Comparison of Point and Line Features and Their Combination for Rigid Body Motion Estimation

Florian Pilz[1], Nicolas Pugeault[2,3], and Norbert Krüger[3]

[1] Department of Medialogy and Engineering Science
Aalborg University Copenhagen, Denmark
fpi@imi.aau.dk

[2] School of Informatics
University of Edinburgh, United Kingdom
nicolas.pugeault@edinburgh.ac.uk

[3] The Maersk Mc-Kinney Moller Institute
University of Southern Denmark, Denmark
norbert@mmmi.sdu.dk

Abstract. This paper discusses the usage of different image features and their combination in the context of estimating the motion of rigid bodies (RBM estimation). From stereo image sequences, we extract line features at local edges (coded in so called multi-modal primitives) as well as point features (by means of SIFT descriptors). All features are then matched across stereo and time, and we use these correspondences to estimate the RBM by solving the 3D-2D pose estimation problem. We test different feature sets on various stereo image sequences, recorded in realistic outdoor and indoor scenes. We evaluate and compare the results using line and point features as 3D-2D constraints and we discuss the qualitative advantages and disadvantages of both feature types for RBM estimation. We also demonstrate an improvement in robustness through the combination of these features on large data sets in the driver assistance and robotics domain. In particular, we report total failures of motion estimation based on only one type of feature on relevant data sets.

1 Introduction

The knowledge about the egomotion of the camera or the motion of objects in a scene is crucial for many applications such as driver assistance systems, object recognition, collision avoidance and motion capture in animation. In case we deal with a rigid object, such a motion can be understood as a 'Rigid Body Motion' (RBM), which is defined as a continuous motion preserving the distance between any two points of the object. The mathematical structure of this motion is well known and has been studied for over 100 years (see e.g., [1]).

The estimation of such a motion from images faces two sub-problems. First, based on a certain mathematical formulation of the RBM, constraint equations need to be defined using correspondences between image features. This problem can become complex, in particular when correspondences of different features (e.g., line and point correspondences) become mixed. The state of the art in this field is described in Section 1.1. Second, correspondences between image features need to be computed from image sequences. Here, different kinds of local image structures are observable that lead to constraints of different structure and strength (e.g., point and line correspondences). Point correspondences are more valuable since they lead to stronger constraint equations. However, in general there are much fewer point correspondences compared to line correspondences available in natural scenes due to the dominance of edges (see, e.g., [2]). Work addressing the correspondence problem for different feature types is discussed in Section 1.2.

The intention of this paper is to analyze the consequences of using two different kinds of correspondences (i.e., point and line correspondences) as well as their combination for motion estimation on a large set of indoor and outdoor sequences, with large variety across the sequences as well as within the sequences. We evaluate and compare the results of line features (primitives) and point features (SIFT) as constraints for RBM estimation. We discuss the qualitative advantages and disadvantages of both feature types for RBM estimation and also demonstrate an improvement in robustness through the combination of these features on large data sets in the driver assistance and robotics domain. In particular, we report total failures of motion estimation based on only one type of feature on relevant data sets stressing the importance of their combination for robust motion estimation. Hence, our main contribution is giving empirical evidence to argue that for stable motion estimation both kinds of features need to be used. As a consequence, we state that (1) a rich feature representation needs to be provided by the visual processing[4] and (2) a mathematical framework needs to be used that allows for such a mixing of correspondences.

The paper is structured as followed. First, we discuss related work on the formalization of the RBM estimation problem in Section 1.1 and about the finding of feature correspondences in Section 1.2. Section 2 describes all methods used in this work, including the formulation of the class of RBM. Furthermore, the section briefly describes the process of feature extraction and matching, and the 3D-2D pose estimation algorithm. Section 3 is concerned with the settings under which all methods are applied and presents the results. Section 4 discusses the results and briefly summarizes the contribution of this paper.

1.1 Mathematical aspects of RBM estimation

Motion estimation in different scenarios has been approached successfully (e.g. [4–6]), a recent overview on monocular rigid pose estimation can be found in [7]. The different methods for RBM estimation that have been proposed can be separated into feature based (see, e.g., [8, 9]), optic flow based (see, e.g., [10, 11]) and direct methods where no explicit representations as features or optic flow

[4] In this context we coined the term 'early cognitive vision' in [3].

vectors are used but image intensities are directly matched (see, e.g., [12–15]). A summary of the comparison of optic flow and feature based methods can be found in [16]. Solutions for feature based methods can be divided into linear and iterative algorithms. Linear algorithms use a limited number of feature correspondences (usually $n \geq 4, 5$). Proposed solutions include the use of 3 points [17], 6 points [18], 4 points in general position [19] and 3 lines in general position [20]. Iterative algorithms (see, e.g. [9, 21, 22]) make use of large correspondence sets, where closed form solutions no longer perform efficiently. The main difference to linear algorithms is that nonlinear algorithms require a starting point (an initial guess of the motion parameters) and need multiple iterations to find the motion parameters. Furthermore, feature based motion estimation algorithms also differ in the usage of different mathematical representations and error functions to be minimized. An example using points is proposed in [21], using 2D to 2D, 2D to 3D and 3D to 3D point correspondences. Other examples make use of line correspondence as in [4, 5].

All feature based solutions mentioned above require the extraction and matching of visual features, most commonly point or line features. However, as we will show, the RBM estimation problem can not be completely solved based one feature type alone, since there are cases where standard methods fail due to certain particularities in the data set. In addition, since different feature types are localized with different accuracy, the precision of the estimated motion depends on the feature types used. As a consequence, we argue that a general, precise and robust motion estimation algorithm makes use of an set of features as complete as possible for describing the scene.

The joint use of point and line correspondences requires a mathematical framework that allows for such a combination. This is mathematically non-trivial since a system of constraint equations on different entities need to be defined and, indeed, most algorithms are based on point correspondences only (see, e.g., [9, 18, 19, 21]). In recent years, some linear solutions have been proposed for the pose estimation problem making use of n points or n lines [6, 23]. These solutions provide algorithms and mathematical formulations allowing to use either points or lines, but not a combination of both mathematical representations within the same system of constraint equations.

However, there exist some algorithms which allow for a combination of correspondences of different types of constraints (see, e.g., [24, 25]). For our work, we chose the algorithm [24], since, in addition of being able to deal with different visual entities, it does optimization on 3D constraint equations, using a twist formulation (see, e.g., [4, 26]). This formulation directly acts on the parameters of rigid-body motions, (i.e., $SE(3)$) avoiding the use of additional non-linear constraints (e.g. enforcing rotation quaternions) to make sure that the found solution actually is in $SE(3)$.

A similar approach is [27], which uses a non-linear least squares optimization to optimize the 2D re-projection error (instead of a 3D error as in [24]). As shown in [28], this approach performs better than the POSIT (**P**ose from **O**rthography and **S**caling with **IT**erations) algorithm [29] in tracking scenarios. The work [27] extends the original publication [30] by the possibility to include 3D-point to 2D-line correspondences.

Note that some recent work, including [31, 32], proposed monocular, multiple view evaluation of a constrained motion (line or conic section). In this work we are interested in the unconstrained case. Recent work in a similar driving environment as the outdoor sequences used this work include [33] concerning the estimation of a three dimensional velocity vector using stereo data. The proposed approach uses two consecutive image pairs in stereo sequences, where the main concept is the decoupling of the postion estimation and velocity estimation, allowing both the use of sparse and dense disparity estimation algorithms. Similar work in indoor robotic environments include the work of [34] addressing the inverse kinematics problem, providing a model for tracking kinematic chains with restriced degrees of freedom.

1.2 Computing feature correspondences

Relevant research does not only concern the mathematical aspects of RBM estimation, but also the extraction and matching of features. Visual scenes contain different kinds of local image structures (e.g., texture, junctions, local edges, homogeneous image patches) with different properties with regards to the correspondence problem, in particular line structures suffer from the aperture problem. The statistical distribution of these structures can be efficiently represented by making use of the concept of intrinsic dimensionality (see [2, 35]). In general, it can be stated that approximately 10% of the local image structures correspond to edges, 1% to junction-like structures, and the remaining 90% to texture or homogeneous areas. There is no strict delimitation, but a continuum between texture and homogeneous image areas since recording noise on homogeneous areas already represents some weak texture (that of course is unsuitable for correspondence finding) and also most textured areas have very low contrast making them unsuitable for reliable correspondence finding. The distribution of occurrences of different image structures can vary significantly across images. Texture and junctions lead to point correspondences, which result in a stronger mathematical constraint (two constraint equations) than edge/line correspondences (one single constraint equation). However, in particular in man-made environments, edge features can be dominating. SIFT features [36] and their derivatives (for a review see [37]) describe semi-local textures and allow for very robust (but not necessarily precise, see below) correspondence finding, resulting in point correspondences that provide two constraint equations each. In contrast to that, edges are frequently occurring features that allow for robust and precise line–correspondences, that are however in mathematical terms 'weak', since they result in only one constrain equation.

