

# REAL-TIME VISUAL ODOMETRY FOR GROUND MOVING ROBOTS USING GPUS

Michael Schweitzer, Alois Unterholzner and Hans-Joachim Wuensche

*Institute for Autonomous Systems Technology, Universität der Bundeswehr München, Germany*

Keywords: Robot Vision, GPGPU, Structure from motion.

Abstract: This paper introduces a novel visual odometry framework for ground moving robots. Recent work showed that assuming non-holonomic motion can simplify the ego motion estimation task to one yaw and one scale parameter. Furthermore, a very efficient way of computing image frame to frame correspondences for those robots was presented by skipping rotational invariance and optimizing keypoint extraction and matching for massive parallelism on a GPU. Here, we combine both contributions to a closed framework. Long term correspondences are preserved, classified and stabilized by motion prediction, building up and keeping a trusted map of depth-registered keypoints. We also allow other ground moving objects. From this map, the ego motion is inferred, extended by constrained rotational perturbations in pitch and roll. A persistent focus is on keeping algorithms suitable for parallelization and thus achieving up to one hundred frames per second. Experiments are carried out to compare against ground-truth given by DGPS and IMU data.

## 1 INTRODUCTION

Reconstructing the motion from the 2D image stream of a moving camera is an important task in computer vision. It is commonly known as *visual odometry* and often concerned with real-time robot vision. The idea is to compute image correspondences of consecutive image frames connected to static points in 3D space and infer the camera motion between both frames. Early work needed at least eight points to solve the equations given by epipolar constraints (Longuet-Higgins, 1981; Hartley, 1994). (Nister et al., 2004) proposed a more efficient way by using only five points. Recent work by (Scaramuzza et al., 2009b; Scaramuzza et al., 2009a) reduced this to an one point method by constraining the camera motion. Once the motion is known, the depth (*structure*) of the correspondences can be determined, commonly known as *Structure from Motion (SfM)*. By keeping this structure and using longterm correspondences, (Civera et al., 2009) introduced a novel SfM-method within the extended kalman filter framework (*EKF*). The state vector of the proposed camera-centered EKF contains the location in the world frame plus a map of 3D-registered image features in the camera frame. Measuring around 20 features leads to a state vector size around 300. Real-time capa-

bility at 30 frames per second comes from a 1-point RANSAC algorithm to reject outliers updating the filter. Opposite to (Scaramuzza et al., 2009b) additional information to achieve 1-point relations comes from the probability distribution provided by the EKF instead of constraining the robot's motion. With a larger amount of tracked features (around one hundred), the frame rate drops to 1Hz and goes beyond real time. However, this contribution shows good accuracy in its results, similar to non-linear optimization techniques like bundle adjustment (Triggs et al., 2000).

Using EKF frameworks it turns out that the *inverse depth parameterization*, briefly *IDP*, presented in (Civera et al., 2008) is a good choice. Initializing unregistered 3D points with an IDP and estimating the inverse depth instead of euclidean coordinates reduces linearization errors within the EKF framework. Thereby the EKF converges faster and is more stable.

In this paper, we deal with ground moving robots described in (Scaramuzza et al., 2009b).

The robot is assumed to move in an ideal ground plane with a locally circular motion. Only two parameters are needed for describing this motion model: the yaw angle  $\Psi$  of the circle segment and the scale  $\rho$  which denotes the chord. (Scaramuzza et al., 2009b) shows that  $\Psi$  is computable from only one correspondence in a monocular stream. Outliers are removed

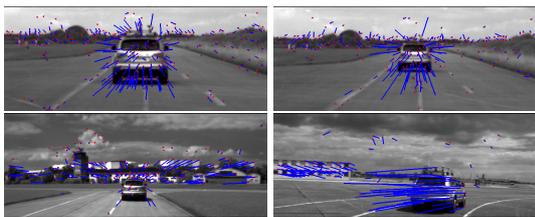


Figure 1: Three classes of correspondences w.r.t. the PIDM. Points with no blue line belong to *static on ground or infinite*, points with long blue lines belong to *static above ground or dynamic*.

with a 1-point RANSAC (Fischler and Bolles, 1981). The scale  $\rho$  in (Scaramuzza et al., 2009b) is given by a velocity sensor. Then, in (Scaramuzza et al., 2009a), a method for scale estimation is proposed. But this is only possible, when a circular motion is detected ( $\Psi > 0$ ) and leads to a sparse distribution of working points. Here, we describe a method for estimating  $\Psi$  and  $\rho$  in every phase of the robot’s motion. By explicitly solving optical flow equations for image keypoints for a locally planar, circular motion and extracting the main effecting terms, we derive a measurement model for image correspondences. We call this measurement model *linearized planar measurement model*, *LPMM*. The structure of the scene is approximated by an IDP model. Here, we assume a *planar inverse depth model*, *PIDM*.

