Proceedings of the 2006 IEEE/RSJ
International Conference on Intelligent Robots and Systems
October 9 - 15, 2006, Beijing, China

Robust Feature Correspondences for Vision-Based Navigation with Slow Frame-Rate Cameras

Darius Burschka

Lab for Robotics and Embedded Systems/Institute for Robotics and Mechatronics
Technische Universität München/German Aerospace Agency (DLR), Germany
Email: burschka@cs.tum.edu

***Abstract*— We propose a vision-based inertial system that overcomes the problems associated with slow update rates in navigation systems based on high-resolution cameras. Due to bandwidth limitations in current camera interfaces, like Firewire or USB, an increase in the camera resolution results in a drop of the effective frame-rate from the sensor. This increases the correspondence problem for point features in consecutive images due to significant motion of the features. We solve the correspondence problem for very significant motion of point features between consecutive images by analyzing the motion blur to estimate the current motion parameters.**

The proposed algorithm can be used to track the position of point features in video images originating from slow frame-rate cameras. The system is validated on real images from a camera moved by a mobile system, but it can be used for any type of motion ranging from flying systems to mobile robots operating in outdoor or indoor environments.

I. Motivation

Localization is an essential task in majority of applications that involve a sensor moving in space. In a typical navigation task, the 6DoF motion parameters are reconstructed from changes in the projection of physical points in the world in consecutive images taken from a video camera [8] or a laser scanner [10].

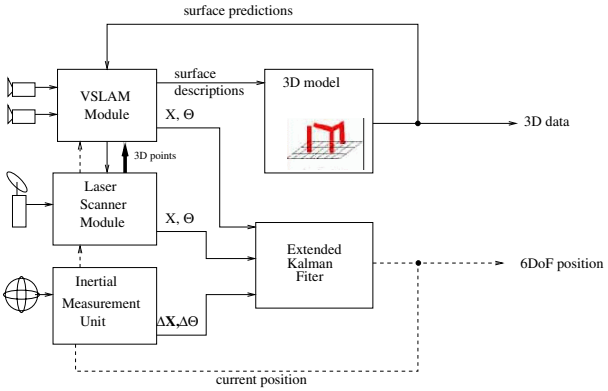

Fig. 1. Information flow in a typical navigation system.

In our case, a typical system used for navigation and 3D reconstruction consists of a laser scanner and a camera system (Fig. 1). The system estimates the motion parameters in a Kalman Filter [13] from a configurable set of sensors. A minimum system configuration consists of a video camera, but its performance can be enhanced by adding additional sensors like inertial units and other sensor types.

In case of an operation in an unknown environment, the system needs to reconstruct a 3D model of the environment in parallel to the localization task to use it as a reference for localization (SLAM - Simultaneous Localization and Mapping). In many applications, a high accuracy of the reconstructed model is required and this implies an accurate localization in all 6DoF.

The localization is based on motion of the corresponding feature points in consecutive frames. The challenge is to robustly identify these points. The motion of the camera relative to the environment results in a depth-dependent motion of the projected points in the camera image. A projected point is represented by its image coordinates as $p = (u, v)^T$. Typically, the largest motion in the image results from a rotation around the focal point or a translation parallel to the image plane. In this case, the horizontal motion Δu (in pixels) of a projection of a point $P = (X, Y, Z)^T$ corresponds to a 3D horizontal motion by ΔX. This relation is described by

$$\Delta u = \Delta X \cdot \frac{f}{p_x \cdot Z} \tag{1}$$

with p_x being the pixel-size and f being the focal length of the camera. We assume a pin-hole camera model here [9]. We can see in (1) that the motion ΔX in 3D is scaled by the intrinsic camera parameters, where high-resolution cameras with small pixel-size p_x result in a larger motion in the image Δu. Features close to the camera (small Z) experience also a larger motion in the projection. Additionally, the parameter $\Delta X = v\Delta t$ grows with the increasing sampling time Δt between the frames given a constant speed v.