Besides the frequency of occurrence, there exist also differences in the precision with which these features can be localized. We observed problems in the precision of SIFT correspondences in particular in the sequences in the driver assistance domain where motion blur and other sources of noise (such as rain drops, fog, etc) influence the recording process. In contrast to SIFT features, edge/line features represent more simple structures for which sub-pixel accuracy can be achieved with higher precision. Moreover, we observed that for scenes taken in an indoor environment, there occur cases where not enough SIFT features could be extracted, leading to an underconstrained estimation problem.[5] This also occurred in some (although very few) cases in the outdoor scenario.

Since feature matching is a fundamental problem in computer vision, many feature detectors and techniques for matching have been proposed over the years (for an overview see [37]). The term of 'feature detector' covers the process of extracting of wide range of interest points, where each of them usually defines a location in an image with large gradients in several directions. The approach of using a corner detector for stereo matching was initially proposed by Moravec [38] and later improved by Harris and Stephens [39], and has since been used for a wide range of computer vision applications involving point matching. Harris also showed the value of corner features for recovering structure from motion [40]. Zhang [41] showed successful matching of Harris corners over large image ranges and motions by using a correlation window for identifying potential correspondences. Other examples of feature points used for motion estimation include junctions [42,43] and SIFT features [44]. As an alternative to point features, line features have been used for motion estimation [23,45].

Regardless of the type of feature or constraints introduced to matching, finding correspondence remains a real challenge due the change of the 3D viewpoint, resulting in the perspective distortions of features. The Harris corner detector, for example, is very sensitive to changes in image scale. In the field of RBM estimations, where objects move and thereby change size, Harris corners will encounter difficulties for temporal matching. Therefore, a good feature descriptor for matching is invariant towards change in rotation and scale. The SIFT descriptor [36] provides these properties, and is therefore used in this evaluation (for a discussion of different descriptors in the context of correspondence finding, we refer again to [37]).

2 Methods

In this section, we explain the different parts of the motion algorithm. In Section 2.1, we briefly explain the mathematical formulation of the RBM estimation problem. In Section 2.2, we describe the features and their extraction. The stereo matching as well as the temporal matching is described in Section 2.3. Finally, in Section 2.4, we describe the complete motion estimation process. In this section, we will write 2D entities in lower case e, 3D entities in upper case E, predicted entities as $\hat{e}$ and the matched ones as $\check{e}$. When relevant, we denote in which image 2D entities belong to using subscript, e_l corresponding to a 2D entity in the left image and e_r to a 2D entity in the right image. Furthermore, when relevant we

[5] We are confident that the reason is not a bad parameter choice of the SIFT processing, since another group confirmed our experience on the specific data set using their motion estimation algorithm.

make use of superscript e^t for describing the instant of time t at which a given entity was observed.

2.1 Rigid Body Motion

A Rigid Body Motion $\mathcal{M}$ consisting of a translation $\boldsymbol{t}$ and a rotation $\boldsymbol{r}$ is described by six parameters, three for the translation $\boldsymbol{t} = (t_1, t_2, t_3)$ and three for the rotation axis $\boldsymbol{r} = (r_1, r_2, r_3)$ (see Figure 1). This allows for the formulation of the transformation between a visual entity according to this motion.

$$\mathcal{M}^{(\boldsymbol{t},\boldsymbol{r})}(E) = \hat{E} \tag{1.1}$$

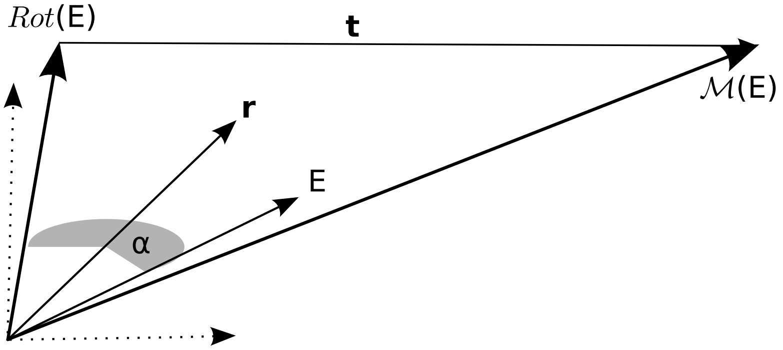

Fig. 1. Representations of a Rigid Body Motion by a combination of rotation (coded in axis-angle representation) and translation. First a rotation $\mathrm{Rot}(E)$ is performed around the axis $\boldsymbol{r}$. Then the 3D entity E is moved according to the translation vector $\boldsymbol{t}$. This allows for the formulation of the transformation between a visual entity in one frame, and the same entity in the next frame. The norm of this rotation axis codes the angle of rotation $\alpha = \|\boldsymbol{r}\|$.

The problem of computing the RBM from correspondences between 3D objects and 2D image entities is referred as 3D-2D pose estimation [46,47]. The 3D entity (3D object information) needs to be associated to a 2D entity (2D correspondence of the same object in the next image) according to the perspective projection $\mathcal{P}$.

$$\mathcal{P}(\mathcal{M}^{(\boldsymbol{t},\boldsymbol{r})}(E)) = \hat{e} \tag{1.2}$$

There exist approaches (in the following called projective approaches) that formalize constraints directly on equation (1.2) (see e.g., [30]). An alternative is, instead of formalizing the pose estimation problem in the image plane, to associate a 3D entity to each 2D entity: a 2D image point together with the optical center of the camera spans a 3D line (see figure 2a) and an image line together with the optical center generates a 3D plane (see figure 2b). In case of a 2D point $\check{\boldsymbol{x}}$, we denote the 3D line that is generated in this way by $\boldsymbol{L}(\check{\boldsymbol{x}})$. The RBM estimation problem can be formulated for 3D entities as:

$$\mathcal{M}^{(\boldsymbol{t},\boldsymbol{r})}(\boldsymbol{X}) \in \boldsymbol{L}(\check{\boldsymbol{x}}) \tag{1.3}$$

where $\boldsymbol{X}$ is the 3D point and $\check{\boldsymbol{x}}$ the 2D point it is matched with. Such a formulation in 3D has been applied by, e.g., [47,48], coding the RBM estimation problem in a twist representation that can be computed iteratively on a linearized approximation of the RBM. For that, we want to formulate constraints between 2D image entities and 3D object entities, where a 2D image point together with the optical center of the camera spans a 3D-line (see Figure 2a) and an image line together with the optical center generates a 3D-plane (see Figure 2b).

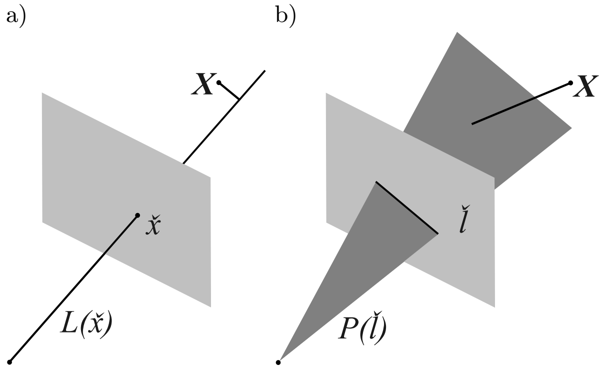

Fig. 2. Geometric interpretation of constraint equations (assuming the camera geometry is known): **a)** From an image point and the camera optical center, a 3D-line can be generated The 3D-point 3D-line constraint realizes the shortest Euclidean distance between the 3D-point and the 3D-line. **b)** From an image line and the camera optical center, a 3D-plane can be generated. The 3D-point 3D-plane constraint realizes the shortest Euclidean distance between the 3D-point and the 3D-plane.

A Rigid Body Motion $\mathcal{M}^{(\boldsymbol{t},\boldsymbol{r})}$ can be formalized in different ways, describing the same motion in different formulations. One of them is the formulation of *twists* [26], which is used in this work and described briefly in the following. Twists have a straightforward linear approximation (using a Taylor series expansion) and lead to a formalization that searches in the six dimensional space of RBMs (i.e, SE(3)). In the twist formulation, an RBM is understood as a rotation of angle α around a line $\boldsymbol{L}$ in 3D space with direction $\boldsymbol{w = (w_1, w_2, w_3)}$ and moment $\boldsymbol{w} \times \boldsymbol{q}$, where $\boldsymbol{q} = (q_1, q_2, q_3)$ is a point on that line. The vectors $\boldsymbol{w}$ and the cross-product of $\boldsymbol{w} \times \boldsymbol{q}$ are referred to as Plücker coordinates. In addition to the rotation, a translation with magnitude λ along the line $\boldsymbol{L}$ is performed. Then, an RBM can be represented as follows:

$$\hat{E} = e^{\alpha\tilde{\xi}}(E) = \mathcal{M}^{(\boldsymbol{t},\boldsymbol{r})}(E) \tag{1.4}$$

with

$$e^{\alpha\tilde{\xi}} = \sum_{n=0}^{\infty} \frac{1}{n!}(\tilde{\xi}\alpha)^n \tag{1.5}$$

with $\tilde{\xi}\alpha$ being the 4×4 matrix with 6 motion parameters to be estimated

$$\tilde{\xi}\alpha = \begin{pmatrix} 0 & -\alpha w_3 & \alpha w_2 & \alpha v_1 \\ \alpha w_3 & 0 & -\alpha w_1 & \alpha v_2 \\ -\alpha w_2 & \alpha w_1 & 0 & \alpha v_3 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{1.6}$$

with

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} w_3 q_2 - w_2 q_3 + \lambda w_1 \\ w_1 q_3 - w_3 q_1 + \lambda w_2 \\ w_2 q_1 - w_1 q_2 + \lambda w_2 \end{pmatrix} \tag{1.7}$$

