Efficient Feature Parameterisation for Visual SLAM Using Inverse Depth Bundles

Tobias Pietzsch
Technische Universität Dresden, 01062 Dresden, Germany
Tobias.Pietzsch@inf.tu-dresden.de

Abstract

Flexibility and robustness of visual SLAM systems have been shown to benefit from an inverse depth parameterisation of features. However the increased number of 6 parameters per feature presents a problem to real-time EKF SLAM implementations because their computational complexity scales quadratically with the size of the state vector. Recent work tackles this for instance by converting the representation of well-established features from inverse to regular depth. In this paper, we propose a parameterisation where bundles of features share a common representation of the view-point they were initially observed from. According to the experiments performed, a feature occupies effectively about 1.5 state parameters in the proposed approach, allowing real-time performance for maps with more than 200 features.

1 Introduction

Simultaneous localisation and mapping (SLAM) is concerned with estimating the pose of a mobile robot while simultaneously building a map of the environment it is navigating. The problem is formulated in a Bayesian framework where noisy measurements are integrated over time to update a probability distribution of the state of a dynamical system, consisting of landmark positions and the robot's pose. Since the seminal work by Davison [3] visual SLAM, tackling this problem with a camera (monocular or stereo, typically hand-held) as the only sensor, has received a lot of attention from both the vision and robotics communities. Davison's approach of using the Extended Kalman Filter (EKF) as the underlying probabilistic mechanism has been adopted widely, e.g. [3, 5].

In [3] and related systems 3D landmarks (or features) are parameterised by their Euclidean scene coordinates. From the beginning, it was well understood that the Euclidean parameterisation is not well suited to the low-parallax situations occurring with very distant or newly initialised features whose depth estimate has not yet converged. The shape of the uncertainty region for such features is not approximated well by a Gaussian in Euclidean space. Montiel et al. [7] proposed an inverse depth parameterisation which successfully handles these cases. However, one issue with this parameterisation is that with 6 parameters a inverse depth feature occupies a portion of the state vector that is twice as large as for the Euclidean representation. Given the quadratic complexity of the EKF with respect to state size this leads to severely restricted map sizes (60-80 features) feasible for real-time operation. Civera et al. [1] address this issue by converting inverse depth features to the Euclidean parameterisation once their uncertainty region approaches

Gaussianity. An approach to further reduce the state size has been presented by Gee et al. [5]. They detect groups of features lying in a common plane. These features can then share a representation of the plane, requiring only two additional state entries per feature to describe its location within the plane.

In this paper, we propose a new feature parameterisation which is based on a similar idea. Instead of grouping features by co-planarity, we form groups of features which have been initialized from the same camera frame, i.e., from the same point of view. The result is an inverse depth parameterisation where a group of features shares a common 6 parameter anchor. Only one additional state entry per feature is required, making the representation more efficient than Euclidean parameters when 4 or more features are initialised from the same frame.

Pupilli and Calway [9] use a similar representation in the context of a particle filtering SLAM framework. They also point out the potential decrease in state size, although they make no attempt to actively exploit this. To keep the state small we try to minimize the number of camera frames used for feature initialisation, and initialise many features in each of these frames. This is related to the ideas of using keyframes [6] and representing features in local coordinate frames [4]. Klein and Murray [6] perform mapping on a sparse set of keyframes using bundle adjustment. Our work differs in that we use all feature measurements from all frames to refine the map. Eade and Drummond [4] partition measurements into a set of nodes where inverse depth features are represented with respect to local coordinate frames. These nodes form a graph which is globally optimized. In contrast to their work, we represent the map in a single state vector, maintaining full correlations between all features.

In the next section we review the general EKF framework for visual SLAM. Forming the main contribution of this paper, Section 3 introduces the inverse depth bundle parameterisation starting from an alternative inverse depth parameterisation. In Section 4 we provide some details about the complete visual SLAM system used for experimentation. After presenting experimental results in Section 5, we conclude with Section 6.

