Real-time Vision-based Lateral Drift Correction

T. Hübner [1] and R. Pajarola[1]

[1]Visualization and MultiMedia Lab, University of Zürich, Switzerland

Abstract

A major drawback in many robotics projects is the dependance on a specific environment and the otherwise uncertain behavior of the hardware. Simple navigation tasks like driving in a straight line can lead to a strong lateral drift over time in an unknown environment. In this paper we propose a fast and simple solution for the lateral drift problem for vision guided robots by real-time scene analysis. Without an environment-specific calibration of the robot's drive system, we balance the differential drive speed on the fly. Therefore, a feature detector is used on consecutive images. Detected feature points determine the focus of expansion (FOE) that is used for locating and correcting the robot's lateral drift. Results are presented for an unmodified real-world indoor environment that demonstrate that our method is able to correct most lateral drift, solely based on real-time vision processing.

Categories and Subject Descriptors (according to ACM CCS): Artificial Intelligence [I.2.10]: Vision and Scene Understanding—Image Processing and Computer Vision [I.4.8]: Scene Analysis—Pattern Recognition [I.5.4]: Applications—

1. Introduction

The adapted behavior of robots in an unknown environment still represents a challenging research task. Robots are most commonly tested in man-made or specifically prepared environments, whereby external impact parameters are known and the robot's behavior is predefined. Alteration of the environment can strongly affect the robot and cause an undefined behavior. Hence environment-adapted behavior based on a regulatory feedback system, e.g. vision, similar to the humans visual and cognitive system is desirable, especially for robot navigation. Visual feedback and a defined navigation objective should control the robot's hardware behavior.

In this paper we are focusing on one specific part of the navigation problem, the lateral drift that occurs over time while trying to drive in a straight line and the vision based correction by real-time scene analysis.

2. Related work

While navigating, a lateral drift over time occurs as the consequence of several impact factors. The mechanical design of the robot is imperfect through the natural variability in materials. Drive motors, though having the same electrical specification, might still turn at slightly different speeds under the same conditions. Additionally, external parameters like wheel traction on different surfaces, distribution of weight on the robot platform, and internal parameters like battery charge influence the robot's drive system behavior.

Thus the challenging task is to design a system that balances all parameters of the drive system in real-time. Two common approaches exist for correcting or preventing lateral drift over time: A sensor-based that relies on hardware to sense and directly correct the mechanic deficiencies, and a vision-based approach that analyzes the scene in real-time and feedbacks the results to control the hardware.

Sensor-based: The simplest sensor-based method is the experimental calibration of the robot's drive system according to external and internal parameters. Lateral drift as well as battery performance are manually measured and fed back as curve offsets into the drive controller. As the drive speed is not balanced on the fly, this method still causes drift and has to be repeated for each environmental change.

Constantly monitoring and updating the motor speeds improves reliability and significantly reduces lateral drift. However, a complex synchronization between the different drive controllers is required, otherwise the robot might drive in a wriggly line. Furthermore, as there is no visual feedback, balancing mechanically the drive speed doesn't necessarily mean that the robot is moving at this speed.

More evolved sensors such as compasses or gyroscopes can determine in real-time the robot's heading direction. This information can be used to adjust for lateral drift, but compasses are strongly influenced by local magnetic fields, thus they are inadequate for indoor environments. Gyroscopes provide more precise heading information but nevertheless generate an accumulation error over time [Con08].

Currently the most convincing sensor-based method is the use of optical mouse sensors [BLB07, PVP06]. Originally intended for high precision input devices, these optical sensors are able to observe a tiny area of the ground surface, e.g. 16×16 pixels at a very high frame rate ≥ 1500 fps, so that even the slightest movements are detected by an optical flow algorithm. As the flow calculation is done on the sensor chip, precise movement information is directly available in real-time. Recent results show a minimal lateral drift, even over a long time. The major disadvantage of these sensors is their fixed focus that defines the distance of the robot platform to the ground surface and limits their application.

Vision-based: It was shown in [SZLC96] that even less complex organisms like bees use cues derived from optical flow for navigational purposes to fly in a straight line by balancing the optical flow field information. Based on this idea, a robot should be able to navigate by optical flow. Calculation of optical flow in real-time is challenging [Cam95] but can be used for navigational purposes [Tel01]. The level of detail for calculating the optical flow in real-time is quite limited, and moreover, the environment needs to be highly textured. In realistic environments it is only possible to determine the optical flow roughly but still use this information for navigation [Seb03].