By using the exponential representation in Equation 1.5, a straightforward linearization is given by:

$$e^{\tilde{\xi}\alpha} \approx I_{4\times 4} + \tilde{\xi}\alpha \tag{1.8}$$

A 3D-line $\boldsymbol{L}$ can be expressed as two 3D vectors $(\boldsymbol{\nu}, \boldsymbol{\mu})$. The vector $\boldsymbol{\nu}$ describes the direction and $\boldsymbol{\mu}$ describes the moment which is the cross product of a point $\boldsymbol{X}$ on the line and the direction $\boldsymbol{\mu} = \boldsymbol{X} \times \boldsymbol{\nu}$. The null space of the equation $\boldsymbol{X} \times \boldsymbol{\nu} - \boldsymbol{\mu} = \boldsymbol{0}$ is the set of all points on the line, and can be expressed in matrix form as follows:

$$\mathcal{F}^{\boldsymbol{L}}(\boldsymbol{X}) = \begin{pmatrix} 0 & \nu_3 & -\nu_2 & -\mu_1 \\ -\nu_3 & 0 & \nu_1 & -\mu_2 \\ \nu_2 & -\nu_1 & 0 & -\mu_3 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{1.9}$$

Combining the above formulation of a 3D-line with a 3D-point $\boldsymbol{X}$ allows the creation of the 3D-point/3D-line constraint using the linearization from Equation 1.8 as:

$$\mathcal{F}^{\boldsymbol{L}(\check{x})}\left((I_{4\times 4} + \alpha\tilde{\xi})\boldsymbol{X}\right) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{1.10}$$

Here, the value $||\mathcal{F}^{\boldsymbol{L}(\check{\boldsymbol{x}})}(\hat{\boldsymbol{X}})||$ can be interpreted as the Euclidian distance between the moved point $\hat{\boldsymbol{X}}$ and the closest point on the line $\boldsymbol{L}$ [47,49]. Note that, although we have 3 equations for one correspondence the matrix is of rank 2 resulting in only 2 constraints.

A 3D-plane $\boldsymbol{P}$ can be expressed by defining the components of the unit normal vector $\boldsymbol{n}$ and a scalar (Hesse distance) δ_h. The null space of the equation $\boldsymbol{n} \cdot \boldsymbol{X} - \delta_h = \boldsymbol{0}$ is the set of all points on the plane, and can be expressed in matrix form as follows:

$$\mathcal{F}^{\boldsymbol{P}(\check{\boldsymbol{l}})}(\boldsymbol{X}) = \begin{pmatrix} n_1 \; n_2 \; n_3 \; -\delta_h \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{pmatrix} = 0 \tag{1.11}$$

Combining the above formulation of a 3D-plane together with a 3D-point $\boldsymbol{X}$ allows for the creation of the 3D-point/3D-plane constraint using the linearization from Equation 1.8:

$$\mathcal{F}^{\boldsymbol{P}(\check{\boldsymbol{l}})} \left((I_{4\times 4} + \alpha\tilde{\xi})\boldsymbol{X} \right) = 0 \tag{1.12}$$

Note that the value $|\mathcal{F}^{\boldsymbol{P}}(\hat{\boldsymbol{X}})|$ can be interpreted as the Euclidean distance between the moved point $\hat{\boldsymbol{X}}$ and the closest point on the plane $\boldsymbol{P}$ (see [47]). These 3D-point/3D-line and 3D-point/3D-plane constraints result in a system of linear equations, the solution of which is found by iterative optimization (for details see [50]).

2.2 Feature Extraction

Visual Primitives form a feature based image representation that has been developed in the European project ECOVISION [51], which was focused on the modeling of early cognitive vision [3]. We believe that this representation provides robust means for finding correspondences across stereo and time, which are necessary for addressing the problem of RBM estimation.
Visual Primitives are extracted at image points representing edge structures, encoded as values of different visual modalities: position $\boldsymbol{m}$, orientation θ, phase ω, color $\boldsymbol{c}$ and local optic flow $\boldsymbol{f}$. Consequently, a multi-modal primitive is described by the following vector:

$$\boldsymbol{\pi} = (\boldsymbol{m}, \theta, \omega, \boldsymbol{c}, \boldsymbol{f}, p)^T \tag{1.13}$$

where p is the size of the image patch represented by the primitive. The problem of matching primitives was discussed in [52, 53], and we make use of the same criteria in the present evaluation. The matching criterion over all modalities is defined as:

$$d(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j) = \sum_k w_k d_k(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j) \tag{1.14}$$

where w_k is the relative weighting of the modality k and d_k being the distance of the modality k modality between the two primitives $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$.

Scale-Invariant Feature Transform feature extraction provides a set of robust features invariant to scaling and rotation [36]. Moreover, SIFT features are also very resilient to the effects of image noise. These properties make SIFT features widely used in many vision application involving the task of feature matching. SIFT features are extracted in a four step process [36].

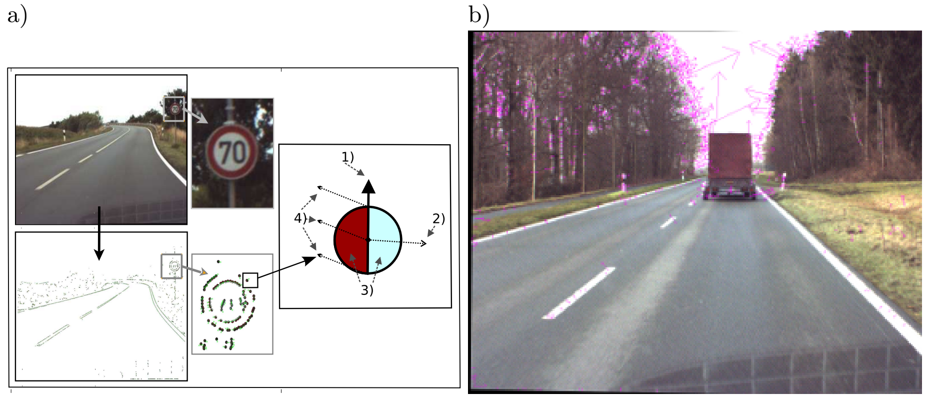

Fig. 3. The feature descriptors used in this work. **a)** Extracted visual primitives, modeling line features using multiple visual modalities, where: 1. stands for the orientation (θ), 2. for the phase (ω), 3. for the color ($\boldsymbol{c}$), and 4. for the optic flow ($\boldsymbol{f}$). **b)** Features extracted using the Scale Invariant Feature Transform (SIFT).

Scale-space extrema detection: The first stage of computation searches over all scales and image locations by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.

Keypoint localization: During the second step a detailed model is fit to determine location and scale for each candidate location, then keypoints are selected based on measures of their stability.

Orientation assignment: One or more orientations are assigned to each keypoint location based on local image gradient directions.

Keypoint descriptor: The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination.

During orientation assignment, the gradient orientation histogram is computed in the neighborhood of the feature point location. This serves as the feature-descriptor defined as a vector containing the values of all the orientation histogram entries, corresponding to the lengths of the arrows shown in Figure 2.2b. All the properties of the feature point are measured relative to its dominant orientation, providing invariance to rotation. Matchings are found by identifying the nearest neighbor from the set of SIFT features, defined as the keypoint with minimum Euclidean distance for the invariant descriptor vector.

2.3 Feature Matching

Having defined the process of feature extraction and metrics for matching, we now want to apply these to the image sequence used in this paper. In the context of RBM estimation from stereo sequences, the correspondence problem is twofold: first stereo correspondences have to be found, then temporal correspondences (as described in Section 2.3). Note that the temporal correspondence problem suffers from higher ambiguity than stereo matching, since the epipolar constraint is not directly applicable, and need to be replaced by a neighborhood search.

One of the sequences used is shown in Figure 4a, where the left column (resp. right) contains the images obtained from the left (resp. right) camera, while the rows show the stereo images taken at different time steps. For image acquisition, a calibrated stereo rig was used and the captured images have been undistorted and rectified. The task of finding correspondences consisting of two steps illustrated in Figure 4b is briefly described in the next two subsections.