2 EKF-Based Visual SLAM

We assume a stereo camera moving freely but smoothly in a static scene. The position of the camera with respect to a fixed scene coordinate frame is to be estimated, while simultaneously building a map of 3D points in the scene. The belief about the joint state $\mathbf{x}$ of the system is modelled as a multivariate Gaussian represented by its mean vector $\mu_{\mathbf{x}}$ and covariance matrix $\Sigma_{\mathbf{x}}$. The state vector can be divided into parts describing the state of the camera $\mathbf{x}_v$ and of map features $\mathbf{y}_i$.

$$\mu_{\mathbf{x}} = \begin{pmatrix} \mu_{\mathbf{x}_v} \\ \mu_{\mathbf{y}_1} \\ \vdots \\ \mu_{\mathbf{y}_n} \end{pmatrix} \qquad \Sigma_{\mathbf{x}} = \begin{bmatrix} \Sigma_{\mathbf{x}_v\mathbf{x}_v} & \Sigma_{\mathbf{x}_v\mathbf{y}_1} & \cdots & \Sigma_{\mathbf{x}_v\mathbf{y}_n} \\ \Sigma_{\mathbf{y}_1\mathbf{x}_v} & \Sigma_{\mathbf{y}_1\mathbf{y}_1} & \cdots & \Sigma_{\mathbf{y}_1\mathbf{y}_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{\mathbf{y}_n\mathbf{x}_v} & \Sigma_{\mathbf{y}_n\mathbf{y}_1} & \cdots & \Sigma_{\mathbf{y}_n\mathbf{y}_n} \end{bmatrix} \tag{1}$$

The state estimate is updated sequentially using the predict-update cycle of the EKF. Whenever a new image is acquired by the camera, measurements of map features can be made and used to update the state estimate, resulting in a decrease of uncertainty in the update step. In the prediction step, a process model is used to project the estimate forward in time. The process model describes how the state evolves during the period of "temporal blindness" between images. Similar to [3] the camera is assumed to be moving with constant linear and angular velocity. The (unknown) accelerations that cause deviation from this assumption are modelled as noise. The camera state is modelled as $\mathbf{x}_v = \begin{pmatrix}\mathbf{r} & \mathbf{q} & \mathbf{v} & \boldsymbol{\omega}\end{pmatrix}^\top$. Position and orientation of the camera with respect to the world frame $\mathscr{W}$ are described by the 3D position vector $\mathbf{r}$ and the quaternion $\mathbf{q}$. Translational and angular velocity are described by $\mathbf{v}$ and $\boldsymbol{\omega}$.

The EKF update step integrates new information from measurements of map features into the state estimate. A generative measurement model

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \delta \tag{2}$$

describes the measurement vector $\mathbf{z}$ as a function of the (true, unknown) state, affected by zero-mean Gaussian measurement noise δ. In the case of a stereo camera a measurement $\mathbf{z} = \begin{pmatrix}u, v, d\end{pmatrix}$ consists of the coordinates u, v of the projection of a feature in the reference camera and the disparity d. The current (prior) state estimate can be used to predict the expected measurement. The difference between the predicted and actual measurement is then used in the EKF update to improve the state estimate.

3 Parameterising Features by Inverse Depth Bundles

In this section we first introduce a view-point based feature parameterisation. This can be seen as an alternative representation to inverse depth features [7]. Using 7 parameters instead of 6 it is slightly less efficient than the "traditional" representation. The advantage here is the fact that 6 of the 7 parameters can be shared among features initialised from the same camera frame, leading to the bundle representation discussed in Section 3.2.

3.1 View-Point Based Feature Parameterisation

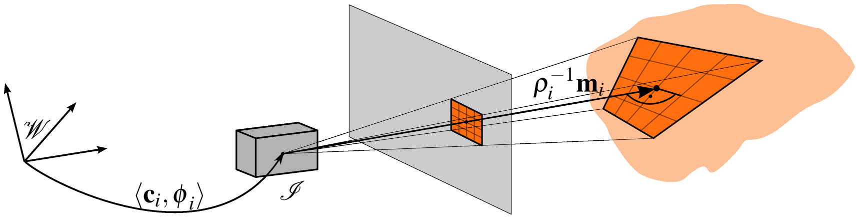
Based on the idea of inverse depth parameterisation [7], we introduce a new feature representation. We will refer to this representation as *view-point based* because it describes features in terms of the initial view-point, i.e., the camera pose at the time of initialisation.

