for dominant-plane detection using an optical-flow field observed through an uncalibrated moving camera. The details of our algorithm are described in ref. [42]. First, we describe the computation of the optical flow. Next, we define the dominant plane in an image, and we describe the relationship between the optical flow and the dominant plane. Finally, we present the algorithm for detecting the dominant plane.

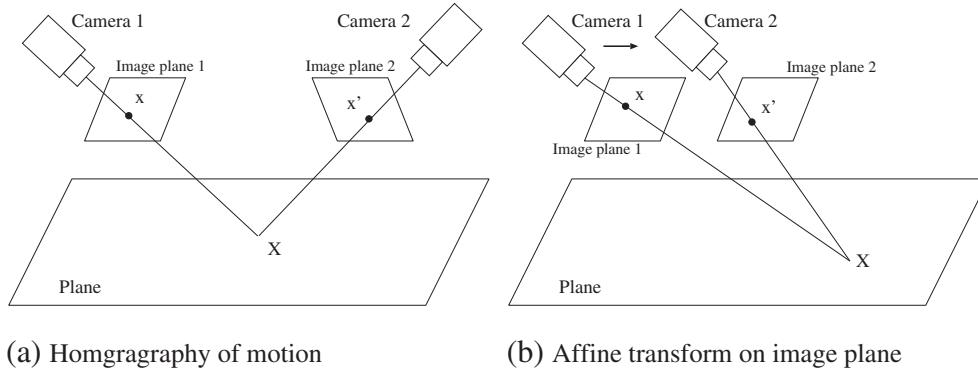
#### 3.1. Definition of dominant plane

We define the dominant plane in an image.

**Definition 1.** The dominant plane as the planar area in the robot workspace corresponding to the largest part of an image or at least the half of an image.

These assumptions are illustrated in Fig. 1. Therefore, if there are no obstacles around the robot, and since the robot does not touch the obstacles, the ground plane corresponds to the dominant plane in the image observed through the camera mounted on the mobile robot. Assuming that the dominant plane in the image corresponds to the ground plane on which the robot moves, the detection of the dominant plane enables the robot to detect the feasible region for

<sup>2</sup> Considering a camera moving along its optical axis with a velocity  $\dot{x}$ , the angular velocity of objects across the image plane of a moving camera can be expressed as  $\dot{\theta} = \dot{x} \sin \frac{\theta}{R}$ , where  $\theta$  is the angular position of an object on the image plane with respect to the direction of motion,  $\dot{\theta}$  is the angular velocity of the object image, and  $R$  is the distance to the object.



**Fig. 6.** Approximation of dominant plane motion by affine transformation. (a) Homography is expressed by  $\xi' = H\xi$ . (b) Affine transformation is expressed by  $x' = Ax$ .

navigation in its workspace. In this figure, the white region in the captured image is the dominant plane, which corresponds to the ground plane in the robot workspace.

Since we assume that the optical axis of the camera mounted on the robot intersects with the ground plane as shown in Fig. 1, when the robot moves on the ground plane. Therefore, the dominant plane possesses the following property.

**Property 1.** *The dominant plane that corresponds to the ground floor is incident to the bottom line of the image-frame.*

From Definition 1 and item 5 of the assumptions, an image captured by the camera mounted on the robot has the following property.

**Property 2.** *The appearance area of the obstacles on an image is less than half of an image.*

From this property, the distribution of obstacles is space in the workspace.

### 3.2. Optical flow computation

Setting  $I(x,y,t)$  and  $\dot{x} = (\dot{x}, \dot{y})^\top = (\frac{dx}{dt}, \frac{dy}{dt})^\top$  to be the time-varying grey-scale-valued image at time  $t$  and the optical flow  $\dot{x}$  of the point  $x = (x, y)^\top$ , respectively, the optical flow  $\dot{x}$  at each point  $x$  satisfies

$$\frac{\partial I}{\partial x} \dot{x} + \frac{\partial I}{\partial y} \dot{y} + \frac{\partial I}{\partial t} = 0. \quad (1)$$

The computation of  $\dot{x}$  from  $I(x,y,t)$  is an ill-posed problem. Therefore, additional constraints are required to compute  $\dot{x}$ . The most commonly used constraints are those indicated by Horn and Schunck [20], Lucas and Kanade [21], and Nagel and Enkelmann [22].

### 3.3. Optical flow and homography

In this section, we present the geometrical properties of the optical flow on the dominant plane.

Setting  $H$  to be a  $3 \times 3$  matrix [44], the homography between two images of a planar surface can be expressed as

$$\xi' = H\xi, \quad (2)$$

where  $\xi = (x, y, 1)^\top$  and  $\xi' = (x', y', 1)^\top$  are the homogeneous coordinates of corresponding points in two successive images. Assuming that the camera displacement is small, the matrix  $H$  can be approximated by affine transformations. These geometrical and mathematical assumptions are valid when the camera is mounted on a mobile robot moving

on the dominant plane. Therefore, the corresponding points  $x = (x, y)^\top$  and  $x' = (x', y')^\top$  on the dominant plane are related by

$$x' = Ax + b, \quad (3)$$

where  $A$  and  $b$  are a  $2 \times 2$  affine-coefficient matrix and a two-dimensional vector, respectively, which are approximations of  $H$ . Fig. 6 shows the homography transform of images caused by the translation of the camera. (a) shows the camera motion and (b) shows the homography transform on the image plane caused by the robot motion.

### 3.4. Optical flow on dominant plane

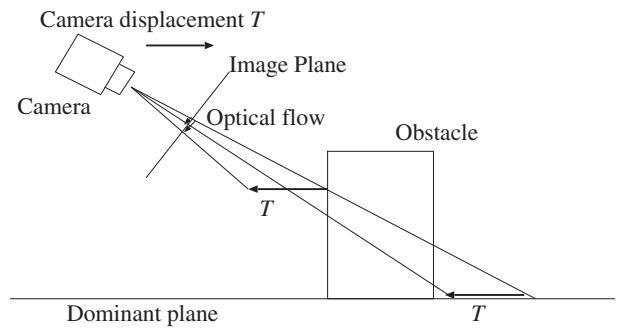
We can estimate the affine coefficients using the RANSAC-based algorithm below [42,43].

Using the estimated affine coefficients, we can estimate the optical flow on the dominant plane  $\hat{x} = (\hat{x}, \hat{y})^\top$  where

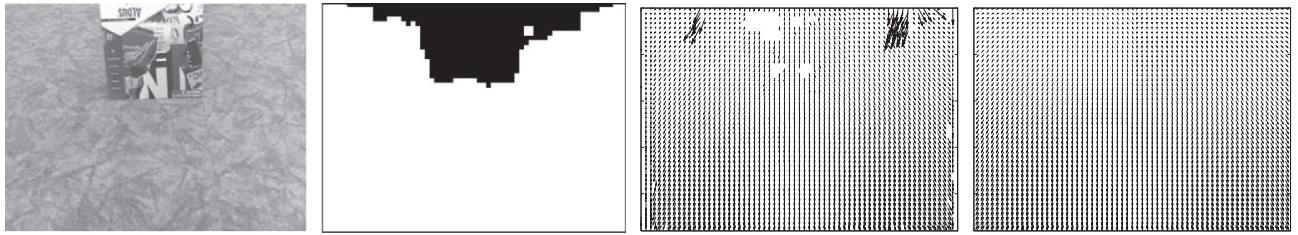
$$\hat{x} = \{Ax + b\} - x = x', \quad (4)$$

for all points  $x = (x, y)^\top$  in the image. We call  $\hat{x}$  the *planar flow* and  $\hat{u}(x, y, t)$  the *planar-flow field* at time  $t$ , which is a set of planar flows  $\hat{x}$  computed for all pixels in an image.

At each point on the dominant plane, the planar flow and the flow vector of the planar-flow field coincide. And from the definition, the dominant plane is the area on the plane which occupies at most half of the imaging plane. Therefore, we have the Algorithm 2 for the estimation of the affine coefficients which describe the homography transform between a pair of two successive image frames in the image sequence, using the RANSAC-based algorithm [41–44].



**Fig. 7.** Difference in optical flow between dominant plane and obstacles. If the camera moves a distance  $T$  approximately parallel to the dominant plane, the optical flow vector on the obstacle and on the dominant plane areas are the same distance  $T$ . However, they differ at the same time. Therefore, the camera can observe the difference in the optical flow vectors between the dominant plane and obstacles.



**Fig. 8.** Examples of dominant plane and planar flow field. Starting from the left: captured image  $I(x, y, t)$ , detected dominant plane  $d(x, y, t)$ , optical flow field  $u(x, y, t)$ , and planar flow field  $\hat{u}(x, y, t)$ .

#### Algorithm 1. Affine coefficient estimation

---

```

1 Repeat
2   Randomly select three points  $x_i, x_j, \xi_k$  and  $x'_i, x'_j, x'_k$  from  $\{x\}$  and  $\{x'\}$ , respectively;
3   Estimate  $A$  and  $b$  in Eq. (3) from  $x_i, x_j, x_k$  and  $x'_i, x'_j, x'_k$ ;
4   Compute the planar flow field  $\hat{x}$  using Eq. (4);
5 until  $x \neq \hat{x}$  for more than the half of the image.

```

---

#### 3.5. Dominant-plane detection

If an obstacle exists in front of the robot, the planar flow on the image plane differs from the optical flow on the image plane, as shown in Fig. 7. Since the planar-flow vector  $\hat{x}$  is equal to the optical-flow vector  $\dot{x}$  on the dominant plane, we use the difference between these two flows to detect the dominant plane. We set  $\varepsilon$  to be the tolerance of the difference between the optical-flow vector and the planar-flow vector.

If an obstacle exists in front of the robot, the planar flow on the image plane differs from the optical flow on the image plane. Since the planar flow vector  $\hat{x}$  is equal to the optical flow vector  $\dot{x}$  on the dominant plane, we use the difference between these two flows to detect the dominant plane. We set  $\varepsilon$  to be the tolerance of the difference between the optical flow vector and the planar flow vector. Therefore, if the inequality

$$|\dot{x} - \hat{x}| < \varepsilon, \text{ for } \hat{x} = (Ax + b) - x, \quad \nabla I^\top \dot{x} + \partial_t I = 0 \quad (5)$$

is satisfied, we accept point  $x$  as a point on the dominant plane [42]. Then, the image is represented as a binary image containing the

dominant-plane region and the obstacle region. Therefore, we set  $d(x, y, t)$  to be the dominant plane, where

$$d(x, y, t) = \begin{cases} 255, & \text{if } x \text{ is on the dominant plane.} \\ 0, & \text{if } x \text{ is on the obstacle area} \end{cases}$$

We call  $d(x, y, t)$  the dominant-plane map. Our algorithm is summarised as follows:

#### Algorithm 2. Dominant-plane detection

---

```

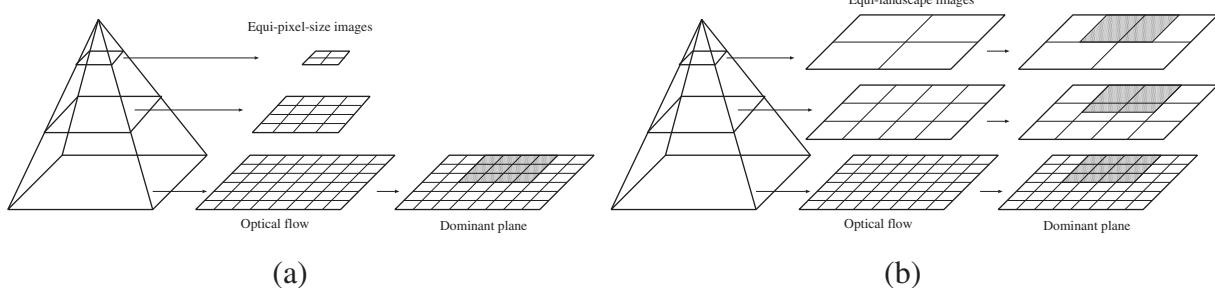
1 Compute optical-flow field  $u(x, y, t)$  from two successive images;
2 Repeat
3   Compute optical-flow field  $u(x, y, t)$  from two successive images;
4   Compute affine coefficients in Eq. (3) from three randomly selected
      points;
5   Estimate planar-flow field  $\hat{u}(x, y, t)$  from affine coefficients;
6   Match the computed optical flow field  $u(x, y, t)$  and estimated
      planar-flow field  $\hat{u}(x, y, t)$  using Eq. (5);
7   if  $|\dot{x} - \hat{x}| < \varepsilon$  then assign these points as the dominant plane;
8   if the dominant plane occupies more than the half of the image then
      output the dominant plane  $d(x, y, t)$  as a binary image;
9 until predetermined number of times N.

