We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

21

Vision-Based Path Following Without Calibration[1]

Zhichao Chen and Stanley T. Birchfield,
Clemson University
United States of America

1. Introduction

Route-based knowledge, in which the spatial layout of an environment is recorded from the perspective of a ground-level observer, is an important component of human and animal navigation systems (Shelton & Gabrieli, 2002). In this representation, navigating from one location to another involves comparing current visual inputs with a sequence of views captured along the path in a previous instance. Applications that would benefit from such a path-following capability include courier and delivery robots (Burgard et al., 1999), robotic tour guides (Shen & Hu, 2006), or reconnaissance robots following a scout (Crawford et al., 2004). Furthermore, a solution to this problem would be useful for the general problem of navigating between two arbitrary locations in an environment by following a sequence of such paths.

One approach to path following is visual servoing, in which the robot is controlled to align the current image with a reference image, both taken by an onboard camera (Hutchinson et al., 1996). Such an approach generally employs a Jacobian to relate the coordinates of world points to their projected image coordinates (Burschka and Hager, 2001), a homography or fundamental matrix to relate the coordinates between images (Sagüés & Guerrero, 2005, Remazeilles & Chaumette, 2007, Liang & Pears, 2002, Šegvić et al., 2007), or bundle adjustment to minimize the reprojection error over multiple image frames (Royer et al., 2007). As a result, the camera usually must be calibrated (Burschka and G. Hager, 2001, Remazeilles & Chaumette, 2007, Royer et al., 2007, Šegvić et al., 2007), and even uncalibrated systems require lens distortion to be removed. Alternative vision-based algorithms make strong assumptions about the environment or the sensor, such as a flat ground plane (Burschka & Hager, 2001, Liang & Pears, 2002, Guerrero & Sagüés 2001, 2005), a man-made environment in which vertical straight lines are present (Kosaka & Kak, 1992, Sagüés & Guerrero, 2001, 2005, Tang & Yuta 2001), or an omnidirectional camera (Gaussier, 1997, Ulrich & Nourbakhsh, 2002, Tang & Yuta 2001, Kröse et al., 2001).

---

[1] Based on "Qualitative Vision-Based Path Following", by Z. Chen and S. Birchfield, which appeared in *IEEE Transactions on Robotics*, 25(3):749-754, June 2009. © 2009 IEEE.

To overcome these limitations, we consider the problem from a novel viewpoint in which there is no equation relating image coordinates to world coordinates. Such a direct approach is motivated by the observation that the problem is vastly overdetermined, with tens of thousands of image pixels available to determine a single turning command output. We present a simple algorithm that uses a single, off-the-shelf camera attached to the front of the robot. The technique follows the teach-replay approach (Burschka & Hager, 2001) in which the robot is manually led through the path once during a teaching phase and then follows the path autonomously during the replay phase. Without any camera calibration (even calibration for lens distortion), the robot is able to follow the path by making only qualitative comparisons between the feature coordinates in the two phases. All that is needed is a single controller gain parameter to convert pixel coordinates to turning angles. We demonstrate the technique on several indoor and outdoor experiments, showing its robustness with respect to slanted surfaces, changing lighting conditions, and dynamic occluding objects. This paper extends the applicability and improves upon the robustness of our earlier work (Chen & Birchfield, 2006) by incorporating odometry information and correcting for camera roll. We also demonstrate the ability of the technique to work with wide-angle and omni-directional cameras, with only slight modification in the latter case to ignore the bottom half of the image which views the scene behind the robot.

The proposed approach falls within the category of mapless algorithms (DeSouza & Kak, 2002). As such, it is closely related to the view-sequenced route representation (VSRR) of Matsumoto et al. (Matsumoto et al., 1996, 2000, Jones et al., 1997) in which the turning angle is computed by cross-correlating images acquired during the replay phase with those captured during training. However, VSRR requires large amounts of memory to store the views and is sensitive to occlusions by dynamic objects. Along with a homography-based extension using vertical lines (Sagüés & Guerrero, 2005), it has only been demonstrated for short sequences on flat terrains. An alternate mapless approach is to learn the mapping from images to turning commands based on their classification (Weng & Chen, 1996, Ackerman & Itti, 2005). While this method can successfully follow a specific pattern such as a road or hallway, it will have difficulty generalizing to environments in which the images cannot be categorized into a small number of classes known at training time. Another approach that has received considerable attention (Gaussier, 1997, Wolf et al., 2002, Zhou et al., 2003, Sim & Dudek, 2001, Košeckā, 2003, Ulrich & Nourbakhsh, 2002, Horswill, 1993) is to store an example image with each specific location of interest. At run time, the image database is searched to find the image that most closely resembles the current one (or, alternatively, the current image is projected onto a manifold learned from the database (Nayar, 1994, Kröse et al., 2001)). Such approaches require extensive training and have difficulty providing sufficient spatial resolution to determine actual turning commands in large environments. Similarly, sensory-motor learning has been used to map visual inputs to turning commands, but the resulting algorithms have been too computationally demanding for real-time performance (Giovannangeli et al., 2006). Other researchers have developed mapless algorithms for low-level functionality like corridor following or obstacle avoidance (Nelson & Aloimonos, 1988, Santos-Victor et al., 1995, Barrows et al., 2002, LeCun et al., 2005, Michels et al., 2005, Murali & Birchfield, 2008), but these techniques are not applicable to following a specific arbitrary path.

2. Qualitative mapping from feature coordinates to turning direction

Consider a mobile robot equipped with a camera whose optical axis is parallel to the heading direction of the robot. Suppose we wish to move the robot from location $C=(x_C,y_C,\theta_C)$ to a previously encountered location $D=(x_D,y_D,\theta_D)$, where (x_i,y_i) and θ_i are the position and orientation, respectively, in the xy plane, $i\in\{C,D\}$. The robot has access to a current image I_C, taken at C, and a destination image I_D, taken previously at the destination D.

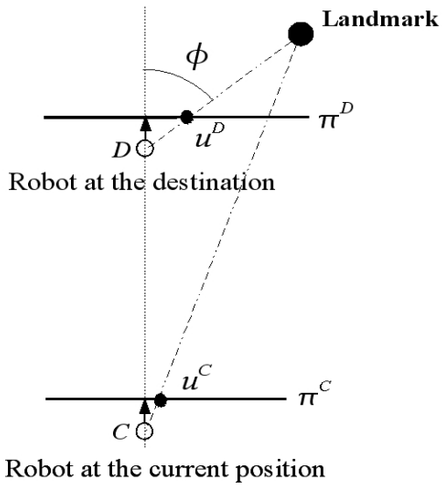


Fig. 1. The robot is at ***C*** moving toward the destination ***D*** with the same heading direction. The open circle coincides with both the camera focal point and the robot position, the arrow indicates the heading direction, π is the image plane, and ϕ is the angle between the optical axis and the projection ray from the landmark.

We start with a simple observation. Suppose the robot views a fixed landmark in both images yielding image feature coordinates of u^C and u^D, as shown in Fig. 1. The features are computed with respect to a coordinate system centered at the principal point (the intersection of the optical axis and the image plane), so that positive coordinates are on the right side of the image while negative coordinates are on the left side. If the robot moves toward the destination in a straight line with the same heading direction as that of the destination (i.e., $\theta_C=\theta_D$), then the point u^C will move away from the principal point toward u^D, reaching u^D when the robot reaches D. This observation is made more precise in the following theorem.

Theorem 1: Let a mobile robot move in a straight line toward location D on a flat surface. Let u^j be the horizontal image coordinate, relative to the principal point, of a monotonic projection at location j of a fixed landmark. For any location C along the line such that $\theta_C=\theta_D$, $\left|u^C\right|<\left|u^D\right|$ and $\operatorname{sign}\left(u^C\right)=\operatorname{sign}\left(u^D\right)$.

The theorem can be easily proved by geometry. Note that the image projection function is only required to be monotonic (i.e., perspective projection is not necessary), so the result applies equally to a camera with radial lens distortion. The primary assumption is that the optical axis of the camera passes through the axis of rotation of the robot. Other assumptions include the alignment of the optical axis with the robot heading direction, zero roll and tilt angles of the camera with respect to the robot, and a flat ground plane. In practice, misalignment is not an issue because the camera alignment can be learned automatically by estimating the focus of expansion as the robot drives forward. Similarly, rough terrain is easily handled by measuring image rotation to compensate for a non-zero roll angle of the robot and by recognizing that a non-zero tilt angle has a negligible effect on the horizontal feature coordinates.