In the probabilistic state, the i-th 3D point feature is parameterised by the 7-dimensional vector

$$\mathbf{y}_i = \begin{pmatrix}\mathbf{c}_i & \phi_i & \rho_i\end{pmatrix}^\top . \tag{3}$$

Here, the 3-vector $\mathbf{c}_i = \begin{pmatrix}x_i, y_i, z_i\end{pmatrix}$ is the camera position at the time of the first observation of the feature. The 3-vector $\phi_i = \begin{pmatrix}\phi_i^x, \phi_i^y, \phi_i^z\end{pmatrix}$ is an exponential rotation representing the camera rotation for this first observation. Finally ρ_i is the inverse depth of the feature on a ray in direction $\mathbf{m}_i$.

The ray to the feature is represented in the initialisation camera coordinate frame $\langle \mathbf{c}_i, \phi_i \rangle$. Thus, the unit vector $\mathbf{m}_i$ simply encodes the direction to the pixel where the feature was detected in the initial image. With respect to the initialisation camera frame there is no uncertainty about where the projection of the feature was observed, thus $\mathbf{m}_i$ is a fixed component of the model and not part of the probabilistic state vector. Furthermore, a template T of the appearance of the feature in the reference image is stored. The view-point based feature model is illustrated in Figure 1.



$\mathscr{W}$
$\langle \mathbf{c}_i, \phi_i \rangle$
$\mathscr{I}$
$\rho_i^{-1}\mathbf{m}_i$

Figure 1: View-point based feature model: The relative orientation of world frame $\mathscr{W}$ and initialisation camera frame $\mathscr{I}$ is given by translation $\mathbf{c}_i$ and rotation ϕ_i. The unit vector $\mathbf{m}_i$ defines a ray to the feature centre, and ρ_i is the inverse depth along this ray.

In comparison to the the classical inverse depth model [7], by parameterising the full camera rotation ϕ_i one additional degree of freedom is introduced, namely rotation about the ray to the feature. This is not observable from point measurements $\left(u,v,d\right)$ of the feature directly. However, via its (perfect) correlation to the camera rotation estimate at the time of initialisation it becomes correlated to other state variables. Hence, additional information on rotation about the ray to feature $\mathbf{y}_i$ is provided by measurements of *other* features. For future measurements, prior to correlation search, the template T is warped to account for varying appearance caused by view-point changes. Because the full rotation ϕ_i is used in warping the template, updating it's estimate can improve the accuracy of the predicted feature appearance for correlation search.

The second difference occurs with respect to initializing uncertainty of the feature ray. In [7], this results from a combination of uncertainty in the initial camera position and measurement uncertainty in the initial $\left(u,v\right)$ observation. This rests on the assumption that the initial measurement is subject to the same measurement error as any other measurement. This is justifiable if measurements are of some directly observable physical quantity, like laser range finder measurements of the distance to a wall. We argue, that for the case of making image measurements by correlation search the situation is different. The feature template is the projection of a scene surface in the initial camera image, *not* the scene surface itself. The measurement process proceeds by back-projecting the feature template to the (uncertain) scene surface and then projecting it to the (uncertain) current camera frame, where the projection is used for correlation search. However if the current camera pose is the initialisation pose this will always result exactly in the observed initial template *regardless* of feature depth or scene structure. The location of the template in the initial image is known with absolute certainty. Hence, we model the initial uncertainty of the feature ray as resulting from the camera pose uncertainty only. We assume no uncertainty in the pixel position $\left(u,v\right)$ for the initial observation (and thus, $\mathbf{m}_i$ is fixed).[1]

Having established the view-point based model, we note that the initial camera poses for features initialised in the same frame are the same and perfectly correlated to each other. Hence, those features can share their representation of $\mathbf{c}$, ϕ requiring only one additional parameter for each feature, namely its inverse depth ρ_i. This leads to the feature

[1] If we would correctly model the initialisation errors for the case of correlation search, we should include the pixel intensities of the template in the state vector and initialise their uncertainty with the variance of the intensity noise introduced by the camera.