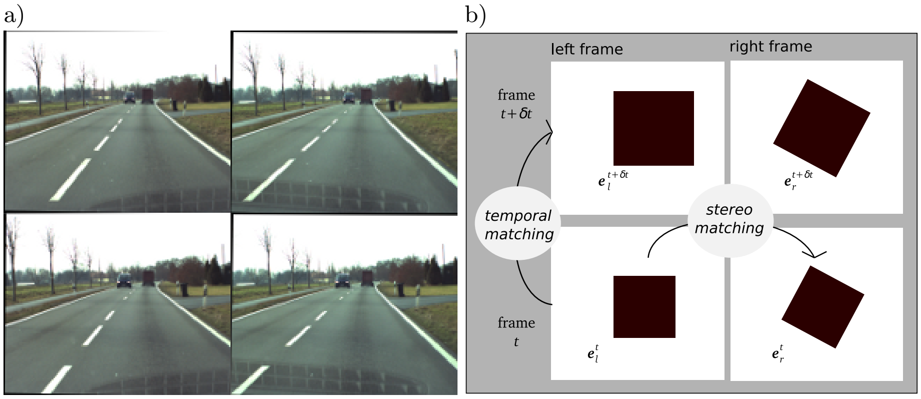

Fig. 4. Feature matching: **a)** Example frames from one of the sequences, columns showing images from the left (resp. right) camera, and the rows depicting images from different frames over times. **b**) The two tasks of feature matching required in the context of 3D-2D pose estimation. First stereo matches have to be found (at time t, see bottom row). If a stereo match was found for a given feature, the corresponding feature in next frame (time $t + \delta t$) has to be identified (illustrated in the first column).

Stereo Matching: Considering a feature e_l in the left image, the corresponding feature e_r in the right image has to be found, as illustrated by the bottom row in Figure 4b. Since the epipolar geometry is known and the images are rectified (see Figure 4a), the search space can be limited to horizontal scan lines. Primitives are matched using the similarity constraint defined by their modalities (see [52] for details). The matching of SIFT features uses a nearest neighbors search using a k-d tree [54] and an approximation algorithm, called the Best-Bin-First (BBF) [55]. This is an approximation algorithm in the sense that it returns the closest neighbor with a high probability. For an evaluation of the matching performance, we refer the reader to [36] for SIFT features, and [52] for primitives. An example of stereo matching from SIFT features is illustrated in Figure 5.

Having found stereo correspondences of the different features (representing lines or points), a 3D point is computed for each correspondence. The 3D point is regarded as valid if it is located in front of the camera.

Temporal Matching: Temporal matching involves finding correspondences between the 3D-points reconstructed from stereo at time t and the 2D features (primitives and SIFT) extracted at time $t + \delta t$ (upper row in Figure 4a). This is achieved by matching all e_l^t with all $e_l^{t+\delta t}$ for which there exists a valid 3D point. Temporal matching is done in the very same manner as stereo-matching with the exception that the epipolar constraint is not applicable. Therefore, the computational complexity is significantly higher than for stereo matching. Fortunately, the number of features for temporal matching has already been considerably reduced. Furthermore, we constrain the search of temporal matches within a neighborhood of the feature's previous position. The size of this neighborhood is called temporal disparity threshold. For the experiments, we both use a maximum and minimum threshold to this disparity. The minimum disparity threshold disregards temporal feature matches which do not move sufficiently in the image. These normally correspond to structures very distant to the camera, which would not serve as valuable contributions to the set of temporal features matches. During the experiments a threshold of minimum 2 and maximum of 150 pixels have been found adequate for all outdoor sequences, allowing to match nearby features at speed over 100km/h.

For the indoor sequence (see Figure 6d) where an object is moving, rather than the camera, a segmentation of background and the object becomes necessary. Using a minimum disparity threshold solves this problem for all point features (including SIFT), since the background is static, and therefore all non-moving temporal matches can be disregarded. For primitives the problem is different. Since primitives often are part of a global edge structure, temporal matches for a given primitive may be found at any point along the global edge structure, since primitives located on a global edge have similar attributes. Therefore, we constrain the temporal matching of primitives to a region around the robot, reducing the number of incorrect matched primitives.

After temporal matching, each correspondence contains a left entity in the current frame e_l^t, and the entity matched in the next left frame $e_l^{t+\delta t}$, where e_l^t has an associated 3D point from stereopsis (matched with e_r^t). An example of temporal matching of SIFT features is illustrated in Figure 5.

2.4 Applying the Pose Estimation Algorithm

Having 3D-2D feature correspondences we can now apply the pose estimation algorithm for estimating the motion in stereo sequences. The first step is to project the 2D temporal correspondence matches from time $t + \delta t$ from the image plane to 3D coordinates, by using the information from camera calibration. The corresponding 2D-points (SIFT) or lines (primitives) from the next left frame generate 3D-lines or 3D-planes, respectively. So, from the 3D-2D correspondences, we derive constraints describing the motion between two 3D entities

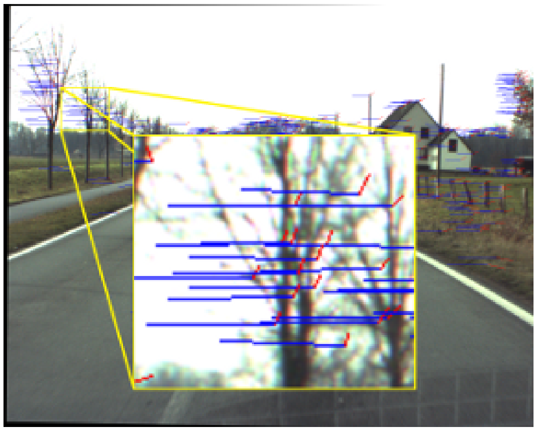

Fig. 5. Results for the matching of SIFT features (corresponding to the left-top frame in figure 4a), depicting stereo and temporal matches. The disparity of stereo correspondences between e_l^t and e_r^t are depicted with blue lines. The disparity of temporal correspondences between e_l^t and $e_l^{t+\delta t}$ are depicted with red lines and show radial disparities in the images, with a point of expansion in the image center indicating a forward translation.

(see Equation(1.10, 1.12). A 3D-point/2D-line correspondence (primitives) leads to one independent constraint equation and a 3D-point/2D-point correspondence (SIFT) leads to two independent constraint equations [50]. Earlier work has shown [56] that 3D-point/2D-point correspondences produce more accurate estimates with the same number of correspondences.
The constraints derived from Equation 1.10 and 1.12 result in a system of linear equations the solution of which is found iteratively (for details see [24, 50]).

Eliminating outliers using RANSAC Another challenge is the selection of correspondences resulting in correct estimates, referred to as inlier. Erroneous correspondences, referred to as outliers, are introduced due to the inherent ambiguity of stereo and temporal matching of local features, and should be neglected. By applying the Random Sample Consensus [17] (RANSAC) approach, we identify a set of inliers resulting in more accurate estimation results. The algorithm explained below uses a correspondence set containing either SIFT, primitives, or a combination of both.

1. The set of correspondences is divided into generation set and evaluation set, where the evaluation set is only to be used for performance statistics.
2. From the generation set, a random set of correspondences is selected, corresponding to 8 independent constraints.
3. The pose estimation algorithm is run with this random set.
4. The inverse of the computed motion is applied to all remaining 3D entities reprojected in the generation set and back into the image. The distance

between the original and re-projected 2D feature, referred as the *deviation*, serves as measure of accuracy for the estimated motion.
5. A consensus set is formed from all correspondences of the generation set, using correspondences with a deviation below a given threshold τ.
6. If the size of the consensus set is above a certain percentage ξ of the of size of the generation (referred to as the consensus threshold), the estimated motion is regarded as correct and the algorithm continues, otherwise the algorithm goes back to step 2.
7. The pose estimation algorithm is re-run with the whole consensus set, which is considered to only contain inliers.

During the experiments a deviation threshold of $\tau = 1$ pixel for SIFT an $\tau = 2$ pixels for primitives have been found as adequate. The consensus threshold of $\xi = 70\%$ has shown to result in precise motion estimates while at the same time succeeding for most of the frames.

3 Results

The four stereo image sequences used are recorded with a calibrated camera system and contain three outdoor scenes and one indoor scene (see Figure 6). Note that the outdoor sequences are available as Set 3 on the web site `http://www.mi.auckland.ac.nz/EISATS/`. On this website, initiated by Reinhard Klette and colleagues (see, e.g., [57]), data sets recorded in the driver assistance domain are provided for comparison of computer vision algorithms. The indoor sequence is provided together with calibration matrices and ground truth on the web site `http://www.mip.sdu.dk/covig/Data/PACO`.

