Monocular Navigation for Long-Term Autonomy

Tomáš Krajník
Lincoln Centre for Autonomous Systems,
University of Lincoln, United Kingdom
tkrajnik@lincoln.ac.uk

Sol Pedre
Comisión Nacional de Energía Atómica,
Div. de Robótica CAREM, Argentina
sol.pedre@cab.cnea.gov.ar

Libor Přeučil
Faculty of Electrical Engineering
Czech Technical University in Prague
preucil@labe.felk.cvut.cz

***Abstract*—We present a reliable and robust monocular navigation system for an autonomous vehicle. The proposed method is computationally efficient, needs off-the-shelf equipment only and does not require any additional infrastructure like radio beacons or GPS. Contrary to traditional localization algorithms, which use advanced mathematical methods to determine vehicle position, our method uses a more practical approach. In our case, an image-feature-based monocular vision technique determines only the heading of the vehicle while the vehicle's odometry is used to estimate the distance traveled. We present a mathematical proof and experimental evidence indicating that the localization error of a robot guided by this principle is bound. The experiments demonstrate that the method can cope with variable illumination, lighting deficiency and both short- and long-term environment changes. This makes the method especially suitable for deployment in scenarios which require long-term autonomous operation.**

I. Introduction

A considerable progress in visual-based systems capable of autonomous navigation of long routes can be observed during the last decades. One can divide these navigation systems to three groups [1]: map-less, map-based and map-building based. Map-less navigation systems such as [2] rely on recognition of the environment structure and use the current image input to generate motion commands for a mobile robot. Map-based navigation systems rely on user-created models of the environment [3]. Map-building-based navigation systems are able to build an environment map and use it for robot navigation and localization. The mapping and localization is usually performed simultaneously, resulting in so called visual SLAM [4]. Another approach is to perform mapping (typically by guiding the robot along the desired path) prior to the autonomous navigation [5]. This technique (called 'map-and-replay') is similar to the common practice in industrial robotics, where a skilled operator guides a robot tip along a desired trajectory, the robot records positions of its joints and then performs the recorded movement repeatedly.

The articles [6], [7], [8], [9], [10] describe mobile robot navigation systems based on the 'map-and-replay' principle. In the article [8], a monocular camera is carried through an environment and a video is recorded. The recorded video is then processed (in a matter of several minutes) and subsequently used to guide a mobile robot along the same trajectory. Authors of paper [9] present an even simpler form of navigation in a learned map. Their method utilizes a map consisting of salient image features remembered during a tele-operated drive. The map is divided into several conjoined segments, each associated with a set of visual features detected along it and a milestone image indicating the segment end. When a robot navigates a segment, its steering commands are calculated from positions of the currently recognized and the already mapped features. The robot moves forward with a constant speed until it detects the segment end by means of comparing the mapped segment's last image with the current view. However, the authors report low reliability of the segment end detection.

978-1-4799-2722-7/13/$31.00 ©2013 IEEE

This paper presents a simple monocular 'map-and-replay' navigation system for an autonomous vehicle. The core of our system is a simple, yet novel method of position estimation based on monocular vision and odometry. Contrary to traditional localization methods, which use advanced mathematical concepts to determine vehicle position, our method uses a more practical approach. The monocular vision technique determines heading of the vehicle only and leaves the traveled distance estimation to odometry. We claim, *that if the robot heading is continuously adjusted to turn it towards the desired path, its position error does not diverge even if the robot self-localization is based solely on odometry.* To prove the claim, we outline a formal mathematical model of the proposed navigation method and show that the robot position uncertainty is bound for closed polygonal paths. Correctness of the mathematical analysis is supported by experimental evidence, which proves method robustness and stability in real-world scenarios.

The proposed navigation system is based on an off-the-shelf equipment (camera and odometry) and standard image processing algorithms. Since the visual information is used for heading estimation only, the system is able to reliably guide a robot while sensing only one image feature at a time. This results not only in system's robustness to variations of the environment, but also in its computational efficiency. Thus, the navigation method provides competitive properties to traditional methods, which makes it suitable especially for long-term operation. The method's accuracy of 0.3 m surpasses the precision of consumer-grade GPS systems.

II. Navigation method description

The navigation system works in two steps: learning and navigation. During the learning phase, a robot is guided by an operator along a path consisting of straight-line segments and creates a landmark map. Once the mapping is finished, the robot can travel autonomously within the mapped environment.