2.1 The funnel lane

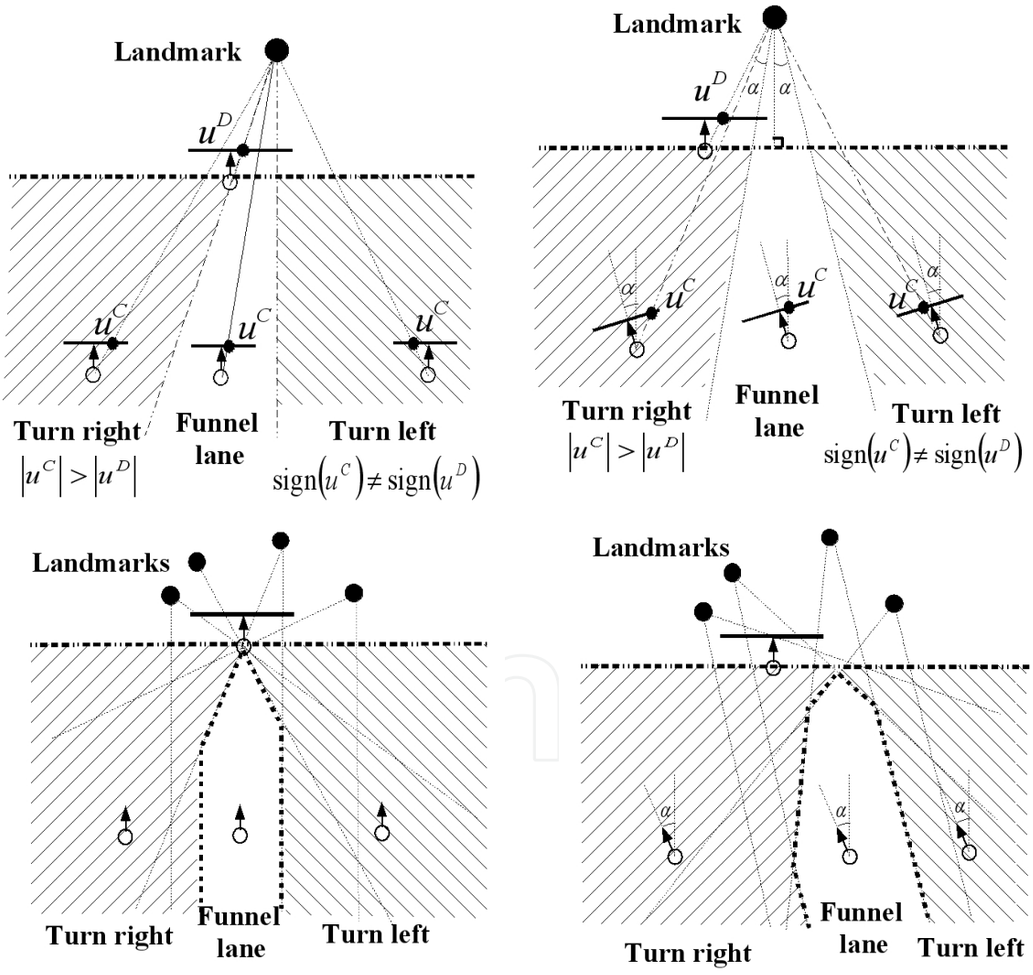


Fig. 2. TOP: The funnel lane created by the two constraints, shown when the robot is facing the correct direction (left) and when it has turned by an angle α (right). BOTTOM: The combined funnel lane created by multiple feature points, shown when the robot is facing the correct direction (left) and when it has turned by an angle α (right).

According to the preceding theorem, if the robot is on the path toward the destination with the same heading direction, then two constraints are satisfied. Conversely, as shown in Fig. 2, if the constraints are satisfied then the robot lies within a trapezoidal region (assuming perspective projection) for any given relative robot angle $\alpha = \theta_C - \theta_D$. For $\alpha = 0$, the sides of the trapezoid are defined by two lines passing through the landmark, one through D and another that is parallel to the destination direction. These lines are rotated about the landmark by α if the relative angle is nonzero. We call the trapezoidal region the *funnel lane* associated with the landmark, destination, and relative angle. The terminology arises from the analogy of pouring liquid into a funnel: The liquid moves in a straight line until it hits the sides of the funnel, which cause it to bounce back and forth until it eventually reaches the spout. In a similar manner, the sides of the trapezoid act as bumpers, guiding the robot toward the goal. The notion of the funnel and the funnel lane are captured in the following definitions.

Definition 1: The *funnel* of a fixed landmark λ and a robot location D is the set of locations $\mathbb{F}_{\lambda,D}$ such that, for each $C \in \mathbb{F}_{\lambda,D}$, the two funnel constraints are satisfied:

$$\left|u^C\right| < \left|u^D\right| \qquad \text{(Constraint 1)}$$

$$\operatorname{sign}\left(u^C\right) = \operatorname{sign}\left(u^D\right) \qquad \text{(Constraint 2)}$$

where u^C and u^D are the coordinates of the image projection of λ at the locations C and D, respectively.

Definition 2: The *funnel lane* of a fixed landmark λ, a robot location D, and a relative angle α is the set of locations $\mathbb{F}_{\lambda,D,\alpha} \subset \mathbb{F}_{\lambda,D}$ such that $\theta_C - \theta_D = \alpha$ for each $C \in \mathbb{F}_{\lambda,D,\alpha}$.

Multiple features yield multiple funnel lanes, the intersection of which is the set of locations for which both constraints are satisfied for all the features. This intersection, which we call the *combined funnel lane*, is depicted in Fig. 2. Notice the importance of having features on both sides of the image in order to narrowly constrain the path of the robot, thus achieving more robust and accurate results. Features can be at any depth, and there need not be any relationship between the depths of the various features as long as they remain visible.

2.2 Qualitative control algorithm

The funnel constraints lead to a simple control algorithm, illustrated in Fig. 3. The robot continually moves forward, turning to the right whenever Constraint 1 is violated and to the left whenever Constraint 2 is violated, given a feature on the right side of the image ($u^D > 0$). If the feature is on the left side ($u^D < 0$), then the directions are reversed.

For each feature *i*, a desired heading is obtained by

$$\theta_d^{(i)} = \begin{cases} \gamma \min\left\{u^C, f\left(u^C, u^D\right)\right\} & \text{if } u^C > 0 \text{ and } u^C > u^D \\ \gamma \max\left\{u^C, f\left(u^C, u^D\right)\right\} & \text{if } u^C < 0 \text{ and } u^C < u^D \\ 0 & \text{otherwise} \end{cases}$$

where $f\left(u^C, u^D\right) = \frac{1}{\sqrt{2}}\left(u^C - u^D\right)$ is the signed distance to the line $u^C = u^D$. Here we approximate the conversion of pixels to radians with a constant gain γ.

At any given time, the desired heading of the robot is given by

$$\theta_d = \eta \frac{1}{N} \sum_{i=1}^{N} \theta_d^{(i)} + \left(1 - \eta\right)\theta_0 \tag{1}$$

where N is the total number of feature points, θ_o is the desired heading obtained by sampling a third-order polynomial that is fit to the initial and destination odometry measurements of the segment in the teaching phase, and the factor $0 \le \eta \le 1$ determines the relative importance of visual measurements versus odometry measurements. We set η = 0.5 in our system.