```

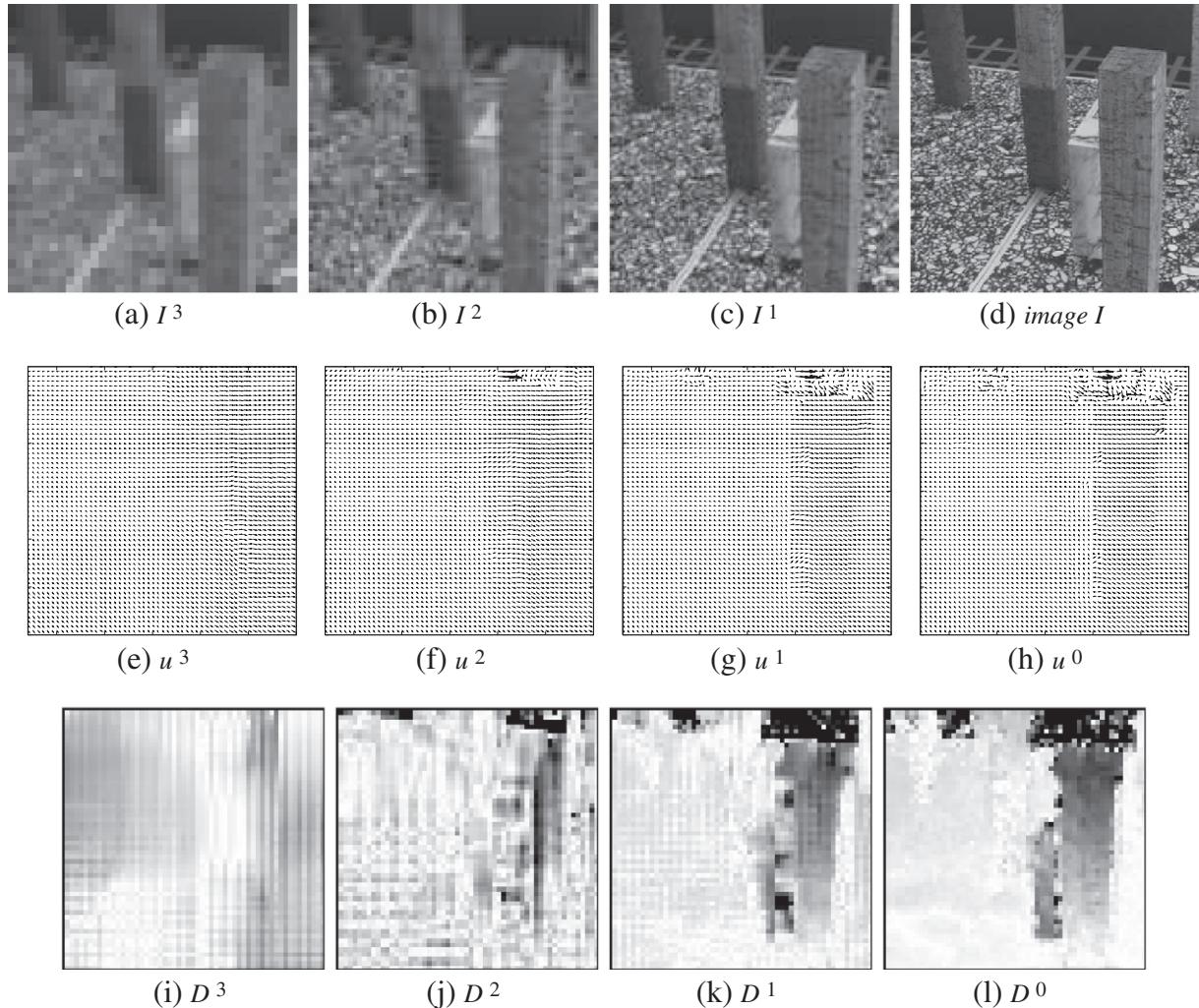
---

If the image size is  $320 \times 240$ , the random sampling procedure in Algorithm 2 is iterated at most  $\frac{1}{2} \times 320 \times 240$  times. Moreover, in the experiments, we set the error-bound  $\varepsilon$  as  $2 \times 10^{-2}$ .

Fig. 8 (top) shows examples of the captured image  $I(x, y, t)$  and the detected dominant plane  $d(x, y, t)$ . Fig. 8 (bottom) shows the optical-flow field  $u(x, y, t)$  and the planar-flow field  $\hat{u}(x, y, t)$ .



**Fig. 9.** Hierarchical extraction of the dominant planes. (a) Dominant plane detection using the pyramid transform with equi-pixel size in each layer. (b) Hierarchical dominant plane detection using the pyramid transform with equi-landscape size in each layer. The upper layer images and lower layer images in this hierarchy of dominant planes provide global features and precise features for robot navigation, respectively. The global features define the feasible region for controlling and topological configuration of the obstacles, and precise features derive geometrical configuration for real-time map generation from visual information captured by a camera mounted on the robot.

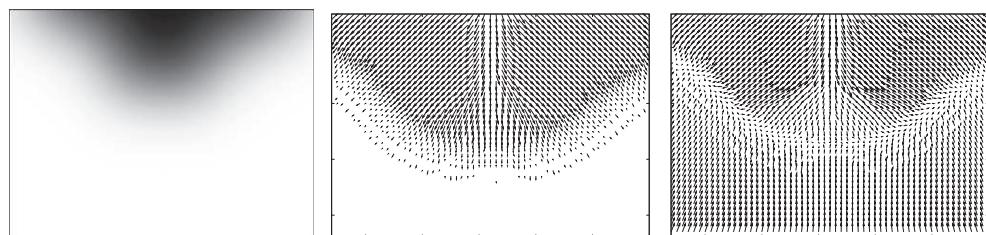


**Fig. 10.** Detected obstacle at each layer. Pyramidal representation of the marbled-block images in a simulated environment. The layer optical flow fields.  $u^i$  is the optical flow field from  $I^i$ .

In the top horizontal line of the optical-flow-field image in the bottom-left image of Fig. 8, outliers of flow vectors appear as the computational results. However, the algorithm extracts the dominant plane as the white region of the top-right image comparing the planar flow and the optical flow field in the bottom-right image. The algorithm finally yields the top-right image as a binary image. Since the dominant plane is a binary feature, the algorithm is robust against the outliers which are derived in the process of optical-flow computation [42].

### 3.6. Dominant planes in layer optical flow fields

Our algorithm uses the pyramid-transform-based multiresolution optical flow computation, which detects hierarchy optical flow fields. If the algorithm is processed at each layer, we detect obstacles in an image sequence, from the optical flow fields  $u^l(x, y, t)$  at the layer  $l$  for  $l=0,\cdots,L$  in the pyramid hierarchy of images. This multiresolution method allows the robot to detect robustly the obstacle region. The obstacle region is an essential part for the



**Fig. 11.** Examples of potential field  $p(x, t)$  computed from the examples in Fig. 8. Starting from the left: the dominant plane after Gaussian operation  $G(x) \cdot d(x, t)$ , gradient vector field  $g(x, t)$  as repulsive force from obstacles, and visual potential field  $p(x, t)$ .

computation of the control force. On the feature propagation steps from the coarse grid to the fine grid in the image pyramid hierarchy, this multiresolution method removes unstable effects caused by small structural perturbations in images such as lines and small markers on the floor. Fig. 9 shows the process of the hierarchical extraction of the dominant planes.

Setting DPD to be the algorithm for detecting the dominant plane  $\mathbf{d}(x, y, t)$  from the optical flow  $\mathbf{u}(x, y, t)$ , the hierarchical dominant plane detection algorithm is summarised as follows:

**Algorithm 3.** Hierarchical dominant plane detection algorithm

---

```

Data :  $I_t^l, I_{t+1}^l, \dots, 0 \leq l \leq$  the maximum of the layers;
Result :  $d(x, y, t)^l$ 
1 := the maximum of the layers ;
2 while  $1 \neq 0$ , do
3    $u_t^l := u(I_t^l, I_{t+1}^l);$ 
4    $d(x, y, t)^l := \text{DPD}(u_t^l);$ 
5    $I_{t+1}^{l-1} := w(I_{t+1}^{l-1}, u_t^l);$ 
6    $l := l - 1;$ 
7 end

```

---

We set  $O^l$  to be the obstacle region on the  $l$ -th layer. From the geometrical properties of the pyramid transform, as shown in Fig. 9, for  $O^l$  we have the inclusive relation

$$O^0 \supset O^1 \supset \dots \supset O^L \quad (6)$$

as the hierarchical expression of obstacles regions in layers. This relation implies the inclusive relation between the dominant planes

$$D^0 \supset D^1 \supset \dots \supset D^L \quad (7)$$

for the dominant plane  $D^l$ , since  $D^l \cap O^l = \emptyset$  and  $O^l \cup D^l = R$ , where  $R$  is the total region of the image. These relations imply that a pair  $C^l = (D^l, O^l)$  shows global and local configuration in the workspace for a larger and a smaller  $l$ , respectively. This hierarchical relation is automatically detected from a pyramid-based hierarchical expression of images for optical flow computation. The system uses selectively  $C^l$  for navigation and spatial perception.

The marbled-block image sequence and captured images in a real environment are used for the experiment. Fig. 10 shows the marbled-block images at each layer. The computed optical flow fields at each layer from each image, and the detected obstacle at each layer. In this figure, the black and white region indicate the obstacle and dominant plane, respectively. This example shows that in each layer the obstacle-regions are detected. Therefore, the algorithm detects the global configuration of obstacles from higher layer images,

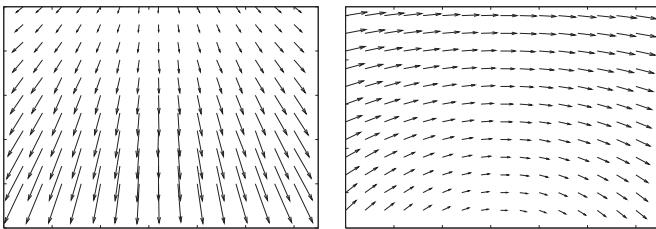


Fig. 12. Optical flow fields as attractive force fields. (Left) Translational planar flow field  $u_t(x)$ . (Right) Rotational planar flow field  $u_r(x)$ .

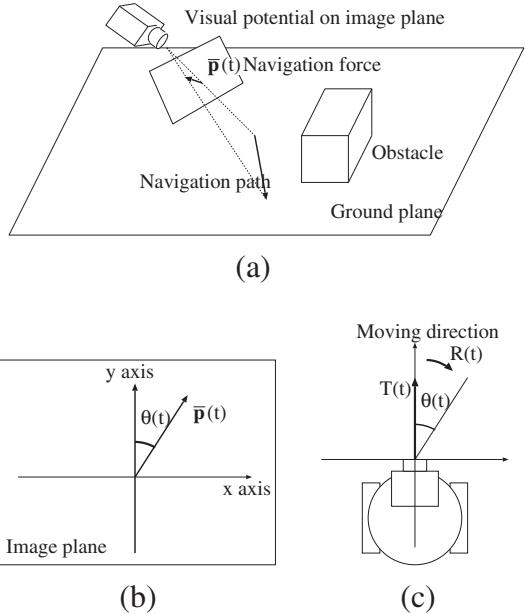


Fig. 13. Navigation from potential field. (a) Navigation force  $\bar{p}(t)$ . (b) Angle between navigation force  $\bar{p}(t)$  and y axis is  $(t)$ . (c) Robot displacement  $T(t)$  and rotation angle  $R(t)$  at time  $t$  determined using  $\theta(t)$ .

though the lower layer images allows us to detect the detailed configuration of obstacles.

#### 4. Determination of robot motion using visual potential

In this section, we describe the algorithm used for the determination of robot motion from the dominant plane image  $\mathbf{d}(x, t)$  and the planar flow field  $\hat{\mathbf{u}}(x, t)$ .

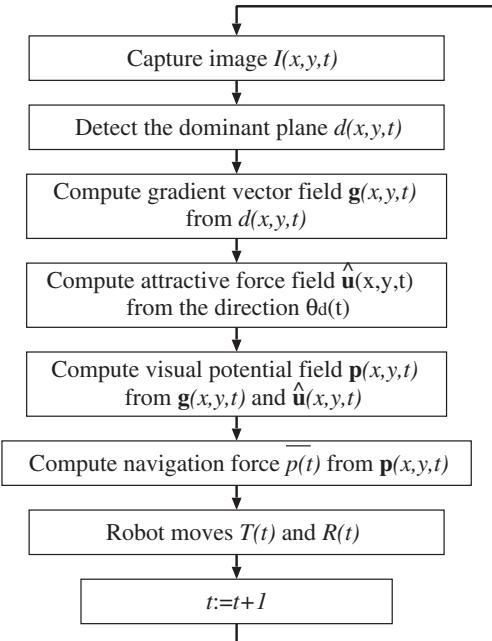
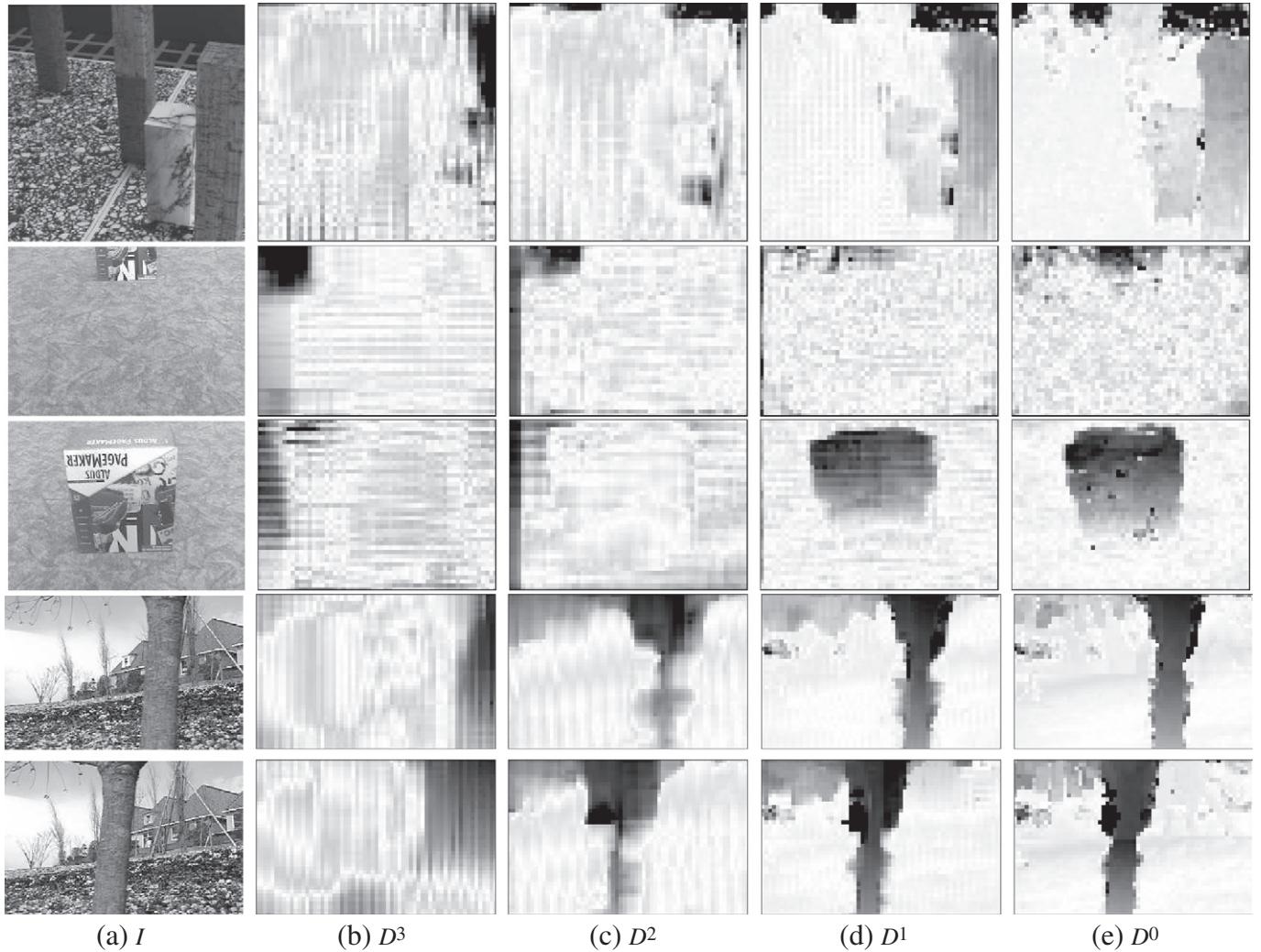


Fig. 14. Closed loop for autonomous robot motion. Computations of  $\mathbf{g}(x, t)$ ,  $\mathbf{p}(x, t)$ , and  $T(t)$  and  $R(t)$  are described in Sections 3.1, 3.2, and 3.3, respectively. The algorithm for the detection of dominant plane  $d$  is described in Section 2.