Combining LPMM and PIDM, we can derive 1-point relations for image correspondences. By massive parallel computing on a GPU, image correspondences are classified w.r.t. the PIDM into three classes: *static on ground or infinite*, *static above ground* and *dynamic*. This vectorized classification can be done independently to other keypoints exploiting the 1-point relations for thousands of correspondences. Deviations from the depth model can be recognized and corrected by motion stereo of the integrated motion signal. Therefore, individual image keypoints 3D-locations are *not* estimated within an EKF. Here we see the main difference to the approach of (Civera et al., 2009). Our EKF layout is much leaner because it only contains the motion parameters and their temporal derivatives. Only selected and categorized correspondences update the EKF which therefore converges faster. See Figure 1 to get an idea of this classification task.

Using a *generalized disparity equation* splits the motion into a rotational (*infinite homography*  $\mathbf{H}_\infty$ ) and a transitional component (Hartley and Zisserman, 2006). A similar approach is proposed in (Tardif et al., 2008) but using omnidirectional cameras. Our paper only covers perspective monocular or stereo cameras heading into the direction of the ego motion. Besides giving accurate results, (Tardif et al., 2008)

also gives a good overview of recent visual odometry, visual SLAM and SfM techniques. It turns out that pitch ( $\Theta$ ) and roll ( $\Phi$ ) perturbations heavily effects the quality of the  $(\rho, \Psi)$  estimation. In the same way we extract  $(\rho, \Psi)$ , we derive equations for  $(\Theta, \Phi)$  to reduce this effect.

For correspondence calculation we use a recently proposed efficient and parallelized method for extracting and matching keypoints (Schweitzer and Wuensche, 2009). We enhance this approach to long term correspondences. This allows to correspond up to 2000 keypoints in a stereo stream ( $2 \times 752 \times 480$ ) for both monocular streams plus stereo matching in about *5ms* on a NVidia Tesla GPU. As a result, three closed lists of long term correspondences reside in GPU memory which are then used for the vectorized PIDM-classification mentioned above.

This paper is organized as follows. In section two we describe the GPU computation of long term correspondences and list generation. Section three introduces the PIDM/LPMM and derives the 1-point relations for the motion parameters from them. The use for classification and EKF estimation is explained. and results are evaluated against IMU and DPGS ground truth in section four. We conclude with section five.

## 2 GPU IMAGE CORRESPONDENCES

(Schweitzer and Wuensche, 2009) proposed a novel method for an efficient extraction and matching of corner-based keypoints. It is based on a dense computation of three normalized haar wavelet responses  $(I_x, I_y, I_{xy})_t$  per pixel at scale  $t$ , the so-called *SidCell-image* (*Scale Invariant Descriptive Cell*), shown in Figure 2. Haar wavelets can be computed very efficiently by using integral images (Viola and Jones, 2001) and are also employed by SURF correspondences (Bay et al., 2006). From the SidCell-image, keypoints and descriptors are derived. Keypoints are

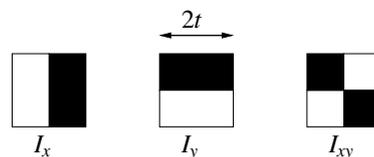


Figure 2: Sidcell Components.

extracted by a non-maximum suppression on the absolute  $|I_{xy}|$ -component of the SidCell-image. The sensitivity of the keypoint extraction is adjusted by a noise threshold of  $|I_{xy}|$  between  $[0, 1]$ . (Schweitzer and

Wuensche, 2009) define two strength tests to remove weak keypoints. For correspondence calculation of consecutive frames two step are carried out. First, a prematching w.r.t. the sign-signature of the keypoints (8 classes, different colors in Figure 3) is done. Second, only if the first step matches, the descriptors are compared. This avoids *heavy-weight* descriptor distance calculations where it is not promising and increases matching efficiency a lot. The descriptors themselves are built from distributed and weighted SidCells around the keypoint, while the complexity of the distribution is user-defined (see Figure 3).