The image processing method which detects salient objects in the robot's field of view is a critical part of the navigation system because it is the only mechanism which the robot employs to reduce its position uncertainty. We have decided to use Speeded Up Robust Features (SURF) [11], [12] to identify features in the image. However, the system has been tested with other feature extraction algorithms as well [13].

A. Mapping phase

During this phase, the robot is driven through the environment by an human operator in a turn-move manner and creates a map, which consists of a sequence of straight segments. Each segment is described by the initial robot orientation α, the segment length s, and the feature set $\mathcal{L}$. The set $\mathcal{L}$ consists of salient features detected in images captured by the robot's forward looking camera. Each feature $l \in \mathcal{L}$ is associated with its SURF descriptor, the image coordinates where it has been spotted for the first and last time and the robot distance from the segment start at these time instants.

The mapping algorithm maintains three sets of features: a set of tracked features $\mathcal{T}$, a set of currently detected features $\mathcal{S}$ and a set of saved features $\mathcal{L}$. Each time a picture is processed by the SURF algorithm, the set $\mathcal{S}$ is populated and correspondences between the sets $\mathcal{S}$ and $\mathcal{T}$ are established. The descriptions (image coordinates and the current robot position) of the associated features in the set $\mathcal{T}$ are then updated based on the data of their counterparts in the set $\mathcal{S}$. The un-associated features in the set $\mathcal{T}$, i.e. features which have been tracked but are not visible any more, are added to the set $\mathcal{L}$. Similarly, the unmatched features in $\mathcal{S}$, i.e. features seen for the first time, are added to the set $\mathcal{T}$. When the mapping is terminated by the operator, the features in $\mathcal{T}$ are added to $\mathcal{L}$. Thus, each feature in the set $\mathcal{L}$ is described not only by the SURF descriptor, but also by its image coordinates and values of the robot odometric counter in moments of its first and last occurrence.

B. Navigation phase

During autonomous navigation, the robot maintains constant forward speed until its odometry indicates that its distance from the segment start is equal to the segment length. The algorithms retrieves the features which were visible at the same position from the map and matches them to the visible ones. The horizontal displacement (in image coordinates) of the expected and detected features' positions is then used to determine the robot steering speed.

First, a relevant set of features $\mathcal{U}$ is retrieved from the set of mapped features $\mathcal{L}$. The set $\mathcal{U}$ contains features that were detected in the mapping phase at the same robot distance from the segment start. For each feature in the set $\mathcal{U}$, the best matching feature in the set $\mathcal{S}$ (features detected in the current image) is found. A difference in horizontal image coordinates of the paired features is then computed and added to a set $\mathcal{H}$. After that, the most frequent value h of the set $\mathcal{H}$ is found by histogram voting and the robot's steering speed ω is set proportionally to h. Thus, the robot is steered to keep the features at their expected image coordinates. The current onboard camera image, positions of the detected and expected features, established correspondences and histogram are logged and displayed on a GUI, see Figure 1. A detailed description of the algorithm is given in [14].

C. Embedded realization

While the process of heading estimation by histogram voting is fairly simple and computationally inexpensive, the SURF algorithm requires significant computational resources. The CPU version of the algorithm takes about 1300 ms (on an Intel Core2Duo2.0GHz and 1024×768 pixel image) and provides about 1000 features. The GPU version of the SURF algorithm [11] is more efficient due to massive parallel processing - detection of 1000 features takes about 40 ms (on the nVidia Quadro NVS 320M). Even though these methods are applicable, their main drawback is that they require a complete PC, which might be simply too large or power consuming to be used with smaller robots. That is why we have created a small-sized FPGA-based device, which is capable to extract the SURF features while consuming significantly less power [15].

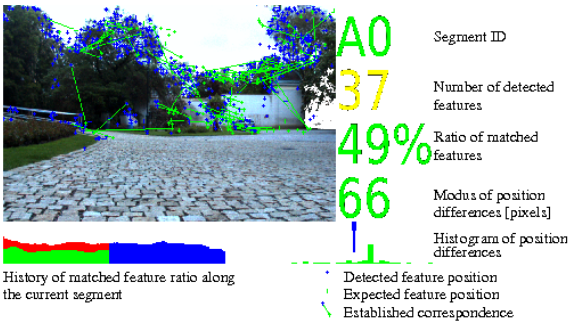


Fig. 1: Robot GUI during navigation phase

III. Navigation Stability

In this section, we show how the heading correction influences the overall position error. At first, we introduce a model of robot position uncertainty and show, how it changes as the robot moves along one path segment. After that, we will extend the model for more segments and show that if the robot traverses a closed path repeatedly, its position error does not diverge.