**Fig. 15.** Experimental results. (a) Original image. (b) (c) (d) and (e) are detected obstacles at the layers 3, 2, 1, and 0, respectively.

#### 4.1. Gradient vector of image sequence

The robot moves on the dominant plane without collision with obstacles. Therefore, we generate an artificial repulsive force from the obstacle area in image  $\mathbf{d}(\mathbf{x}, t)$  using the gradient vector field. The potential field on the image is an approximation of the projection of the potential field in the workspace to the image plane. Therefore, we use the gradient vector of the dominant plane as the repulsive force from obstacles.

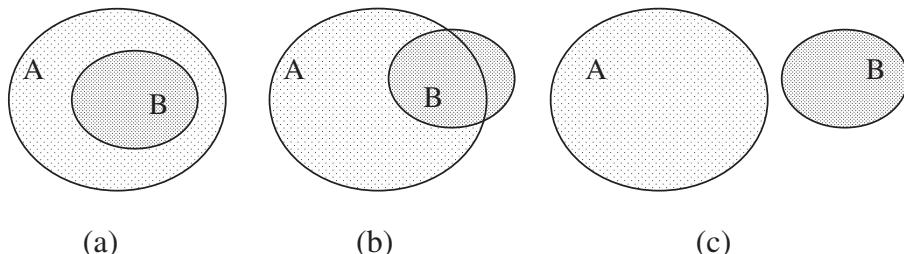
Since the dominant plane map  $\mathbf{d}(\mathbf{x}, t)$  is a binary image sequence, the direct computation of the gradient is numerically unstable and unrobust. Therefore, as a preprocess to the computation of the gradient,

we smooth the dominant plane map  $\mathbf{d}(\mathbf{x}, t)$  by convolution with Gaussian  $G_*$ , that [9,24,29] is, we adopt  $\mathbf{g}$  such that

$$\mathbf{g}(\mathbf{x}, t) = \nabla(G * d(\mathbf{x}, t)) = \left( \begin{array}{l} \frac{\partial}{\partial x}(G * d(\mathbf{x}, y, t)) \\ \frac{\partial}{\partial y}(G * d(\mathbf{x}, y, t)) \end{array} \right) \quad (8)$$

as the potential generated by obstacles. Here, for a 2D Gaussian,

$$G(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}. \quad (9)$$



**Fig. 16.** Geometric expressions of inclusive relations. (a)  $A \supset B$ . (b)  $(A \cap B) \subset A$ . (c)  $A \cap B = \emptyset$ .

$G \cdot d(\mathbf{x}, t)$  is given as

$$G(x) * d(x, t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G(u-x, v-y) d(x, y, t) du dv. \quad (10)$$

In this paper, we select the parameter  $\sigma$  to be half the image size. An example of the gradient vector field is shown in Fig. 11 (middle).

#### 4.2. Optical flow as attractive force

From the geometric properties of the flow field and the potential, we define potential field  $p(\mathbf{x}, t)$  as

$$p(\mathbf{x}, t) = \begin{cases} g(\mathbf{x}, t) - \hat{u}(\mathbf{x}, t), & \text{if } d(\mathbf{x}, t) = 255 \\ g(\mathbf{x}, t), & \text{otherwise,} \end{cases} \quad (11)$$

where  $\hat{u}(\mathbf{x}, t)$  is the attractive force field to the destination.

In Eq. (11), the gradient vector field  $\mathbf{g}(\mathbf{x}, t)$  is a repulsive force from obstacles and the planar flow field  $\hat{u}(\mathbf{x}, t)$  is an artificial attractive force to the destination. Since the planar flow  $\hat{u}(\mathbf{x}, t)$  represents the camera motion, the sum of the sign-inversed planar flow field  $-\hat{u}(\mathbf{x}, t)$  and the gradient vector field  $\mathbf{g}(\mathbf{x}, t)$  is the potential field  $\mathbf{p}(\mathbf{x}, t)$ . However, in the obstacle area in the image, the planar flow field  $\hat{u}(\mathbf{x}, t)$  is set to zero, since the planar flow field represents the dominant plane motion.

#### 4.3. Optical flow as guidance force

If obstacles do not exist in the image, Eq. (11) becomes

$$p(\mathbf{x}, t) = -\hat{u}(\mathbf{x}, t). \quad (12)$$

Then, the robot moves according to the planar flow field. An example of the potential field  $\mathbf{p}(\mathbf{x}, t)$  computed from the examples in Fig. 8 is shown in Fig. 11 (right).

This relation suggests us to use optical flow as the guidance force to the destination. Therefore, using the visual potential, we introduce a mechanism for guiding the robot to the destination. Setting  $\theta_d(t)$  to be the direction angle between the destination and the robot direction, we define the guiding force to the destination as the mixture of

$$\hat{u}_d(\mathbf{x}, t) = \hat{u}_t(\mathbf{x}) \cos \theta_d(t) + \hat{u}_r(\mathbf{x}) \sin \theta_d(t), \quad (13)$$

where  $\hat{u}_t(\mathbf{x})$  and  $\hat{u}_r(\mathbf{x})$  are the pure translational and rotational planar flow fields, respectively, as shown in Fig. 12.

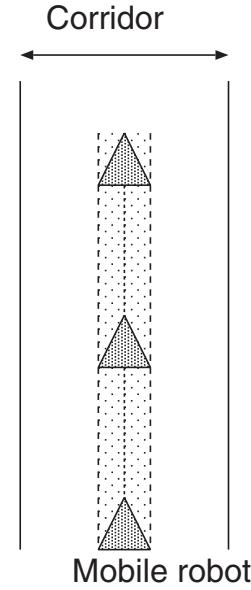
The optical flow  $\hat{u}_d(\mathbf{x})$  is the desired flow generated from images captured by the camera mounted on the robot, when the robot is moving towards the destination in the workspace without obstacles. Pure optical flow fields  $\mathbf{u}_t$  and  $\mathbf{u}_r$  are computed when the velocity of the robot is constant. Therefore, we use the control force

$$p_d(\mathbf{x}, t) = \begin{cases} g(\mathbf{x}, t) - \hat{u}_d(\mathbf{x}, t), & \text{if } d(\mathbf{x}, t) = 225 \\ g(\mathbf{x}, t), & \text{otherwise,} \end{cases} \quad (14)$$

for the guiding the robot to the destination.

**Table 1**  
Experimental results of probabilistic inclusive relation for three image sequences.

Layer	$P(D^0 \subset D^1)$	$P(D^1 \subset D^2)$	$P(D^2 \subset D^3)$
Marbled block	0.980	0.738	0.700
Flower garden	0.878	0.921	0.508
Mobile robot	0.896	0.637	0.697



**Fig. 17.** Corridor centreing for mobile robot navigation. The grey triangle is the mobile robot, that moves in the corridor without collision with walls.

#### 4.4. Navigation by potential field

We define control force  $\bar{p}(t)$ , using the average of potential field  $p_d(\mathbf{x}, t)$ , as

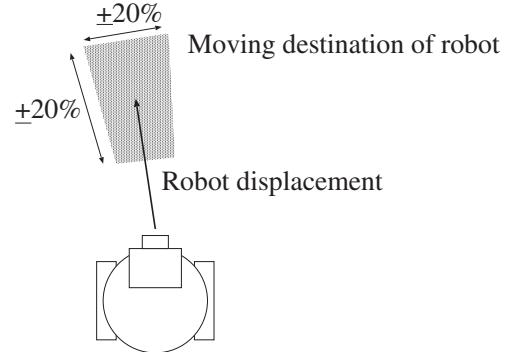
$$\bar{p}(t) = \frac{1}{|A|} \int_{(x,y)^T \in A} p_d(\mathbf{x}, t) dx dy, \quad (15)$$

where  $|A|$  is the size of the region of interest in the image captured by the camera mounted on the mobile robot.

Since we apply the control force  $\bar{p}(t)$  to the nonholonomic mobile robot, we require the control force to be transformed into the translational velocity and rotational velocity. We determine the ratio between the translational and rotational velocities using the angle of control force  $\bar{p}(t)$ .

We assume that the camera is attached to the front of the mobile robot. Therefore, we set parameter  $\theta(t)$  to be the angle between the navigation force  $\bar{p}(t)$  and  $y$  axis  $y = (0, 1)^T$  of the image, which is the forward direction of the mobile robot, as shown in Fig. 13. That is,

$$\theta(t) = \arccos \frac{\langle \bar{p}(t), y \rangle}{|\bar{p}(t)| |y|}. \quad (16)$$



**Fig. 18.** Displacement and rotation of the mobile robot have errors. Error of 20% was added to both robot displacement  $T(t)$  and rotation  $R(t)$ .



**Fig. 19.** Example of image added with random noise. Left: original image of dominant plane. Right: noisy image of dominant plane.

We define the robot translational velocity  $T(t)$  and the rotational velocity  $R(t)$  at time  $t$  as

$$T(t) = T_m \cos\theta(t), \quad R(t) = R_m \sin\theta(t), \quad (17)$$

where  $T_m$  and  $R_m$  are the maximum translational and rotational velocities, respectively, of the mobile robot between time  $t$  and  $t+1$ . Setting  $X(t) = (X(t), Y(t))^\top$  to be the position of the robot at time  $t$ , from Eq. (17), we have the relations

$$\sqrt{\dot{X}(t)^2 + \dot{Y}(t)^2} = T(t), \quad \tan^{-1} \frac{\dot{Y}(t)}{\dot{X}(t)} = R(t). \quad (18)$$

Therefore, we have the control law

$$\dot{X}(t) = T(t) \cos R(t), \quad (19)$$

$$\dot{Y}(t) = T(t) \sin R(t). \quad (20)$$

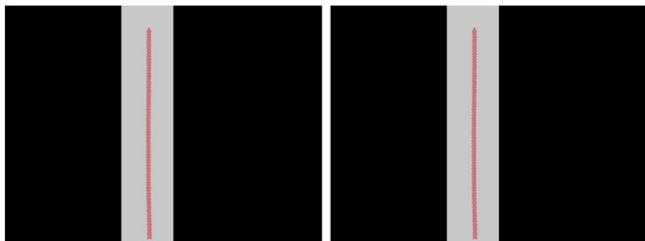
The algorithm for computing robot motion  $T(t)$  and  $R(t)$  and the closed loop for autonomous robot motion is shown in Fig. 14.

## 5. Experimental results

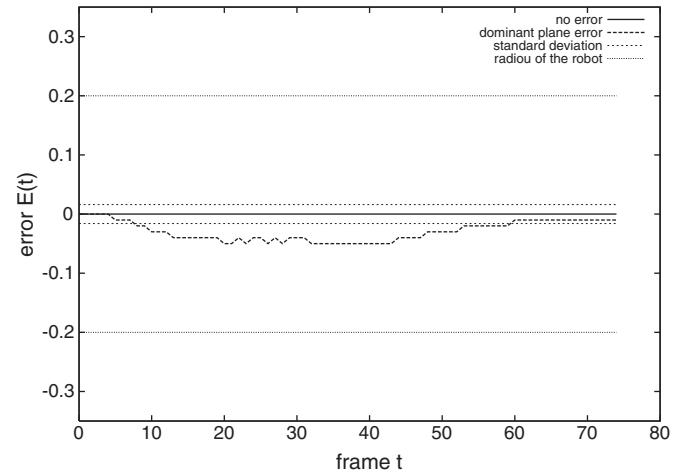
In this section, we show some experimental results for autonomous robot navigation in a synthetic environment and a real environment. We experimented with three kinds of navigation of a mobile robot: corridor centreing, progress to the destination, and homing. If the mobile robot can perform corridor centreing and progress to the destination, it can perform homing, because the homing problem is expressed as progress to the home position without collision with obstacles.

### 5.1. Layer dominant plane detection

We set the maximum layer  $L=3$ . For the visual representation of the results of obstacle detection, the value of  $d(x, y) = |u - \hat{u}|$  is



**Fig. 20.** Overhead view of the result of corridor centreing in the straight pathway. Black regions are walls, and the grey region is the corridor within which the robot is assumed to move. The triangles are the trajectory of the mobile robot. The mobile robot started at the bottom of the image. Left: result computed from original images. Right: result computed from noisy images.



**Fig. 21.** Numerical expression of the trajectories. The solid line is the trajectory computed from original images. The dashed line is the trajectory computed from noisy images. The width of the robot is 0.2 unit, and the standard deviation of the difference between these two trajectories along the time axis is 0.0168.

normalised in the range from 0 to 255. The image of the detected obstacle  $D^l(x, y)$  at the  $l$ -th layer is defined as

$$D^l(x, y) = \frac{d^l(x, y) \times 255}{\max(d^l(x, y))}, \quad (21)$$

where  $d^l(x, y)$  is  $d(x, y)$  at the  $l$ -th layer.