In case of a high-resolution camera, the frame-rate can drop to 15, 7.5 or even less frames per second. This results in large displacements between corresponding points in the video stream making the matching more difficult. The search window for possible corresponding points becomes intolerably large reducing the real-time capability of the algorithm. The position of a tracked feature can range from its previous position for a resting system to a position modified by the maximum possible motion ΔX (Δu). This makes the matching more ambiguous, because many similar features may appear in the search region.

For slow frame-rates, the search window can be restricted significantly by estimating the motion of the camera. The motion estimation is done by a Kalman Filter that estimates the position and velocity parameters from sensor readings in the system. Depending on the system costs, the system may operate in different configurations ranging from a single monocular camera to a system equipped with multiple sensors, like inertial sensor (IMU), that support better the search for corresponding points. In case of the reduced system configuration with just a camera, the filter estimates the velocity values from position updates between consecutive frames. This method predicts a future position based on the past motion assuming a smooth change in motion parameters. Sudden changes in the direction of motion cannot be predicted correctly. They require to enlarge the search window for corresponding points to compensate for these unexpected changes.

An addition of an inertial unit (IMU) to the system simplifies this problem by adding motion information about the current change between two image frames (Fig. 1). This allows a robust tracking under significant rotations or translations between the images. An inertial unit operating accurately in 6DoF increases the system costs significantly. We suggest in this paper a cost efficient replacement with a video camera. It uses the *motion blur* effect to predict the motion of the system between the high-resolution acquisitions (Fig. 2).

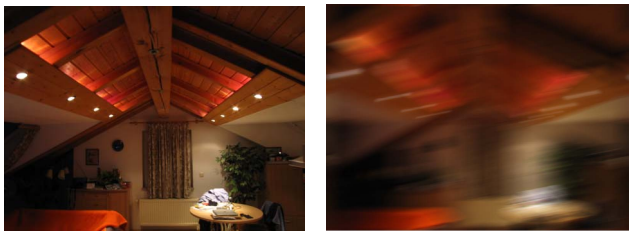

Fig. 2. High-speed shutter and low-speed shutter image of a scene taken from a moving camera.

This usually undesired effect of the *motion blur* helps here to decide the motion of the camera over a longer period of time helping to maintain the correct point correspondences between images.

II. Related Work

In many cases when a scene is observed by a camera there exists motion created either by the movement of the camera or by the independent movement of objects in the scene. In both cases, the goal is to assign a velocity vector D to each visible point in the scene. Such an assignment is called the velocity map. In general, it is impossible to infer from one view the velocity map, however, most motion estimation algorithms calculate the projection of the velocity map onto the imaging surface. A large number of different algorithms have been developed in order to solve this problem. Most motion estimation algorithms consider optical flow with displacements of only a few pixels per frame. Also, most of these algorithms work on a series of images by calculating the displacement of every pixel from image to image ignoring any information about motion that exists within each single image.

One of the first papers on motion estimation from images is the paper of Horn and Schunck in 1980 [5]. The algorithm in this paper can be defined as a differential method. It assumes that the brightness in any particular point in the scene is constant. The velocity constraints are derived from the constant brightness constraint and the smoothness assumption.

In 1988, Aggarwal and Nandhakumar presented a review paper [1] on the calculation of motion. In this paper, they divide the methods used to solve the problem into two different categories: the feature based methods and the optical flow methods. The feature based methods compute the velocities in the scene only in some areas of the image where features, like lines, points or edges, have already been found. Although this kind of method does not generate a continuous field, it is faster and it estimates the velocity of an object by extrapolating from the velocities at its boundary. In general, this approach assumes that all the objects in the scene are rigid and that their movement consists of a translation and a rotation. The algorithms in this class try to define the motion that exists in the scene based on a set of features. Therefore, they use a set of lines and/or points that match during the series of images and calculate the velocities. Variations exist considering the number of features and the number of consecutive images used.