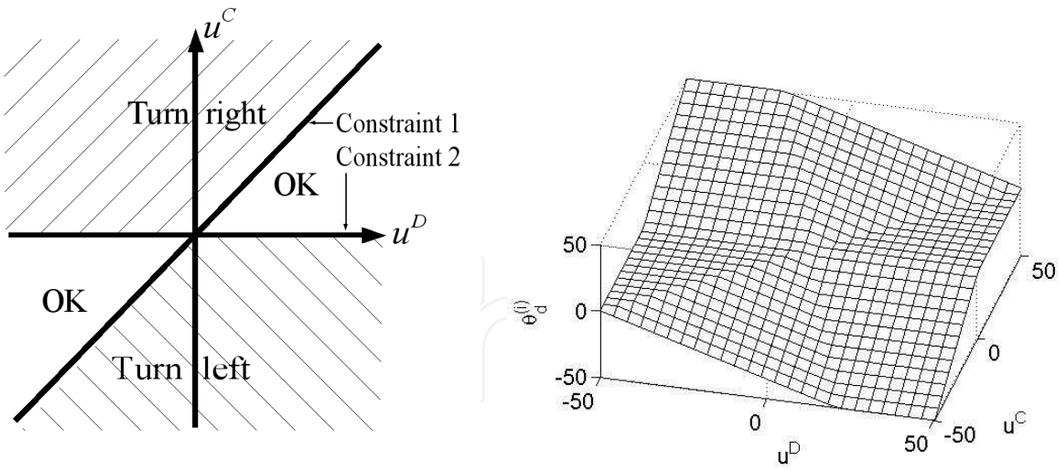


Fig. 3. Qualitative control decision space. The horizontal coordinates of the feature point in the current and destination images (u^C and u^D, respectively) are compared to determine whether to turn the robot to the right, to the left, or not at all. LEFT: Top-down view of decision space. RIGHT: 3D view of decision space, showing the desired angle $\theta_d^{(i)}$ versus u^C and u^D.

2.3 Analysis of qualitative control algorithm

Fig. 4 illustrates the qualitative control algorithm with an example. In its initial position the robot is outside the funnel lane, violating Constraint 1 (Fig. 4a). The robot turns to the right, causing the funnel lane to rotate as well, and the robot moves forward a small amount until the constraint is violated again (Fig. 4b). The robot turns a second time to the right, finds itself with a much clearer opening, and moves forward until the constraint is violated (Fig. 4c). Finally, the robot turns again and moves forward until it reaches a point close to the goal (Fig. 4d).

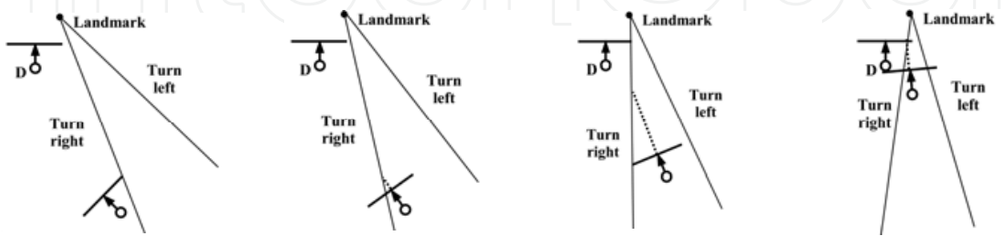


Fig. 4. Four snapshots of a robot making progress toward a destination D using the qualitative control algorithm. The two solid lines indicate the funnel lane, while the dashed line indicates the path of the robot.

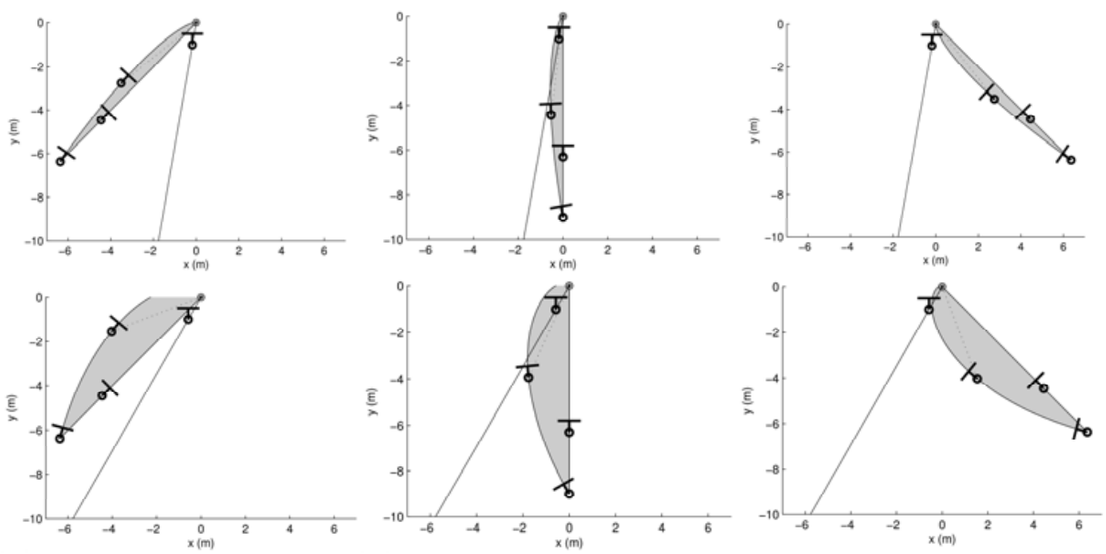
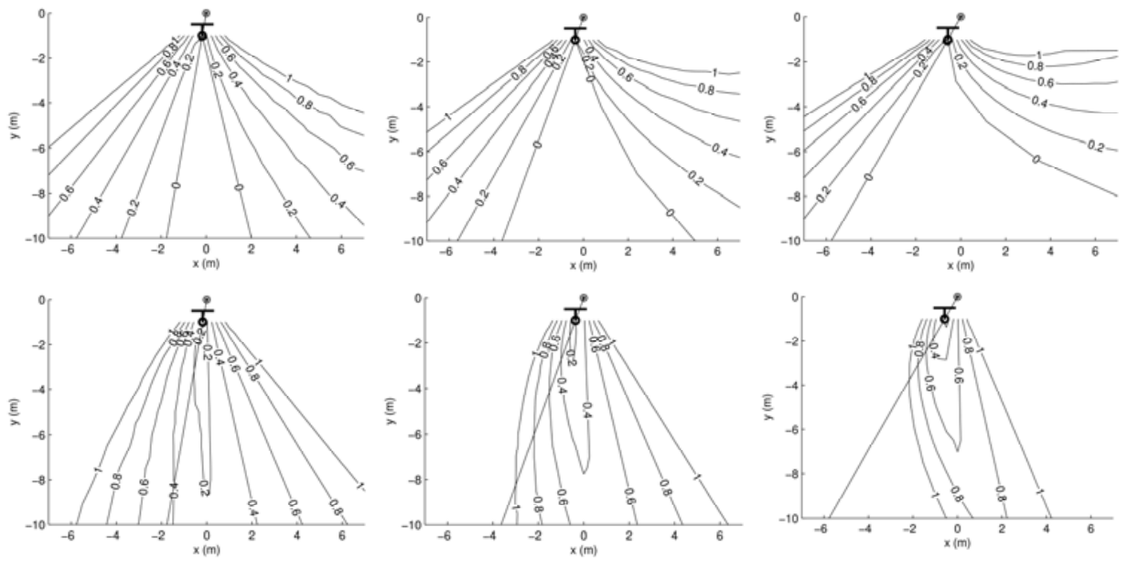


Fig. 5. The reachable set of positions (gray region) from three different initial positions (left, middle, and right) and two different values of ϕ (top: 10 degrees, bottom: 30 degrees). The landmark is at (0, 0), the initial position of the robot is at the bottom tip of the gray region, and the projection ray from the landmark to the destination is the angled line. Two possible robot locations along edges of the reachable set are shown, along with a possible location for the destination.

To better understand the behavior and accuracy of the algorithm, simulations were ran in Matlab, the results of which are shown in Fig. 5. A single landmark was placed at the origin, and the robot was placed at various initial positions for different values of ϕ (the angle of the landmark with respect to the optical axis). From any initial position the robot may turn and drive straight toward the landmark, in which case it will barely satisfy Constraint 2. Alternatively it may turn away from the landmark and drive along a curve so that Constraint 1 is always barely satisfied. In both cases the other constraint is automatically satisfied. This line and curve define a region of positions, shown as gray in the figure, that are reachable from the initial position by a non-holonomic vehicle without violating either constraint. Notice that the actual location of the destination along the projection ray is irrelevant for the plots, which depend only upon the starting location, the landmark location, and the angle ϕ that the light ray makes with respect to the destination optical axis. The reachable set is wider for increasing values of ϕ.