For a pair of sets  $A$  and  $B$  such that  $|A| \geq |B|$ , where  $|A|$  is the measure of the set  $A$ , we define

$$P(A \supseteq B) = \frac{|A \cap B|}{|B|}. \quad (22)$$

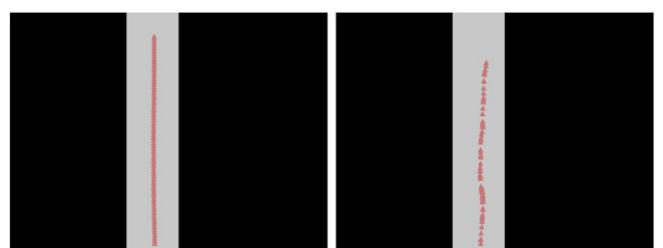
$P(A \supseteq B)$  satisfies the relations

$$\begin{aligned} P(A \supseteq B) &= 0 \text{ if } A \cap B = \emptyset, \\ P(A \supseteq B) &= 1 \text{ if } A \supseteq B, \\ 0 < P(A \supseteq B) &< 1 \text{ if otherwise.} \end{aligned} \quad (23)$$

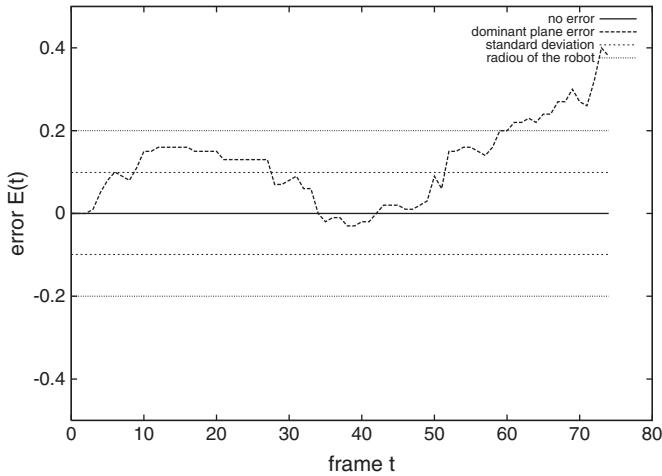
Fig. 16 shows geometric relations  $A$  and  $B$ . Therefore, we can use  $P(A \supseteq B)$  as a probabilistic inclusive relation of a pair of sets in  $\mathbb{R}^n$ .

**Definition 1.** If  $1 \geq P(A \subset B) \geq 1 - \varepsilon$ , we call  $A$  is included in  $B$  with the dissimilarity  $\varepsilon_i$ . If  $A$  is included in  $B$  with the dissimilarity  $\varepsilon$  we express

$$A \sqsupseteq_\varepsilon B. \quad (24)$$



**Fig. 22.** Overhead view of the result of corridor centreing. Black regions are walls, and the grey region is the corridor within which the robot is assumed to move. The triangles are the trajectory of the mobile robot. The mobile robot started at the bottom of the image. Left: result of the trajectory of the robot without any noise. Right: result of the trajectory of the robot with the noise in the robot position.



**Fig. 23.** Numerical expression of the trajectories. The solid line is the trajectory of the robot without any noise. The dashed line is the trajectory of the robot with noise in the robot position. The width of the robot is 0.2 unit, and the standard deviation of the difference between these two trajectories along the time axis is 0.099.

Using this probabilistic inclusive relation defined by Eq. (24), we examine that the dominant plane in the lower layer is included in the higher level for most parts of the image. Setting

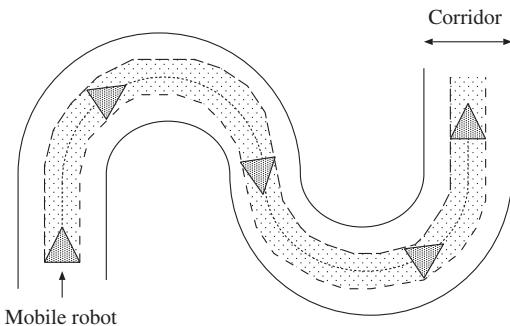
$$P(D_i \subset D_{i+1}) = \frac{|D_i \cap D_{i+1}|}{|D_{i+1}|} = \frac{|D_i \cap D_{i+1}|}{|D_{i+1}|} \quad (25)$$

the  $P(D_i \subset D_{i+1})$  satisfies the relation  $1 \geq P(D_i \subset D_{i+1}) \geq 1 - \varepsilon_i$ , for  $0 < \varepsilon_i \ll 1$ . If the  $D_{i+1}$  on the upper or coarse layer  $L_{i+1}$  is completely included in the  $D_i$  on lower or finer layer  $L_i$ , this measure falls in to 1. Conversely if the  $D_{i+1}$  on the upper or coarse layer  $L_{i+1}$  is completely exclusive to the  $D_i$  on lower or finer layer  $L_i$ , this measure falls into 0. This property comes from the property of  $P(D_i \subset D_{i+1})$  that the term over the bar is the area measure of the common region between  $D_i$  and  $D_{i+1}$ . Using this expression, the relation of  $\{D_i\}_{i=0}^N$  is expressed

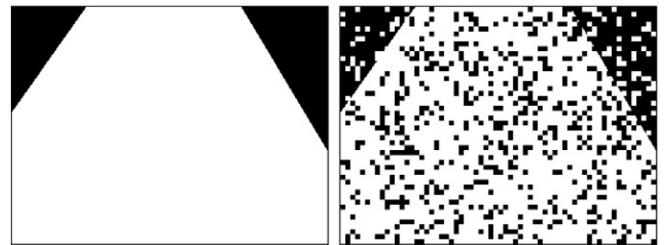
$$D_0 \supseteq_{\varepsilon} D_1 \supseteq_{\varepsilon} D_2 \supseteq_{\varepsilon} \cdots \supseteq_{\varepsilon} D_L \quad (26)$$

for  $\varepsilon_i \leq \varepsilon$ .

We calculate the value of Eq. (26) for three image sequences in Fig. 15. In order to calculate the probabilistic inclusive relation of the dominant planes, we operate the threshold processing to each image. The binary dominant plane images after threshold processing are shown in Fig. 15. The results for calculating the probabilistic inclusive relations are shown in Table 1. We can accept the inclusive relation of  $\{D_i\}_{i=0}^3$ .



**Fig. 24.** Corridor centreing for mobile robot navigation in a S-shaped corridor. The grey triangle is the mobile robot that moves in the corridor without collision with walls.



**Fig. 25.** Example of image with added random noise. Left: original image of dominant plane. Right: noisy image of dominant plane.

## 5.2. Robustness evaluation for corridor centreing

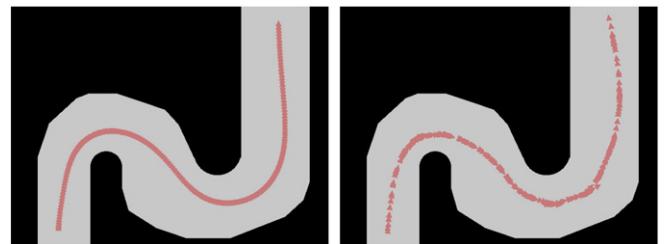
### 5.2.1. Robustness evaluation for synthetic image sequences

We show the validity and robustness of our algorithm by the corridor centreing experiment, as shown in Fig. 17. We evaluate the robustness of the algorithm for a straight trajectory. At each point on the trajectory, we can consider the tangent direction along which the robot moves forward infinitesimally. If the robot moves along the curve by the same mechanism as in forward motion, we can assume that the robot moves within the shaded corridor in Fig. 17. We call this corridor the feasible corridor.

There are basically two kinds of perturbation errors in our robot navigation system. The first one is noise in the images that are the input to the system, since our robot is controlled by a sequence of pictorial data. The second one is the perturbation on the geometric position of the robot caused mainly by mechanical errors in the robot. For the evaluation of the robustness of the controlling system, we have evaluated the effects of these two types of perturbations using two kinds of synthetic image sequences.

For the second kind of error, at each step of robot motion, we assume that displacement  $T(t)$  and rotation  $R(t)$  of the mobile robot have errors of 20%, as shown in Fig. 18. Additionally, we added an error of 20% to the detected dominant plane  $d(x, y, t)$ , that is, we randomly inverted the grey value of pixels in the image. The robot moves forward at a constant speed. The noise of the robot position is caused mainly by mechanical errors in the robot. Therefore, these errors directly affect robot motion and the trajectory.

The first kind of synthetic image sequence is in Fig. 19, which is generated by adding random noise to each image in an image sequence. The noise ratio of each image in this sequence is 20%. The sequence is observed when the robot moves forward at a constant speed. The noise in the images causes instability of the optical flow vectors, computed from a sequence of images, which affects the potential field. Therefore, these errors in the images affect the control direction of the robot. Fig. 20 shows the trajectory of robot viewed from the top. Fig. 21 shows the numerical expression of the trajectory. The graph shows the deviation of the trajectory. In this figure, the



**Fig. 26.** Overhead view of the result of corridor centreing in the S-shaped pathway. Black regions are walls, and the grey region is the corridor within which the robot is assumed to move. The triangles are the trajectory of the mobile robot. The mobile robot started at the bottom of the image. Left: result of the trajectory of the robot without any noise. Right: result of the trajectory of the robot with noise in the robot position.



**Fig. 27.** Experimental result for first frame. Starting from the left, the configuration of the robot and obstacles, captured image  $I(x, y, t)$ , detected dominant plane  $G \cdot d(x, y, t)$ , visual potential field  $p(x, y, t)$ , and estimated navigation force  $\bar{p}(t)$ .

width of the robot is 0.2 unit, and the width of the corridor is 3.5 units. Furthermore, the standard deviation between these two trajectories along the time axis is 0.0168. From this result, the effects of the errors in the image sequence are found to be small. Therefore, we can conclude that our robot control algorithm is robust against noise in the images if the robot moves forward at constant speed.

The second kind of synthetic image sequence includes the perturbation of the geometric position of the robot. At each step the robot motion, we added geometric noise to displacement  $T(t)$  and rotation  $R(t)$ . Fig. 22 shows the trajectory of robot viewed from the top. Fig. 23 shows the numerical expression of the trajectory. The graph shows the deviation of the trajectory. The standard deviation between these two trajectories along the time axis is 0.099. From this result, the effects of the errors in the geometric position are found to be small by comparison with the width of the robot. Therefore, we can conclude that our robot control algorithm is also robust against noise in the robot position if the robot moves forward at a constant speed.

These two experimental propagation-error analyzes are for the forward motion of the robot. Using the variance of these results, we evaluate the robustness of the algorithm for a curved trajectory. At each point on the trajectory, we can consider the tangent direction along which the robot moves forward infinitesimally.

Next, we show the corridor centreing experiment in a S-shaped corridor of Fig. 24. The synthetic image sequence is shown in Fig. 25, and was generated by adding random noise to each image in an image sequence. The noise ratio of each image in this sequence is 20%. The sequence is observed when the robot moves forward at a constant speed. The noise in the images causes instability of the optical flow vectors, computed from a sequence of images, which affects the potential field. Therefore, these errors in the images affect the control direction of the robot. In Fig. 26, we show the robot trajectories superimposed on the feasible corridor of each trajectory. As shown in Fig. 26, if the robot moves along the curve by the same mechanism as in forward motion, we can assume that the robot moves within the shaded corridor in Fig. 17.

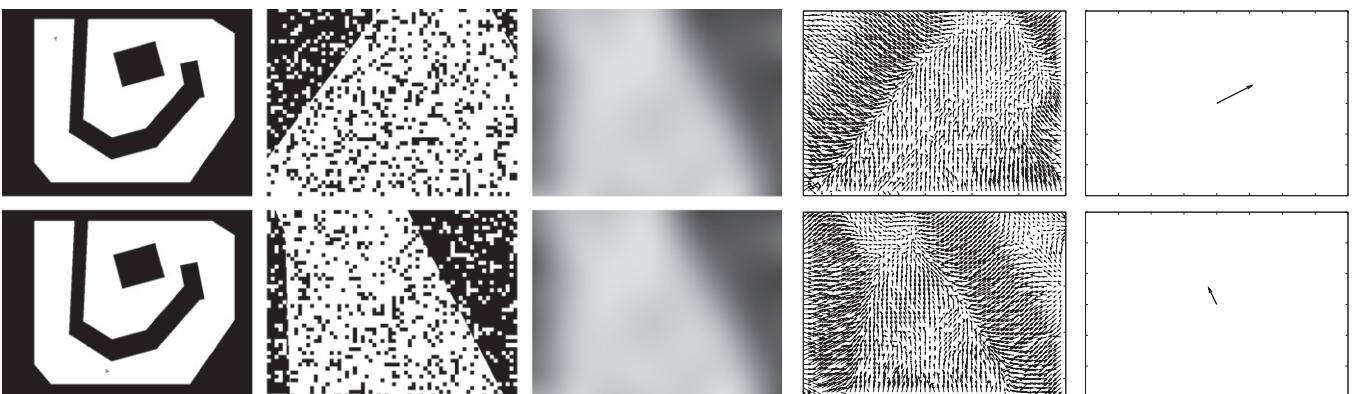
### 5.2.2. Geometrical properties of algorithm

Fig. 27 shows an experimental result for the first frame. In this figure, the triangle in the map is the position of the mobile robot and the black regions are obstacles. Fig. 28 shows results at frame 0 and frame 100. Starting from the left, the configuration of the robot and obstacles, captured image  $I(x, y, t)$ , detected dominant plane  $G \cdot d(x, y, t)$ , visual potential field  $p(x, y, t)$ , and estimated navigation force  $\bar{p}(t)$ . Fig. 29 shows the path determined using our method. This result indicates that our method enables the robot to avoid collision with obstacles without using an environmental map. Fig. 29 shows navigation force  $\bar{p}(t)$  in whole frame. At each step of robot motion, displacement  $T(t)$  and rotation  $R(t)$  of the mobile robot have errors of 20%, as shown in Fig. 18. Additionally, we added an error of 20% to the detected dominant plane  $d(x, y, t)$ . The results of the experiment with these errors are shown in Figs. 28 and 29. The robot moves forward at a constant speed.