The main difference between these two types of approaches is that the feature based methods require the existence of a match of features among consecutive images before the algorithm is applied. The optical flow methods do not need any feature correspondence to be established. Another difference is that optical flow techniques are very sensitive to noise and this makes their application to real world situations difficult.

In 1992, Barron, Fleet and Beauchemin made a quantitative analysis of the different algorithms that exist for solving the optical flow problem [4]. There are four different categories according to this analysis: one is the differential method, which starts with the Horn and Schunck algorithm and continues with the Lucas and Kanade algorithm. The other category is a region based method where a correlation type algorithm of Anandan [6] is used, which is iterative and calculates the optical flow from a coarser to finer result. The third category is an energy based approach, which is using the output of velocity tuned filters with the calculations in the frequency space.

In opposite to these approaches, we propose an new approach to the problem of visual motion estimation. The algorithm is based on interpreting the motion blur to estimate the optical flow field in a single image and to use this information to predict the position of the tracked features in consecutive images of a video stream. The motion estimation is not used to estimate the 3D motion in the environment. It merely helps to reduce the search areas for corresponding landmarks.

The remainder of this paper is structured as follows. In Section II, we compare our system to other systems based on motion effects in images for motion estimation. In Section III,

we present two alternatives how to estimate the motion parameters from the motion blur. In Section III-B, we present a conventional method to estimate motion parameters from the Power Spectrum of the point spread function (PSF) followed by an alternative approach that uses point feature traces to estimate the motion directly in the image domain (Section III-C). We conclude in Section V with an evaluation of the presented system and we outline our future goals.

III. Approach

We propose a solution to robust estimation of correspondences between camera images taken at a very slow frame-rate. As we mentioned already in Section I, we propose to replace an expensive inertial unit with a standard video camera operating with a slow shutter speed. We use a daylight filter to reduce the brightness of the scene to prevent the CCD chip from over-saturating during operations under bright light conditions, since we need to switch the camera to a long exposure time.

A. Underlying Motion Estimation

In our global navigation approach, we track natural landmarks using their image position to calculate the pose changes of the camera. The system does not depend on any specific tracking algorithm, and has been used with a variety of tracking algorithms relying on color, pattern or depth information encoded in the image. These algorithms are discussed in [12]. The only assumption is that the result of tracking a landmark is reported as a position of a fixed point of a target in the image. The proposed extension allows to establish the correct correspondences even with very long time intervals between the image acquisitions. The scope of this paper is an enhancement of the tracking to cope with the low update rates. Details of the navigation system are described in [7] in more detail.

There have been a number of papers on the process of selecting useful feature points or natural markers in image data [14], [15], [16], [17], [18], [19], [20].

The key problem is to estimate the motion of the camera as it observes the scene. Several approaches have been proposed to recover motion from a set of correspondences [2], [3]. Our system makes use of the pose estimation algorithm described in [11] adapted to deal with 2D projections of points instead of their 3D coordinates. The algorithm is described in [7] in more detail. We give here a short summary for better understanding of the processing in the system.

The position of any three-dimensional point P_i in space can be described as

$$\begin{aligned} P_i^*(\alpha_i, \beta_i, r_i) &= \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = r_i \cdot \begin{pmatrix} \cos\beta_i \cos\alpha_i \\ \sin\beta_i \\ \cos\beta_i \sin\alpha_i \end{pmatrix} \\ &= r_i \cdot \mathbf{n_i} \end{aligned} \tag{2}$$

where (α_i, β_i) are the azimuth and elevation angles of the point P_i projected onto the spherical projection plane of the sensor and r_i is the distance to the point along the ray of projection $\mathbf{n_i}$.

The projection of a 3D-point P_i changes due to the motion of the camera. The 6 DoF motion can be described with a rotation matrix $\tilde{\mathbf{R}}$ around the three-axes of the coordinate frame and a 3D translation vector $\mathbf{T}$ as

$$P_i^* = r_i \cdot n_i^* = \tilde{R} \cdot P_i + \vec{T} \tag{3}$$

We estimate the rotation matrix R and the translation vector T in a recursive approach. In this way, we reconstruct the 3D trajectory in space with all 6DoF. The presented extension to establish robust correspondences helps to predict the correct region in the image where to search for corresponding features. The motion of the features can be very significant because of the slow frame-rate of the system used for image acquisition.