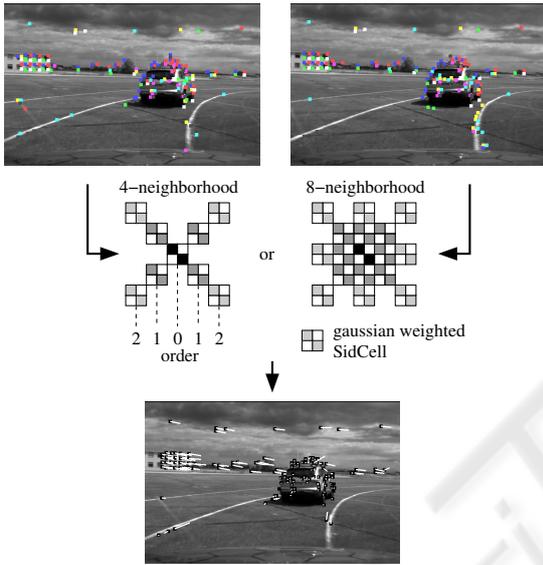


Figure 3: SidCell Matching with  $|I_{xy}| > 0.2$ .

## 2.1 List Processing on GPUs

Collecting keypoints into a closed list is mostly done by an intermediate CPU step. This requires additional data transfer between GPU and CPU and reduces the speed gain. Using modern atomic GPU-functions like `atomicInc()` or `atomicAdd()` those disturbing CPU-steps are avoidable. List data structure operations like `merge`, `split` or `concatenate` all need an `addElem` operator. A parallelized version of `addElem` needs a synchronization. Figure 4 shows how `addElemSync` is implemented. The vectorized versions of `merge`, `split`, `concatenate` are essential for this work. Assuming that the keypoints can be processed independently of each other, full parallelism is given. Atomic GPU functions are used in the same way for building histograms needed later in this paper.

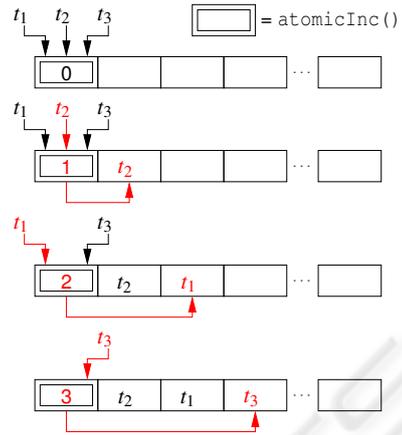


Figure 4: Implementation of `addElemSync` on a GPU. The first element of the list is its size. In this example, three threads want to add a new element to an empty list simultaneously. Each thread gets its number of the list cell where it can put its data by applying an `atomicInc()` on the size element.

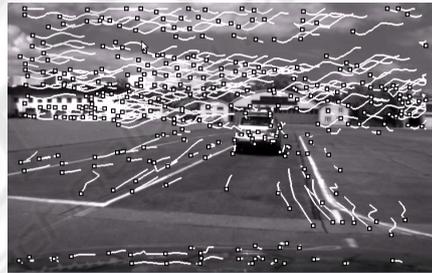


Figure 5: SidCell long term correspondences.

## 2.2 Long Term Correspondences

Long term correspondence (*lt-correspondences*) are held in a list. To build this list, initially all frame to frame correspondences (*f2f-correspondences*) are added. Subsequent cycles search this list whether two *f2f-correspondences* can be concatenated. If a *lt-correspondence* does not have a successor, it is removed from the list. *F2f-correspondences* which do not have a predecessor are added as new element. Figure 5 shows the result. Selected *f2f-correspondences* are the measurements for the EKF of the motion parameters. *Lt-correspondences* support the selection and adjust deviations to the assumed inverse depth model. This is explained in detail in the next section.

### 3 OUR APPROACH

#### 3.1 Planar Inverse Depth Model

In case of a known rigid motion  $[\mathbf{R}|\mathbf{t}]$  between two cameras  $c_0 \rightarrow c_1$  the image point  $m_1$  in  $c_1$  can be computed from  $m_0$  in  $c_0$ :

$$\tilde{\mathbf{m}}_1 = \mathbf{A} \mathbf{R} \mathbf{A}^{-1} \cdot \tilde{\mathbf{m}}_0 + X_0^{-1} \mathbf{A} \mathbf{t} \quad (1)$$