Because of limited computational resources on compact and low-cost robots, we decided to implement a method that locates and corrects the lateral drift over time by using distinctive feature points and the estimated focus of expansion (FOE).

3. Implementation

3.1. Overview

Figure 1 gives an overview of our method used for determining and correcting the robot's lateral drift.

Our current robot uses a Bumblebee 2 stereo camera from Point Grey Research as optical sensor that is capable of capturing 20 fps with a resolution of 1024×768 pixels. Captured frames are run through a difference filter that limits the number of frames that need to be processed. The frame rate varies after this filter between 0-20 fps depending on the image differences and thus the speed of the robot.

Images that reach the pre-processing stage are decoded and converted to 8-bit grayscale images. Lens distortions are corrected with bilinear-interpolation through a per-pixel look-up table. As the performance of current feature detectors is insufficient for high-resolution images we resize the images from 1024×768 to 512×384. Pre-processing steps are set-up as a multi-threaded pipeline that heavily uses SSE instructions to allow for a maximal throughput.

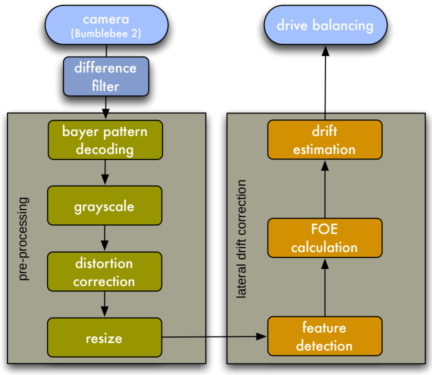


Figure 1: *Pre-processing steps and drift estimation*

Feature points: Our lateral drift correction based on FOE needs good feature point detection. We implemented three different feature detectors and evaluated each for performance and detected feature quality:

KLT: Kanade-Lucas-Tomasi feature tracker [TK91, ST94].
SIFT: Scale Invariant Feature Tracker [Low99].
SURF: Speed Up Robust Features [BETG08].

Our findings correspond mainly to [JB07]. While the detected feature quality is slightly better with SIFT, the performance is, by a factor of 5-6, significantly higher with KLT and SURF. Detected feature quality is lower for KLT as it detects more features but also contains more outliers. As they are performing at a similar speed, we therefore decided to use SURF as a fast and reliable feature detector.

Figure 2 shows the scheme we are using for continuous feature detection and matching. For each sequence of n images, we detect features on the sequence's first and at three other distinctive offsets (i.e. at 10, 15, 20). Feature matches are generated for three sets each involving the first image. This matching procedure results in strongly pronounced movement vectors and an additional robustness against outliers or wrong movement prediction. Outliers are easily recognized by their vector length that should fall in the range defined by the chosen distinctive image positions and their misalignment compared to its closest neighbors.

Based on the three feature-match sets, the focus of expansion is estimated and the image sequence moved to the next position $n+1$.

Focus of expansion: The focus of expansion is the particular point that determines the heading direction for a translational motion. It is equivalent to the epipole, a fixed point

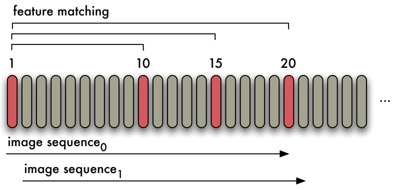


Figure 2: *Frame sequence feature matching*

that has the same coordinates in both images. Feature points in the image sequence can only move along lines emerging from the epipole [CPMH03].

As illustrated in Figure 3, point p_i and p_i' must lie on the same epipolar line arising from the FOE. Thus $(p_i \times p_i') \cdot v = 0$, while v being the focus of expansion. For each pair of corresponding points we obtain a linear equation. The FOE can be calculated from at least two equations. If the data is not exact because of noise in the point coordinates or false detected feature points, then sufficiently many pairs of matching points p_i, p_i' are needed for a good fit of the resulting equation system $A \rightarrow \forall_i : (p_i \times p_i') \cdot v = 0$.