We are planning to estimate a guess of the 3D trajectory directly from the small motion segments that were detected in the blurred image.

B. FFT-based Motion Segmentation

In an unblurred case, each pixel in a camera image should represent the intensity of a single stationary point in front of the camera. If the shutter speed is too slow and the camera is in motion, a given pixel will be an amalgam of intensities from points along the line of the camera's motion (Fig. 2). The two-dimensional mathematical expression for blurring can be best described in the frequency domain to

$$G(u, v) = I(u, v) \cdot H(u, v), \tag{4}$$

where $I(u, v)$ represents the unblurred image, $H(u, v)$ represents the blurring function also known as point spread function (PSF), and $G(u, v)$ describes the blurred result. The PSF represents in the spatial domain the path which the camera took during the exposure. It is a known property of the Fourier Transform that a multiplication in the frequency domain corresponds to a convolution in the spatial domain. The PSF shows which neighboring pixels contributed to the blurred pixel representation at each image point.

Given the angle of motion in the image α and the length of the projected motion $d = V \cdot \Delta t$, which is the number of scene points that affect a specific pixel during exposure, the point spread function (PSF) in the spatial domain can be estimated to

$$h(x, y) = \begin{cases} \frac{1}{d}, & 0 \leq |x| \leq d \cdot \cos\alpha \wedge y = \sin\alpha \cdot x \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

For the case of horizontal motion along the u axis, the Fourier transform of $h(u, v)$ from (5) can be estimated to

$$H(\omega, \nu) = \frac{\sin \pi d\omega}{\pi d\omega} = \mathrm{sinc}(\pi d\omega) \tag{6}$$

In general, for an arbitrary direction of the motion the FFT of the PSF is a ripple as shown in Fig. 3.

It is clear that $H(\omega, \nu)$ is a periodic function with period T=1/d and, therefore, every 1/d there exists a zero crossing of the function. The convolution operation in the space domain

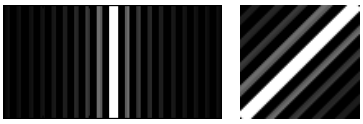

Fig. 3. Power Spectrum of the PSF for a horizontal and a 45° angle motion.

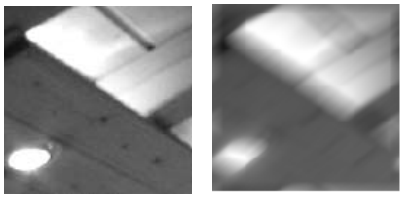

Fig. 4. Magnified patch and its blurred representation during camera motion.

is transformed into the multiplication of two matrices in the frequency domain. As a result, the periodic Power Spectrum of the blur PSF appears as a ripple in the Power Spectrum of the blurred image. This ripple can be identified by a negative peak in the Cepstrum domain. The Cepstrum C_f is the Fourier transformation of the log spectrum of an image. Therefore, it is a tool for analyzing the frequency domain of an image

$$C_f = \mathcal{F}^{-1}\left\{log|F(u,v)|\right\}, \tag{7}$$

with $F(u,v) = \mathcal{F}\{f(x,y)\}$ being the Fourier Transform of $f(x,y)$.

For a tracked feature, like the one in Fig. 4, we can calculate the Cepstrum of the blurred image to Fig. 5. We can see from the width of the peak that the blur was caused by a motion in the image plane by approx. 20 pixels at an angle of approx. 40°.

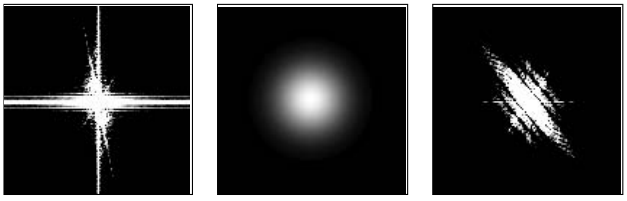

Fig. 5. (left) the Power Spectrum for the unfiltered blurred patch from Fig. 4, (middle) the Gaussian mask used to reduce windowing artifacts in the spectrum, (right) the ripple structure visible in the Power Spectrum of the masked patch.