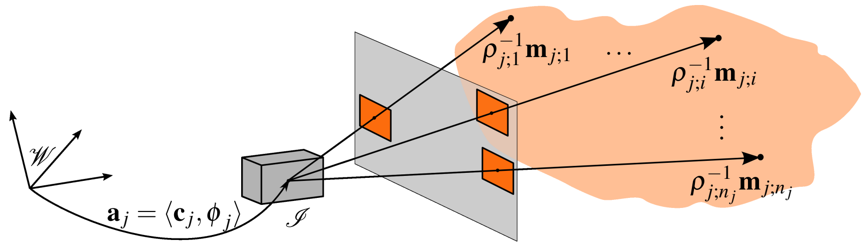

$\rho_{j;1}^{-1}\mathbf{m}_{j;1}$
$\dots$
$\rho_{j;i}^{-1}\mathbf{m}_{j;i}$
$\vdots$
$\rho_{j;n_j}^{-1}\mathbf{m}_{j;n_j}$
$\mathscr{W}$
$\mathbf{a}_j = \langle \mathbf{c}_j, \phi_j \rangle$
$\mathscr{I}$

Figure 2: Inverse depth feature bundle model: The anchor $\mathbf{a}_j$ represents the relative orientation of world frame $\mathscr{W}$ and initialisation camera frame $\mathscr{I}$ by translation $\mathbf{c}_j$ and rotation ϕ_j. Features $\mathbf{y}_{j;i}$ initialised with respect to this anchor are each represented by their inverse depths $\rho_{j;i}$ along rays $\mathbf{m}_{j;i}$.

bundle representation discussed next.

3.2 Inverse Depth Bundle Parameterisation

Given n view-based features initialised from the same camera frame, we can split their state representation into the 6 parameter anchor

$$\mathbf{a}_j = \begin{pmatrix} \mathbf{c}_j & \phi_j \end{pmatrix}^\top, \tag{4}$$

and n feature states

$$\mathbf{y}_{j;i} = \begin{pmatrix} \rho_{j;i} \end{pmatrix} \tag{5}$$

(with $i \in \{1, \dots, i\}$) wich are represented relative to the anchor $\mathbf{a}_j$. For each feature a unit vector $\mathbf{m}_{j;i}$ encodes the direction to the feature with respect to the initial camera frame that is defined by $\mathbf{a}_j$. The model is illustrated in Figure 2. The state vector then takes the form

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_v & \mathbf{a}_1 & \mathbf{y}_{1;1} & \dots & \mathbf{y}_{1;n_1} & \dots & \mathbf{a}_m & \mathbf{y}_{m;1} & \dots & \mathbf{y}_{m;n_m} \end{pmatrix}^\top \tag{6}$$

In the next subsection we will discuss the measurement model for the bundle representation. This is followed by a discussion of the inverse measurement model, i.e., how to initialise new anchors and features. Finally we discuss the state size reduction that can be achieved using inverse depth bundles and initialisation heuristics to maximize this effect.

3.2.1 Measurement Model

A measurement of a feature $\mathbf{y}_{j;i}$ from an inverse depth bundle can be modelled as a function of the current camera state, the anchor state, and the feature state

$$\mathbf{z}_{j;i} = \mathbf{h}(\mathbf{x}_v, \mathbf{a}_j, \mathbf{y}_{j;i}) + \delta, \tag{7}$$

where $\delta \sim N(0, \mathsf{R})$ is measurement noise with covariance $\mathsf{R} = \mathrm{diag}(\sigma_u^2, \sigma_v^2, \sigma_d^2)$. We proceed by deriving the function $\mathbf{h}(\mathbf{x}_v, \mathbf{a}_j, \mathbf{y}_{j;i})$ which gives the predicted measurement (u, v, d).Given the 3D coordinates $\rho_{j;i}^{-1}\mathbf{m}_{j;i}$ of the feature with respect to the anchor frame $\langle \mathbf{c}_j, \phi_j \rangle$ we compute the world coordinates of the feature as

$$\mathbf{y}_{j;i}^w = \mathbf{c}_j + \frac{1}{\rho_{j;i}} \mathsf{R}_{\phi_j} \mathbf{m}_{j;i}, \tag{8}$$