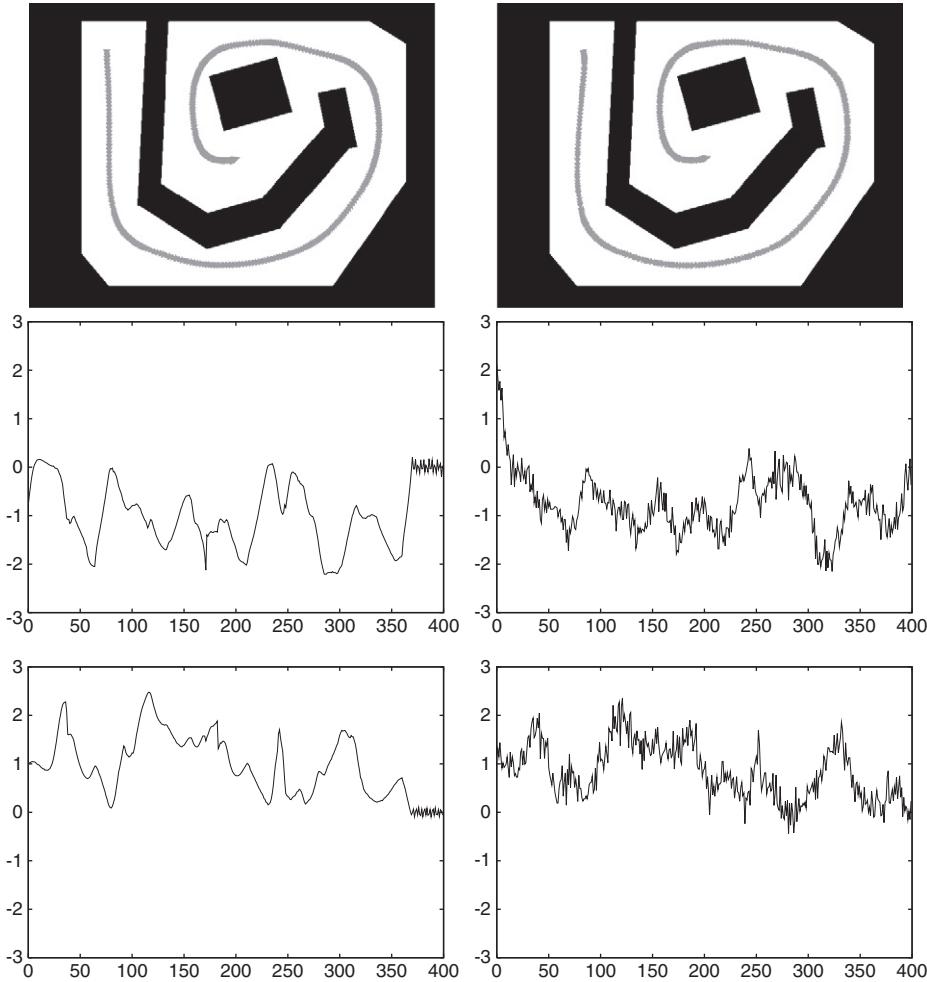
In Fig. 29, left and right images are results from ideal and noisy synthetic sequences. From top to bottom, the paths, the vertical axes are  $x$  component, and  $y$  component of navigation force  $\bar{p}(t)$ . Two coefficients of control forces computed from an ideal image sequence and a noisy sequence are comparative. Although we can observe small oscillation on coefficients of control forces computed from a noisy sequence, the whole profile of the trajectory of coefficients and paths are comparative for both ideal and noisy image sequences. These results indicate that our method is robust against robot-displacement error and obstacle-detection error. These experiments for a synthetic environment confirm that our method is effective for robot navigation using visual information captured by the robot itself.

### 5.2.3. Experiments on corridor centreing in real environment

We show collision avoidance by the mobile robot using our algorithm in a real environment. The specifications of the mobile robot are described in Table 2. The environments for the experiment are the straight and curved pathways. In these environments, the robot moved without collision with the obstacles and walls of the corridor.



**Fig. 28.** Experimental results at frame 0 and frame 100. Starting from the left, the configuration of the robot and obstacles, captured image  $I(x, y, t)$ , detected dominant plane  $G \cdot d(x, y, t)$ , visual potential field  $p(x, y, t)$ , and estimated navigation force  $\bar{p}(t)$ .



**Fig. 29.** Paths and navigation force  $\bar{p}(t)$  in whole frame. Left and right images are results from ideal and noisy synthetic sequences. From top to bottom, the paths, the vertical axes are  $x$  component, and  $y$  component of navigation force  $\bar{p}(t)$ . The horizontal axis of control force is frame number  $t$ .

The experimental results for the straight pathway and the curved pathway are shown in Figs. 30 and 31, respectively. In these figures, the top row shows snapshots of the experiment. In the middle row, starting from the left, the captured image  $I(x, y, t)$ , the estimated optical flow field  $(\dot{x}, \dot{y})^\top$ , and the detected dominant plane  $d(x, y, t)$  at one frame are shown. In the detected dominant plane, the white and black regions represent the dominant plane and the obstacle, respectively. In the bottom row, starting from the left, the Gaussian-operated image  $G * d(x, y, t)$ , the potential field  $\mathbf{p}(x, y, t)$ , and the estimated control force  $\bar{p}(t)$  at one frame are shown.

Fig. 30 shows that the estimated control force in the straight pathway indicates forward direction. Therefore, the robot moves forward. Fig. 31 shows that the estimated control force in the clockwise pathway indicates the right direction. Therefore, the robot rotates clockwise. These results show that our algorithm enables the mobile robot to move without collision with obstacles without any predetermined robot trajectories or environmental maps.

### 5.3. Experiments for navigation

#### 5.3.1. Experiments for navigation in synthetic environment

Fig. 32 shows experimental results for navigation in a synthetic environment with three different destinations. In this figure, the triangles in the map are the trajectories of the mobile robot, the black regions are obstacles, and the grey circle is the destination. Fig. 33 shows the robot position in the environment, the captured image

$I(x, t, t)$ , the attractive force field  $\hat{u}(x, y, t)$ , the visual potential field  $\mathbf{p}(x, y, t)$ , and the control force  $\bar{p}(t)$  at 10th, 30th, and 60th frames in Fig. 32(a). Several view of Fig. 32 are shown in Figs. 33, 34, and 35.

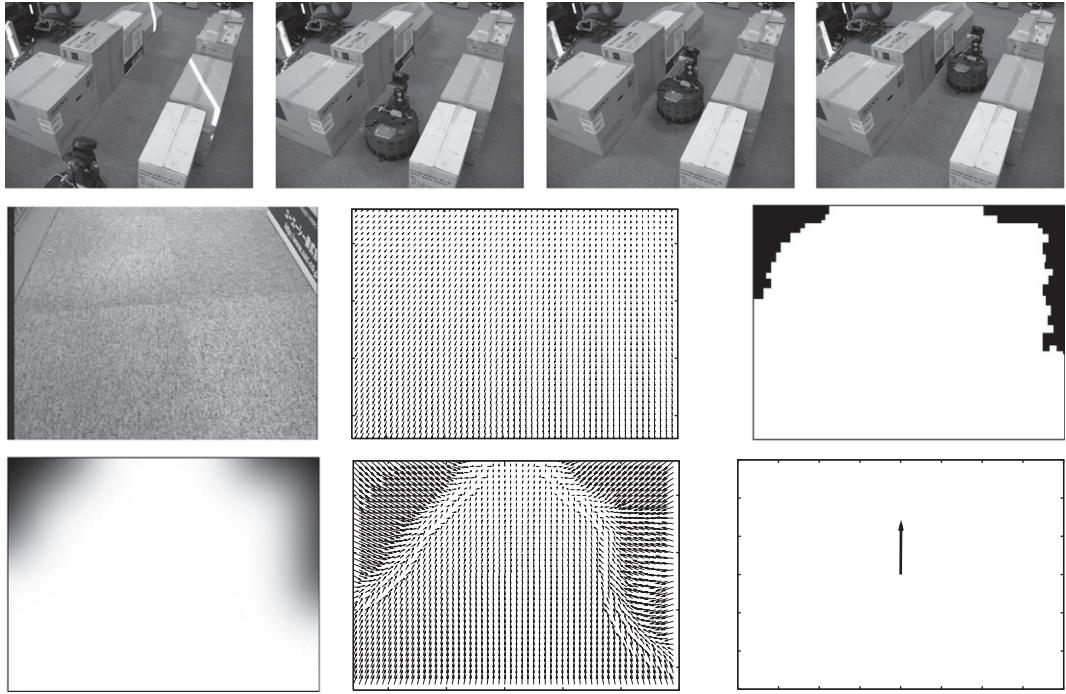
These experimental results show that our method of using visual information captured by the mobile robot itself is effective for robot navigation.

#### 5.3.2. Experiments for navigation in real environment

We show an experimental result for a real environment with one obstacle as shown in Fig. 36. The destination is set beyond the obstacle, that is, the obstacle divides the geodesic path from the origin to the destination. Therefore, on the image, the destination appears on the obstacle.

**Table 2**  
Specifications of our mobile robot.

Name	Magellan Pro, AAI Systems, Inc.
Size	Circular – 16-inch diameter
Weight	50 lb
Drive	2-wheel
CPU	800 MHz, AMD-K6 processor
Main memory	256 MB
OS	Red Hat Linux
Compiler	GNU C++ Compiler
Camera	SONY EVI-D30



**Fig. 30.** Experimental result for the straight pathway. The images in the top row are snapshots of this experiment. In the middle row, starting from the left: captured image  $I(x, y, t)$ , estimated optical flow field  $(\bar{x}, \bar{y})^T$ , and detected dominant plane  $d(x, y, t)$  at one frame. In the bottom row, starting from the left: Gaussian image  $G \cdot d(x, y, t)$ , potential field  $p(x, y, t)$ , and estimated control force  $\bar{p}(t)$  at one frame.

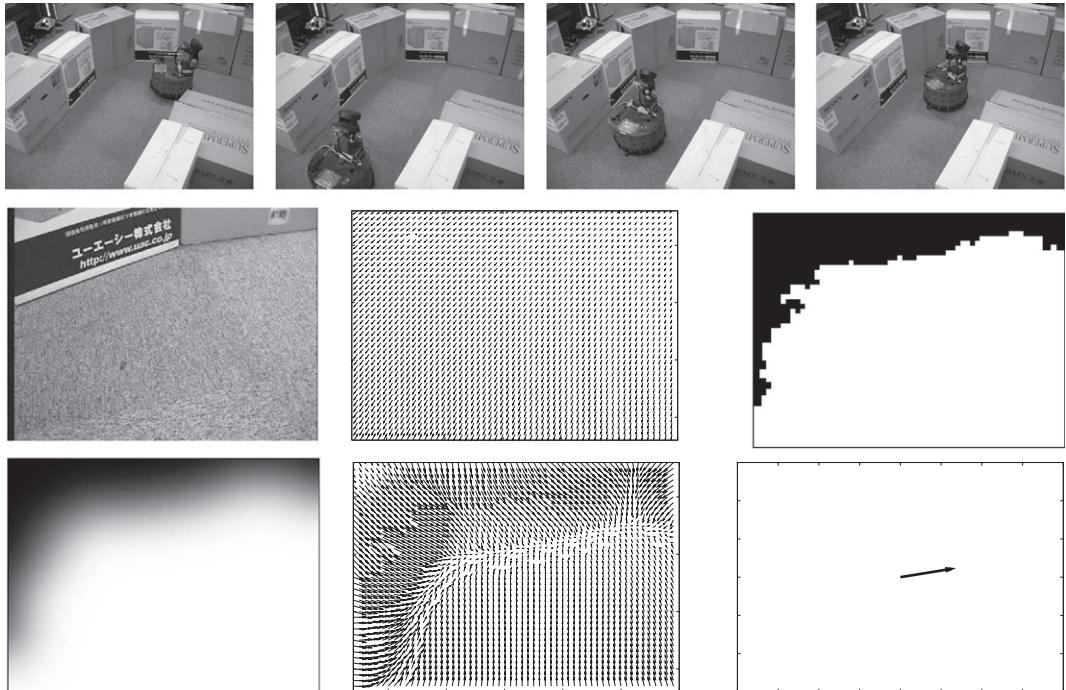
The direction angle  $\theta_d(t)$  between the destination and the robot direction is computed by

$$\theta_d(t) = \angle[X_D(t), X_D(0)] \quad (27)$$

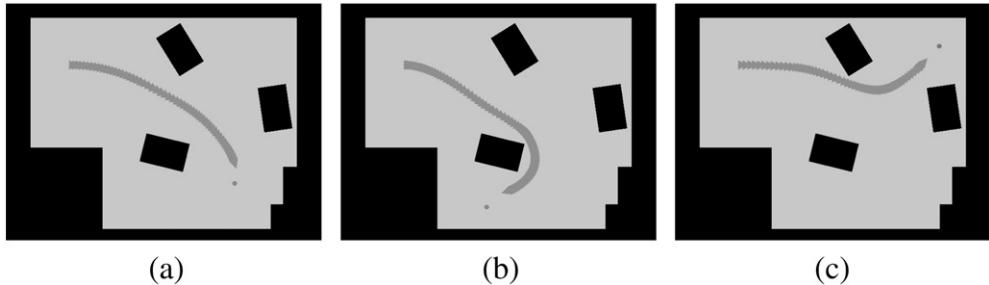
for  $\mathbf{X}_D(t) = \mathbf{X}(t) - \mathbf{D}$ , where  $\mathbf{D}$  is the coordinate of the destination point at  $t=0$ , which is given to the robot at the initial position (Fig. 35).

**Fig. 36** shows the snapshot of the robot in the environment, the captured image  $I(x, t, t)$ , the detected dominant plane  $d(x, t, t)$ , the visual potential field  $\mathbf{p}(x, y, t)$ , and the control force  $\bar{p}(t)$  at 10th, 50th, 100th, 150th, and 200th frames.