$\mathbf{A}$  is the camera calibration matrix and assumed to be the same for both cameras. It is obvious that this equation also needs the inverse depth  $X_0^{-1}$  of the homogeneous image point  $\tilde{\mathbf{m}}_0$ . Additionally it is not linear in case of  $X_0^{-1} \neq 0$  or  $\mathbf{t} \neq 0$ . We discuss this equation later in this paper, for now it is important how to get  $X_0^{-1}$ . The approach here is to imply a structural model from which the inverse depth can be computed for every pixel. In case of ground moving robots we imply a ground plane model. Then, the idea is to separate correspondences to those which fit to this model and those which do not fit. For those which fit, the inverse depth is known and the motion can be observed. For those which don't fit, the inverse depth can be computed from motion. The importance of the PIDM fit is high at the beginning but should decrease over time, when more and more new depth-registered keypoints are available in a trusted map. When the system fails, a re-initialization by the PIDM can be done. To express a 3D point  $M_0 = (X_0, Y_0, 0)^\top \in E(Z_c, \Theta_c)$  in camera coordinates (see Figure 6) the homogeneous transformation  $\mathbf{R}_Y(\Theta_c) \cdot \mathbf{T}(0, 0, Z_c)$  is applied.

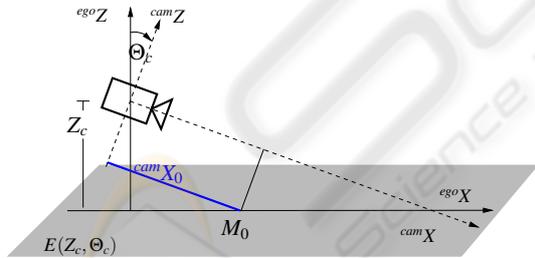


Figure 6: Depth  $X_0$  of a 3D Point  $M_0$  which lies on the plane  $E(Z_c, \Theta_c)$  given in camera coordinates.

The corresponding image point  $m_0$  is then given by

$$\begin{aligned} \tilde{\mathbf{m}}_0 &= \mathbf{R}_Y(\Theta_c) \mathbf{T}(0, 0, Z_c) \mathbf{P}_c(k) \cdot \tilde{\mathbf{M}}_0 \\ &= \begin{bmatrix} kY_0 \\ k(X_0\Theta_c - Z_c) \\ Z_c\Theta_c + X_0 \end{bmatrix} = \begin{bmatrix} \tilde{y}_0 \\ \tilde{z}_0 \\ \tilde{w}_0 \end{bmatrix} \end{aligned}$$

As one can see, every image line  $z = \tilde{\lambda} \cdot \tilde{z}$  corresponds to a constant depth, with  $\tilde{\lambda} = 1/\tilde{w}$ . Now we can derive the PIDM:

$$\begin{aligned} X^{-1}(z) &= \frac{1}{kZ_c}(z_\infty - z) \\ \text{PIDM}(z) &= \begin{cases} X^{-1}(z) & \text{if } X^{-1} > 0 \\ 0 & \text{else} \end{cases} \quad (2) \end{aligned}$$

where  $z_\infty = k\Theta_c$  is the image line at infinity (horizon). Note that the PIDM is a linear mapping without singularity at infinite points.

#### 3.2 Non-holonomic Robot Motion

Recent work (Scaramuzza et al., 2009b) introduced a simple circular motion model with two parameters  $(\rho, \Psi)$  for wheeled ground moving robots (see Figure 7). The transitional component  $\mathbf{t}$  of the rigid camera

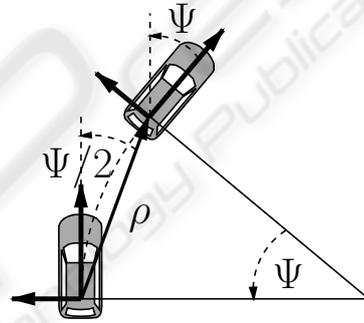


Figure 7: Local circular motion model by (Scaramuzza et al., 2009b).

motion  $[\mathbf{R}|\mathbf{t}]$  is then given by:

$$\mathbf{t} = \rho \cdot \begin{bmatrix} \cos(\Psi/2) \\ \sin(\Psi/2) \\ 0 \end{bmatrix} \quad (5)$$

In addition to this model, our model is augmented by pitch ( $\Theta$ ) and roll ( $\Phi$ ). The motion  $(\Theta, \Phi)$  is unconstrained and modeled by assuming constant velocities.