where R_{ϕ_j} is the 3×3 rotation matrix corresponding to the exponential rotation ϕ_j. Then, we transform the feature's world coordinates to the current camera coordinate frame given by $\langle \mathbf{r}, \mathbf{q} \rangle$, and obtain

$$\mathbf{y}^c_{j;i} = \mathsf{R}(\mathbf{q}^{-1})\,(\mathbf{y}_{j;i} - \mathbf{r}) = \frac{1}{\rho_{j;i}}\,\mathsf{R}(\mathbf{q}^{-1})\left(\rho_{j;i}\,(\mathbf{c}_j - \mathbf{r}) + \mathsf{R}_{\phi_j}\mathbf{m}_{j;i}\right) = \frac{1}{\rho_{j;i}}\,\vec{\mathbf{y}} \tag{9}$$

Finally, we obtain the projection of this point by the camera. For a monocular projection only the direction $\vec{\mathbf{y}} = \left(x, y, z\right)^\top$ to the feature is important. Thus, the $1/\rho_{j;i}$ factor can be dropped from Equation 9. In a stereo setting this must be compensated when computing the disparity, resulting in the following projection function

$$\Pi(\vec{\mathbf{y}}, \rho_{j;i}) = \begin{pmatrix} u \\ v \\ d \end{pmatrix} = \begin{pmatrix} f_u \frac{x}{z} + u_0 \\ f_v \frac{y}{z} + v_0 \\ \rho_{j;i}\, f_u \frac{b}{z} \end{pmatrix}, \tag{10}$$

where b is the stereo baseline, (u_0, v_0) is the principal point, and f_u resp. f_v is the focal length in multiples of pixel width resp. height. In summary, we obtain the measurement function

$$\mathbf{h}(\mathbf{x}_v, \mathbf{a}_j, \mathbf{y}_{j;i}) = \Pi\left(\mathsf{R}(\mathbf{q}^{-1})\left(\rho_{j;i}\,(\mathbf{c}_j - \mathbf{r}) + \mathsf{R}_{\phi_j}\mathbf{m}_{j;i}\right),\, \rho_{j;i}\right). \tag{11}$$

3.2.2 Initialising new Anchors and Features

We now consider the initialisation of a bundle of features into the state. Assume a set of newly detected features in the current frame is given by their initial observations $\{\mathbf{z}^0_1, \ldots, \mathbf{z}^0_n\}$ with $\mathbf{z}^0_i = \left(u_i, v_i, d_i\right)^\top$.

We start by augmenting the state with a new anchor $\mathbf{a}_j$ which represents the current camera position. The anchor state is obviously not dependent on any of the measurements. It is a function of the camera state only, representing a copy of the current camera pose

$$\mathbf{a}_j = \mathbf{g}_a(\mathbf{x}_v) = \begin{pmatrix} \mathbf{c}_j \\ \phi_j \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ \log(\mathbf{q}) \end{pmatrix}, \tag{12}$$

where $\log(\cdot)$ represents the conversion of a quaternion rotation to exponential coordinates. The anchor is appended to the state vector and the covariance matrix is updated as

$$\Sigma_{\mathbf{x}} := \mathsf{J}\Sigma_{\mathbf{x}}\mathsf{J}^\top \quad \text{with } \mathsf{J} = \begin{pmatrix} & \mathsf{I} & \\ \frac{\partial \mathbf{g}_{\mathbf{a}}(\mathbf{x}_v)}{\partial \mathbf{x}_v} & \mathbf{0} \cdots \mathbf{0} \end{pmatrix}. \tag{13}$$

This is followed by the initialisation of the features $\mathbf{y}_{j;i}$. For each feature, we first compute the unit vector $\mathbf{m}_{j;i}$ as the ray from the projection centre through the pixel $\left(u_i, v_i\right)$. As discussed above, we do not assume that u_i, v_i are subject to measurement error.