A. A Model of Robot Movement

Let us define the position and heading of the mobile robot as x, y, φ. Assume that the robot was taught a straight path, which starts at coordinate origin and leads in the direction of the x-axis. Let us assume, that the robot is heading approximately in the direction of the segments, so it can detect some previously mapped features. Let the navigation algorithm assume the robot location to be at $(0, 0)$, but the real robot position is (a_x, a_y), where the values of (a_x, a_y) are small compared to the segment length. The displacement of the robot causes the image features to appear not at their expected positions. Rather than that, the features would be shifted to the right side of the image for $a_y > 0$ or vice versa. This will be reflected by the result of the histogram voting, which will steer the robot towards the segment axis. Larger robot displacement will cause larger displacement of the image features which will result in stronger heading correction calculated by the histogram voting method. Thus, we can estimate the robot heading φ by the formula $\varphi \approx -ky$, where k is a positive nonzero constant. On the contrary, the robot forward speed controller maintains a constant speed v_k until the robot has traveled the distance equal to the segment length and therefore, the robot initial position (a_x, a_y) does not affect the distance the robot travels. Assuming that the robot heading φ is small, we can state that $dx/dt = v_k$ and $dy/dt = v_k\varphi = -v_k ky$.

Solving these differential equations with boundary conditions $x(0) = a_x$, $y(0) = a_y$ allows us to compute the robot position:

$$x(t) = a_x + v_k t, \quad y(t) = a_y e^{-v_k k t}. \tag{1}$$

Taking into account that the time t to traverse a segment of length s is $t = s/v_k$ we can calculate the robot position (b_x, b_y) after it traverses the entire segment as:

$$b_x = a_x + s, \quad b_y = a_y e^{-ks}. \tag{2}$$

Equation (2) would hold for an error-less odometry and noiseless camera. Considering the camera and odometry noise, the Equation (2) will be rewritten to

$$\begin{pmatrix} b_x \\ b_y \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{-sk} \end{pmatrix} \begin{pmatrix} a_x \\ a_y \end{pmatrix} + \begin{pmatrix} s\upsilon \\ \xi \end{pmatrix}, \tag{3}$$

where υ is a random variable drawn from the Gaussian distribution with the mean equal to 1 and the variance equal to ϵ and ξ is a random variable of the Gaussian distribution with the zero mean and the variance τ. A compact form of (3) is

$$\mathbf{b} = \mathbf{M}\mathbf{a} + \mathbf{s}. \tag{4}$$

For an arbitrarily oriented segment, one can rotate the coordinate system by the rotation matrix $\mathbf{R}$ (the initial robot position in the rotated coordinate system will be $\mathbf{R}\mathbf{a}$), apply (4) and then rotate the result back by $\mathbf{R^T}$. Thus, Equation (4) can be rewritten as

$$\mathbf{b} = \mathbf{R^T}(\mathbf{M}\mathbf{R}\mathbf{a} + \mathbf{s}) = \mathbf{R^T}\mathbf{M}\mathbf{R}\mathbf{a} + \mathbf{R^T}\mathbf{s}. \tag{5}$$

Using (5), the robot position $\mathbf{b}$ at the end of the segment can be computed from its starting position $\mathbf{a}$. Equations (5) and (1) allow us to calculate how the robot position error evolves as the robot traverses the intended path.

B. Position Error

Let the robot position $\mathbf{a}$ be a random variable drawn from a two-dimensional normal distribution with the mean $\hat{\mathbf{a}}$ and the covariance matrix $\mathbf{A}$. Since equation (5) has only linear and absolute terms, the position $\mathbf{b}$ will constitute a normal distribution with a covariance matrix $\mathbf{B}$. Let $\mathbf{a} = \hat{\mathbf{a}}+\tilde{\mathbf{a}}$, where $\tilde{\mathbf{a}}$ is a random variable of a normal distribution with a zero mean and covariance $\mathbf{A}$. Assuming the same notation for $\mathbf{b}$ and $\mathbf{s}$, equation (5) can be rewritten as

$$\tilde{\mathbf{b}} = \mathbf{R^T}\mathbf{M}\mathbf{R}(\hat{\mathbf{a}} + \tilde{\mathbf{a}}) + \mathbf{R^T}(\hat{\mathbf{s}} + \tilde{\mathbf{s}}) - \hat{\mathbf{b}}. \tag{6}$$