#### 3.3 Linearized Planar Measurement Model LPMM

Using the disparity equation (1) and the our motion model (5) it is possible to derive the displaced homogeneous image point  $\tilde{\mathbf{m}}_1$  as given in (3). The measurement is the displacement  $\partial m_1$  of the correspondence of the image point  $m_0$ . Therefore, the partial derivatives  $\partial m_1 / \partial(\Psi, \Theta, \Phi, \rho)$  in euclidean coordinates must be determined. Using (3) and the transformation from homogeneous coordinates into euclidean  $m_1 = \tilde{\lambda} \cdot (\tilde{y}_1, \tilde{z}_1)^\top$  the derivatives  $\partial m_1 / \partial(\dots)$  yield to:

$$\begin{aligned}
 \tilde{\mathbf{m}}_1 &= \mathbf{A}\mathbf{R}_\Phi\mathbf{R}_\Theta\mathbf{R}_\Psi\mathbf{A}^{-1} \cdot \tilde{\mathbf{m}}_0 + X_0^{-1}\mathbf{A}\rho \begin{bmatrix} \cos(\Psi/2) \\ \sin(\Psi/2) \\ 0 \end{bmatrix} \\
 &= \begin{bmatrix} k(c_1c_2c_3 - c_3s_1) + y_0(c_1c_3 + s_1s_2s_3) + z_0c_2s_3 + \frac{1}{2}X_0^{-1}k\Psi\rho \\ k(s_1s_3 + c_1s_2c_3) - y_0(c_1s_3 - s_1s_2c_3) + z_0c_2c_3 \\ \frac{1}{k}y_0s_1c_2 - \frac{1}{k}z_0s_2 + c_1c_2 + X_0^{-1}\rho \end{bmatrix} = \begin{bmatrix} \tilde{y}_1 \\ \tilde{z}_1 \\ 1/\tilde{\lambda} \end{bmatrix} \\
 &\text{with } s_1 = \sin\Psi, c_1 = \cos\Psi, s_2 = \sin\Theta, c_2 = \cos\Theta, s_3 = \sin\Phi, c_3 = \cos\Phi
 \end{aligned} \tag{3}$$

$$\begin{aligned}
 \frac{\partial m_1}{\partial(\Psi, \Theta, \Phi)} &= \tilde{\lambda} \begin{bmatrix} \tilde{y}_1 \\ \tilde{z}_1 \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial}{\partial\Psi} & \frac{\partial}{\partial\Theta} & \frac{\partial}{\partial\Phi} \end{bmatrix} \\
 &\approx \lambda \begin{bmatrix} (\frac{1}{2}X_0^{-1}\rho - 1)k - \Psi y_0 & \Phi k & \Theta k - \Phi y_0 + z_0 \\ \Phi k + \Theta y_0 & k + \Psi y_0 - \Theta z_0 & \Psi k - y_0 - \Phi z_0 \end{bmatrix} + \frac{\partial\tilde{\lambda}}{\partial(\Psi, \Theta, \Phi)} \begin{bmatrix} \tilde{y}_1 \\ \tilde{z}_1 \end{bmatrix} \\
 \frac{\partial m_1}{\partial\rho} &= \tilde{\lambda} \frac{\partial}{\partial\rho} \begin{bmatrix} \tilde{y}_1 \\ \tilde{z}_1 \end{bmatrix} + \frac{\partial\tilde{\lambda}}{\partial\rho} \begin{bmatrix} \tilde{y}_1 \\ \tilde{z}_1 \end{bmatrix} \\
 &\approx \lambda \begin{bmatrix} \frac{1}{2}X_0^{-1}\Psi k \\ 0 \end{bmatrix} - X_0^{-1} \begin{bmatrix} (1 - \Psi + \frac{1}{2}X_0^{-1}\rho)k + y_0 + \Theta z_0 \\ \Theta k - \Phi y_0 + z_0 \end{bmatrix} \\
 \text{with } \tilde{\lambda} \approx \lambda &= \frac{k}{(1 + X_0^{-1}\rho)k + \Psi y_0 - \Theta z_0}
 \end{aligned} \tag{4}$$

$$\frac{\partial m_1}{\partial(\dots)} = \tilde{\lambda} \frac{\partial(\tilde{y}_1, \tilde{z}_1)^\top}{\partial(\dots)} + (\tilde{y}_1, \tilde{z}_1)^\top \frac{\partial\tilde{\lambda}}{\partial(\dots)} \tag{6}$$

This term gets very complex. We assume that the image displacement are comparatively small which allows for a small angle approximation:  $c_i$  gets to 1 and  $s_i$  gets to  $\{\Psi|\Theta|\Phi\}$ . Also we assume that products of two or more angles get to zero. Note that this approximation is only valid *after* the derivatives are calculated.

It turns out that the terms for  $\partial\tilde{\lambda}/\partial(\Psi, \Theta, \Phi)$  can also be approximated to zero in case of  $X_0^{-1} \rightarrow 0$ . Furthermore, the approximation  $\partial\tilde{\lambda}/\partial\rho \approx -X_0^{-1}$  is valid. The result is given by the set of equations in (4). This set is what we call the *linearized planar motion model*, briefly *LPMM*.