ϕ = 10 ϕ = 20 ϕ = 30

Fig. 6. Contour plots of the minimum (top) and maximum (bottom) error in reaching a destination from any point in the plane. The destination is 1 m in front of the landmark, which is placed at the origin. To reduce clutter, contours with a value greater than 1.0 are not shown. The curves labeled 0 enclose the region of zero error.

Fig. 6 displays the data in a different format, showing the minimum and maximum error in reaching the goal from various initial positions. With the landmark still at the origin, we placed the destination at the intersection of the line y = −1 m with the projection ray from the landmark at a given ϕ. The set of possible initial locations was densely sampled in order to generate contour plots of the error, as shown. The error was computed as the distance from the robot to the destination when the robot crossed the line y = −1 m. As can be seen, the probability of reaching the destination with zero error increases with larger values of ϕ, but the probability of large errors increases as well. As long as the robot starts from a position nearly behind the destination at a reasonable distance, the minimum error is zero and the maximum error is approximately 20-50% of the distance from the destination to the

landmark. Keep in mind that in a real system, the use of multiple landmarks dramatically reduces this error.

4. Tracking feature points

Feature points are automatically selected and tracked using the Kanade-Lucas-Tomasi (KLT) feature tracker (Birchfield, 1997), which computes the displacement $\mathbf{d} = \begin{bmatrix} d_x & d_y \end{bmatrix}^T$ that minimizes the sum of the squared differences between consecutive image frames I and J:

$$\iint_W \left[I\left(\mathbf{x} - \frac{\mathbf{d}}{2}\right) - \left(\alpha\ J\left(\mathbf{x} + \frac{\mathbf{d}}{2}\right) + \beta\right)\right]^2 d\mathbf{x}, \tag{2}$$

where W is a window of pixels around the feature point and $\mathbf{x} = \begin{bmatrix} x & y \end{bmatrix}^T$ is a pixel in the image. This nonlinear error is minimized by repeatedly solving its linearized version by Taylor series expansion:

$$\begin{aligned} Z &= \sum_{\mathbf{x} \in W} g(\mathbf{x}) g^T(\mathbf{x}), \\ \mathbf{e} &= \sum_{\mathbf{x} \in W} g(\mathbf{x}) \left[I(\mathbf{x}) - (\alpha\ J(\mathbf{x}) + \beta) \right], \end{aligned} \tag{3}$$

where $g(\mathbf{x}) = \frac{1}{2} \frac{\partial \left[I(\mathbf{x}) + \alpha\ J(\mathbf{x}) \right]}{\partial \mathbf{x}}$ is the spatial gradient of the weighted average image. These equations are the standard Lucas-Kanade equations (Baker & Matthews, 2004, Shi & Tomasi, 1994, Tomasi & Kanade, 1991) with geometric symmetry between the two images and an affine model of brightness to model the dynamic lighting conditions encountered by the mobile robot, particularly when moving outdoors (Chen & Birchfield, 2006, Negahdaripour & Yu, 1993). A coarse-to-fine pyramidal strategy is used to allow large image motions. As in (Shi & Tomasi, 1994, Tomasi & Kanade, 1991), features are automatically selected as those points in the image for which both eigenvalues of Z are greater than a specified minimum threshold. This feature selection mechanism is a slight variation of the Harris corner detector which has been shown to be effective for both its repeatability rate, information content, and theoretical properties (Harris & Stephens, 1988).

5. Teach-and-replay navigation

The navigation system involves two phases. In the teaching phase, an operator manually moves the robot along a desired path to gather training data. The path is divided into a number of non-overlapping segments. Within each segment, feature points are automatically detected in the first image and tracked in subsequent images. When the percentage of features that have been successfully tracked falls below 50% of the original features in the segment, a new segment is declared. For each feature that is successfully tracked throughout a segment, its gray-level intensity pattern and x-coordinate in the first and last images of the segment are stored in a database for use in the replay phase. We also

store the length of each segment and the change of heading direction of the robot in each segment by odometry, which are used in determining the desired heading and the segment transitions.

In the replay phase, the robot automatically proceeds sequentially through the segments starting from approximately the same initial location as that of the teaching phase. At the beginning of each segment, correspondence is established between feature points in the current image and those of the first teaching image of the segment. Then, as the feature points are tracked in the incoming images, their coordinates are compared with those of the *milestone image* (i.e., the last teaching image of the segment) in order to determine the turning direction for the robot. Prior to comparison, feature coordinates are warped to compensate for a non-zero roll angle about the optical axis by applying the RANSAC algorithm (Fischler & Bolles, 1981) to pairs of random features. This compensation removes the undesirable in-plane image rotation that occurs due to unpaved, rough terrain. Note that this is the only place in the algorithm where the *y*-coordinates of the features are used.

A crucial component of the technique is determining when to transition to a new segment. To solve this problem, we continually monitor the probability that the robot at time *t* is at the end of the current segment:

$$\delta(t) = \underbrace{\exp\left\{-\frac{\varepsilon_f^2(t)}{2\sigma_f^2}\right\}}_{\text{feature}} \underbrace{\exp\left\{-\frac{\varepsilon_d^2(t)}{2\sigma_d^2}\right\}}_{\text{distance}} \underbrace{\exp\left\{-\frac{\varepsilon_h^2(t)}{2\sigma_h^2}\right\}}_{\text{heading}} \tag{4}$$

assuming that the feature, distance, and heading measurements are independent. In this equation $\varepsilon_f(t)$ is the mean squared error of the feature coordinates between the current and milestone images; $\varepsilon_d(t)$ is the difference between the distance traveled in the current segment and the corresponding segment in the teaching phase, calculated by odometry; and $\varepsilon_h(t)$ is the difference between the current heading and the heading at the end of the teaching segment. These errors are normalized by values computed automatically by the system: σ_f is the mean squared error of the feature points at the beginning of the segment; σ_d is the length of the segment calculated by odometry in the teaching phase; and σ_h is the maximum variation in heading encountered during the teaching segment.

Two values are actually computed: $\delta(t)$ using the current milestone image and $\delta^-(t)$ using the previous milestone image. If $\delta(t-1) - \delta(t) > \tau$ and $\delta^-(t-1) - \delta^-(t) > \tau$, where τ = 0.05, then the system advances to using the next milestone image. The rationale is that both $\delta(t)$ and $\delta^-(t)$ increase as the robot approaches the end of the segment then decrease afterward. Therefore, when both values have decreased by a significant amount, the end has been reached. We have found that using both values yields improved results compared with using a single value. To reduce the effects of noise, both signals are first smoothed by a low-pass nonlinear filter.

6. Experimental results

The qualitative algorithm was implemented in Visual C++ on a Dell Inspiron 700m laptop (1.6 GHz) controlling an ActivMedia Pioneer P3-AT mobile robot with an inexpensive Logitech QuickCam Pro 4000 webcam mounted on the front. The 320×240 images were acquired at 30 Hz and processed by the KLT algorithm with the default 7 × 7 feature window size (Birchfield, 1997). In all experiments a maximum of 60 features were detected and tracked throughout each segment. On average 85% of the features survive the initial correspondence in the first image of the segment during replay.

The algorithm was tested in a number of indoor and outdoor environments. (Videos of the results can be found at http://www.ces.clemson.edu/~stb/research/mobile_robot .) Figs. 7 and 8 show two typical runs in which the robot successfully navigated between chairs and desks along a 10 m path in our laboratory, as well as along a 380 m loop trajectory in a parking lot of our university campus. The driving speed of the robot was 100 mm/s and the turning speed was 4 degrees per second during both the teaching and replay phases of the indoor experiments. Outdoors, the additional maneuvering room enabled the driving and turning speeds to be increased to 750 mm/s (the maximum driving speed of the robot) and 6 degrees per second, respectively. The error was less than 1 m for two-thirds of the sequence and remained below 3.5 m for the entire sequence.