The ray $\mathbf{m}_{j;i}$ is represented with respect to the current camera coordinate frame. Consequently, the inverse depth $\rho_{j;i}$ of the feature along this ray is not dependent on the current camera, anchor, or any other features. The new feature state is a function of the observed disparity d_i only[2]

$$\mathbf{y}_{j;i} = \mathbf{g}_{\mathbf{y}}(d_i) = \left(\rho_{j;i}\right) = d_i\,\frac{\mathbf{m}^z_{j;i}}{f_u b}. \tag{14}$$

[2] For the monocular case, no disparity is available, and the inverse depth would be initialised to some heuristically determined value with a large uncertainty.

Here, $\mathbf{m}^z_{j;i}$ denotes the z component of the unit vector $\mathbf{m}_{j;i}$. The new feature is appended to the state vector. Because it is initially uncorrelated to the rest of the state, the covariance update is

$$\Sigma_{\mathbf{x}} := \begin{pmatrix} \Sigma_{\mathbf{x}} & \mathbf{0} \\ \mathbf{0}^\top & \sigma^2_{\mathbf{y}_{j;i}} \end{pmatrix} \text{ with } \sigma^2_{\mathbf{y}_{j;i}} = \frac{\partial \mathbf{g}_{\mathbf{y}}(d_i)}{\partial d_i} \sigma^2_d \frac{\partial \mathbf{g}_{\mathbf{y}}(d_i)}{\partial d_i}^\top . \tag{15}$$

3.2.3 Benefits of the Bundle Parameterisation

A bundle of n features (together with their shared anchor) occupies $6+n$ entries in the state. This means, that for $n \geq 2$ the bundle parameterisation will be more efficient than a straightforward inverse depth parameterisation. For $n > 3$ the bundle parameterisation is even more efficient than a Euclidean parameterisation of features by their 3D coordinates.

The actual benefits depend on the strategy employed to decide when to initialise new features. The strategy should be designed to minimize the number of anchors and ensuring that each anchor is shared by many features. As long as there is a minimum of 3 features per anchor, the bundle representation is at least as efficient as the Euclidean parameterisation. Currently we use the following initialisation heuristic. The camera image area is divided into a 4×4 grid. While making feature measurements in each new image we determine the number of empty grid cells. A grid cell is counted as empty if either, there are no features predicted to be visible in this cell, or, all attempts to measure visible features in this cell failed. If the fraction of empty cells is larger than a threshold (70% in our experiments) a new bundle of features is initialised. New feature candidates are selected to lie on salient image areas and to be evenly distributed in the image. At most 20 new features are initialised per bundle. In Section 5 we experimentally show that even with this simple strategy the bundle parameterisation allows to sustain real-time operation for maps of more than 200 features.

More complex strategies can be envisioned to further increase efficiency. For instance, the map could be divided into fixed features and temporary features. The map is continually augmented with temporary features until a camera pose is reached where no fixed features are observable. Then temporary features are removed from the state and a new bundle of fixed features is initialised. In this way “spatial overlap” between bundles would be reduced, while some accuracy would be sacrificed (hopefully only temporarily).

4 Description of the complete stereo SLAM system

This section provides some details on the stereo SLAM system that was used for experimentation. Images are acquired by a Point Grey Bumblebee® stereo camera with a resolution of 640 × 480 at 30 Hz. The raw images are then Bayer decoded and rectified on the GPU. Somewhat deviating from pure top-down SLAM methodology, we proceed by exhaustively searching both images for corners using the FAST corner detector [10]. The detected corners serve two purposes. First, they are used to determine candidate locations for new features (should we choose to initialise a new feature bundle in this frame). Second, they help to further reduce the number of pixels considered for correlation search during the measurement process.

After the camera pose for the current frame has been predicted the image projection and visibility of features is predicted. For each of the visible features a template for correlation search is then obtained by warping the feature template according to the homography induced by the current estimates of camera pose, anchor, and feature state. To restrict the camera poses from which each feature can be observed to a meaningful range, the size in pixels of the warped template is compared to that of the original template. Measurements are only attempted for features where the sizes diverge by no more than a empirically fixed threshold.