Substituting $\mathbf{R^T}\mathbf{M}\mathbf{R}\hat{\mathbf{a}} + \mathbf{R^T}\hat{\mathbf{s}}$ for $\hat{\mathbf{b}}$, equation (6) becomes

$$\tilde{\mathbf{b}} = \mathbf{R^T}\mathbf{M}\mathbf{R}\tilde{\mathbf{a}} + \mathbf{R^T}\tilde{\mathbf{s}}, \tag{7}$$

where $\tilde{\mathbf{a}}, \tilde{\mathbf{b}}, \tilde{\mathbf{s}}$ are Gaussian random variables with zero mean. The $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{b}}$ represent the robot position error at the start and end of the traversed segment. To obtain the covariance matrices of random variables $\mathbf{a}$ and $\mathbf{b}$, equation (7) can be rewritten as

$$\tilde{\mathbf{b}}\tilde{\mathbf{b}}^{\mathbf{T}} = (\mathbf{R^T}\mathbf{M}\mathbf{R}\tilde{\mathbf{a}} + \mathbf{R^T}\tilde{\mathbf{s}})(\mathbf{R^T}\mathbf{M}\mathbf{R}\tilde{\mathbf{a}} + \mathbf{R^T}\tilde{\mathbf{s}})^{\mathbf{T}}.$$

Assuming $\tilde{\mathbf{s}}$ and $\tilde{\mathbf{a}}$ are independent and uncorrelated,

$$\tilde{\mathbf{b}}\tilde{\mathbf{b}}^{\mathbf{T}} = \mathbf{R^T}\mathbf{M}\mathbf{R}\tilde{\mathbf{a}}\tilde{\mathbf{a}}^{\mathbf{T}}\mathbf{R^T}\mathbf{M^T}\mathbf{R} + \mathbf{R^T}\tilde{\mathbf{s}}\tilde{\mathbf{s}}^{\mathbf{T}}\mathbf{R},$$

which rewritten in terms of covariance matrices is

$$\mathbf{B} = \mathbf{N}\mathbf{A}\mathbf{N^T} + \mathbf{T}, \tag{8}$$

where $\mathbf{N} = \mathbf{R^T}\mathbf{M}\mathbf{R}$ and $\mathbf{T} = \mathbf{R^T}\mathbf{S}\mathbf{R}$. Equation (8) allows determination of the robot position uncertainty after traversing one path segment.

C. Traversing multiple segments

Let the robot path is a closed polygonal chain consisting of n segments denoted by numbers from 0 to $n-1$. Let a segment i be oriented in the direction α_i and its length be s_i. Let the robot positions at the start and end of the i^{th} segment are $\mathbf{a_i}$ and $\mathbf{b_i}$ respectively. The segments are joined, so $\mathbf{b_i}$ = $\mathbf{a_{i+1}}$ and the Equation (8) for the i^{th} traveled segment is

$$\mathbf{A_{i+1}} = \mathbf{N_i}\mathbf{A_i}\mathbf{N_i^T} + \mathbf{T_i}.$$

The robot position uncertainty after traversing the entire path consisting of the n segments will be

$$\mathbf{A_n} = \breve{\mathbf{N}}\mathbf{A_0}\breve{\mathbf{N}}^{\mathbf{T}} + \breve{\mathbf{T}},$$

where

$$\breve{\mathbf{N}} = \prod_{j=n-1}^{0} \mathbf{N_j} \text{ and } \breve{\mathbf{T}} = \sum_{\mathbf{j=0}}^{\mathbf{n-1}} \left(\prod_{\mathbf{k=n-1}}^{\mathbf{j}} \mathbf{N_k}\mathbf{T_j} \prod_{\mathbf{k=j}}^{\mathbf{n-1}} \mathbf{N_k^T} \right).$$

If the robot traverses the entire path k-times, its position uncertainty can be calculated in a recursive way by

$$\mathbf{A_{(k+1)n}} = \breve{\mathbf{N}}\mathbf{A_{kn}}\breve{\mathbf{N}}^{\mathbf{T}} + \breve{\mathbf{T}}. \tag{9}$$

which has the form of Lyapunov discrete equation. Therefore, the covariance matrix $\mathbf{A_{kn}}$ is bound for $k \to +\infty$ if all eigenvalues of $\breve{\mathbf{N}}$ lie within a unit circle and $\breve{\mathbf{T}}$ is symmetric.

