

# Deviation from Trajectory Detection in Vision based Robotic Navigation using SURF and Subsequent Restoration by Dynamic Auto Correction Algorithm

Debraj Ray<sup>a</sup> and Amit Kumar

*PES University, Bangalore 560085, Karnataka, India*

**Abstract.** Speeded Up Robust Feature (SURF) is used to position a robot with respect to an environment and aid in vision-based robotic navigation. During the course of navigation irregularities in the terrain, especially in an outdoor environment may deviate a robot from the track. Another reason for deviation can be unequal speed of the left and right robot wheels. Hence it is essential to detect such deviations and perform corrective operations to bring the robot back to the track. In this paper we propose a novel algorithm that uses image matching using SURF to detect deviation of a robot from the trajectory and subsequent restoration by corrective operations. This algorithm is executed in parallel to positioning and navigation algorithms by distributing tasks among different CPU cores using Open Multi-Processing (OpenMP) API.

## 1 Introduction

Terrain irregularities have been shown to affect wheeled mobile robot navigation [1]. Ground irregularities, slipping of robot wheels or unequal speed of left and right wheels may cause the robot to stray from the trajectory. In this paper we enhance a previous work, which is vision-based robotic navigation using Speeded Up Robust Feature (SURF) [2], by adding a deviation correction algorithm that can restore a robot back to the track. This algorithm is dynamic since it is active throughout the trajectory and auto-corrective because it is initiated automatically, when a deviation occurs.

Image processing itself is a resource expensive and computation intensive task. In order to reduce response time we have designed a parallel architecture using OpenMP [3] which executes the dynamic auto-correction algorithm in parallel to position determination and navigation control.

The rest of the paper is organized as follows; Section II contains a review of previous works. Section III contains the Experiment section where we discuss the dynamic auto-correction algorithm to recover the robot from a deviation and optimization methods to reduce overall response time. Section IV contains the observations and a detailed analysis. Section V contains the conclusion of our work.

## 2 Previous Works

Vision based mobile-robot navigation has emerged as an important topic because of its wide applicability. Mobile

robot navigation involving a teaching and a navigation mode has been demonstrated using SURF [2]. During the teaching mode the robot memorizes a sequence of images of the environment while traversing a path. In the navigation mode it compares the SURF features of omnidirectional images of the environment with the SURF features of the saved images to estimate self-location and direction control. A straight track with a single right turn having a total length of 15m has been used. The maximum error in navigation is recorded as 46.8cm (3.12 %). By deploying the deviation detection algorithm as a parallel task to robot navigation and direction control we aim to improve this work by reducing the error in navigation.

While robotic navigation in a well-defined lane may be relatively easy, traversal in irregular terrains is quite challenging because of the unsteadiness of the robot. A high speed mobile robot has effectively navigated an uneven terrain by determining an optimal desired trajectory using a layered control strategy [4]. Navigation strategy such as obstacle collision avoidance, local path planning and determination of track attack angle has been deployed in navigation of a miniature wheel-track hybrid mobile robot to overcome rough terrain [5]. Navigation planning using 2.5 D egocentric height map to assess drivability based on local height differences on multiple scales has enabled a robot to successfully navigate a challenging arena of rough terrain [6]. Legged robots have been used for traversal of unknown rough terrain [7]. The algorithm is based on stereo images from which depth images are computed using the semi-global matching (SGM) method [7]. Autonomous navigation system has been designed for off road traversal. It uses

<sup>a</sup> Corresponding author: [debrajpesit@gmail.com](mailto:debrajpesit@gmail.com)

long-range vision module to detect and avoid dead-ends and predict the traversability of areas up to 30 meters away [8]. In another work, an optimal landmark placement method has been discussed for minimizing the number of landmarks for which a bound on the maximum deviation of the robot from its desired trajectory can be guaranteed with high confidence [9].

### 3 Experiment

In this experiment we have used a two-wheeled robot as shown in Fig. 1. The wheel motors are controlled by an Arduino board which has a connected Bluetooth module. Mounted on top of the robot is a smart phone which streams live video captured by its camera to a PC via Wi-Fi. The PC performs image processing and decides robot's movement by determining the robot's location in the environment. The PC then sends control information to the Arduino via Bluetooth. Like [2], the robot is taught routes during learning stage which it traverses during navigation stage.

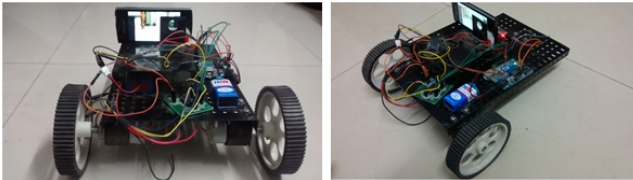


Figure1. Image of the robot

The principal difference between our design of the robot and the design used in [2] is that we have used a smart phone camera which captures a flat or planar image instead of an omnidirectional camera that can capture a 360 degrees cylindrical image. Thus in our present work we locate a robot using only the front view which corresponds to the front region of an omnidirectional image. The front view image is also analyzed and checked for deviation from the trajectory. In case of deviation corrective operations are triggered to bring the robot back to the track. We use an algorithm called dynamic auto correction algorithm for this purpose.