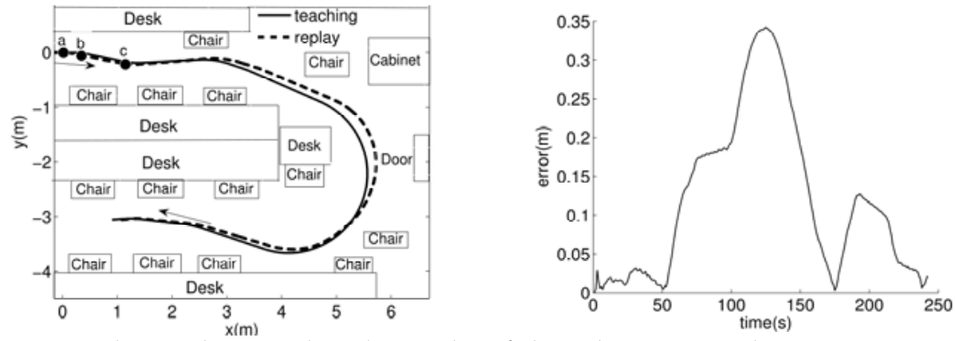


Fig. 7. LEFT: The teaching and replay paths of the robot in an indoor environment. The locations a, b, and c are used in Fig. 9. RIGHT: Error versus time.

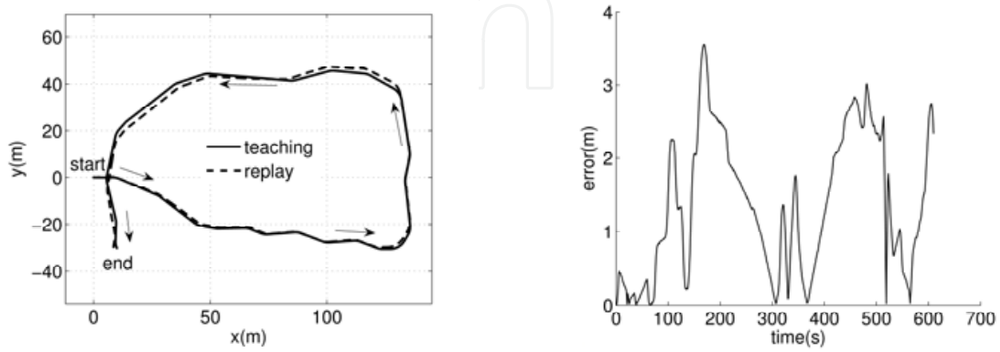


Fig. 8. LEFT: The teaching and replay paths of the robot in an outdoor environment (parking lot). RIGHT: Error versus time.

The decision process during three segments of the experiment is shown in Fig. 9. In the first segment all the features lay in the OK region, so the robot drove forward. In the remaining two segments a majority of the features told the robot to turn right and left, respectively, which caused the features to move toward the OK region. Notice the near unanimity in voting: Except for a lone feature in the second segment that votes incorrectly to turn left, all the features were in agreement in all segments. In this figure the display is simplified by assuming $\tau = 0$ and by normalizing the feature coordinates:

$$\xi = \begin{cases} u_i^C / u_i^D & \text{if } u_i^D \geq 0 \\ 1 - u_i^C / u_i^D & \text{otherwise} \end{cases} \tag{5}$$

so that the interval $0 \leq \xi \leq 1$ indicates "do not turn", larger values ($\xi > 1$) indicate "turn right", and smaller values ($\xi < 0$) indicate "turn left".

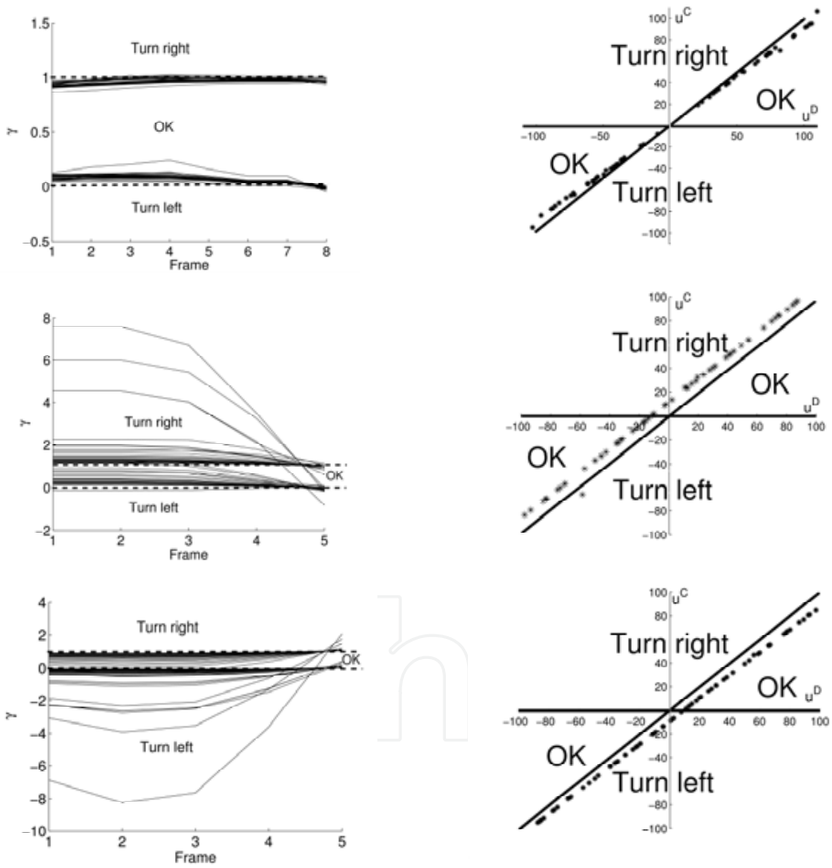


Fig. 9. LEFT: The normalized feature coordinates of all the features plotted versus the image frame number for three segments of the indoor experiment. Features below 0 vote for "turn left", while those above 1 vote for "turn right". RIGHT: A snapshot of the features from frame 2 of each segment plotted on the qualitative control decision space to show the instantaneous decision. The three segments correspond to the points a, b, and c from Fig. 7, in which the robot respectively did not turn, turned right, and turned left.

Fig. 10 shows sample images from two experiments demonstrating the robustness of the algorithm. In the first, the robot navigated a slanted ramp in a 40 m run, thus verifying that the algorithm does not require a flat ground plane. In the second, the robot navigated a narrow road for 80 m while a pedestrian walked by the robot and later a van drove by it. Because the milestone images change frequently, the algorithm quickly recovered from the loss of features due to the occlusion caused by the dynamic objects.