Since matrix $\mathbf{S_i}$ is constructed as diagonal, $\mathbf{T_i} = \mathbf{R_i^T}\mathbf{S_i}\mathbf{R_i}$ is symmetric and $\breve{\mathbf{T}}$ is symmetric as well.

As every $\mathbf{N_i}$ equals to $\mathbf{R_i^T}\mathbf{M_i}\mathbf{R_i}$, its eigenvalues are equal to the diagonal of $\mathbf{M_i}$ and eigenvectors are columns of $\mathbf{R_i}$. Therefore, each matrix $\mathbf{N_i}$ is positive-definite and symmetric. Since the dominant eigenvalue of every $\mathbf{N_i}$ is one, eigenvalues of $\breve{\mathbf{N}}$ are smaller or equal to one. The dominant eigenvalue of $\breve{\mathbf{N}}$ is equal to one if and only if the dominant eigenvalues of products $\mathbf{N_{i+1}}\mathbf{N_i}$ equal 1 for all i. However, dominant eigenvalue of a product $\mathbf{N_{i+1}}\mathbf{N_i}$ equals 1 only if the dominant eigenvectors of both $\mathbf{N_i}$ and $\mathbf{N_{i+1}}$ are linearly dependent, which corresponds to collinearity of i^{th} and $(i+1)^{th}$ segment. Thus, the dominant eigenvalue $\breve{n}$ of the matrix $\breve{\mathbf{N}}$ equals 1 if and only if all path segments are rotated in the same direction, i.e. the entire robot path is a straight line. In any other case, the dominant eigenvalue $\breve{n}$ is lower than 1 and Equation (9) has a finite solution. This means that if the robot travels the trajectory repeatably, its position uncertainty $\mathbf{A_{kn}}$ does not diverge. A detailed analysis of the matrix $\breve{N}$ spectral radius is presented in [16] and [14].

□

IV. Discussion of the Stability Assumptions

The described model of the robot position uncertainty leading to the navigation stability proof stands on several assumptions, which might not always be met in a real situation. First of all, the robot position error, odometry noise υ and camera noise ξ might not have a Gaussian distribution as assumed by the mathematical model. Moreover, the heading correction calculated by the histogram voting might fail in some situations. Besides, the odometry is usually considered as unreliable for long term navigation due to its drift. In this section, we discuss the practical aspects of the heading correction and the odometry error in our navigation system.

A. Incorrect Correspondences

In order to correct the robot heading, we assume that a number of reliable correspondences between the mapped and currently detected features have been established. This assumption might not be met because of four factors: viewpoint changes caused by differences in expected and real robot position, variable illumination due to unstable weather conditions, feature deficiency due to lack of light and naturally occurring seasonal environment changes.

The difference between the expected and real position of the robot leads to different viewpoints of the currently seen and previously learned scene. This affects not only feature image positions, but also their descriptors. However, the viewpoint changes are not significant in our case, because the vehicle keeps itself close to the path it learned before. Moreover, most image feature descriptors are designed to be robust to viewpoint changes.

Varying daytime illumination, which is caused by weather conditions and changing position of the sun causes a significant amount of the image features to be detected in places of shadows rather than real objects. These features are often matched with those of the map, which causes contamination of the set $\mathcal{H}$ with values not corresponding to actual robot heading.

Moreover, the sheer lack of daylight causes the number of detected image features to be too low to establish the robot position properly.

Another significant factor threatening the correct correspondences are naturally occurring seasonal variations, which simply change the environment appearance rendering any map obsolete. This effect is especially significant in non-urban environments, where the natural vegetation grows over time.

The advantage of our method is that it works even in cases, when most of the features are matched incorrectly. The reason for that lies in the determination of the heading correction from the established correspondences. The robustness of the method can be demonstrated by the following example:

Consider a situation, when the robot enter an area that has completely changed since the mapping phase. The established correspondences will be random and thus the set $\mathcal{H}$ will contain random, uniformly distributed numbers. Therefore the histogram, which is used to estimate the modus of $\mathcal{H}$, will have its bins filled approximately equally. Now consider, that the scene has not changed completely and a small portion of the correspondences is correct. Each correctly established correspondence increases the value of the bin, which represents the robot's true heading deviation. Therefore the chance that the correct heading is established increases with each correct correspondence. Thus, only small portion of the correctly established correspondences is sufficient for reliable navigation. Another advantage of histogram voting is that it produces robot heading estimation even in an extreme case of just one visible image feature.