This algorithm should work equally well for the design used in [2], where we can analyze all the four regions of an omnidirectional image separately and in parallel to check for deviation from the trajectory.

#### 3. 1 Dynamic auto-correction algorithm

Fig. 2 illustrates this algorithm. It consists of three sub-images which are themselves composed of two sub-images. The image on the left side is of a landmark that is saved during the learning stage. The image on the right is of the same landmark captured when the robot was navigating to it. The landmark is detected by using SURF based image matching. Both the images are put on an OpenCV canvas (Fig. 2). The matched portion is indicated by enclosing it within a quadrilateral as shown in Fig. 2. Image 1, 2, 3 of Fig. 2 shows the different instances where the robot is on track, deviated towards left and towards right respectively. During navigation the

robot compares the corner coordinates of the matched portion with left and right boundary limits determined by experimentation to decide if there is any deviation. It is observed that when the right corner coordinates are lesser than 810, it indicates a deviation towards right (Image 2; left corner coordinates mean: 266, right corner coordinates mean: 695.5) and when the left corners exceeds 410, it indicates a deviation towards left (Image 3; left corner coordinates mean: 466, right corner coordinates mean: 876). Image 1 indicates the robot is on track with its corner coordinates (left corner coordinates mean: 300, right corner coordinates mean: 850) within boundary values.

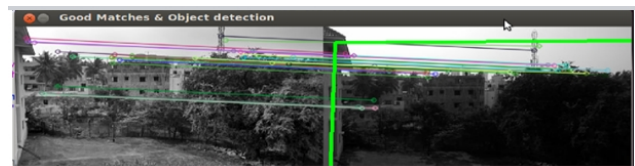


Image 1: Robot is on track



Image 2: Robot has deviated towards right. Corrective action is "turn left".

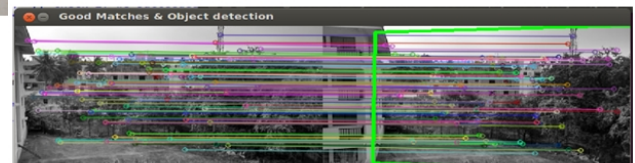


Image 2: Robot has deviated towards left. Corrective action is "turn right".

**Figure 2.** Each of the images shown above consists of two sub images. The left image represent the landmark image which is saved during the learning stage, and the right image is a frame from a live video stream during the navigation stage. Image matching is performed to locate the landmark. Image 2 indicates the right corners of the matched portion are lesser than the right boundary value and Image 3 indicates that the left corners of the matched portion exceeded the left boundary value. In image 1 the scene corners are within left and right boundaries. The boundary values are computed purely by trial and error during experimentation.

#### 3.2 Dynamic auto correction operations

In case of a deviation towards left, the corrective operation performed is turn right until the scene corners are within the boundary values. Similarly, in case of a deviation towards right, the corrective operation performed is turn left until the scene corners are within the boundary values.

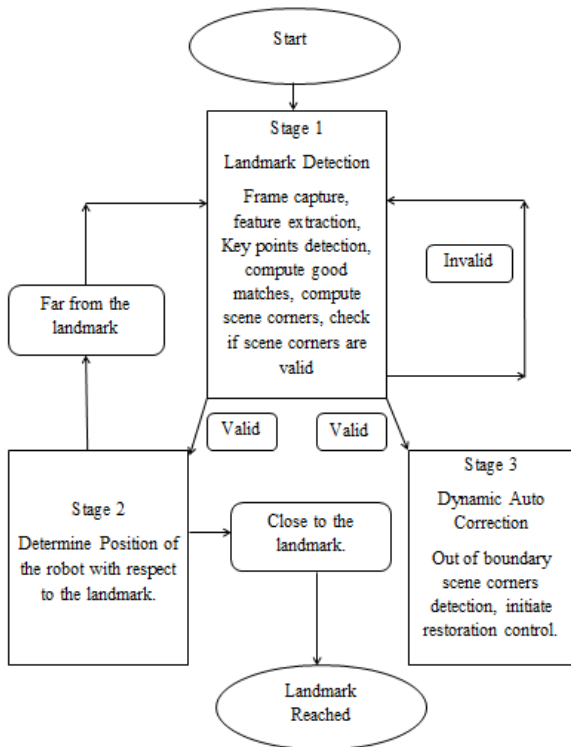


Figure 3. Parallel Architecture

### 3.3 Execution

The main operations during navigation are landmark detection, robot positioning with respect to the landmark, and dynamic auto-correction. These operations are grouped into stages and executed sequentially and with parallel stages. In sequential execution the operations are in the same order as mentioned, i.e. landmark detection, robot positioning, and dynamic auto-correction. In case of parallel execution (Fig. 3) robot positioning and dynamic auto-correction are executed in parallel. Physical Parallelism is achieved by splitting independent stages and assigning them to different CPU Cores. A maximum of two CPU cores process at a time in parallel execution.