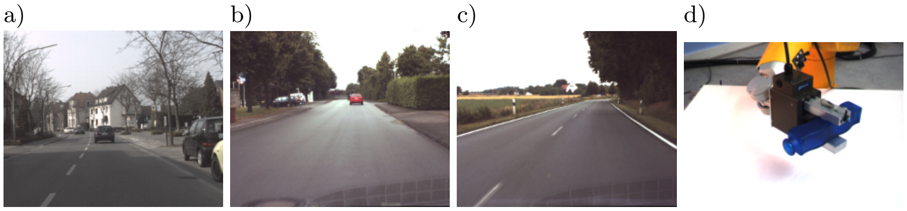

Fig. 6. Example frames from each of the four sequences: a) Lippstadt-Town2, b) s_03a_01#000, c) s_06a_01#000, d) PACO-5

As illustrated in Figure 6a-c the outdoor scene consists of images recorded through a car's windscreen. For these outdoor sequences, logging of car data by an onboard computer provides information about velocity and steering angle. Hence, this data does not describe the full motion (only translation and rotation around the Y-axis) and is subject to noise. Therefore, it can only be used as reference and should not be considered as actual ground truth. The car data is shown in Figure 8. The left column of plots shows the car's translation in mm. and the right column shows the car's rotation angle in radians. In these figures, the rotation along the Y-axis corresponds to steering input and the translation on Z-axis represents the forward motion of the car. During turns, the car moves forward, sideways, and rotates along the Y-axis. Notice the slight correlation between rotation (right columns) and translation along the X-axis (see left column). For the Lippstadt-Town2 sequences no rotation data is available and the forward translation (T_Z) is equal to the velocity provided by the car's onboard computer.

Image Sequence	number of frames	Primitives extracted	SIFT extracted	Primitives matched	SIFT matched
Lippstadt-Town2	437	1679(*1427*)	4818(*2500*)	665(*194*)	572(*136*)
s_03a_01#000	1180	1223(708)	1347(*70*)	669(*49*)	395(*0*)
s_06a_01#000	848	779(250)	1182(*89*)	449(*291*)	216(*0*)
PACO-5	60	1857(1565)	442(*335*)	394(*239*)	15(*6*)

Table 1. Average and minimal numbers (in italics) of extracted and matched (stereo and temporal) features per frame.

The indoor scene depicts a robot holding and moving a blue vase, and consists of a sequence of 70 stereo-frames. The camera is situated in front of the robot and, throughout the entire sequence, the robot rotates the vase 360 degrees around one axis. In this experiment, we used an industrial robot, meaning that the robot's controllet provide high accuracy ground truth data, plotted alongside with the motion estimation results (see Figure 7)

Since the motion estimation uses statistical methods for removing outliers, the number of extracted features is directly related to the robustness of the estimated motion. If too few features are extracted and matched, RANSAC will no longer be applicable. The number of extracted and matched features is shown in Table 1. The reason for the low number of correspondences in the PACO-5 sequence is explained by the relatively small size of the moving object in the images, whereas all outdoor sequences undergo ego-motions and therefore displays an apparent world motion over the whole images (apart from other moving cars). Furthermore we see significant changes in the number of extracted features between the different outdoor sequences. As depicted in Table 1, the number of extracted SIFT features is significantly higher in an urban environment (Lippstadt-Town2) than on a countryroad (s_03a_01#000 and s_06a_01#000).

Indoor sequence

Results are shown in Figure 7, depicting both ground truth and estimated motion. Furthermore, the mean error over whole sequences is recorded for all three

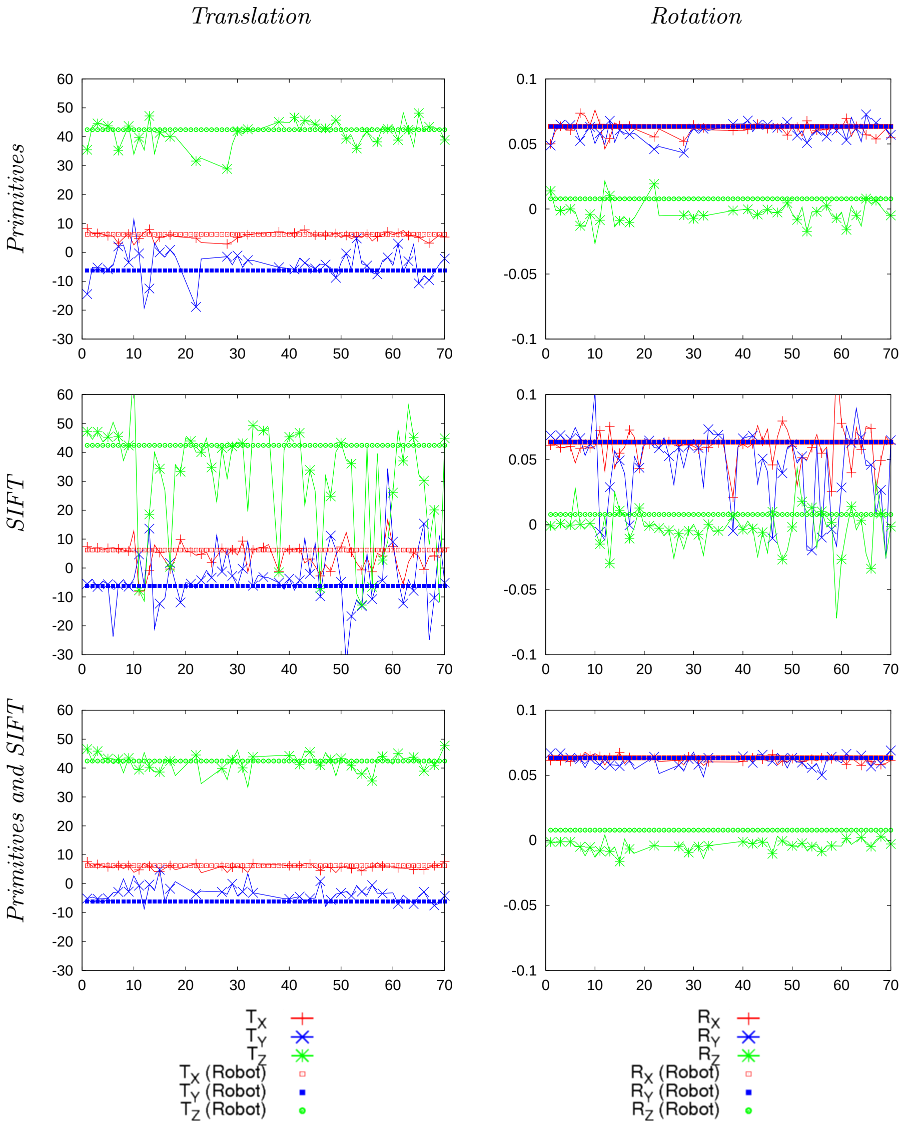

Fig. 7. Results for ego-motion estimates for indoor sequence PACO-5. Left column contains plots translation in mm and right column contains the plots displaying rotation angle in radians. Each row corresponds to a different feature set: **a)** Primitives **b)** SIFT **c)** Combination of primitives and SIFT.

feature sets in Table 2. In this table, the primitives lead to more accurate motion estimates than SIFT features, and the combination of the two lead to the best results. In particular, the motion estimation based on SIFT features shows large deviation from the true motion for many frames (see Figure 7b), due to the comparatively small amount of matched SIFT features (see Table 1). The estimation would basically fail in terms of most applications that make use of the estimated motion data. The reason for this is the small number of extracted and matched SIFT features (see Tab.1 and Figure 12).

Feature Set	Tx (mm)	Ty (mm)	Tz (mm)	Rx (rad)	Ry (rad)	Rz (rad)
Primitives	1.0	4.1	3.2	0.003	0.005	0.005
SIFT	3.2	6.9	14.1	0.009	0.018	0.008
Primitives + SIFT	0.8	3.0	2.5	0.002	0.003	0.004

Table 2. Mean error for image sequence PACO-5.

Outdoor Sequences

For the outdoor sequences with a sufficient number of extracted and matched features (see Table 1), SIFT features alone lead in general to better or similar results than primitives alone (compare the top to the middle rows in Figure 9, 10 and 11). This is explained by the fact that line correspondences fail to constrain the motion in all directions. Especially, in the outdoor scenes the lane structure (in particular the lane markers) dominate the scenes, and due to their particular orientation (i.e., nearly radial from the car's heading), they do not constrain the ego-motion in the z–direction. Therefore, the estimation results of the forward translation, based on primitives alone (see Figure 9a, 10a and 11a) are very unprecise. However, it can be seen that primitive correspondences constrain effectively the 5 other motion parameters, and even in a slightly better way than the SIFT correspondences. We assume that this is caused by the higher precision of the primitive localization compared to SIFT.

Furthermore we observe that usage of SIFT features in an urban environment, provides very accurate estimations, when comparing the reference data with the estimated motion (first row in Figure 8 and 9). Combining SIFT with additional primitive correspondences does not further improve the, since the great number of extracted and matched SIFT features already serves as a sufficient correspondence set.