B. Odometry Error

An odometry is usually regarded as suitable only for short-term localization because its error is cumulative. However, the odometry error is caused mainly by the fact that the robot heading cannot be determined precisely. If the odometry is used to measure the traveled distance only, its precision is much higher. Moreover, the odometric error is mostly systematic, which can be solved by calibration. The accumulation of odometric errors can be prevented by using more advanced localization methods at locations, which are not likely to contain wrong correspondences or by taking into consideration the nature of the localization error when planning the robot path [17]. In our navigation method, the odometry is used only for estimation of the traveled distance along (relatively short) straight line segments and therefore its cumulative error is not an issue. The experiments in [14] show that the proposed method is stable even when using an optical odometry with 10% non-systematic error.

V. Experimental evaluation

This section provides an overview of experimental evaluation of the navigation method verifying its reliability and robustness. The performed experiments show that the navigation method can deal with illumination variations, feature deficiency, seasonal and long-term environment changes. The evaluation method is based on experiments in which the robot has to traverse a previously taught closed path several times. Every time the robot finished one path loop, its distance relative to path start was measured. The accuracy of the navigation method is then calculated as a Root Mean Square (RMS) of these distances like in article [9].

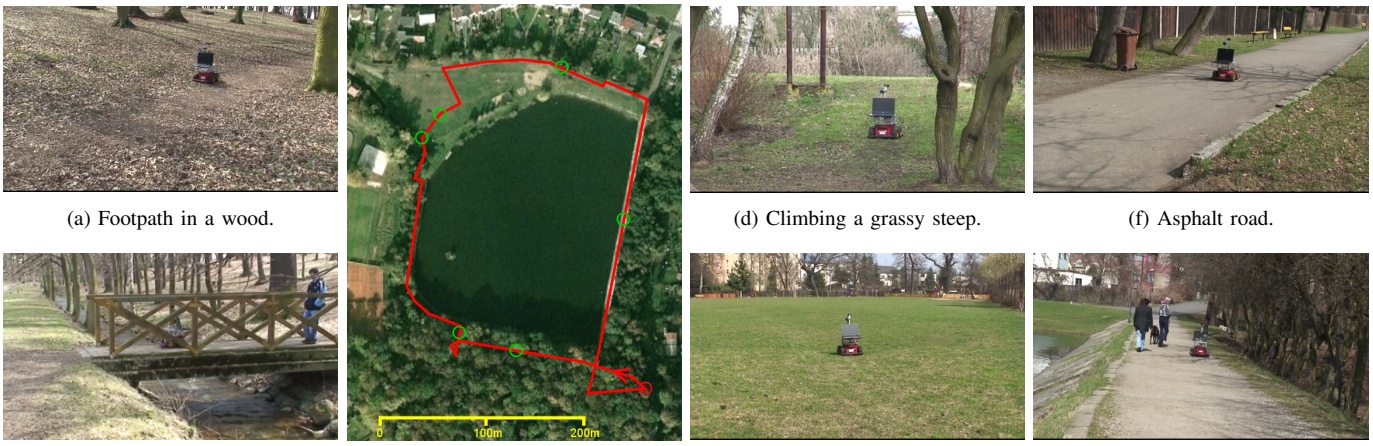
A. Robustness to variable outdoor illumination

The navigation system performance was tested in two all-day experiments. The robot path was 1023 m long and went through variable nonflat terrain consisting mostly of dirtroads and footpaths. The path was learned in the morning and the robot has autonomously traversed it until evening. During the day, the weather has changed from light rain to sunny, which had a significant impact on the environment appearance. Despite the variable lighting conditions, the robot has traversed the path six times with 0.26 m accuracy. One week later, the experiment (without the mapping phase) was repeated with accuracy of 0.31 m. To illustrate on the test, the various parts of the environment during the robot autonomous navigation are shown in the Figure 2.