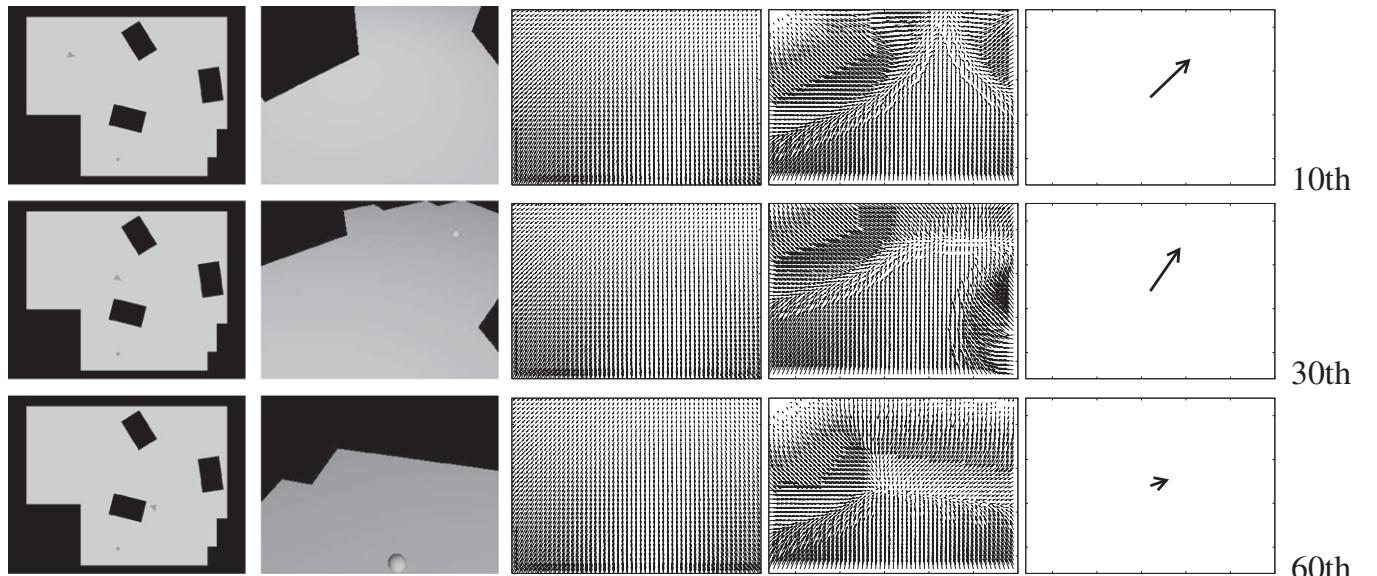
To avoid collision with the obstacle on the geodesic path from the origin to the destination, the robot first turned to the left and then to the right after passing beside the obstacle. The mobile robot avoided the obstacle and reached the destination.



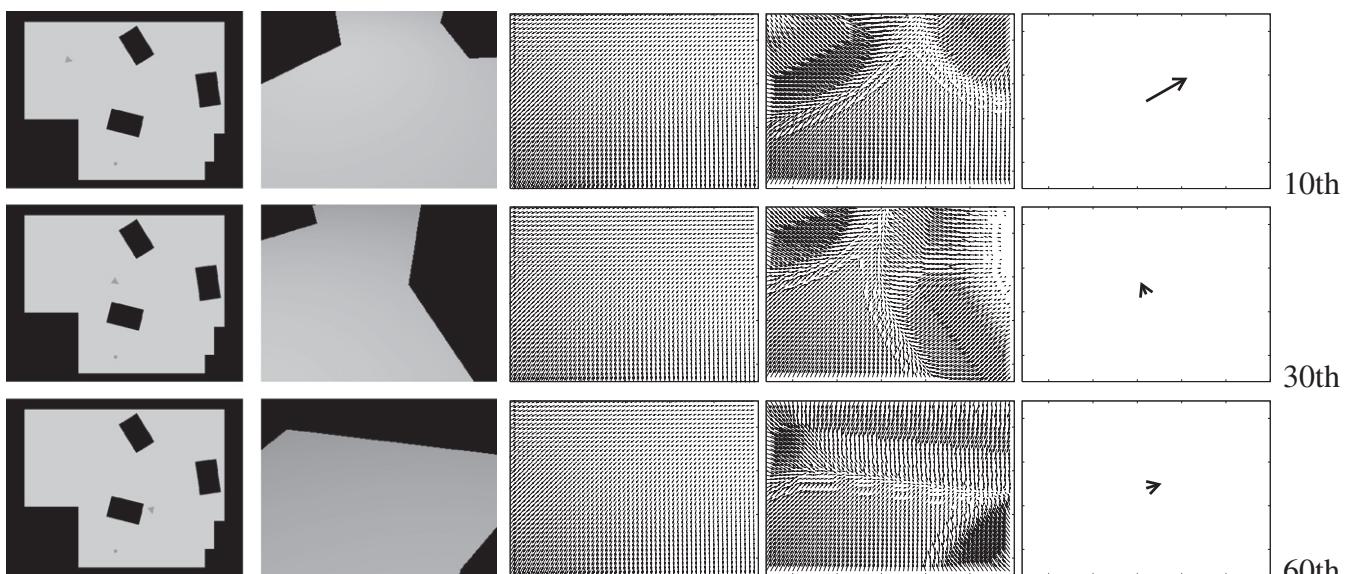
**Fig. 31.** Experimental result for the curved pathway in clockwise direction. The snapshots, captured image, and the results are shown.



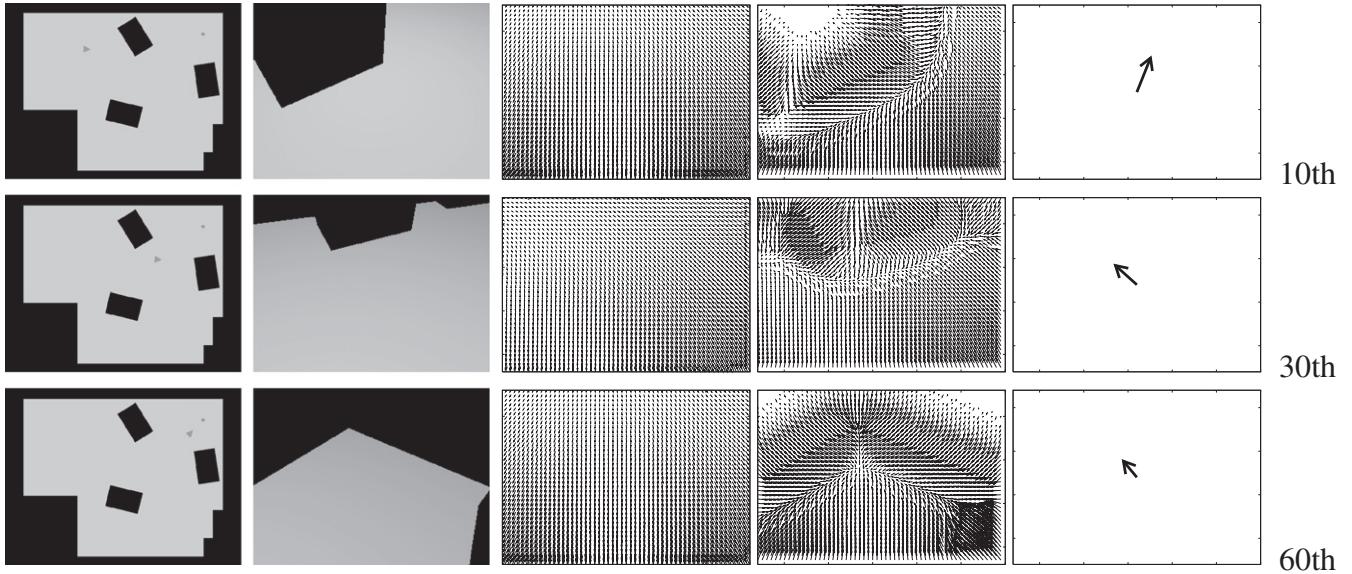
**Fig. 32.** Experimental results of navigation in synthetic environment with three different destinations. The triangles in the map are the trajectory of the mobile robot, the black regions are obstacles, and the grey circle is the destination. (a) The destination is set at the bottom-right corner. (b) The destination is set at the bottom-left corner. (c) The destination is set at the top-right corner.  $\circ$  and  $d$  indicate the start and destination points, respectively.



**Fig. 33.** Robot position and its images at several frames in Fig. 32(a). Starting from the left, the robot position in the environment, the captured image  $I(x, t, t)$ , the attractive force field  $\hat{u}(x, y, t)$ , the visual potential field  $p(x, y, t)$ , the control force  $\bar{p}(t)$ , and the frame number.



**Fig. 34.** Robot position and its images at several frames in Fig. 32(b). Starting from the left, the robot position in the environment, the captured image  $I(x, t, t)$ , the attractive force field  $\hat{u}(x, y, t)$ , the visual potential field  $p(x, y, t)$ , the control force  $\bar{p}(t)$ , and the frame number.

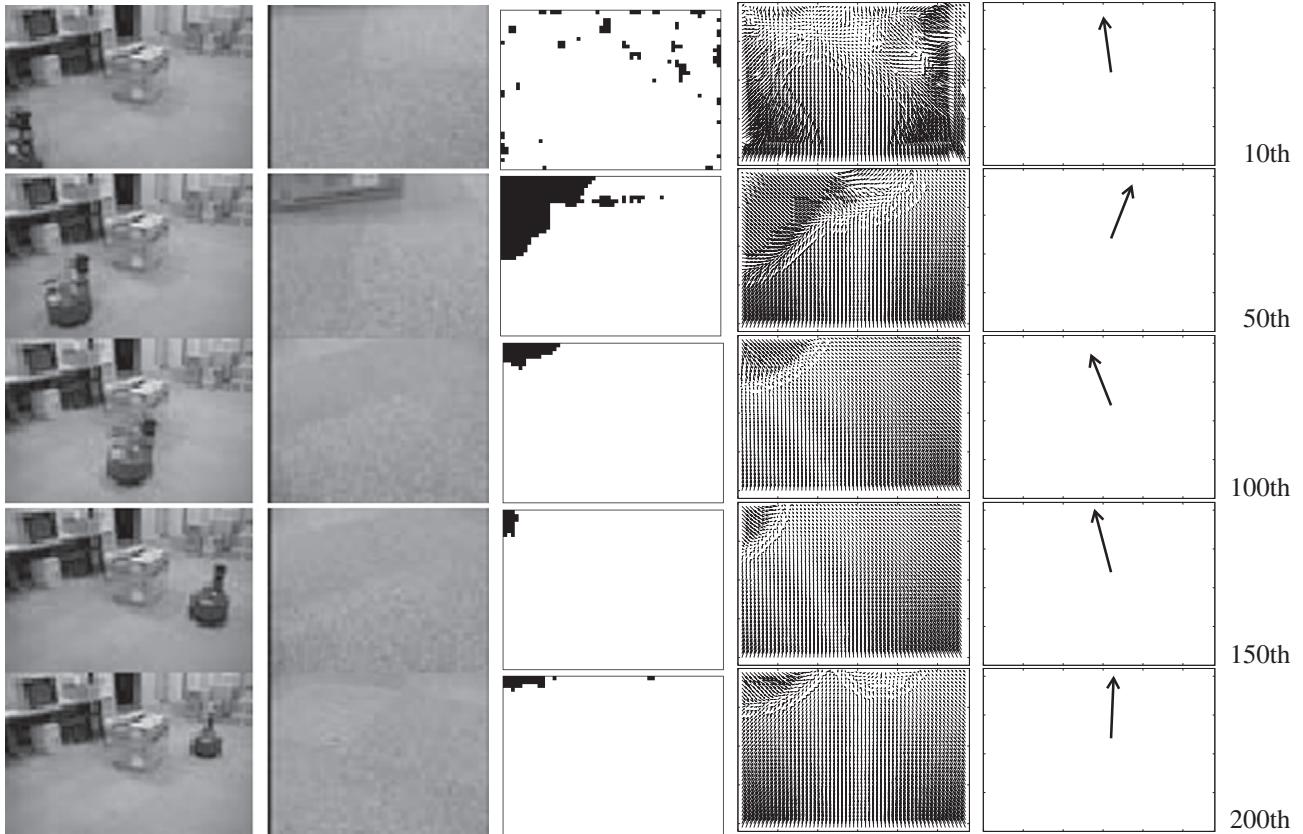


**Fig. 35.** Robot position and its images at several frames in Fig. 32(c). Starting from the left, the robot position in the environment, the captured image  $I(x, t, t)$ , the attractive force field  $\hat{u}(x, y, t)$ , the visual potential field  $p(x, y, t)$ , the control force  $\bar{p}(t)$ , and the frame number.

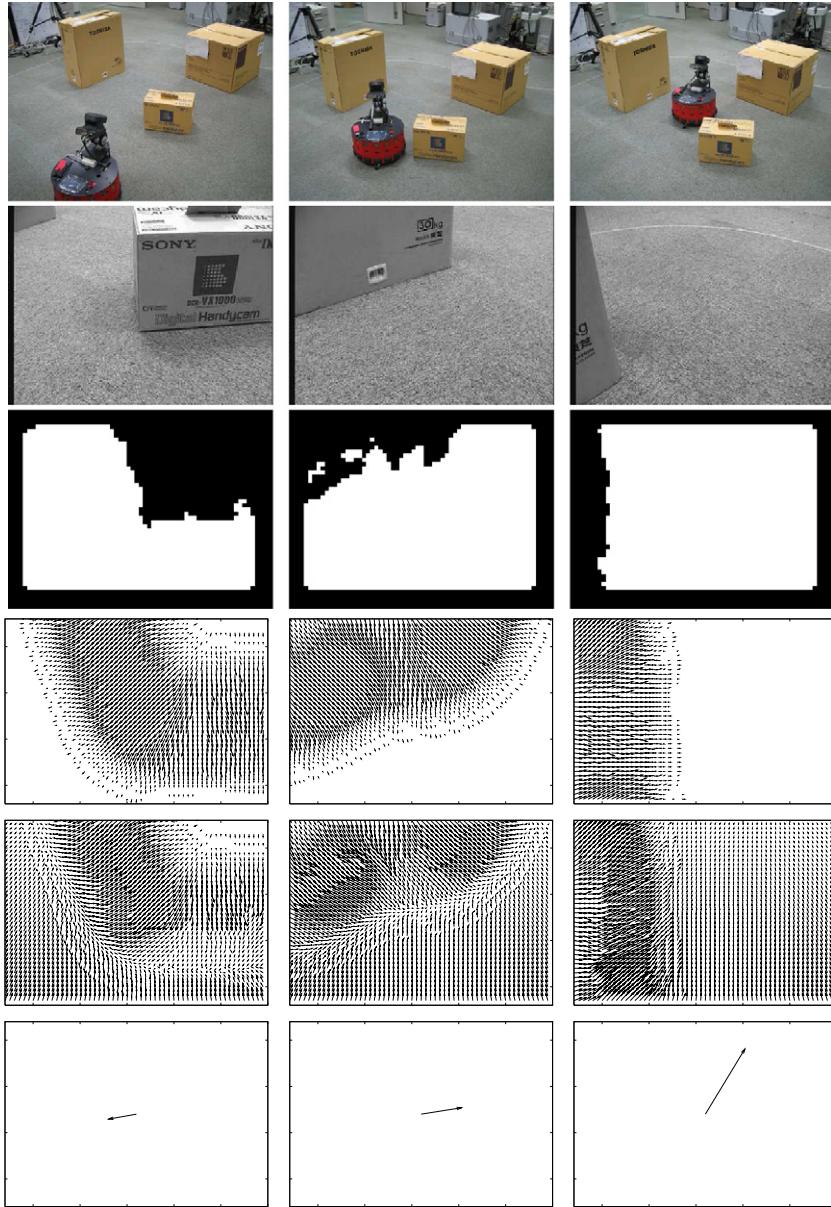
Next, we present an experimental result of obstacle avoidance. Fig. 37 shows the results for the estimation of visual potential and the navigation force. In Fig. 37, starting from the first row, the images show the configuration of the robot and obstacles, the captured image, detected dominant plane  $d(x, y, t)$ , gradient vector field  $\mathbf{g}(x, y, t)$ , visual potential field  $\mathbf{p}(x, y, t)$ , and estimated navigation force  $\bar{p}(t)$ .