### 4 Observation and Analysis

Table 1. Information pertaining to different trajectories through which the robot has navigated  
F-Move forward; L-Turn Left; R- Turn Right; S- Stop

Sl. No.	No of Landmarks	Operation	Distance Travelled in m	Radial displacement	Percentage error
1	5	F,F,R,F,S	8.36	10.032	1.2
2	8	F,F,F,R,F,L,F,S	11.97	21.522	1.78
3	10	F,F,F,F,F,L,F,R,F,S	10.69	15.072	1.41
4	14	F,F,F,F,F,L,F,R,F,F,F,L,F,S	12.23	13.68	1.12

Significant improvement was observed by incorporating dynamic auto correction as a parallel task

in robotic navigation using SURF. Table 1 indicates the radial displacement of the robot from the final destination point, for different number of landmarks and different distances traversed. The maximum percentage error was observed to be 1.78%.

$$\text{Percentage Error} = (\text{Radial displacement from destination}) / (\text{Distance Travelled}) * 100$$

Inaccuracy in positioning introduced due to a deviation from the trajectory tends to propagate. Thus after multiple deviations, the probability that the robot will detect subsequent landmarks falls sharply. By restoring the robot to the trajectory during each deviation, we ensure deviation errors are not propagated and are automatically corrected whenever produced. Thus landmark detection probability remains consistently high and hence high navigation accuracy can be achieved.

Table 2. Average time to process each frame obtained from a live video stream for different architectures incorporated in vision-based robotic navigation with dynamic auto correction

Sl. No.	Architecture Type	Average Time for 1 Frame( in seconds)	Average Time for 50 Frames (in seconds)
1	Parallel	0.662	41.56
2	Sequential	0.758	55.46

Table 2 indicates the execution time for various modes of execution. Execution time was least in the case of sequential execution with parallel stages. An improvement of 0.096 seconds in processing a single frame was observed in parallel architecture compared to a simple sequential run. Whereas for 50 frames, the average processing time in the parallel architecture was 41.56 seconds and in the sequential was 55.46 seconds. Execution time reduced in the case of parallel execution since independent stages executed on separate CPU cores thus achieving physical parallelism.

In the previous work [2], the trajectory was a straight route with a single right turn. However in our experiment we deployed complicated routes with multiple left and right turns (refer to Table 1). Since the robot could reach the destination with good accuracy and minimum failure, this clearly indicates the success of the dynamic auto-correction algorithm.

### 5 Conclusion

Introduction of dynamic auto-correction as a separate task executed in parallel to position detection and navigation control led to a remarkable improvement in mobile robot navigation using SURF. Compared to the previous work [2] an improvement of 1.34% accuracy is obtained. Dynamic auto-correction can ensure traversal of uneven terrain and complicated trajectories with minimum chance of failure. The parallel architecture reduced overhead of processing and time delay in response from the robot. For further enhancement, we can use on board computer for the robot to eliminate time delay involved in Bluetooth and Wi-Fi communication. Upgrading the robot to use an omnidirectional camera

may significantly reduce the error rate in traversal of long complicated trajectories, potentially in irregular terrain.

## References

1. Prado, Maria, Antonio Simón, Ana Pérez, and Francisco Ezquerro. "Effects of terrain irregularities on wheeled mobile robot." *Robotica* 21, no. 02 (2003): 143-152.
2. M. Tabuse, Dai Nakai, "Mobile Robot Navigation using SURF features" in ACS'10 Proceedings of the 10th WSEAS international conference on Applied computer science, October 2010, Wisconsin, USA, pp. 276-279
3. Venkatesan Packirisamy, Harish Barathvajasankar, "OpenMP in Multicore Architectures" Available on: [www.cs.umn.edu/~harish/reports/openMP.pdf](http://www.cs.umn.edu/~harish/reports/openMP.pdf).
4. Iagnemma, Karl, Shingo Shimoda, and Zvi Shiller. "Near-optimal navigation of high speed mobile robots on uneven terrain." In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 4098-4103. IEEE, 2008.
5. Kim, Yoon-Gu, Jeong-Hwan Kwak, Dae-Han Hong, Dong-Hwan Shin, and Jinung An. "Navigation strategy to overcome rough terrain for miniature wheel-track hybrid mobile robot platform." In *Ubiquitous Robots and Ambient Intelligence (URAI), 2012 9th International Conference on*, pp. 571-572. IEEE, 2012.
6. Schwarz, Max, and Sven Behnke. "Local navigation in rough terrain using omnidirectional height." In *ISR/Robotik 2014; 41st International Symposium on Robotics; Proceedings of*, pp. 1-6. VDE, 2014.
7. Stelzer, Annett, Heiko Hirschmüller, and Martin Görner. "Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain." *The International Journal of Robotics Research* 31, no. 4 (2012): 381-402.
8. Hadsell, Raia, Ayse Erkan, Pierre Sermanet, Jan Ben, Koray Kavukcuoglu, Urs Muller, and Yann LeCun. "A multi-range vision strategy for autonomous offroad navigation." *Proc. Robotics and Applications (RA'07)* 1 (2007):7.
9. Beinhofer, Maximilian, Jörg Müller, and Wolfram Burgard. "Landmark Placement for Accurate Mobile Robot Navigation." In *ECMR*, pp. 55-60. 2011.