We have already shown in (1) that different points in the image experience different shifts between consecutive frames. Therefore, the motion analysis needs to be performed on small image regions that represent a part of a rigid object. Different masking functions can be used in order to extract only a patch from an image. The more abrupt is the change into the zero level at the boundary of a patch, the more severe are the artifacts that are going to appear in the frequency domain. There are several masking functions known from the signal processing theory. A masking of the patch region with a Gaussian mask appears to give good results in removing the artifacts due to the periodical extension of the signal (Fig. 5).

C. Direct Recovery of PSF

As we mentioned in Section III-B, the PSF describes the motion of the camera. It shows the path of pixels contributing to the pixel appearance (5) of every pixel in the image. Ideal features for the estimation of the PSF are point features with high contrast to their surroundings.

The best features in indoor scenes are for example lamps, like the light sources shown in Fig. 6, but it is not the only feature that can be used for the PSF estimation.

Fig. 6. Natural point features, like e.g. the lights in the presented images, allow a direct recovery of the PSF function through thresholding and erosion.

For example, the trace of the clock on the wall can be used as a hint for the motion of the camera as well (Fig. 7). We can see that the diameter of this feature does not allow to recover the details of the camera motion during the exposure, but in this application, we are not interested in deblurring the image with the estimate of the PSF. The brightness change in the high-resolution image is used to define an appropriate threshold to generate the trace of the tracked object. like the clock in Fig. 7. Our goal is to capture the motion of the projection of a geometrical point in the scene. For this task, the detection even from the trace of the clock gives sufficient information.

The starting point of each trace is the position of the investigated feature in the high-resolution image taken without blurring.

Fig. 7. Magnified region around the wall clock from the top left image in Fig. 6 after a thresholding operation (see Fig. 2 for a high-speed image of the scene).

D. Properties of the PSF

The reader should notice that the PSF is not constant for the entire image, but it varies for different parts of the image. For example, a forward motion creates an ”explosion field” with the motion vectors pointing radial away from the vanishing point of the motion (Fig. 8). The magnitude of the vectors depends on the distance of the imaged point to the camera and the magnitude of its motion component in 3D that is coplanar to the image plane (1). For this reason, the estimation of the

PSF that is described in Sections III-B and III-C needs to be performed for several points in the image to get the correct estimation of the motion of a tracked feature. In our system, the motion vectors are estimated directly for the point features used in the tracking algorithm to estimate the pose change of the camera in a later processing. The patches are 64x64 pixel large.

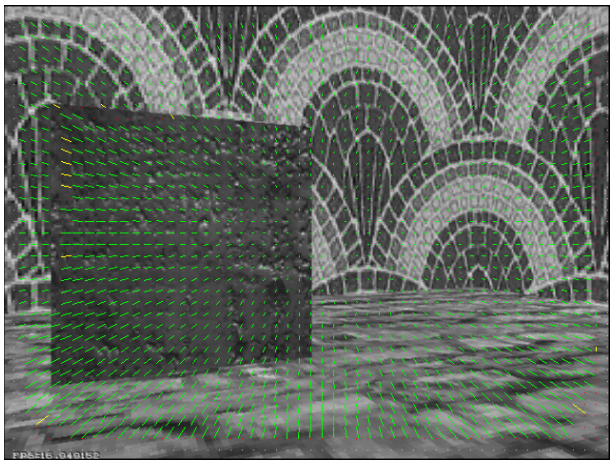

Fig. 8. The motion vectors have different direction and magnitude depending on the distance of the point to the camera (image source – MIT).