#### 5.4. Experiments for homing in real environment

Finally, we show the homing of the mobile robot using the guidance algorithm. By converting the start point and the destination, the robot can return to the start point. Fig. 38 shows the homing process. The frame number is a sequel to Fig. 36. At the destination, the



**Fig. 36.** Experimental results for navigation in a real environment. Starting from the left, the snapshot of the robot in the environment, the captured image  $I(x, t, t)$ , the detected dominant plane  $d(x, y, t)$ , the visual potential field  $p(x, y, t)$ , the control force  $\bar{p}(t)$ , and the frame number. To avoid collision with the obstacle on the geodesic path from the origin to the destination, the robot first turned left and then turned right after passing beside the obstacle.



**Fig. 37.** Experimental results for estimation of visual potential. Starting from the top row, images show the configuration of the robot and obstacles, captured images  $I(x, y, t)$ , detected dominant plane  $d(x, y, t)$ , gradient vector field  $g(x, y, t)$ , visual potential field  $p(x, y, t)$ , and estimated navigation force  $\bar{p}(t)$ .

robot first turns  $180^\circ$  to change the destination, and then it uses the same algorithm.

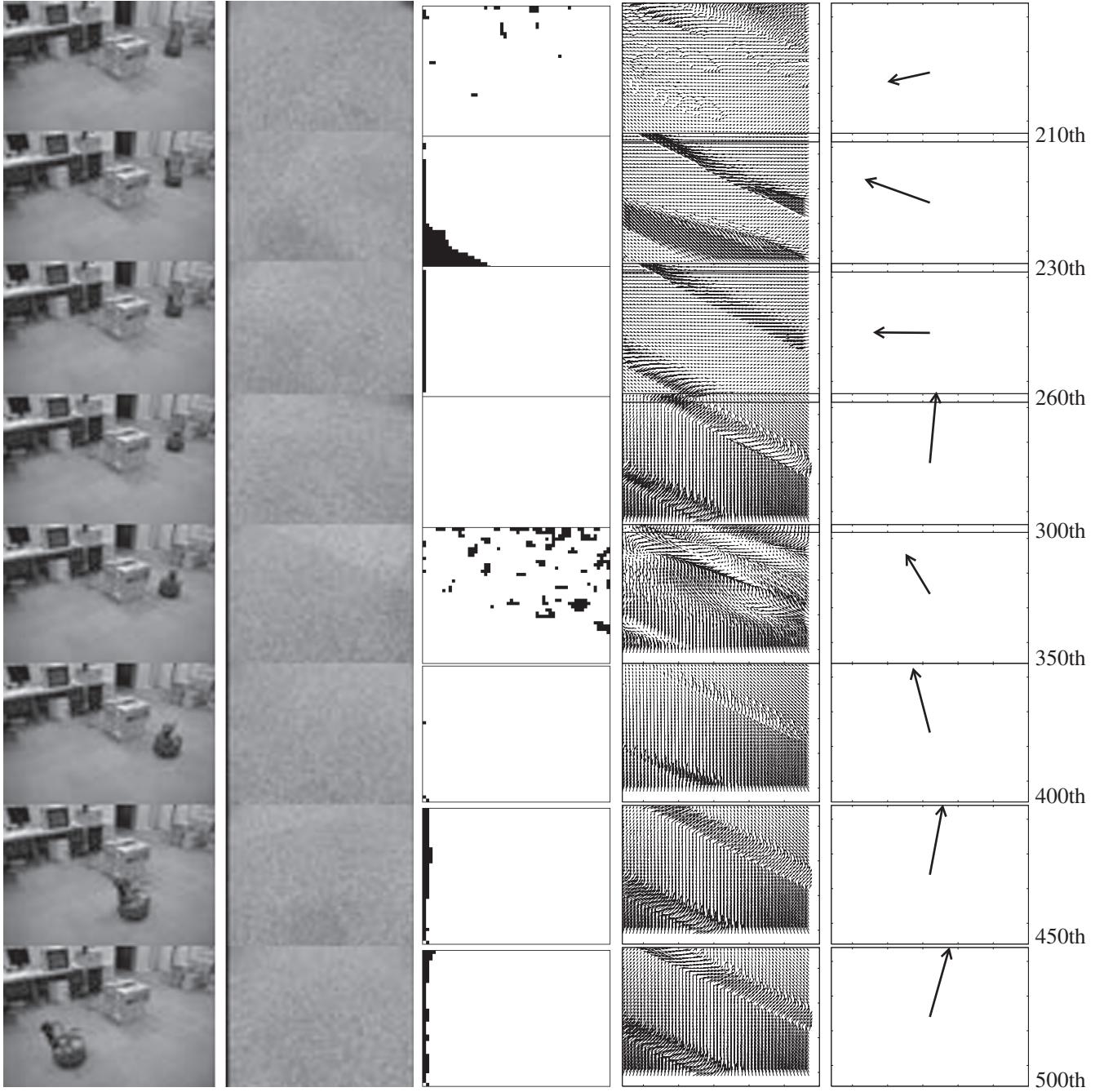
To achieve this control sequence for the guidance to the destination point and homing to the start point, the robot is required to store the geometric location of both start point and destination point in the memory mounted on the robot. After arriving at the destination point, the robot swaps the positions of the destination point and the start point in the memory. Then, the robot starts the guiding to the start point from the destinations, using them as the start point and destination point reversely. This process is shown in Figs. 39 and 40. The robot turned left to avoid collision with the obstacle and then turned right after passing beside the obstacle to go the home position.

Fig. 39 shows snapshots in homing process. Starting from the top left, 10th, 50th, 100th, 150th, 200th, 210th, 230th, 260th, 300th, 350th, 400th, 450th, and 500th frames. The bottom right number in the image is the frame number. Snapshots from 0th to 200th frames are views in the navigation process. From 200th to 300th frames, the robot turned  $180^\circ$  to change the destination. Snapshots from

300th to 500th frames are views in the homing process. Figs. 40(a) and 40(b) show the top views of the navigation and homing processes, respectively. The circle and arrow show the position and direction of the mobile robot at a frame, respectively. The hatched grey regions are obstacles. The numbers correspond to the frame numbers of images in the sequence.

## 6. Conclusions

We developed an algorithm for navigating a mobile robot to its destination using the optical flow and visual potential field captured by a camera mounted on the robot. We generated a potential field of repulsive force from obstacles to avoid collision, using images captured with the camera mounted on the robot. We asserted that optical flow vectors indicate the direction of robot motion to the local destination. Therefore, the optical flow field is suitable as the guide field to the destination of the robot. Our algorithm enabled a mobile robot to avoid obstacles without referring to an environmental map.



**Fig. 38.** Experimental results for homing in a real environment. Starting from the left, the snapshot of the robot in the environment, the captured image  $I(x, t, t)$ , the detected dominant plane  $d(x, t, t)$ , the visual potential field  $p(x, y, t)$ , and the frame number. The frame number is a sequel to Fig. 36. From 200th to 300th frames, the robot turned 180° to change the destination. Next, the robot turned left and then turned right after passing beside the obstacle to go the home position.

Experimental results showed that our algorithm is robust against the fluctuation of the displacement of the mobile robot.

The visual potential field used for guiding the robot was generated from visual information captured with the camera mounted on the robot without the use of any maps that describe the workspace. Therefore, our algorithm is an appearance-base navigation algorithm without a reconstruction of the three-dimensional environment.

Our navigation algorithm enables the robot to reach the destination using only the direction to the destination. Therefore, the algorithm allows to homing then it uses the same algorithm. Furthermore, using the geometrical location of the start point and the destination point stored in the memory, the robot achieves the navigation to the start point as homing process.

For navigation and path planning by robots, our final purposes are to determine a control force  $\dot{x}$  at each point and a path  $\mathcal{C}$  from the starting point  $s$  to the destination  $d$  from the configuration space  $\mathcal{A}$ . Setting

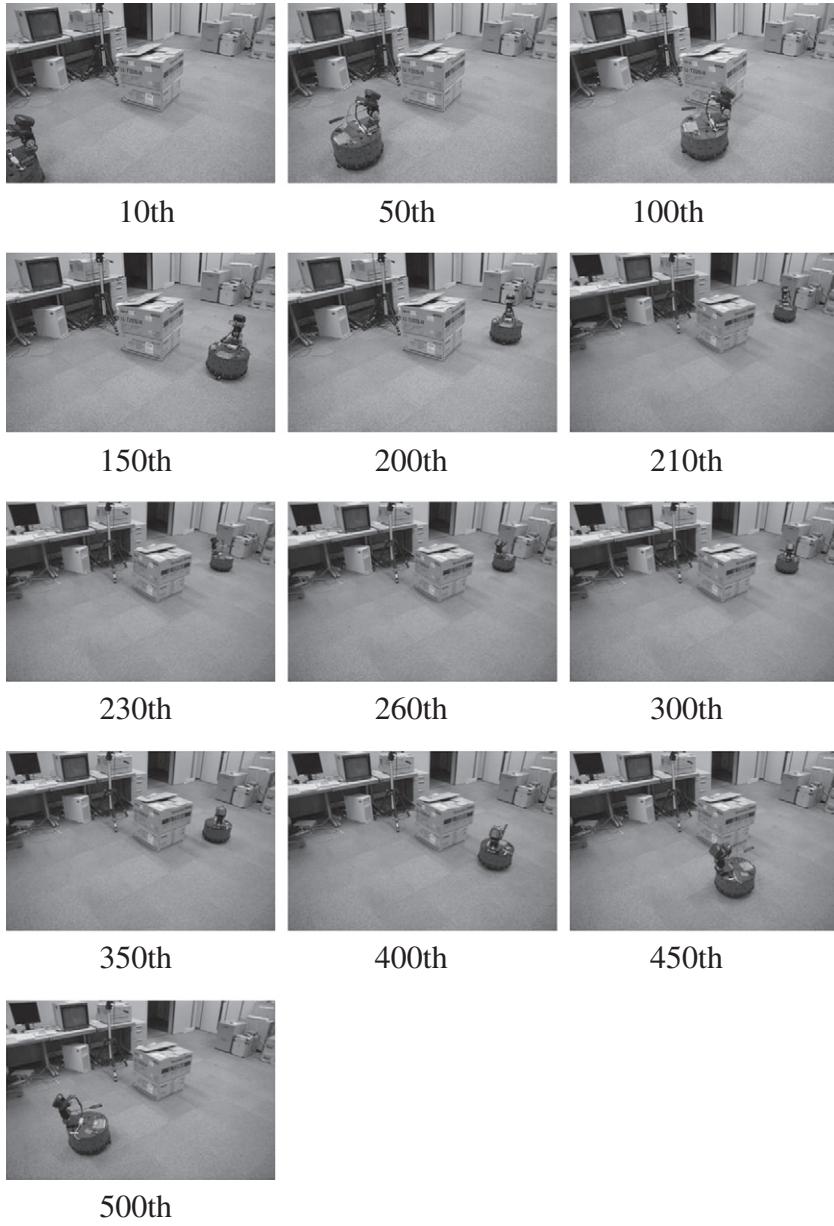
$$\mathcal{C} = \{x(t) | 0 \leq t \leq 1, x(0) = s, x(1) = d\}, \quad (28)$$

path planning  $\mathcal{P}$  is described symbolically as

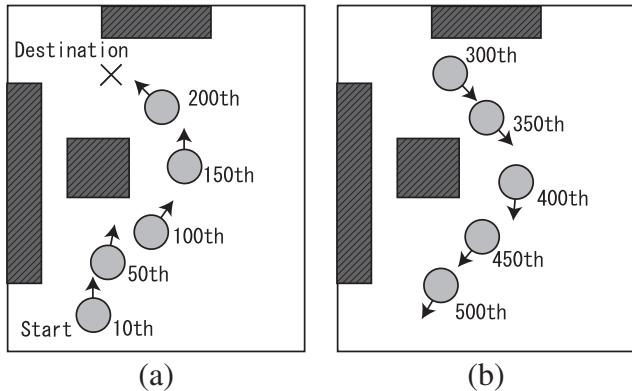
$$\mathcal{C} = \mathcal{P}(\mathcal{A}). \quad (29)$$

Furthermore, navigation  $\mathcal{N}$  is described symbolically as

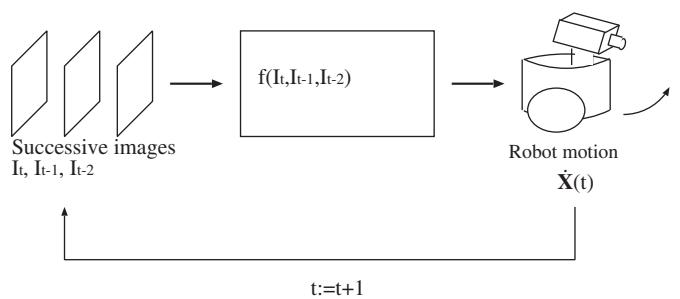
$$\dot{x} = \mathcal{N}(f(\mathcal{A}), x), \quad (30)$$



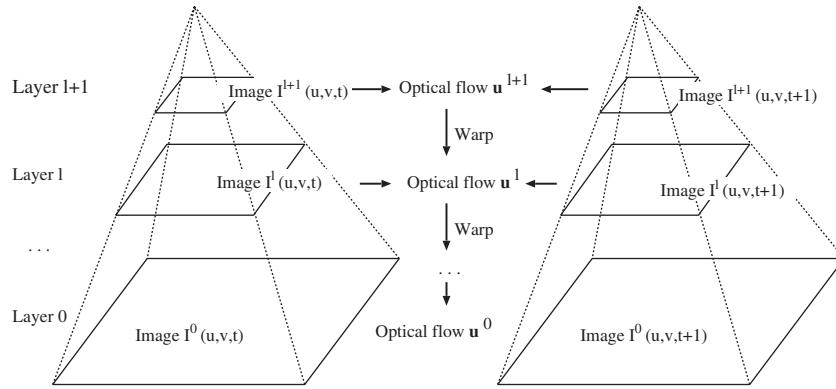
**Fig. 39.** Snapshots in homing process. Starting from the top left, 10th, 50th, 100th, 150th, 200th, 210th, 230th, 260th, 300th, 350th, 400th, 450th, and 500th frames.