However in the other environmets, some frames do not contain enough SIFT features, due to the small amount of texture within the scene (see Figure 12c). (see Figure 12a). Then, few spurious features can very quickly result in large errors in the estimated RBM. Another source of outliers in this type of scenario are pot holes or speed bumps (see Figure 12b). These sudden and violent vertical motions result in blurred images, which make feature extraction and matching a

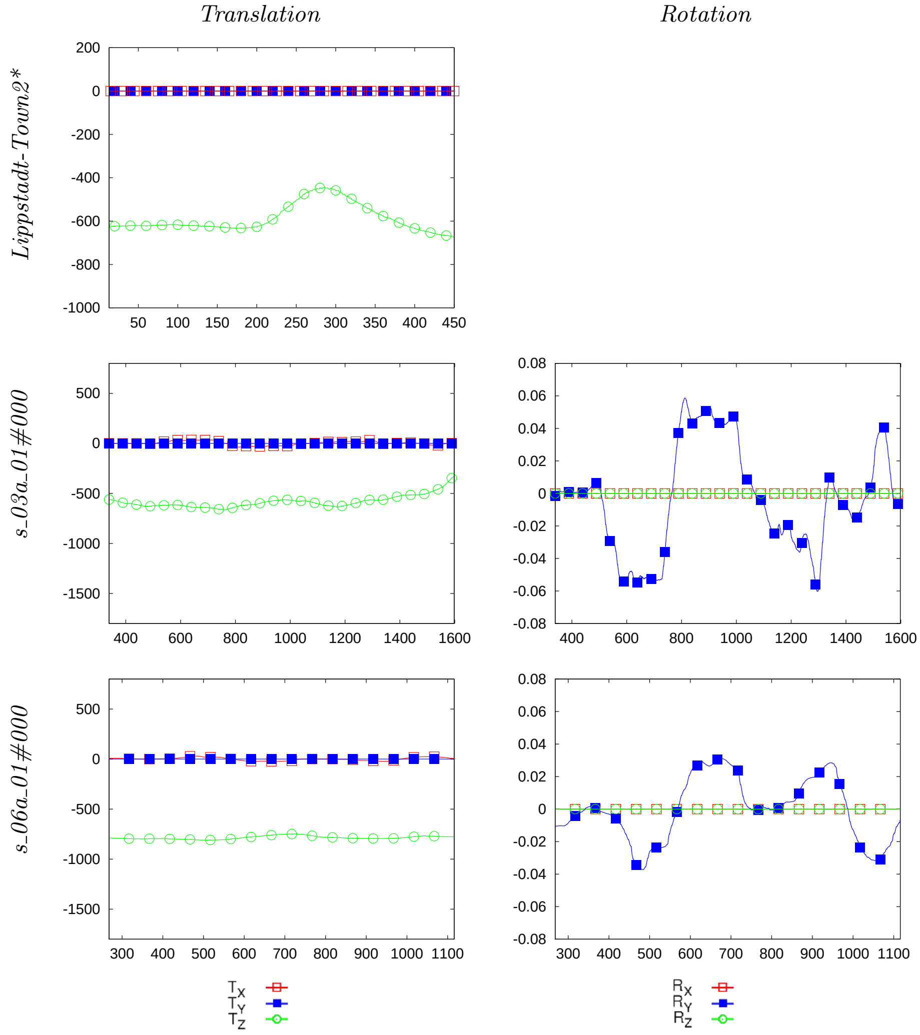

Fig. 8. Reference data computed from car data for all outdoor sequences. Left column contains plots for the car translation in mm. and right column contains the plots displaying rotation angle in radians. Each row corresponds to a different reference sequence. * For the Lippstadt-Town2 sequence (first row) no rotation data is available and T_Z is computed directly from the car's velocity.

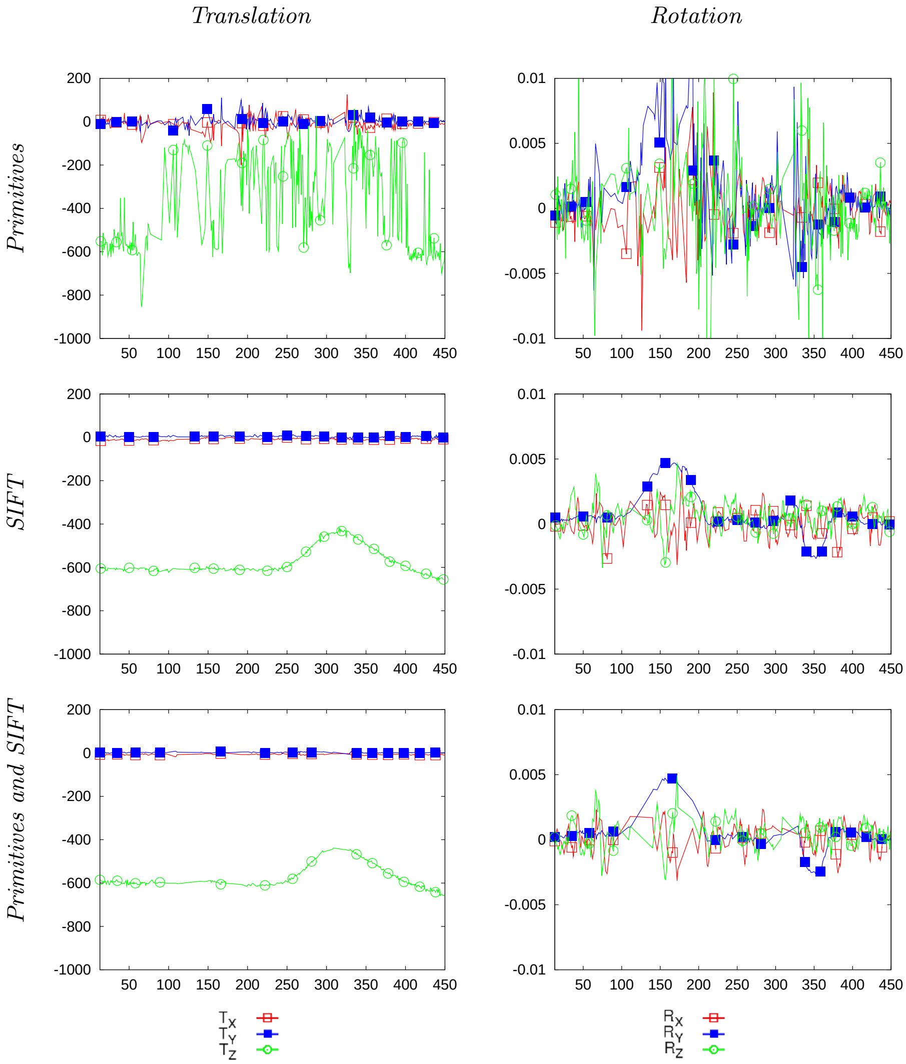

Fig. 9. Results of ego-motion estimates for the outdoor sequence 'Lippstadt-Town2' over 437 frames in a city environment. The left column shows translation estimates, in mm., and the right column shows rotation angle estimates, in radians.

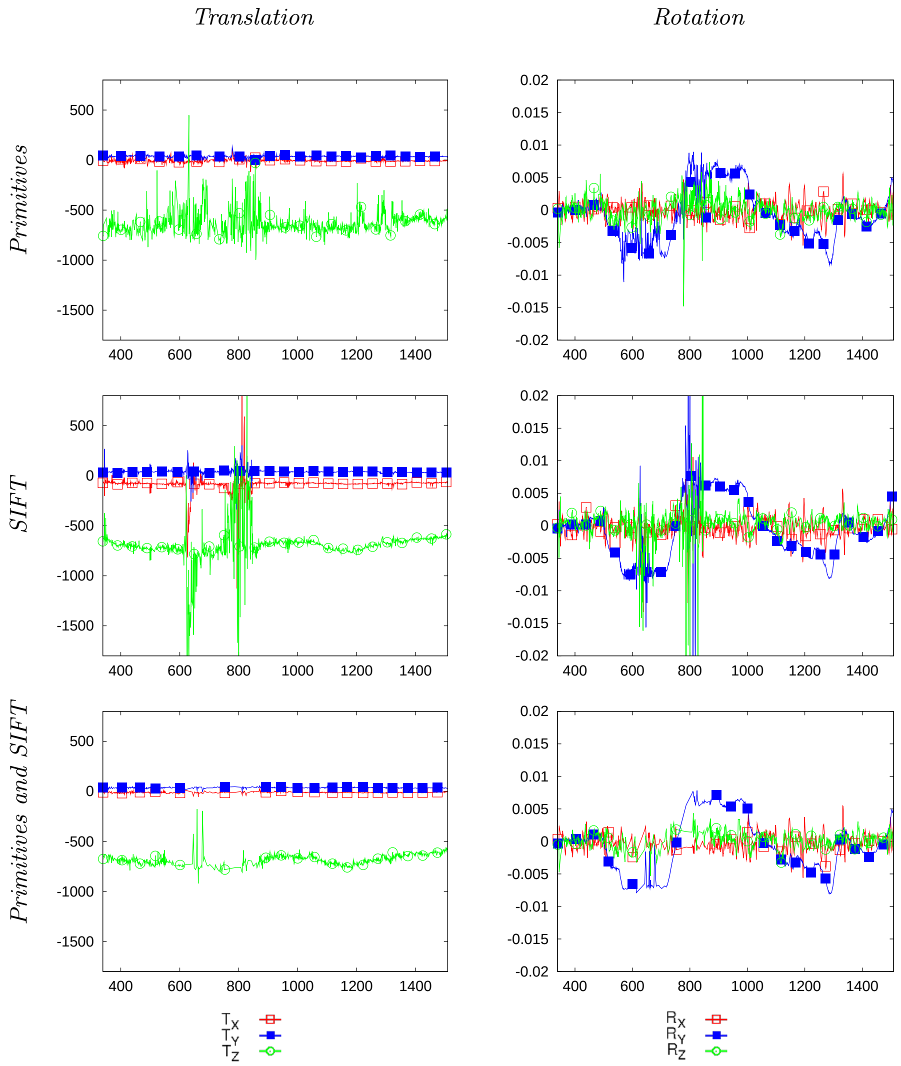

Fig. 10. Results of ego-motion estimates for outdoor sequence 's_03a_01#000' over 1180 frames. The left column shows translation estimates, in mm., and the right column shows rotation angle estimates, in radians. Each row corresponds to a different feature set.