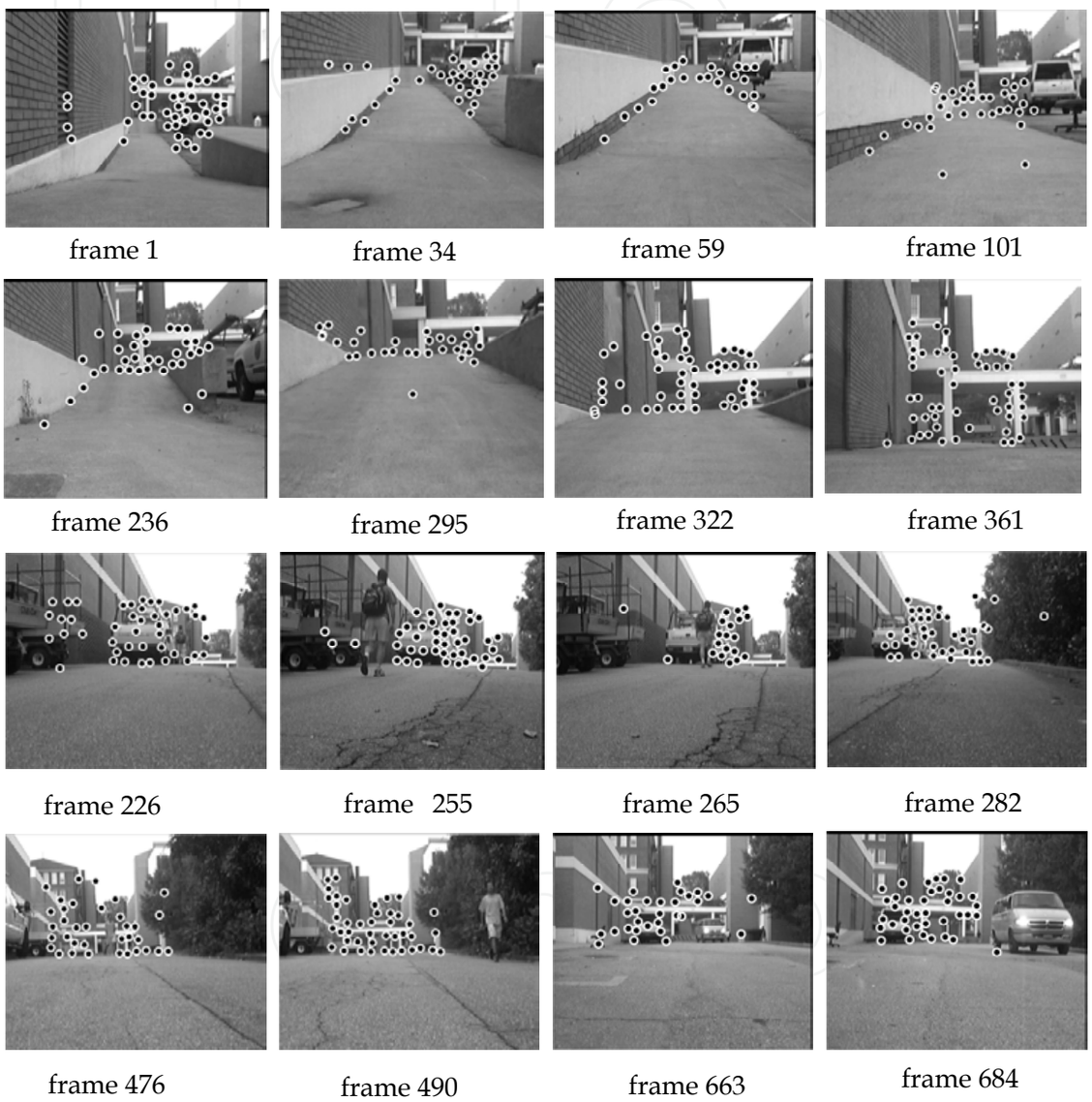


Fig. 10. Sample image frames from two different sequences, one in which the robot traveled down and up a ramp (top 2 rows), and the other containing dynamic objects (bottom 2 rows). The circles indicate the features.

Similarly, Fig. 11 shows the results of the approach using cameras with severe lens distortion. In one experiment we used a wide-angle camera with a 3.5 mm focal length and

110-degree field of view. The other experiment utilized an omnidirectional camera with a 360-degree field of view. For both experiments we used the same parameters as the previous experiments. The only change made to the code was to discard the bottom half of the omnidirectional donut image. This step was necessary because features behind the robot (whether viewed by an omnidirectional or standard camera) move in a way that violates the fundamental assumptions of our approach. In contrast, features in front of the camera obey the funnel constraints sufficiently to be of use in keeping the robot on the path, despite their moving in curved image paths due to the severe lens and catadioptric distortion. The average error of the two experiments was 0.04 m and 0.04 m, respectively, while the maximum error was 0.13 m and 0.09 m.

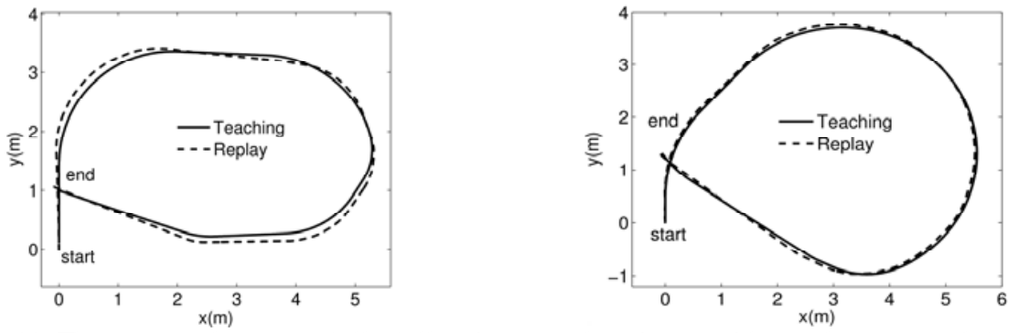


Fig. 11. The approach successfully following a path using a wide-angle camera (left) and an omnidirectional camera (right).

To further illustrate the lack of calibration, we conducted an outdoor experiment in which the robot navigated the same 50 m path twice. In the first run the robot used the Logitech Quickcam Pro 4000 camera, while in the second run it used an Imaging Source DFK21F04 Firewire camera with an 8.0 mm F1.2 lens. The same camera was used for both teaching and replay. As shown in Fig. 12, the algorithm was able to successfully follow the path using either camera, without changing any parameters between runs.

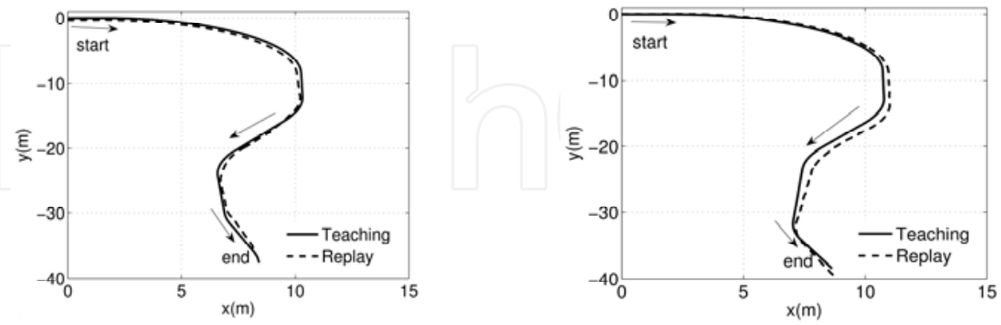


Fig. 12. Teaching and replay paths for the robot using two different uncalibrated cameras, with the same system parameters. LEFT: Logitech QuickCam Pro 4000 USB webcam, RIGHT: Imaging Source DFK 21F04 Firewire camera.

Three additional experiments are shown in Fig. 13. In the first, a scout robot was sent along an outdoor path. Another robot, which received the transmitted path information, was then able to follow the same path as the scout. This demonstrates the natural application of

swarm robotics, where calibrating dozens or hundreds of cameras would be prohibitive, especially if recalibration is needed whenever the lenses are refocused or the cameras adjusted. The second experiment shows the robot following a path along rough terrain, in which roll and tilt angles up to 5 degrees were encountered. The roll angle compensation described earlier was sufficient to enable the robot to remain on the path. In the third, a path with several sharp turns is demonstrated. This ability is achieved by setting the replay driving speed to be that of the teaching driving speed, which is decreased during a turn.

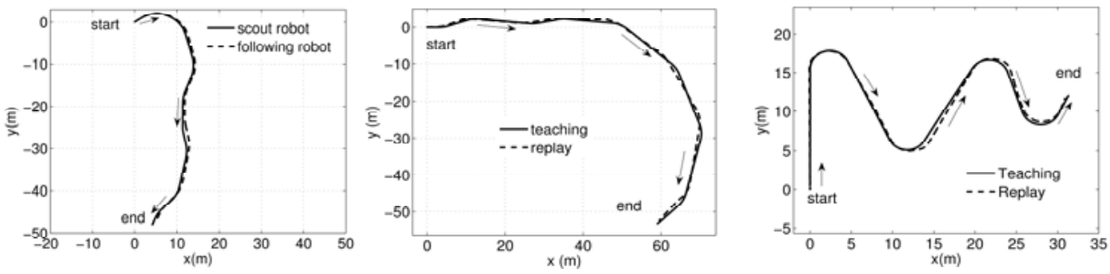


Fig. 13. LEFT: The robot followed a path taken earlier by a scout robot. MIDDLE: A path on rough terrain. RIGHT: A path with sharp turns.

Additionally, the algorithm was tested in various scenarios to quantitatively measure its accuracy and repeatability. Table I displays the results of the algorithm compared with those of the earlier version (Chen & Birchfield, 2006) which did not use odometry, relied upon a bang-bang control scheme, and did not compensate for the camera roll angle. The algorithms were tested in three environments: a 15 m path in an indoor laboratory environment with rich texture for feature tracking, a 60 m trajectory in an outdoor paved parking lot, and a 40 m path along unpaved terrain. In each case, we conducted ten trials and recorded the final 2D location of the robot for each trial: $\{\mathbf{x}_i\}_{i=1}^{n}$, where $\mathbf{x}_i \in \Re^2$ and n = 10. Accuracy was measured as the RMS Euclidean distance to the final ground truth location: $\sqrt{\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i - \mathbf{x}_{gt}\|^2}$. Repeatability was measured as the standard deviation of the final locations: $\sqrt{\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i - \mu\|^2}$, where $\mu = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$. While the earlier algorithm works well when the ground is paved and the scenery is rich in texture, the improved algorithm is more robust, achieving maximum errors of only 0.23 m, 1.20 m, and 1.76 m, respectively, compared with 0.45 m, 1.20 m, and 5.68 m for the earlier algorithm.