B. Robustness to feature deficiency

Due to the system principle, it should be able to operate in an environment which contains only a low number of landmarks. To verify this assumption, we have taught the robot a 300 m long path on paved roads in a residential area. The path has been taught at midnight so more than 90% of the mapped landmarks were streetlamps and illuminated windows, see Figure 3. Due to lack of light, the on-board camera iris has been fully opened and the camera exposure time has been set to a fixed value 0.37 s.

After the path has been learned, the robot has been placed 1.5 m away from the path start and requested to traverse it ten times. On the contrary to day experiments, in which the robot typically used 150-300 landmarks for navigation, during night, the typical number of detected landmarks was 2. The robot has traversed 3 km with the accuracy of 0.32 m.



(a) Footpath in a wood. (b) Crossing a wooden bridge. (c) Allday experiment path. (d) Climbing a grassy steep. (e) Traversing an open area. (f) Asphalt road. (g) Pond dam dirtroad.

Fig. 2: The one-day experiment path and terrain examples.

(a) The most SURF populated view. (b) Approaching car (c) Streetlamp alley (d) Typical night view

Fig. 3: Views from the robot camera during the night experiment.

C. Robustness to short-term environment changes

To verify if the navigation system is able to cope with short-term environment changes, we have performed a one year experiment. In this experiment, the robot was required to traverse a 50 m long path in a park environment using one month old map. This test has been performed repeatedly (on a monthly basis) from November 2009 to October 2010. During this time period, the environment has varied significantly, mainly due to seasonal changes in foliage, see Figure 4. Despite of these changes, the system was able to traverse the path every time with an average accuracy of 0.28 m.

D. Robustness to seasonal environment changes

To examine the map decay effects due to seasonal changes, the robot has taken extra snapshots every month at five key locations of the path it traversed in the aforementioned scenario[1]. While the locations have been almost identical, the robot heading has been slightly different every time. The histogram voting scheme of the algorithm presented was used to estimate the relative heading of the robot between all images taken at the same location. If the estimation has differed more than by 2 degrees from the ground truth, it was considered incorrect and vice versa. In this way, we have verified if a map created during a particular month can be used for navigation of the robot during another month. The success rate of the algorithm in estimating the true robot heading between individual months is shown in Table I.

Note, that the all the fields next to the diagonal indicate a 100% success rate, which means, that one-month old map is suitable for visual based navigation performed by our method. Also note, that the most of the 100% success rate is contained in two square areas around the diagonal from November to April and May to August. This means, that the seasonal changes during these months are not so significant and an autonomous robot operating outdoors throughout the entire year would require just two environment maps.

[1]The dataset capturing seasonal variations is available online at [18]

TABLE I: Heading estimation success rate [%] amongst maps created at individual months

	No foliage							*Foliage*				
	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct
Nov	100	100	60	100	100	100	40	40	80	60	40	80
Dec	100	100	100	100	100	100	100	60	100	80	100	60
Jan	80	100	100	100	100	100	80	40	20	20	40	40
Feb	80	100	100	100	100	100	60	60	60	40	80	60
Mar	80	100	100	100	100	100	40	40	60	40	80	60
Apr	100	100	100	100	100	100	100	80	40	60	60	40
May	60	80	60	80	20	100	100	100	100	100	100	100
Jun	40	80	40	40	60	40	100	100	100	100	100	80
Jul	60	80	20	40	60	40	100	100	100	100	100	80
Aug	60	80	60	60	40	60	100	100	100	100	100	80
Sep	60	80	80	60	60	40	100	100	100	100	100	100
Oct	60	60	60	60	80	20	100	100	100	100	100	100

E. Robustness to long-term environment changes

On April 2011 and March 2012, we have let the robot autonomously navigate the same path again using the maps created during the years 2009 and 2010. Prior to running the navigation algorithm, the robot has matched its currently perceived features to every map from the dataset. The resulting histograms of horizontal feature displacements were evaluated

(a) November 2009 (b) December 2009 (c) January 2010 (d) February 2010

(e) March 2010 (f) April 2010 (g) May 2010 (h) June 2010

(i) July 2010 (j) August 2010 (k) September 2010 (l) October 2010

Fig. 4: The robot's view of the same scene on different months during the long-term experiment.

in terms of their entropy [19], [14] and the map, which produced a histogram with the lowest entropy was selected. After that, the navigation method used the selected map to traverse the learned path. Both times, the robot was able to complete the path 20 times with an average precision of 0.26 m and 0.34 m.