E. Imaging System

Our imaging system consists of two physical detectors (cameras) pointing in the same direction. One camera is a slow shutter-speed camera that is used to evaluate the motion blur to estimate the current motion parameters in the image. The other camera is a high-resolution system with a short shutter time to reduce the motion blur as much as possible.

Our current implementation for the hybrid imaging system consists essentially of two separate cameras pointing in the same direction (Fig. 9). It is important to share the view, because the motion estimation from motion blur estimates only the motion of the image projections. The motion prediction directly in the image space makes the system less sensitive to errors in the estimation of the 3D positions of the tracked points. In case of different view directions of both cameras, the system would need to rely on an exact 3D reconstruction of the tracked points.

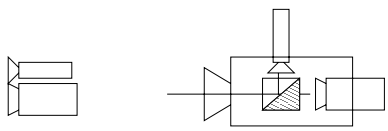

Fig. 9. Possible implementations of the imaging system: a) two separate cameras with different physical properties mounted next to each other; b) beam-splitter allowing exact co-registration of the views.

IV. Preliminary Results

We tested the system on indoor images taken with a moving camera. A Firewire camera Sony DFW-V500 in combination with our image processing library XVision2 [12] allowed to control the shutter time of the camera. We put a developed film as a daylight filter in front of the lens to prevent the camera from over-saturating during the relatively long exposure time. The system is running under LinuxOS on a Pentium 4 with 3GHz CPU frequency. The two-dimensional FFT is performed on 64x64 image patches that are used to estimate the motion. This limits the possible motion in the image to approx. 40 pixels.

The FFT based approach (Section III-B) was used to guess the motion of the camera $v = (v_u, v_\nu)^T$ during the shutter time t_s. We used this speed v to predict the image position of the landmark relative to the previous position between consecutive acquisitions in the time interval t_c.

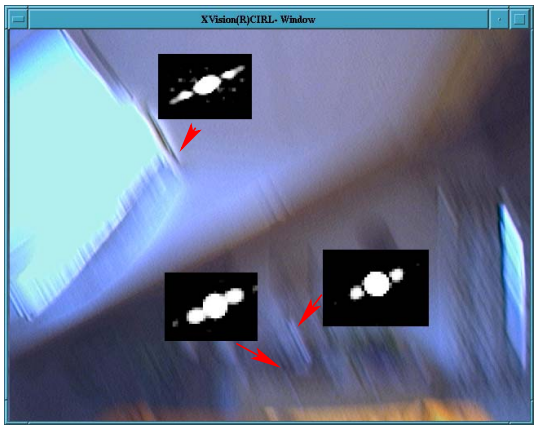

Fig. 10. The velocity and direction of motion can be estimated from the distance between the zero transitions of the sinc functions depicted for a few features in the image (6).

The validation of the Eq. (1) can be found in Fig. 11. Features in different distances result in different sinc functions.

Fig. 11. The velocity estimate for two features in strongly varying distance from the image can be read from the distance between the zero transitions of the sinc function.

The direct approach estimates directly the motion parameters without any domain conversions, but it is not as generic as the FFT version, because it requires single point features with high contrast boundaries in the images. It is a very interesting extension reducing the computational needs significantly and it is of great interest as an inertial system under special condition, like navigation in the dark, based on positions of the light sources (car navigation). The missing texture information in the dark does not allow to distinguish between the different light spots. This extension helps to maintain the correct correspondences.

Without the presented extension, the system cannot track the feature points because of the too large time t_c anymore. It had to switch to an identification task, where entire image was scanned for possible correspondences. The displacements in the image were larger than the allowed displacement in the multi-resolution tracking approach. The motion estimation from the motion blur allows to switch back to tracking approaches, because the prediction comes close to the true position of the tracked feature.

V. Conclusions and Future Work

We presented a system that can replace an expensive 6DoF inertial unit with a single video camera. This camera is operated in a slow shutter speed mode to achieve a significant motion blur in the images. This motion blur is used directly to estimate the current motion parameters during the exposure time and to predict correctly the motion between the acquisition points where images are taken with a slower high-resolution camera. This estimation technique allows a better prediction of the position of tracked features than systems deriving the state estimate from past measurements using Kalman Filtering techniques. Sudden changes in the direction of motion are captured directly in the blur of the images and can be processed correctly in the new prediction. Systems using just past measurements for prediction have a clear disadvantage compared to the proposed system.