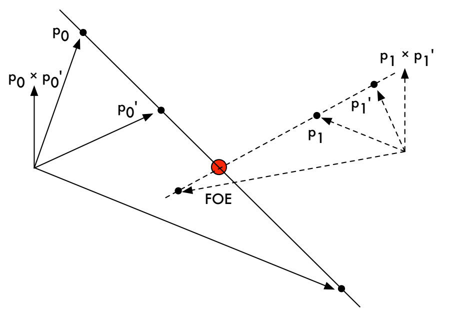


Figure 3: *Calculating the focus of expansion.*

The least squares solution for v of the linear equation system A can be found by singular value decomposition (SVD) and corresponds to the last column of V in: $svd(A) = UDV^T$.

SVD is prone to outliers so we implemented additionally a RANSAC (RANdom Sample Consensus) extension to estimate the FOE. Results (Figure 5) show a higher variance for the RANSAC approach and thereby do not justify the additional computational effort.

3.2. Lateral drift estimation and correction

Each image of a sequence of n images results in the estimation of a FOE vector v, which should vary smoothly over time without abrupt changes between images. To remove falsely estimated FOEs we median filter the three last values of v to enforce such smoothness. Furthermore, minor variations should not result in any unmotivated correction, thus we defined a FOE-center zone in which changes will not result in an immediate drift correction. Outside a value between -1 and 1 is assigned for FOE offsets from the expected center. Hence we have an offset function $f(t) \in [-1, 1]$ over time that indicates the distance of v from the expected image center and a deviation angle. Consequently, the integral over time $F(t) = \int_t f(t)dt$ indicates the actual driving direction and thus its deviation from the intended straight line. If $F(t) \neq 0$ then the robot is experiencing lateral drift. Therefore, $F(t)$ can be used to balance the robot's drive system and limit lateral drift. To drive in a straight line we must keep or continuously strive for $F(t) = 0$. Note that trying to permanently make sure that $f(t) = 0$, which in theory satisfies $F(t) = 0$, in practice only causes the robot to straighten out, but will not correct for the overall deviation in driving direction after an occurrence of $f(t) \neq 0$. Using a feedback system that strives for $F(t) = 0$ can more accurately correct drift.

4. Experimental results

We tested our method in an unmodified indoor environment with a robot platform that features a six wheel differential drive and a MacMini Dual Core 2 for realtime vision processing. Balancing of the drive system is enabled by motor controllers that directly interact with the computer unit.

Examples for FOE estimation are given in Figure 4 and 5. The image center is represented by the yellow line while the dashed white lines delimit the FOE-center zone. Estimated FOEs are shown as colored dots. A lateral drift is not recognized in case of Figure 4 as all FOEs are lying inside the FOE-center zone. This corresponds to the Graph 6(a). The integral $F(t) = 0$ as only small lateral drifts occur. In Graph 6(b) we applied intentionally an unbalanced speed to the robots drive system, resulting in a lateral drift to the left side. The graph illustrates this drift and the integral. Finally, we applied our lateral drift correction method as seen in Graph 6(c). The robot's drive system is configured in the same way as before but as soon as the drift is visually recognized, the robot starts to rebalance its drive system and manages nearly to correct the initial lateral drift. However, balancing the robot's drive system properly has shown to be a challenging problem during our experiments, as oversteering can slightly occur which results in driving in a wriggly line.

5. Conclusion

We have proposed a method for real-time vision-based lateral drift correction based on the focus of expansion that can be used as a feedback system to balance a robot's drive system and to reduce lateral drift over time. Our method is currently limited by the processing power of the robot platform, so we have to compromise between real-time performance and accuracy of the FOE estimation and the number of frames used for the estimation of the drift integral.