The reference image of the stereo pair is divided into a 4×4 grid. For each grid cell measurements are attempted until one feature could be successfully measured, starting with the feature with the largest uncertainty region. Measurements are obtained by correlation search for the warped template in gated 3σ search ellipses in both images of the stereo pair. Additionally, correlation search is restricted to image pixels which have at least one FAST corner in their 8-neighbourhood. For the left image search is further restricted to the epipolar line corresponding to the maximum found in the reference image.

Next we try to detect measurements which are due to erroneous feature matchings. The largest jointly consistent subset of successful measurements is computed using the Joint Compatibility Branch and Bound algorithm [8] in the form described in [2]. This consistent subset is then used in the EKF update of the state estimate. Features that did not pass the Joint Compatibility test are removed from the map.

We determine whether a new bundle of features should be initialised using the heuristic discussed in Section 3.2.2. For this purpose the KLT score is computed for the maxima among the detected FAST corners in the reference image. For pixels whose KLT score is above an empirically fixed threshold we search for a stereo correspondence in the left image. Among the successfully matched pixels a set of new features is selected that are well distributed in the reference image and sufficiently far from existing features. A bundle of these new features is then initialised into the state as discussed in Section 3.2.2.

5 Experimental Results

The proposed representation was evaluated with respect to state size and processing time on real image sequences. The pre-recorded sequences were processed on a 2 GHz Intel Core 2 Duo (using one core). The timing results given below include image acquisition times measured during sequence recording. We present results for two sequences here.

The first, “indoor,” sequence is recorded in a structured office environment. The camera moves in a single room where translational motion is restricted to a volume of approximately $1 \times 1 \times 2$ meters. Here, the need to initialise new feature bundles arises mainly because the camera rotates away from known features. At the end of the sequence the map contains 211 features and 20 anchors. Figure 4 shows views of the map before and after closing a loop. Throughout, the processing time stays within the real-time constraint of 33 ms per frame. The following table gives a breakdown of processing time for a representative frame towards the end of the sequence (the full map size has been reached):

3.5 ms	image acquisition, rectification
4.5 ms	corner detection
8 ms	feature prediction and correlation search
0.3 ms	Joint Compatibility test
10 ms	EKF update

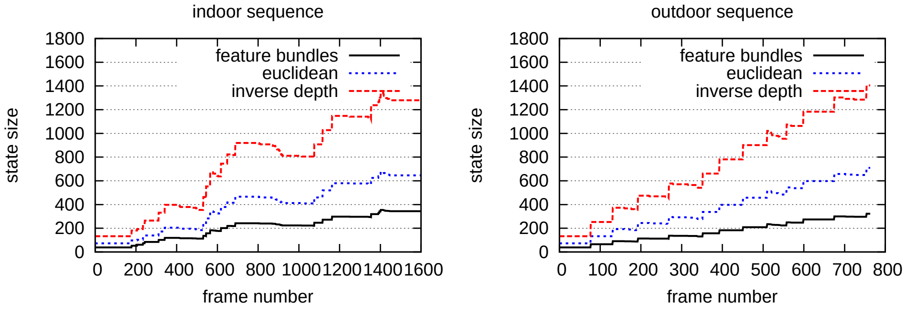


Figure 3: Evolution of the state vector size for the indoor (left) and outdoor (right) experiments. The solid curve on the bottom show the actual state size that was obtained with the bundle parameterisation. The dashed and dotted curves show the (hypothetical) state size when using inverse depth respective Euclidean coordinates.

Occasionally, initialisation of a new feature bundle is required. Depending on the number of new features and current size of the map this takes additional $0.3-3$ ms.

The second, “outdoor,” sequence shows a less structured environment and a more exploratory kind of motion. Here, the camera translates forward on a path of approximately 15 meters, roughly in the viewing direction. Initialisation is required less often in this scenario. New feature bundles have to be initialised, because mapped features have moved too close to, or past the camera. At the end of the outdoor sequence the map contains 232 features and 13 anchors. Processing times are very similar to that of the indoor sequence.

The evolution of the state size for both sequences is shown in Figure 3. The plots include the state size that would arise using 6-parameter inverse depth respective 3-parameter Euclidean features. Clearly, in both experiments the bundle parameterisation is effective in keeping the state vector small. Throughout both sequences the state size remains well below the hypothetical “Euclidean” state size. The effective state size per feature is 1.6 for the indoor respective 1.3 for the outdoor sequence.