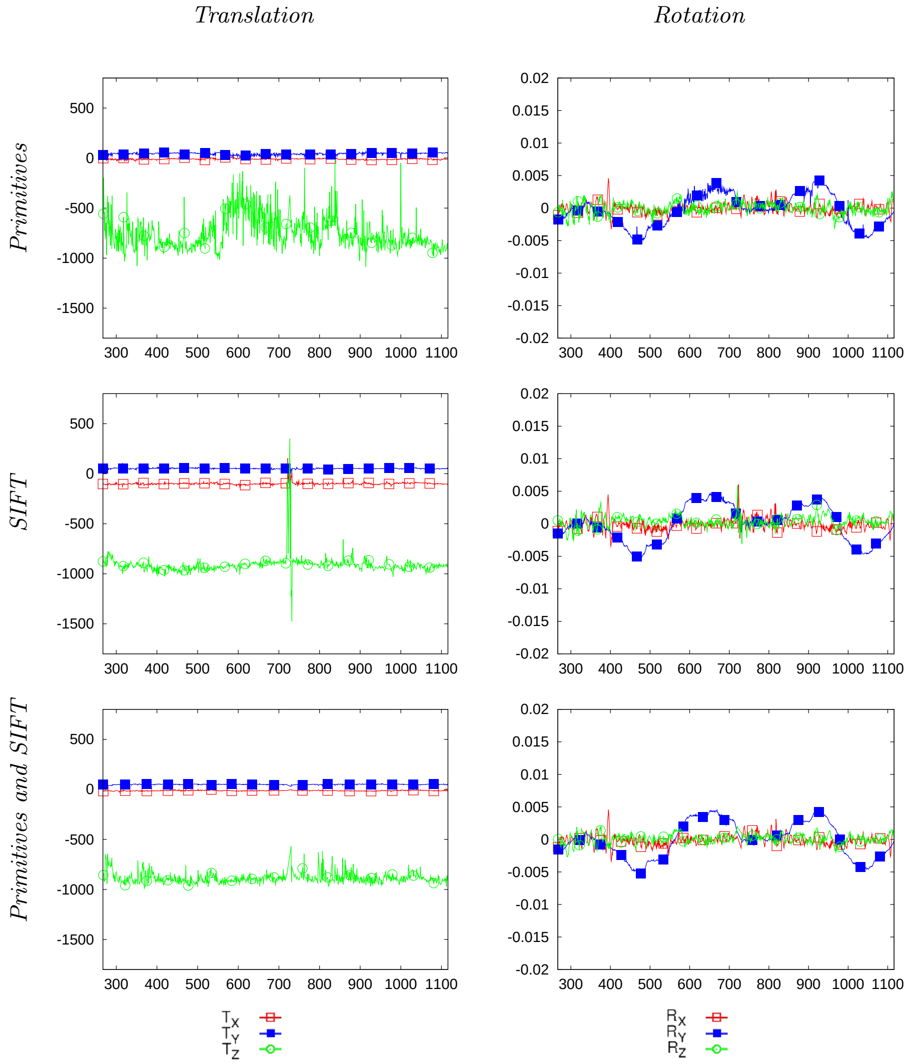

Fig. 11. Results for ego-motion estimates for outdoor sequence 's_06a_01#000' over 848 frames. The left column shows translation estimates, in mm., and the right column shows rotation angle estimates, in radians. Each row corresponds to a different feature set.

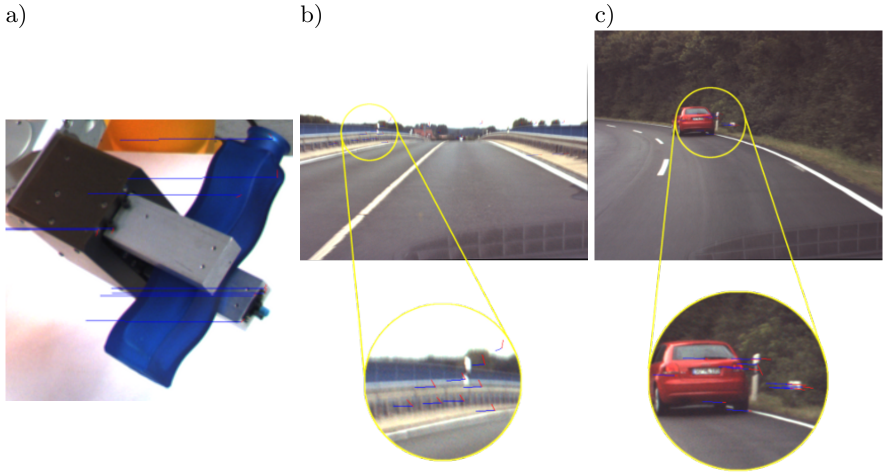

Fig. 12. Examples of problematic image data: **a)** for the indoor sequence only few SIFT features are extracted and matched. **b)** Some frames from the outdoor sequences (as in ‘s_06a_01#000’, frame 723) are blurred due bumps in the road, resulting in large estimation errors (spikes) as shown in Figure 11. **c)** Independently moving objects (e.g., other vehicles on the road) provide temporal matches which are not coherent with the ego motion. This becomes especially a problem if the scene contains few matched features (as in ‘s_03a_01#000’, frame 600-800), resulting in erroneous motion estimates (see Figure 10).

difficult task. In such cases, we observe that a combination of SIFT and primitives make the estimation more stable to outliers. Over all sequences, we observe that a combination of features (see bottom row in Figure 10 and 11) consistently improves the robustness. Note that, we did not make use of any temporal regularization (which of course would improve the results further) since we were interested in investigating the different properties and consequences of different kinds of correspondences. In this context, temporal regularization would effectively hide the occasional failures of the estimation, that are relevant indicators of the estimation's robustness.

4 Conclusion

We have evaluated point and line correspondences on a set of sequences with large variations across and within the sequences. There are a number of issues involved in such an investigation, namely (1) the extraction process of features, (2) the correspondence finding process, (3) the mathematical formulation of constraints allowing for their mixing and (4) the efficient handling of outliers.

For the process of finding correspondences, an evaluation of matching needs to been done, using ground truth data in form of disparity maps.

From a mathematical point of view, line features represent weaker correspondences than point features as they provide only one constraint equation, against two for the point constraints. Because of this, pathologic scene structures, like the road markers in the outdoor sequences can lead to an ill-definition of the motion estimation problem. On the other hand, point features such as SIFT lead to an ill-definition in the case of the indoor scene, due to the lack of texture in the scene, whereas edge features give stable results. The same holds true for certain sub-parts of the outdoor sequences which were dominated by sky and edge structures. Besides the avoidance of severe outliers, we also observed that the additional use of edge-features increases the precision of the estimates due to their frequent occurrence in visual scenes as well as their more accurate localization.

As a consequence, besides the mathematical properties of the constraints, we need to take the actual distribution of features in images into account. Here, we can observe that this distribution changes significantly across as well as within sequences. Hence, a robust motion estimation must rely on the combination of point and line features and hence relies on a *rich feature processing* as well as on a formulation of the motion estimation problem that allows for the mixing of both kinds of correspondences.

SIFT features become extracted at local textured structures. In our current research, we investigate the use of junctions as an additional feature type which is extractable with high local precision and rich semantic. We expect a further improvement of stability and precision by such further enrichment of our representations.

Acknowledgement

This work has been supported by the European Commission - FP6 Project DRIVSCO (IST-016276-2). We also would like to thank Daniel Grest and Emre Başeski for fruitful discussions.