### 3.4 Discussing the LPMM

Getting granular on the LPMM one can draw valuable consequences. One important term is the product of the motion scale  $\rho$  and the inverse depth  $X_0^{-1}$  of a keypoint. If the robot is not moving and therefore  $\rho = 0$ ,  $X_0^{-1}$  also vanishes from the equation. No structure is observable in this case which is consistent with the 3D reconstruction theory. The product also vanishes for far away (or infinite) 3D points. From those measurements no information about  $\rho$  is observable. But it's worth to find those far away points because

the rotational component (or the infinite homography  $\mathbf{H}_\infty$ ) of the camera motion can be measured almost directly. A further consequence can be seen, if every angle-product is set to zero. This rough estimate still contains valuable information because the dominating terms are clearly visible. Another important consequence is, that the individual parameters of the LPMM become independent of each other. Furthermore,  $\lambda$  gets to  $(1 + X_0^{-1}\rho)^{-1}$  and for far away points it gets to 1. One of the most important properties of the LPMM is the fact that the  $z$ -component of  $\partial m_1/\partial\rho$  is *completely independent* of the yaw parameter  $\Psi$ . Exploiting those consequences allows a vectorized classification of image correspondences we discuss later.

## 4 EGO MOTION ESTIMATION

In order to estimate the robots motion we want to extract four signals, one transitional scale ( $\rho$ ) and three rotational angles ( $\Psi, \Theta, \Phi$ ). Each image frame coming with the measured f2f-correspondences updates an EKF designed to output those signals by using the LPMM for jacobian linearization. As still discussed, measurements have to be selected w.r.t. their capability to observe a certain signal. Mainly this depends on their depth configuration and leads to an initialization problem. For our robot we imply the PIDM as a starting point. Note that one can apply any (inverse)

depth model, e.g. a corridor model for indoor robots or depth from stereo. Needless to say, the assumed inverse depth model has to be observable to a certain degree, otherwise it would fail. The aim is to adapt to the real scene by depth registering variations to the model and soften the model assumption.

#### 4.1 Scale Estimation

Using the PIDM (2) within the  $\Psi$ -independent  $z$ -component of LPMM scale (4) leads to the following 1-point equation:

$$\partial\rho = \frac{kZ_c}{(z_0 + \Theta k - \Phi y_0)(z_0 - z_\infty)} \partial z \quad (7)$$

Once again,  $z_0$  is the  $z$ -coordinate of the extracted image keypoint  $m_0$  and  $\partial z$  is the  $z$ -displacement measurement of the correspondence to the previous frame. (7) gets singular for  $z_0 \rightarrow z_\infty$  and on the other side well conditioned for 3D points in the near field ( $X_0^{-1} \gg 0$ ). Figure (8) shows the keypoint correspondence classification and scale estimation without using any additional signal than the image stream. Figure (9) shows the EKF scale output while Figure (10) uses an INS pitch signal for correction. Setting the roll compo-

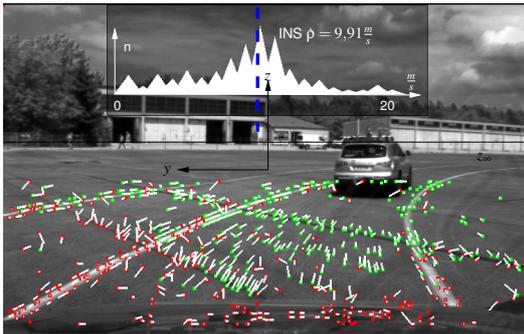


Figure 8: Classification of image correspondences for scale estimation. First, a histogram is built by a vectorized 1-point scale computation for every correspondence independently on the GPU. By histogram voting, the most likely scale value is chosen. The dashed blue line indicates the ground truth scale value measured by the INS. Green correspondences match to the histogram vote, red ones don't. Histogram vote and EKF prediction are combined for a stable inlier classification. See also the image coordinate frame and the line at infinity (black).

nent in equation (7) to zero,  $\partial\rho$  is completely independent from the  $y$ -coordinate in the image. One can say this denotes a  $\frac{1}{2}$ -point relation but in fact this is a PIDM special case.