Concluding, we wish to note that the threshold of initialising at most 20 features was empirically selected to ensure stable tracking while simultaneously keeping the number of anchors small (and not to produce artificially dense maps to the disadvantage of the other parameterisations). Lowering the threshold generates more anchors while producing maps of similar size.

6 Conclusion

In this paper we have presented a new feature parameterisation for visual SLAM. Our inverse depth bundle parameterisation exploits the fact that features initialised from the same camera frame can share large parts of their state representation, requiring only one additional parameter per feature. A simple initialisation heuristic was proposed which has proven successful in keeping the number of anchors small while ensuring stable tracking performance. Experimental results for two real stereo sequences have been presented. In both cases fully correlated maps with more than 200 features were processed in real-time – which would not have been possible using inverse depth or Euclidean parameterisations.

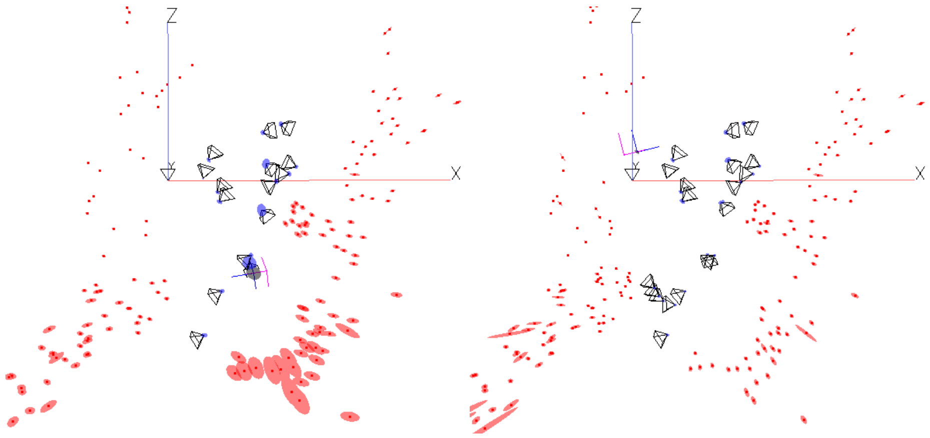


Figure 4: Two views of the map obtained for the indoor sequence, before and after loop closure. Pyramids indicate the location of the anchors with their uncertainties shown as filled ellipses. Feature locations are plotted as dots. The uncertainty ellipses for the features result from the anchor uncertainty and feature uncertainty wrt. the anchor.

References

[1] J. Civera, A. J. Davison, and J. M. M. Montiel. Inverse Depth to Depth Conversion for Monocular SLAM. In *ICRA 2007*, 2007.

[2] L. A. Clemente, A. J. Davison, I. Reid, J. Neira, and J. D. Tardós. Mapping Large Loops with a Single Hand-Held Camera. In *RSS*, 2007.

[3] A. J. Davison. Real-Time Simultaneous Localisation and Mapping with a Single Camera. In *ICCV*, 2003.

[4] E. Eade and T. Drummond. Monocular SLAM as a Graph of Coalesced Observations. In *ICCV 2007*, October 2007.

[5] A. P. Gee, D. Chekhlov, W. Mayol, and A. Calway. Discovering Planes and Collapsing the State Space in Visual SLAM. In *BMVC*, 2007.

[6] G. Klein and D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *ISMAR'07*, 2007.

[7] J.M.M. Montiel, J. Civera, and A. J. Davison. Unified Inverse Depth Parameterization for Monocular SLAM. In *RSS*, 2006.

[8] J. Neira and J. D. Tardos. Data Association in Stochastic Mapping Using the Joint Compatibility Test. *IEEE TRA*, 17:890–897, 2001.

[9] M. Pupilli and A. Calway. Real-Time Visual SLAM with Resilience to Erratic Motion. In *CVPR*, June 2006.

[10] E. Rosten and T. Drummond. Machine Learning for High-Speed Corner Detection. In *ECCV*, volume 1, pages 430–443, 2006.