VI. Conclusion

A navigation system based on monocular camera and an odometry was presented in this paper. The method utilizes a map of the environment that is created by the mobile robot prior the autonomous navigation. During the autonomous navigation, a camera input is used to establish the robot heading and odometry is used to measure the traveled distance. The robot heading is estimated simply by performing a histogram voting method on a set horizontal position differences of previously mapped and currently detected SURF features. The theoretical analysis indicates that this kind of navigation keeps the robot position error bound. The experimental results not only confirm that, but also demonstrate the method's robustness to landmark deficiency, variable illumination and seasonal environment changes. These properties make the method especially suitable for long-term navigation of mobile robots in outdoor environments.

Acknowledgments

The work has been supported by the EU projects ICT-600623 'STRANDS' and Czech-Argentinean project 'CLARS-II' AMB12AR022-ARC/11/11.

References

[1] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2002.

[2] P. De Cristóforis, M. Nitsche, and T. Krajník, "Real-time image-based autonomous robot navigation method for unstructured outdoor roads," *Journal of Real Time Image Processing*, 2013.

[3] A. Kosaka and A. C. Kak, "Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties," *CVGIP: Image understanding*, vol. 56, no. 3, pp. 271–329, 1992.

[4] S. Holmes, G. Klein, and D. W. Murray, "A Square Root Unscented Kalman Filter for visual monoSLAM," in *International Conference on Robotics and Automation (ICRA)*, 2008, pp. 3710–3716.

[5] K. Kidono, J. Miura, and Y. Shirai, "Autonomous visual navigation of a mobile robot using a human-guided experience," in *Proceedings of 6th International Conference on Intelligent Autonomous Systems*, 2000.

[6] G. Blanc, Y. Mezouar, and P. Martinet, "Indoor navigation of a wheeled mobile robot along visual routes," in *International Conference on Robotics and Automation*. IEEE, 2005, pp. 3354–3359.

[7] Y. Matsumoto, M. Inaba, and H. Inoue, "Visual navigation using view-sequenced route representation," in *IEEE International Conference on Robotics and Automation (ICRA)*, Minneapolis, USA, 1996, pp. 83–88.

[8] E. Royer, M. Lhuillier, M. Dhome, and J.-M. Lavest, "Monocular vision for mobile robot localization and autonomous navigation," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 237–260, Sep 2007.

[9] Z. Chen and S. T. Birchfield, "Qualitative vision-based mobile robot navigation," in *International Conference on Robotics and Automation (ICRA)*, IEEE. IEEE, 2006, Proceedings Paper, pp. 2686–2692.

[10] S. Segvic, A. Remazeilles, A. Diosi, and F. Chaumette, "Large scale vision based navigation without an accurate global reconstruction," in *IEEE International Conference on Computer Vision and Pattern Recognition, CVPR'07*, Minneapolis, Minnesota, 2007, pp. 1–8.

[11] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, 2008.

[12] N. Cornelis and L. Van Gool, "Fast scale invariant feature detection and matching on programmable graphics hardware," in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*. IEEE, 2008, pp. 1–8.

[13] P. De Cristóforis, "Vision-based mobile robot system for monocular navigation in indoor/outdoor environments," Ph.D. dissertation, University of Buenos Aires, Buenos Aires, 2012.

[14] T. Krajník, "Large-Scale Mobile Robot Navigation and Map Building," Ph.D. dissertation, Czech Technical University, Prague, 2012.

[15] J. Šváb, T. Krajník, J. Faigl, and L. Přeučil, "FPGA based Speeded Up Robust Features," in *IEEE International Conference on Technologies for Practical Robot Applications*. IEEE, 2009, pp. 35–41.

[16] T. Krajník et al., "Simple yet stable bearing-only navigation," *Journal of Field Robotics*, vol. 27, no. 5, pp. 511–533, 2010.

[17] J. Faigl, T. Krajník, V. Vonásek, and L. Přeučil, "On Localization Uncertainty in an Autonomous Inspection," in *International Conference on Robotic and Automation*. Piscataway: IEEE, 2012, pp. 1119–1124.

[18] "Stromovka dataset," [Cit: 2013-03-25]. [Online]. Available: http://purl.org/robotics/stromovka_dataset

[19] H. Szücsová, "Computer vision-based mobile robot navigation," Master's thesis, Czech Technical University, Prague, Czech republic, 2011.