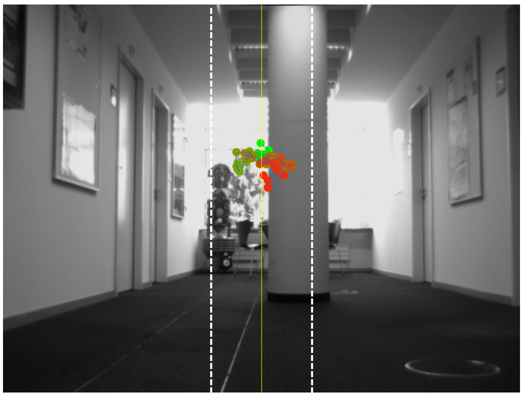


Figure 4: *FOE estimation while driving straight (SVD)*

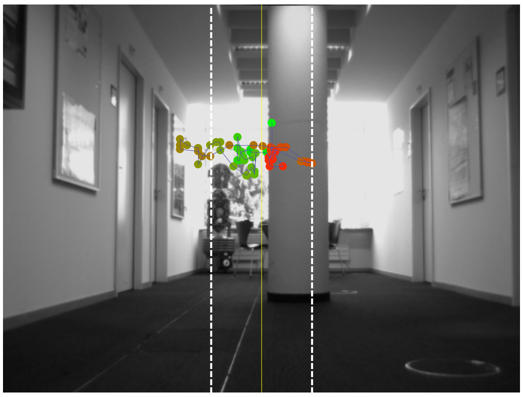
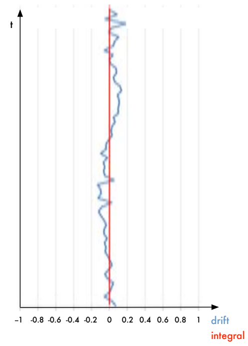
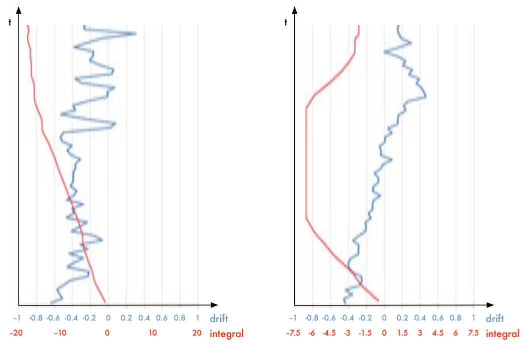


Figure 5: *FOE estimation while driving straight (RANSAC)*



(a) No drift



(b) Drift to left side (c) Real-time drift correction

Figure 6: *Lateral drift and accumulated integral*

References

[BETG08] BAY H., ESS A., TUYTELAARS T., GOOL L. V.: Speeded-up robust features (surf). *Comput. Vis. Image Underst. 110*, 3 (2008), 346–359.

[BLB07] BRADSHAW J., LOLLINI C., BISHOP B.: On the development of an enhanced optical mouse sensor for odometry and mobile robotics education. *System Theory, 2007. SSST '07. Thirty-Ninth Southeastern Symposium on* (March 2007), 6–10.

[Cam95] CAMUS T. A.: Real-time optical flow, 1995.

[Con08] CONRADT J.-A.: A distributed cognitive map for spatial navigation based on graphically organized place agents, 2008.

[CPMH03] CHEN Z., PEARS N., MCDERMID J., HESELTINE T.: Epipole estimation under pure camera translation. In *DICTA* (2003), pp. 849–858.

[JB07] J. BAUER N. SÜNDERHAUF P. P.: Comparing several implementations of two recently published feature detectors. *In Proc. of the International Conference on Intelligent and Autonomous Systems* (Sep 2007).

[Low99] LOWE D.: Object recognition from local scale-invariant features. *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on 2* (1999), 1150–1157 vol.2.

[PVP06] PALACIN J., VALGAŰON I., PERNIA R.: The optical mouse for indoor mobile robot odometry measurement. *Sensors and Actuators A: Physical 126*, 1 (2006), 141 – 147.

[Seb03] SEBASTIEN D.: Low overhead optical flow based robot navigation, 2003.

[ST94] SHI J., TOMASI C.: Good features to track. *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on* (Jun 1994), 593–600.

[SZLC96] SRINIVASAN M., ZHANG S., LEHRER M., COLLETT T.: Honeybee navigation en route to the goal: visual flight control and odometry. *The Journal of Experimental Biology 199*, 1 (1996), 237–244.

[Tel01] TELLZER S.: Optical flow based robot navigation, 2001.

[TK91] TOMASI C., KANADE T.: *Detection and Tracking of Point Features*. Tech. Rep. CMU-CS-91-132, Carnegie Mellon University, April 1991.