Algorithm	indoor acc. / rep. (m) / (m)	outdoor paved ground acc. / rep. (m) / (m)	outdoor rough terrain acc. / rep. (m) / (m)
Vision only (Chen & Birchfield, 2006)	0.30/0.18	0.77 / 0.74	3.87 / 1.85
Combination (this work)	**0.14 / 0.08**	**0.60 / 0.55**	**1.47 / .0.66**

Table 1. Comparison of the accuracy and repeatability of the algorithm with an earlier version, in three different scenarios. The lowest number in each case is in bold.

The algorithm assumes that the robot is placed in the same initial location in both the teaching and replay phases. To test the sensitivity to this assumption, we conducted an experiment with a fairly straight teaching path outdoors, with the background approximately 50 m from the initial location. The robot was then placed at different initial locations for the replay phase, deviating laterally from the initial teaching location by 0 m, 0.5 m, 1.0 m, 1.5 m, and 2.0 m. To ensure overlap between the teaching and replay features, the initial robot orientation was adjusted accordingly. The results, shown in Fig. 14a, show that the robot converged to the teaching path in all cases, reducing the error in half after approximately 20 m. With closer backgrounds, the convergence was faster. In a similar experiment, the robot was placed 0 m, 0.5 m, 1.0 m, 1.5 m, and 2.0 m ahead and behind the initial teaching location. Fig. 14b plots the deviation in the estimate of the position of the robot along the path versus the driving distance along the teaching path. Although these results exhibit more noise, the errors reduce over time, requiring about 6 m for convergence for locations in front, and approximately 20 m for locations behind.

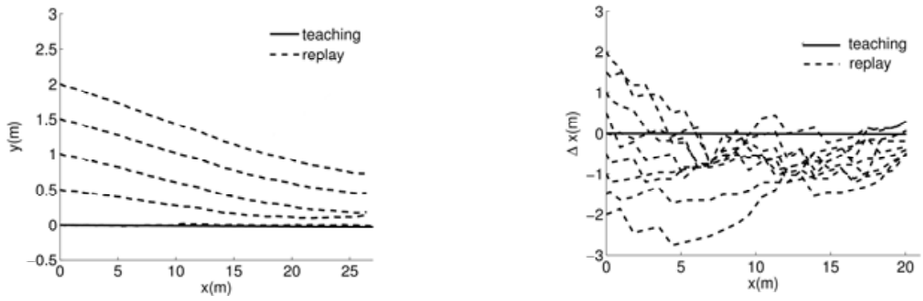


Fig. 14. Sensitivity to the initial location. (a) The replay path of the robot versus time, starting from different locations deviating laterally from the initial teaching location. (b) The deviation of the robot along the replay path versus time, starting from different deviations along the path from the initial teaching location.

Another way for the robot to deviate from the desired path is the presence of an obstacle. While avoiding an obstacle, the robot will rotate in such a way that there may be no overlap between the current image and the milestone image. To solve this problem, we use odometry to return the robot to the original path, after the obstacle has been avoided. (Obstacle detection is accomplished using a ring of sonars.) By comparing the current features with the milestone features, the robot is then able to locate the segment which has the minimum mean squared error between the current features and the milestone features. Once this segment has been determined, the path-following algorithm resumes. An experiment of this capability is shown in Fig. 15.

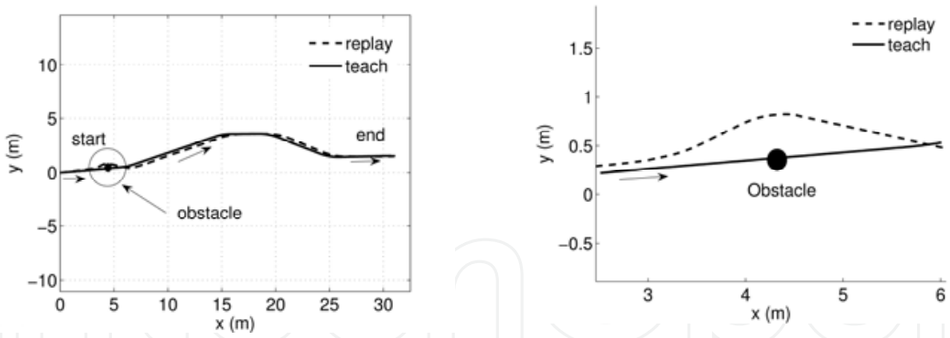


Fig. 15. LEFT: The teaching and replay paths of the robot in an outdoor environment (parking lot), with an obstacle blocking the path. RIGHT: Close-up of the section of the path containing the obstacle.

6. Discussion

Because our system does not explicitly model the geometric world, its geometric accuracy is limited. Therefore, when compared with map-based approaches using calibrated cameras (Royer et al., 2007), the errors exhibited by the simple control scheme of our algorithm are rather large. Nevertheless, the remarkable flexibility and versatility of the system offer some important advantages over more precise techniques. With our approach, one can literally take an off-the-shelf camera, attach it to the robot, align it approximately in the forward direction, and start the system. The algorithm is not perfect, and there are scenarios in which it will fail. For example, occasionally the algorithm does not properly transition to the next milestone image, in which case the overlap between the current and milestone image can decrease to the point that an insufficient number of features are matched. Also, untextured scenes containing distant trees, bushes, or undecorated indoor hallways sometimes prevent the KLT algorithm from successfully tracking enough features to accurately compute the heading direction. While only a handful of features are necessary for the algorithm to succeed, it is important that features exist on both sides of the image, and that some number of features remain visible throughout the milestone. Another source of error is due to distant features. Although features near the center of the image produce a narrow funnel lane even when they are far from the camera, distant features near the side of the image produce much larger funnel lanes which are less useful for navigation. Moreover, image parallax is inversely proportional to the distance to a feature. As a result, distant features are primarily useful for correcting the rotation of the robot and are quite incapable of informing the robot about minor translation errors. This problem is compounded by the inherent ambiguity between rotation and translation in the funnel lane itself. Even though this ambiguity has little effect when the robot is near the path, it hinders the ability of the visual information to correctly determine the correct amount of rotation when the robot has deviated significantly. Odometry helps to overcome this limitation, and we have conducted experiments in which the robot consistently returns to the path after deviating by several meters. However, much larger deviations either initially or during replay cannot be handled by our present system. At any rate, it should be noted that odometry drift is not an issue because we only store odometry values local to the segment, not in a global coordinate frame.

7. Conclusion

In this chapter we have presented a novel approach to the problem of vision-based mobile robot path following using a single off-the-shelf camera. The robot navigates by performing a qualitative comparison of feature coordinates across the teaching and replay phases, utilizing the novel concept of a *funnel lane*. Vision information is combined with odometry for increased robustness. The algorithm does not make use of the traditional concepts of Jacobians, homographies, fundamental matrices, or the focus of expansion, and it does not require any camera calibration, including lens calibration. It only requires implicit calibration in the form of a controller gain. Experimental results on both indoor and outdoor scenes demonstrate the effectiveness of the approach on trajectories of hundreds of meters, along with its robustness to effects such as dynamic objects, slanted surfaces, and rough terrain. The versatility of the algorithm in working with wide-angle and omnidirectional cameras with only minor modification has also been shown. Future work should be aimed at incorporating higher-level scene knowledge to enable obstacle avoidance and terrain characterization, as well as connecting multiple teaching paths in a graph-based framework to enable autonomous navigation between arbitrary points.

Acknowledgements

This work was partially supported through a Ph.D. fellowship from the National Institute for Medical Informatics.

8. References

Ackerman, C. & Itti, L. (2005). Robot steering with spectral image information. *IEEE Transactions on Robotics*, Vol. 21, No.2, pp. 247–251.

Baker, S. & Matthews, I. (2004). Lucas-Kanade 20 years on: A unifying framework. In *International Journal of Computer Vision*, Vol. 56, No. 3, pp. 221–255.