#### 4.2 Rotation Estimation

At a first look, the rotation estimation easily can be done by taking f2f-correspondences of far away key-

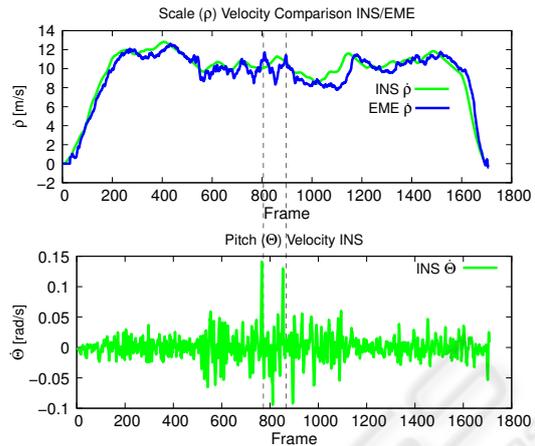


Figure 9: Ego motion estimation (*EME*, blue) scale signal compared to the ground truth INS signal (green). *EME* extracts a near similar signal with two major perturbations. Between frame 700-1000, a strong pitch perturbation occurs, see the green INS pitch signal below. Between frame 1000-1200 the scene is quite keypoint-less. The *constant acceleration* system model of the EKF stabilizes the signal for a certain time.

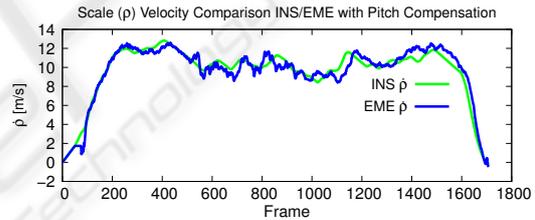


Figure 10: Scale signal with pitch compensation by ground truth INS data. The peaks between frame 700-1000 are flattened.

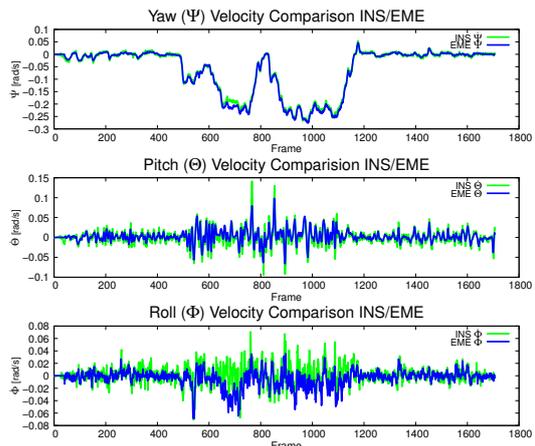


Figure 11: Rotation estimation (*EME*, blue) by far away keypoints compared to the ground truth INS signal (green). The selection of the keypoints is done w.r.t. the PIDM.

points. The structure term  $\rho X_0^{-1}$  of the optical flow measurement  $\partial m$  gets to zero for those points and

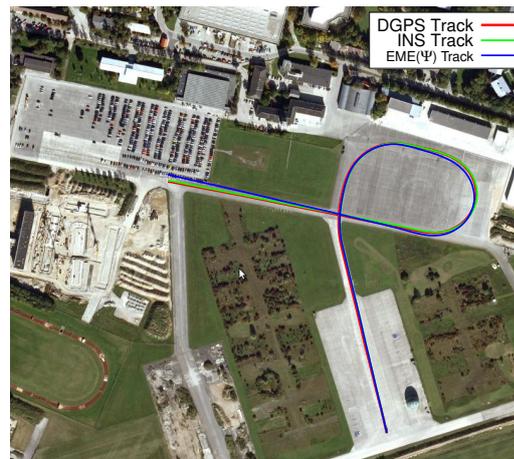
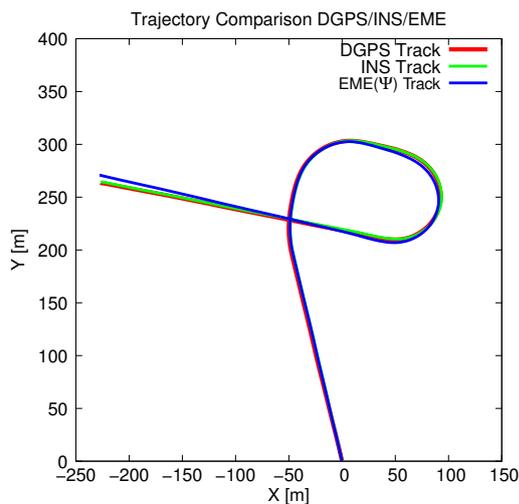


Figure 13: Ego motion estimation (EME, blue) track by using the estimated yaw signal and the locally circular motion model. The scale signal is given by INS ground truth. Red shows the differential GPS track, green shows the motion model using ground truth INS data. The right image shows the tracks referenced in a satellite image by GoogleEarth.