The motion blur is merely used to estimate the future image position of the tracked points in the next frame. A 3D trajectory is calculated in a separate post-processing step. It allows an estimation of the 3D trajectory giving a better prediction of out-of-plane motion towards or away from the camera. This type of motion results in a non-linear motion of the projected image point over time, which is not correctly modeled with constant velocity in the image plane between the acquisition points.

Acknowledgments

The work presented in this paper was sponsored by the Helmholtz Gesellschaft in the context of the Virtual Institute for Sensor Data Fusion and Telepresence. This is a cooperation between the Technische Universität München and the German Aerospace Agency (DLR) in Oberpfaffenhofen. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the HGF or the institutions mentioned above.

References

[1] J.K. Aggarwal and N. Nandhakumar . On the computation of motion from sequences of images-A review. *Proceedings of the IEEE*, (76):8:917–935, 1988.

[2] Adnan Ansar, and Kostas Daniilidis. Linear Pose Estimation from Points or Lines. In IEEE Pattern Analysis and Machine Intelligence, pages 578–589, May 2003.

[3] T. J. Broida, S. Chandrashekhar, and R. Chellappa. Recursive 3D motion estimation from a monocular image sequence. In IEEE Trans. Aero. Elect. Syst., vol. AES-26, no. 4. pp. 639-656, July 1990.

[4] J.L. Barron, D.J. Fleet, S.S. Beauchemin, and T.A. Burkitt. Performance of optical flow techniques. *CVPR*, pages 236–242, 92.

[5] Berthold K. Horn and Brian Schunck. Determining Optical Flow. Technical report, Massachusetts Institute of Technology, 1980. Technical report.

[6] M.J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *ICCV93*, pages 231–236, 1993.

[7] Darius Burschka and Gregory D. Hager. V-GPS – Image-Based Control for 3D Guidance Systems. In *Proc. of IROS*, pages 1789–1795, October 2003.

[8] Darius Burschka and Gregory D. Hager. V-GPS(SLAM): – Vision-Based Inertial System for Mobile Robots. In *Proc. of ICRA*, pages 409–415, April 2004.

[9] O. Faugeras, *Three-Dimensional Computer Vision*, The MIT Press, 1993.

[10] G. Hager and D. Burschka. Laser-based position tracking and map generation. In *Proceedings of RA*, pages 149–155, August 2000.

[11] G. Hager, C-P. Lu, and E. Mjolsness. Object pose from video images. *PAMI*, 22(6):610–622, 2000.

[12] G. Hager and K. Toyama. The XVision System: A General-Purpose Substrate for Portable Real-Time Vision Applications. *Computer Vision and Image Understanding*, 69(1):23–37, 1995.

[13] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[14] S. Se, D. Lowe, and J. Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *IEEE Conf. on Robotics and Automation*, pages 2051–2058, 2001.

[15] R. Sim and G. Dudek. Mobile robot localization from learned landmarks,. In *IROS*, 1998.

[16] Saul Simhon and Gregory Dudek. Selecting targets for local reference frames. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2840–2845, 1998.

[17] Y. Takeuchi, P. Gros, M. Hebert, and K. Ikeuchi. Visual learning for landmark recognition. In *Proc. Image Understanding Workshop*, pages 1467–1473, 1997.

[18] Sebastian Thrun. Finding landmarks for mobile robot navigation. In *IEEE Conf. on Robotics and Automation*, pages 958–963, 1998.

[19] C. Tomasi and J. Shi. Good features to track. In *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, pages 593–600, 1994.

[20] E. Yeh and D. Kriegman. Toward selecting and recognizing natural landmarks. In *IEEE Int. Workshop on Intelligent Robots and Systems*, pages 47–53, 1995.