References

1. Ball, R.: The theory of screws. Cambridge University Press (1900)
2. Zetzsche, C., Barth, E.: Fundamental limits of linear filters in the visual processing of two dimensional signals. Vision Research **30** (1990) 1111–1117
3. Krüger, N., Hulle, M.V., Wörgötter, F.: Ecovision: Challenges in early-cognitive vision. International Journal of Computer Vision **72** (2007) 5–7
4. Bregler, C., Malik, J.: Tracking people with twists and exponential maps. IEEE computer Society conference on Computer Vision and Pattern Recognition (1998) pp.8–15
5. Christy, S., Horaud, R.: Iterative pose computation from line correspondences. Comput. Vis. Image Underst. **73** (1999) 137–144
6. Ansar, A., Daniilidis, K.: Linear pose estimation from points or lines. In: ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV, London, UK, Springer-Verlag (2002) 282–296
7. Lepetit, V., Fua, P.: Monocular model-based 3d tracking of rigid objects. Found. Trends. Comput. Graph. Vis. **1** (2005) 1–89
8. Roach, J., Aggarwall, J.: Determining the movement of objects from a sequence of images. IEEE Transactions on Patterm Analysis and Machine Intelligence **2** (1980) 554–562
9. Lowe, D.G.: Three–dimensional object recognition from single two images. Artificial Intelligence **31** (1987) 355–395
10. Bruss, A., Horn, B.: Passive navigation. Computer Vision, Graphics, and Image Processing **21** (1983) 3–20
11. Horn, B.: Robot Vision. MIT Press (1994)
12. Waxman, A., Ullman, S.: Surface structure and 3-D motion from image flow: A kinematic analysis. International Fournal of Robot Research **4** (1985) 72–94
13. Negahdaripour, S., Horn, B.: Direct passive navigation. IEEE Transactions on Pattern Analysis and Machine Intelligence **9** (1987) 168–176
14. Steinbach., B.G.E.: An image-domain cost function for robust 3-d rigid body motion estimation. In: 15th International Conference on Pattern Recognition (ICPR-2000. Volume 3. (2000) 823–826
15. Steinbach, E.: Data driven 3-D Rigid Body Motion and Structure Estimation. Shaker Verlag (2000)
16. Torr, P.H.S., Zisserman, A.: Feature based methods for structure and motion estimation. In: ICCV '99: Proceedings of the International Workshop on Vision Algorithms, London, UK, Springer-Verlag (2000) 278–294
17. Fischler, R., Bolles, M.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM **24** (1981) 619–638
18. Schaffalitzky, F., Zisserman, A., Hartley, R.I., Torr, P.H.S.: A six point solution for structure and motion. In: ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part I, London, UK, Springer-Verlag (2000) 632–648

19. Horaud, R., Conio, B., Leboulleux, O., Lacolle, B.: An analytic solution for the perspective 4-point problem. Comput. Vision Graph. Image Process. **47** (1989) 33–44
20. Dhome, M., Richetin, M., Lapreste, J.T.: Determination of the attitude of 3d objects from a single perspective view. IEEE Trans. Pattern Anal. Mach. Intell. **11** (1989) 1265–1278
21. Haralick, R., Joo, H., Lee, C., Zhuang, X., Vaidya, V., Kim, M.: Pose estimation from corresponding point data. Systems, Man and Cybernetics, IEEE Transactions on **19** (1989) 1426–1446
22. Liu, Y., Huang, T., Faugeras, O.: Determination of camera location from 2-d to 3-d line and point correspondence. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) **12** (1989) 28–37
23. Phong, T., Horaud, R., Yassine, A., Tao, P.: Object pose from 2-D to 3-D point and line correspondences. International Journal of Computer Vision **15** (1995) 225–243
24. Rosenhahn, B., Granert, O., Sommer, G.: Monocular pose estimation of kinematic chains. In Dorst, L., Doran, C., Lasenby, J., eds.: Applied Geometric Algebras for Computer Science and Engineering. Birkhäuser Verlag (2001) 373–383
25. Bretzner, L., Lindeberg, T.: Use your hand as a 3-d mouse, or, relative orientation from extended sequences of sparse point and line correspondences using the affine trifocal tensor. In: ECCV (1). (1998) 141–157
26. Murray, R., Li, Z., Sastry, S.: A mathematical introduction to robotic manipulation. CRC Press (1994)
27. Grest, D., Herzog, D., Koch, R.: Monocular body pose estimation by color histograms and point tracking. In: DAGM-Symposium. (2006) 576–586
28. Grest, D., Petersen, T., Krüger, V.: A Comparison of Iterative 2D-3D Pose Estimation Methods for Real-Time Applications, to appear. In: SCIA, Oslo, Norway, Springer (2009)
29. Dementhon, D.F., Davis, L.S.: Model-based object pose in 25 lines of code. International Journal of Computer Vision **15** (1995) 123–141
30. Araujo, H., Carceroni, R., Brown, C.: A fully projective formulation to improve the accuracy of lowe's pose–estimation algorithm. Computer Vision and Image Understanding **70** (1998) 227–238
31. Wolf, L., Shashua, A.: Lior wolf and a. shashua. on projection matrices $p^k - > p^2, k = 3, ..., 6$, and their applications in computer vision. In: In Proceedings of the 8th International Conference on Computer Vision, IEEE Computer Society Press (2001) 412–419
32. Avidan, S., Shashua, A.: Trajectory triangulation: 3d reconstruction of moving points from a monocular image sequence. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 348–357
33. Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., Cremers, D.: Efficient dense scene flow from sparse or dense stereo data. In: ECCV '08: Proceedings of the 10th European Conference on Computer Vision. (2008) 739–751
34. Rosenhahn, B., Brox, T., Cremers, D., Seidel, H.P.: Modeling and tracking line-constrained mechanical systems. In Sommer, G., Klette, R., eds.: 2nd Workshop on Robot Vision. Volume 4931 of LNCS. (2008) 98–110
35. Felsberg, M., Kalkan, S., Krüger, N.: Continuous dimensionality characterization of image structures. Image and Vision Computing (accepted for publication in a future issue)
36. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision **2** (2004) 91–110

37. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (2005) 1615–1630
38. Moravec, H.: Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical Report CMU-RI-TR-3, Carnegie-Mellon University, Robotics Institute (1980)
39. Harris, C.G., Stephens, M.: A combined corner and edge detector. In: 4th Alvey Vision Conference. (1988) 147–151
40. Harris, C.G.: Geometry from visual motion. MIT Press (1992)
41. Zhang, Z., Deriche, R., Faugeras, O., Luong, Q.T.: A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Artificial Intelligence **87** (1995) 87–119
42. Kalkan, S., Shi, Y., Pilz, F., Krüger, N.: Improving junction detection by semantic interpretation. In: VISAPP (1). (2007) 264–271
43. Pollefeys, M., Koch, R., van Gool, L.: Automated reconstruction of 3D scenes from sequences of images. ISPRS Journal of Photogrammetry and Remote Sensing **55** (2000) 251–267
44. Lowe, D.G.: Robust model-based motion tracking through the integration of search and estimation. Int. J. Comput. Vision **8** (1992) 113–122
45. Krüger, N., Jäger, T., Perwass, C.: Extraction of object representations from stereo imagesequences utilizing statistical and deterministic regularities in visual data. DAGM Workshop on Cognitive Vision (2002) 92–100
46. Grimson, W., ed.: Object Recognition by Computer. The MIT Press, Cambridge, MA (1990)
47. Rosenhahn, B., Sommer, G.: Adaptive pose estimation for different corresponding entities. In van Gool, L., ed.: Pattern Recognition, 24th DAGM Symposium. Springer Verlag (2002) 265–273
48. Rosenhahn, B., Perwass, C., Sommer, G.: Cvonline: Foundations about 2d-3d pose estimation. In CVonline: On-Line Compendium of Computer Vision [Online]. R. Fisher (Ed). http://homepages.inf.ed.ac.uk/rbf/CVonline/. (2004)
49. Selig, J.: Some remarks on the statistics of pose estimation. Technical Report SBU-CISM-00-25, South Bank University, London (2000)
50. Krüger, N., Wörgötter, F.: Statistical and deterministic regularities: Utilisation of motion and grouping in biological and artificial visual systems. Advances in Imaging and Electron Physics **131** (2004) 82–147
51. ECOVISION: Artificial visual systems based on early-cognitive cortical processing (EU–Project). http://www.pspc.dibe.unige.it/ecovision/project.html (2001–2003)
52. Pugeault, N., Krüger, N.: Multi–modal matching applied to stereo. Proceedings of the BMVC 2003 (2003) 271–280
53. Krüger, N., Felsberg, M.: An explicit and compact coding of geometric and structural information applied to stereo matching. Pattern Recognition Letters **25** (2004) 849–863
54. Freidman, J.H., Bentley, J.L., Finkel, R.A.: An algorithm for finding best matches in logarithmic expected time. ACM Trans. Math. Softw. **3** (1977) 209–226
55. Beis, J.S., Lowe, D.G.: Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In: In Proc. IEEE Conf. Comp. Vision Patt. Recog. (1997) 1000–1006
56. Pilz, F., Shi, Y., Grest, D., Pugeault, N., Kalkan, S., Krüger, N.: Utilizing semantic interpretation of junctions for 3d-2d pose estimation. Proceedings of the ISVC 2007 (2007) 271–280

57. Hermann, S., Klette, R.: A study on parameterization and preprocessing for semi-global matching. Technical report, Computer Science Department, The University of Aukland, New Zealand (2008) http://citr.auckland.ac.nz/techreports/2008/CITR-TR-221.pdf.