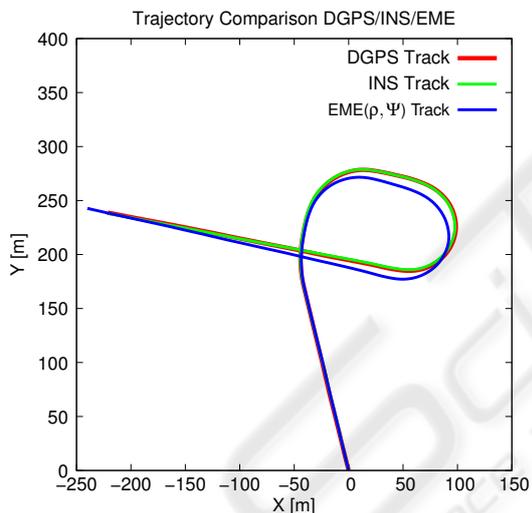


Figure 12: Ego motion estimation (EME, blue) track by using the pure monocular vision scale and yaw signal of Figure 10 and Figure 11.

therefore the LPMM (4) simplifies considerably. Figure 11 shows the EKF rotational signals for yaw  $\Psi$  and pitch  $\Theta$  using the same method as for scale estimation with the difference of taking far away key-points instead of near ones. Figures 12 and 13 show the driven tracks by integrating the estimated motion signals.

## 5 CONCLUSIONS

In this paper we introduced a visual odometry method using the EKF framework and GPU computed im-

age correspondences. We described the measurement model LPMM of the EKF in detail and derived 1-point relations for a vectorized inliers selection on the GPU. We achieve a overall computation time of less than 10ms: 4ms for GPU correspondence calculation and selection and 5ms for the EKF update on a 2GHz single core CPU. Future work will extend this approach to a full structure from motion application.

## ACKNOWLEDGEMENTS

The authors gratefully acknowledge support of this work by the Deutsche Forschungsgemeinschaft (German Research Foundation) within the Transregional Collaborative Research Center 28 *Cognitive Automobiles*.

## REFERENCES

- Bay, H., Tuytelaars, T., and Gool, L. V. (2006). SURF: Speeded up Robust Features. In *Proc. European Conf. on Computer Vision*.
- Civera, J., Davison, A., and Montiel, J. (2008). Inverse Depth Parametrization for Monocular SLAM. *IEEE Transactions on Robotics*, 24(5):932–945.
- Civera, J., Grasa, O. G., Davison, A. J., and Montiel, J. M. M. (2009). 1-Point RANSAC for EKF-Based Structure from Motion. In *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*.
- Fischler, M. A. and Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Comm. of the ACM*, 24:381–395.

- Hartley, R. (1994). Euclidean reconstruction from uncalibrated views. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 908–912.
- Hartley, R. and Zisserman, A. (2006). *Multiple View Geometry*. Cambridge University Press.
- Longuet-Higgins, H. C. (1981). A Computer Algorithm for Reconstructing a Scene From Two Projections. *Nature*, 293(1):133–135.
- Nister, D., Naroditsky, O., and Bergen, J. (2004). Visual Odometry. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages I–652–I–659 Vol.1.
- Scaramuzza, D., Fraundorfer, F., Pollefeys, M., and Siegwart, R. (2009a). Absolute Scale in Structure from Motion from a Single Vehicle Mounted Camera by Exploiting Nonholonomic Constraints. In *Proc. IEEE Int'l Conf. on Computer Vision*.
- Scaramuzza, D., Fraundorfer, F., and Siegwart, R. (2009b). Real-Time Monocular Visual Odometry for On-Road Vehicles with 1-Point RANSAC. In *Proc. IEEE Int'l Conf. on Robotics and Automation*.
- Schweitzer, M. and Wuensche, H.-J. (2009). Efficient Key-point Matching for Robot Vision using GPUs. In *Proc. 12th IEEE Int'l Conf. on Computer Vision, 5th IEEE Workshop on Embedded Computer Vision*.
- Tardif, J., Pavlidis, Y., and Daniilidis, K. (2008). Monocular Visual Odometry in Urban Environments Using an Omnidirectional Camera. In *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, pages 2531–2538.
- Triggs, B., McLauchlan, P., Hartley, R., and Fitzgibbon, A. (2000). Bundle Adjustment – A Modern Synthesis. In *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag.
- Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*.