**Fig. 40.** Top views of the environment in homing process. (a) Navigation process. (b) Homing process. The circle and arrow show the position and direction of the mobile robot at a frame, respectively. The hatched grey regions are obstacles. The numbers correspond to the frame numbers of images in the sequence.



**Fig. 41.** Closed loop function for autonomous robot motion. Our algorithm determines robot motion  $\dot{X}(t)$  from three successive images  $I(t)$ ,  $I(t-1)$  and  $I(t-2)$ . Function  $f$  is our algorithm.



**Fig. 42.** Procedure for computing optical flow in L-K method with pyramids. Optical flow is obtained by integration of optical flows of each level of the pyramid representation.

where  $f$  is an apparent feature of  $\mathcal{A}$ , such as images of  $\mathcal{A}$  obtained by a camera mounted on the robot. Moreover, servoing of the path is described symbolically as

$$\dot{x} = \mathcal{S}(f(\mathcal{A}), x, \mathcal{C}). \quad (31)$$

If  $f(\mathcal{A})$  is visual information of  $\mathcal{A}$ , navigation and servoing are called visual navigation and visual servoing, respectively.

For path planning and visual servoing, some optimisation conditions on  $\mathcal{C}$  and  $\dot{x}$  are additionally utilized. We are focusing on visual navigation. Therefore, in this paper, we dealt with Eq. (30). Since our method use disparity of the optical flow fields, that is, features are computed form successive three images  $I_{t-2}$ ,  $I_{t-1}$ , and  $I_t$ , where  $I_t$  is the image captured by the camera in the configuration space  $\mathcal{A}$  at time  $t$ , we can set

$$f(\mathcal{A}) = \{I_{t-2}, I_{t-1}, I_t\}. \quad (32)$$

Therefore, our model is symbolically described as

$$\dot{X}(t) = \mathcal{N}(I_t, I_{t-1}, I_{t-2}) \quad (33)$$

as shown in Fig. 41.

## Appendix A

### Equation

$$\frac{\partial I}{\partial x} \dot{x} + \frac{\partial I}{\partial y} \dot{y} + \frac{\partial I}{\partial t} = 0.$$

can be solved by assuming that the optical flow vector of pixels is constant in the neighbourhood of each pixel. We set the window size to be  $5 \times 5$ . Then, Eq. (1) is expressed as a system of linear equations,

$$\begin{aligned} I_{\alpha\beta} \dot{x} + I_{\beta\gamma} \dot{y} + I_t = 0, & \quad |\alpha| \leq 2, |\beta| \leq 2 \\ I_{\alpha\beta}(x, y, t) &= I(x + \alpha, y + \beta, t + 1), \end{aligned}$$

where  $I_{\alpha\beta}(x, y)$  is the spatial neighbourhood of the pixel. The optical flow  $\dot{x}$  is solved by the Lucas–Kanade method [21]. Setting this phase as the estimated optical flow at level 0 of the pyramid representation of the image, we estimate the optical flow at levels 0 to 4.

By setting  $I^0(x, y, t) = I(x, y, t)$  for the original image and  $I^l(x, y, t)$  for the pyramid transformation of the image  $I(x, y, t)$  at level  $l$ , the pyramid representation is expressed as

$$I^{l+1}(x, y, t) = \sum_{\alpha, \beta \in N_i} a_{\alpha\beta} I^l(x - \alpha, y - \beta, t),$$

where  $N_i$  is the neighbourhood of the point  $(x, y)^T$  at level  $i$  and  $a_{\alpha\beta}$  is weight parameter of the neighbourhood pixels. We set  $N_i$  to be the  $3 \times 3$  neighbourhood of a point and

$$a_{\alpha\beta} = \begin{cases} \frac{1}{4}, & (\alpha = 0, \beta = 0) \\ \frac{1}{8}, & (\alpha = \pm 1, \beta = 0), (\alpha = 0, \beta = \pm 1) \\ \frac{1}{16}, & (\alpha = \pm 1, \beta = \pm 1). \end{cases}$$

Finally, the optical flow is obtained by the integration of the optical flows at each level of the pyramid representation. We denote the optical flow field at time  $t$  as  $u(x, y, t)$ , which is a set of optical flows  $\dot{x}$  computed for all pixels in an image.

Using the pyramid transform, the optical-flow field is computed by the following algorithm, where  $u$  and  $w$  stand for the optical-flow computational algorithm at each pyramid level and the warping operation, respectively, where  $w(I, u) = I(x + u)$ . In the algorithm,  $u_t^l = (u^l, v^l)^T$  is computed from  $u_t^{l+1} = (u^{l+1}, v^{l+1})^T$  as  $u^l = E(u^{l+1})$  and  $v^l = E(v^{l+1})$ , where

$$E(f) = 4 \sum_{\alpha} \sum_{\beta} a_{\alpha\beta} f_{\frac{m-\alpha}{2}, \frac{n-\beta}{2}},$$

for the function  $f(x, y)$  defined on the plane. The summation is computed, when both  $\frac{m-\alpha}{2}$  and  $\frac{n-\beta}{2}$  are integers. In this algorithm,  $I_t^l$  stands for the pyramidal representation at the level  $l$  of an image  $I(x, y, t)$  at time  $t$ .

Fig. 42 shows the Lucas–Kanade method with pyramid. Applying the dominant-plane-detection algorithm at each layer, we detect dominant plane hierarchically in an image sequence, from the optical flow fields  $u^l(x, y, t)$  at the layer  $l$ .

**Algorithm 4.** Optical flow computation by the Lucas–Kanade method using pyramids

---

```

Data:  $I_t^l, I_{t+1}^l, 0 \leq n \leq \text{the maximum of the layers};$ 
Result:  $x_{ijk} := x_{(0)ijk};$ 
1  $:=$  the maximum of the layers;
2 while  $n \neq 0$ , do
3    $d_t^l := u(I_t^l, I_{t+1}^l);$ 
4    $I_{t+1}^{l+1} := w(I_{t+1}^l, u_t^l);$ 
5    $l := l - 1;$ 
6 end

```

---

## References

- [1] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, Int. J. Robot. Res. 5 (1986) 90–98.

- [2] S. Trihatmo, R.A. Jarvis, Short-safe compromise path for mobile robot navigation in a dynamic unknown environment, Australian Conf. on Robotics and Automation, 2003.
- [3] A. Das, R. Fierro, V. Kumar, J. Southall, J. Spletzer, C. Taylor, Real-time vision-based control of a nonholonomic mobile robot, IEEE International Conference on Robotics and Automation, 2001, pp. 1714–1719.
- [4] Y. Ma, J. Kosecka, S.S. Sastry, Vision guided navigation for a nonholonomic mobile robot, IEEE Trans. Robot. Autom. 15 (1999) 521–536.
- [5] N.D. Guilherme, C.K. Avinash, Vision for mobile robot navigation: a survey, IEEE Trans. PAMI 24 (2002) 237–267.
- [6] N. Ohnishi, A. Imaia, Featureless robot navigation using optical flow, Connect. Sci. 17 (2005) 23–46.
- [7] J.L. Barron, D.J. Fleet, S.S. Beauchemin, Performance of optical flow techniques, Int. J. Comput. Vis. 12 (1994) 43–77.
- [8] J.Y. Bouguet, Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm, Intel Corporation, Microprocessor Research Labs, OpenCV Documents, 1999.
- [9] N. Ohnishi, A. Imaia, Navigation of nonholonomic mobile robot using visual potential field, International Conference on Computer Vision Systems, 2007. (<http://biecoll.ub.uni-bielefeld.de/volltexte/2007/54>).
- [10] N. Ohnishi, A. Imaia, Corridor navigation and obstacle avoidance using visual potential for mobile robot, Canadian Conference on Computer and Robot Vision, 2007, pp. 131–138.
- [11] A. Bur, A. Tapus, N. Ouerhani, R. Siegwar, H. Hiigli, Robot navigation by panoramic vision and attention guided features, International Conference on Pattern Recognition, 2006, pp. 695–698.
- [12] C. Lopez-Franco, E. Bayro-Corrochano, Omnidirectional vision and invariant theory for robot navigation using conformal geometric algebra, International Conference on Pattern Recognition, 2006, pp. 570–573.
- [13] P.E. Lopez-de-Teruel, A. Ruiz, L. Fernandez, Efficient monocular 3D reconstruction from segments for visual navigation in structured environments, International Conference on Pattern Recognition, 2006, pp. 143–146.
- [14] A. Stemmer, A. Albu-Schaffer, G. Hirzinger, An analytical method for the planning of robust assembly tasks of complex shaped planar parts, IEEE International Conference on Robotics and Automation, 2007, pp. 317–323.
- [15] C. Braillon, C. Pradalier, J.L. Crowley, C. Laugier, Real-time moving obstacle detection using optical flow models, Intelligent Vehicles Symposium, 2006, pp. 466–471.
- [16] B. Liang, N. Pears, Visual navigation using planar homographies, IEEE International Conference on Robotics and Automation, 2002, pp. 205–210.
- [17] K. Young-Geun, K. Hakil, Layered ground floor detection for vision-based mobile robot navigation, International Conference on Robotics and Automation, 2004, pp. 13–18.
- [18] P.J. Sobey, Active navigation with a monocular robot, Biol. Cybern. (71) (1994) 433–440.
- [19] N. Ohnishi, Y. Mochizuki, A. Imaia, T. Sakai, On-line planar area segmentation from sequence of monocular monochrome images for visual navigation of autonomous robot, Proc. VISAPP 1 (2010) 435–442.
- [20] B.K.P. Horn, B.G. Schunck, Determining optical flow, Artif. Intell. 17 (1981) 185–203.
- [21] B. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, International Joint Conference on Artificial Intelligence, 1981, pp. 674–679.
- [22] H.H. Nagel, W. Enkelmann, An investigation of smoothness constraint for the estimation of displacement vector fields from image sequences, IEEE Trans. PAMI 8 (1986) 565–593.
- [23] H.A. Mallot, H.H. Bulthoff, J.J. Little, S. Bohrer, Inverse perspective mapping simplifies optical flow computation and obstacle detection, Biol. Cybern. 64 (1991) 177–185.
- [24] D.C. Conner, A.A. Rizzi, H. Choset, Composition of local potential functions for global robot control and navigation, Int. Conf. Robot. Syst. 4 (2003) 3546–3551.
- [25] A.D. Tews, G.S. Sukhatme, M.J. Matarić, A multi-robot approach to stealthy navigation in the presence of an observer, IEEE International Conference on Robotics and Automation, 2004, pp. 2379–2385.
- [26] F. Capparella, L. Freda, M. Malagnino, G. Oriolo, Visual servoing of a wheeled mobile robot for intercepting a moving object, Proc. IROS, 2005, pp. 2021–2027.
- [27] F. Mnif, F. Touati, An adaptive control scheme for nonholonomic mobile robot with parametric uncertainty, Int. J. Adv. Robot. Syst. 2 (2005) 59–63.
- [28] N. Wada, M. Saeki, H. Chen, Robust tracking control of a nonholonomic mobile robot in the presence of disturbances, JSME Int. J. Ser. C 47 (2004) 694–701.
- [29] D. Aarno, D. Krägic, H.I. Christensen, Artificial potential biased probabilistic roadmap method, IEEE Int. Conf. Robot. Autom. 1 (2004) 461–466.
- [30] S. Shimoda, Y. Kuroda, K. Iagnemma, Potential field navigation of high speed unmanned ground vehicles on uneven terrain, IEEE International Conference on Robotics and Automation, 2005, pp. 2839–2844.
- [31] K. Valavanis, T. Hebert, R. Kolluru, N. Tsourveloudis, Mobile robot navigation in 2D dynamic environments using an electrostatic potential field, IEEE Trans. Syst. Man Cybern. 30 (2000) 187–196.
- [32] H. Murase, S.K. Nayar, Visual learning and recognition of 3-D objects from appearance, Int. J. Comput. Vis. 14 (1995) 5–24.
- [33] M.G. Suraj, D.S. Guru, Appearance based recognition methodology for recognising fingerspelling alphabets, International Joint Conference on Artificial Intelligence, 2007, pp. 605–610.
- [34] J.M. Buenaposada, E. Munoz, L. Baumela, Efficient appearance-based tracking, Proceeding CVPRW'04 conjunction with International Conference on Computer Vision and Pattern Recognition, 2004.
- [35] S.D. Jones, C. Andrensen, J.L. Crowley, Appearance based processes for visual navigation, IEEE/RSJ International Conference on Intelligent Robots and Systems, 1997, pp. 551–557.
- [36] I. Ulrich, I. Nourbakhsh, Appearance-based place recognition for topological localization, IEEE International Conference on Robotics and Automation, 2000, pp. 1023–1029.
- [37] D. Coombs, K. Roberts, Centering behavior using peripheral vision, Proc. CVPR, 1993, pp. 400–407.
- [38] M.O. Franz, H.A. Mallot, Biomimetic robot navigation, Robot. Auton. Syst. 30 (2000) 133–153.
- [39] W.E. Green, P.Y. Oh, G. Barrows, Flying insect inspired vision for autonomous aerial robot maneuvers in near-earth environments, IEEE International Conference on Robotics and Automation, 2004, pp. 2347–2352.
- [40] A. Vardy, R. Moller, Biologically plausible visual homing methods based on optical flow techniques, Connect. Sci. 17 (2005) 47–89.
- [41] A. Imaia, I. Fermín, Motion analysis by random sampling and voting process, Comp. Vision Image Underst. (CVIU) textbf{73} (1999) 309–328.
- [42] N. Ohnishi, A. Imaia, Dominant plane detection from optical flow for robot navigation, Pattern Recogn. Lett. 27 (2006) 1009–1021.
- [43] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (1981) 381–395.
- [44] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2000.