Barrows, G. L.; Chahl, J. S. & Srinivasan, M. V. (2002). Biomimetic visual sensing and flight control. In *Bristol Conference on UAV Systems*.

Birchfield, S. (1997). KLT: An implementation of the Kanade-Lucas-Tomasi feature tracker, http://www.ces.clemson.edu/~stb/klt/.

Burgard, W.; Cremers, A. B.; Fox, D.; Hähnel, D.; Lakemeyer, G.; Schulz, D.; Steiner, W. & Thrun, S. (1999). Experiences with an interactive museum guide-robot. *Artificial Intelligence*, Vol. 114, No. 1-2, pp. 3–55.

Burschka, D. & Hager, G. (2001). Vision-based control of mobile robots. In *Proceedings of the International Conference on Robotics and Automation*, pp. 1707–1713.

Chen, Z. & Birchfield, S. (2006). Qualitative vision-based mobile robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2686–2692.

Crawford, S.; Cannon, M.; Letourneau, D.; Lepage, P. & Michaud, F. (2004). Performance evaluation of sensor combinations on mobile robots for automated platoon control. In *ION GNSS Conference*, pp. 706–717.

DeSouza, G. N. & Kak, A. C. (2002). Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 2, pp. 237–267.

Fischler, M. A. & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, Vol. 24, No. 6, pp. 381–395.

Gaussier, P.; Joulain, C.; Zrehen, S.; Banquet, J. P. & Revel, A. (1997). Visual navigation in an open environment without map. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. (IROS)*, pp. 545–550.

Giovannangeli, C.; Gaussier, P. & Désilles, G. (2006). Robust mapless outdoor vision-based navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3293–3300.

Guerrero, J. J. & Sagüés, C. (2001). Uncalibrated vision based on lines for robot navigation. *Mechatronics*, Vol. 11, No. 6, pp. 759–777.

Harris, C. G. & Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pp. 147–151.

Horswill, I. D. (1993). Polly: A vision-based artificial agent. In *Proceedings of the National Conference on Artificial Intelligence*, pp. 824–829.

Hutchinson, S.; Hager, G. & Corke, P. (1996). A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 5, pp. 651–670.

Jones, S. D.; Andersen, C. S. & Crowley, J. L.(,1997). Appearance based processes for visual navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 551–557.

Kosaka, A. & Kak, A. C. (1992). Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. *Image Understanding*, Vol. 56, No. 3, pp. 271–329.

Košeckā, J.; Zhou, L.; Barber, P. & Duric, Z. (2003). Qualitative image based localization in indoors environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3–8.

Kröse, B.; Vlassis, N.; Bunschoten, R. & Motomura, Y. (2001). A probabilistic model for appearance-based robot localization. *Image and Vision Computing*, Vol. 19, No.6, pp. 381–391.

LeCun, Y.; Muller, U.; Ben, J.; Cosatto, E. & Flepp, B. (2005). Off-road obstacle avoidance through end-to-end learning. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 739–746.

Liang, B. & Pears, N. (2002) Visual navigation using planar homographies. In *Proceedings of the International Conference on Robotics and Automation*, Vol. 1, pp. 205–210.

Matsumoto, Y.; Inaba, M. & Inoue, H. (1996). Visual navigation using view-sequenced route representation. In Proceedings of the International Conference on Robotics and Automation, Vol. 1, pp. 83–88.

Matsumoto, Y.; Sakai, K.; Inaba, M. & Inoue, H. (2000). View-based approach to robot navigation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vol. 3, pp. 545–550.

Michels, J.; Saxena, A. & Ng, A. Y. (2005). High-speed obstacle avoidance using monocular vision and reinforcement learning. In Proceedings of the Twenty-second International Conference on Machine Learning, pp. 593–600.

Murali, V. N. & Birchfield, S. T. (2008). Autonomous navigation and mapping using monocular low-resolution grayscale vision. In Workshop on Visual Localization for Mobile Platforms (in association with CVPR).

Nayar, S. K.; Murase, H. & Nene, S. A. (1994). Learning, positioning, and tracking visual appearance. In Proceedings of the International Conference on Robotics and Automation, pp. 3237–3244.

Negahdaripour, S. & Yu, C. H. (1993). A generalized brightness change model for computing optical flow. In *Proceedings of the International Conference on Computer Vision*, pp. 2–11.

Nelson, R. C. & Aloimonos, J. (1998). Using flow field divergence for obstacle avoidance towards qualitative vision. In Proceedings of the International Conference on Computer Vision, pp. 188–196.

Remazeilles, A. & Chaumette, F. (2007). Image-based robot navigation from an image memory. *Robotics and Autonomous Systems*, Vol. 55, No.4, pp. 345–356.

Royer, E.; Lhuillier, M.; Dhome, M. & Lavest, J.-M. (2007). Monocular vision for mobile robot localization and autonomous navigation. *International Journal of Computer Vision*, Vol. 74, No. 3, pp. 237–260.

Sagüés, C. & Guerrero, J. J. (2005). Visual correction for mobile robot homing. *Robotics and Autonomous Systems*, Vol. 50, No. 1, pp. 41–49.

Santos-Victor, J.; Sandini, G.; Curotto, F. & Garibaldi, S. (1995). Divergent stereo in autonomous navigation: From bees to robots. *International Journal of Computer Vision*, Vol. 14, No. 2, pp. 159–177.

Šegvić, S.; Remazeilles, A.; Diosi, A. & Chaumette, F. (2007). Large scale vision based navigation without an accurate global reconstruction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8.

Shelton, A. L. & Gabrieli, J. D. E. (2002). Neural correlates of encoding space from route and survey perspectives. *The Journal of Neuroscience*, Vol. 22, No. 7, pp. 2711–2717.

Shen, J. & Hu, H. (2006). Visual navigation of a museum guide robot. In *Proceedings of the* 6th World Congress on Intelligent Control and Automation (WCICA), Vol. 2, pp. 9169–9173.

Shi, J. & Tomasi, C. (1994). Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600.

Sim, R. & Dudek, G. (2001). Learning environmental features for pose estimation. *Image and Vision Computing*, vol. 19, No. 11, pp. 733–739.

Tang, L. & Yuta, S. (2001). Vision based navigation for mobile robots in indoor environment by teaching and playing-back scheme. In Proceedings of the International Conference on Robotics and Automation, pp. 3072–3077.

Tomasi, C. & Kanade, T. (1991). Detection and tracking of point features. In *Technical Report CMU-CS-91-132, Carnegie Mellon University*.

Ulrich, I. & Nourbakhsh, I. (2002). Appearance-based place recognition for topological localization. In *Proceedings of the International Conference on Robotics and Automation*, pp. 1023–1029.

Weng, J. & Chen, S. (1996). Incremental learning for vision-based navigation. In *Proceedings of the IAPR International Conference on Pattern Recognition*, pp. 45–49.

Wolf, J.; Burgard, W. & Burkhardt, H. (2002). Using an image retrieval system for vision-based mobile robot localization. In *Proceedings of the International Conference on Image and Video Retrieval (CIVR)*, pp. 108–119.

Zhou, C.; Wei ,Y. & Tan, T. (2003). Mobile robot self-localization based on global visual appearance features. In Proceedings of the International Conference on Robotics and Automation, pp. 1271–1276.

Mobile Robots Navigation

Edited by Alejandra Barrera

ISBN 978-953-307-076-6
Hard cover, 666 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

Mobile robots navigation includes different interrelated activities: (i) perception, as obtaining and interpreting sensory information; (ii) exploration, as the strategy that guides the robot to select the next direction to go; (iii) mapping, involving the construction of a spatial representation by using the sensory information perceived; (iv) localization, as the strategy to estimate the robot position within the spatial map; (v) path planning, as the strategy to find a path towards a goal location being optimal or not; and (vi) path execution, where motor actions are determined and adapted to environmental changes. The book addresses those activities by integrating results from the research work of several authors all over the world. Research cases are documented in 32 chapters organized within 7 categories next described.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Zhichao Chen and Stanley T. Birchfield (2010). Vision-Based Path Following Without Calibration, Mobile Robots Navigation, Alejandra Barrera (Ed.), ISBN: 978-953-307-076-6, InTech, Available from: http://www.intechopen.com/books/mobile-robots-navigation/vision-based-path-following-without-calibration

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen