Coarse-to-fine Planar Regularization for Dense Monocular Depth Estimation

Stephan Liwicki[1] Christopher Zach[1] Ondrej Miksik[2] Philip H. S. Torr[2]

[1]Toshiba Research Europe, Cambridge, UK [2]University of Oxford, Oxford, UK
stephan.liwicki@crl.toshiba.co.uk

Abstract. Simultaneous localization and mapping (SLAM) using the whole image data is an appealing framework to address shortcoming of sparse feature-based methods – in particular frequent failures in textureless environments. Hence, direct methods bypassing the need of feature extraction and matching became recently popular. Many of these methods operate by alternating between pose estimation and computing (semi-)dense depth maps, and are therefore not fully exploiting the advantages of joint optimization with respect to depth and pose. In this work, we propose a framework for monocular SLAM, and its local model in particular, which optimizes simultaneously over depth and pose. In addition to a planarity enforcing smoothness regularizer for the depth we also constrain the complexity of depth map updates, which provides a natural way to avoid poor local minima and reduces unknowns in the optimization. Starting from a holistic objective we develop a method suitable for online and real-time monocular SLAM. We evaluate our method quantitatively in pose and depth on the TUM dataset, and qualitatively on our own video sequences.

Keywords: SLAM, monocular odometry, dense tracking and mapping

1 Introduction

Simultaneous localization and mapping (SLAM), also known as online structure from motion, aims to produce trajectory estimations and a 3D reconstruction of the environment in real-time. In modern technology, its application ranges from autonomous driving, navigation and robotics to interactive learning, gaming and enhanced reality [1–7]. Typically, SLAM comprises two key components: (1) a local model, which generates fast initial odometry measurements (which often includes a local 3D reconstruction – *e.g.* a depth map – as byproduct), and (2) a global model, which performs loop closures and pose refinement via large scale sub-real-time bundle adjustment. In our work, we focus on the former, and propose a new strategy for local monocular odometry and depth map estimation.

Estimating the 3D position of tracked landmarks is a key ingredient in any SLAM system, since it directly allows for the poses to be computed w.r.t. a common coordinate frame. Historically, visual landmarks are induced by sparse keypoints, but there is a recent trend to utilize a dense (or semi-dense) set of points (leading to a dense or semi-dense depth map representation) [8, 9].

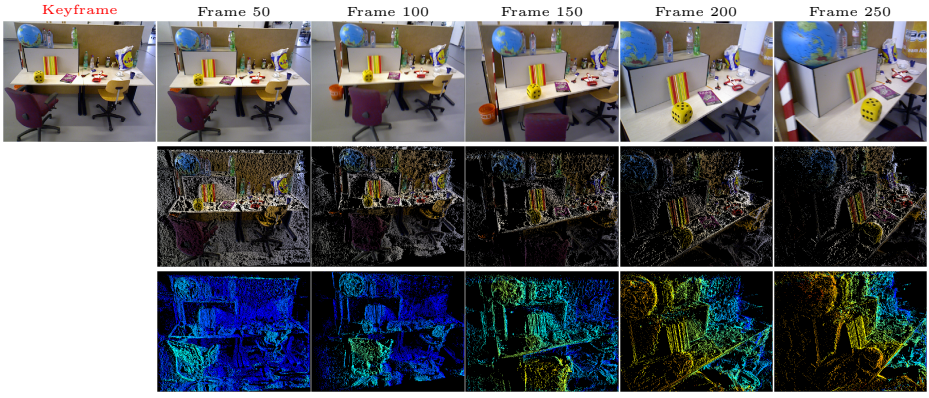


Fig. 1. During keyframe-to-frame comparison a dense depth map is build. Image, point cloud and depth (top to bottom) are shown as they develop, for selected frames from a *single* keyframe. (While depth is dense at the keyframe, their projections may not be.)

Another trend is the inclusion of different sensing modalities for depth estimation. Often, methods exploit (a combination of) alternative sensors, such as infrared, lidar and stereo camera setups, which natively provide fairly accurate depth data [10–13]. Such algorithms are quite advanced and are often employed even in consumer technology where hardware is controllable. Visual SLAM with only monocular camera streams is less common and still challenging in literature [8, 9, 14–21]. Nonetheless, the monocular setup is very suitable for (1) long range estimations, where stereo baselines are negligible, (2) light weight mobile and wearable devices aiming for a minimal amount of sensors to reduce weight and power consumption, and (3) legacy video footage recoded by a single camear.

Classical approaches for monocular visual SLAM are based on keypoint tracking and mapping [15–17], which produces a feature-based sparse depth hypothesis. A number of methods have since been proposed which essentially alternate between tracking (and pose computation) and dense depth map estimation: Most prominently, [8] presents dense tracking and mapping (DTAM) which generates a dense depth map on GPU. Similarly, [18–20] provide dense depth maps, but like [8] also rely heavily on GPU acceleration for real-time performance. In contrast to these methods large-scale direct SLAM (LSD-SLAM) [9] focusses the computation budget on a semi-dense subset of pixels and has therefore attractive running-times, even when run on CPU or mobile devices. As a direct method it computes the odometry measurements directly from image data without an intermediate representation such as feature tracks. Depth is then computed in a separate thread with small time delay. Note that all these methods employ an alternation strategy: odometry is computed with the depth map held fixed, and the depth map is updated with fixed pose estimates. In contrast, we propose joined estimation of depth and pose within a single optimization framework, which runs twice as fast as LSD-SLAM to find structure and motion. In partic-

ular, we introduce minimal additional computational cost compared to that of only the tracking thread of LSD-SLAM.

1.1 Contributions

In this work, we present a local SLAM front-end which estimates pose and depth truly simultaneously and in real-time (fig. 1). We revisit traditional setups, and propose inverse depth estimation with a coarse-to-fine planar regularizer that gradually increases the complexity of the algorithm's depth perception. Note, many systems for stereo vision or depth sensors incorporate local or global planar regularization [12, 13, 22–24]. Similarly, we employ global planar constraints into our monocular setup, and enforce local smoothness by representing each pixel as lying on a plane that is similar to its neighbours'. Furthermore, similarly to many algorithms in stereo (*e.g.* [10, 22]), we reduce depth complexity via discretization, in our case through planar splitting techniques which (in the spirit of graphical methods) create labels "on demand". In summary,

1. we formulate a global energy for planar regularized inverse depth that is optimized iteratively at each frame,
2. we revisit depth and pose optimization normally considered separately, and introduce a coarse-to-fine strategy that refines both truly simultaneously,
3. we establish our method as semi-dense, and find pose *and* depth twice as fast as LSD-SLAM, by adding minimal cost to LSD-SLAM's tracking thread,
4. we evaluate pose and depth quantitatively on the TUM dataset.

Closely related to our work is [25], where depth and pose is optimized simultaneously given the optical flow of two consecutive images. This approach is based on image pairs. Our method considers video input and incrementally improves its belief. In [26, 27] planarity is proposed in conjunction with scene priors, previously learned from data, and [20] presents a hole-filling strategy for semi-dense monocular SLAM. While these methods are real-time, they rely on keypoints at image corners or gradients, which are later enriched with a planar refinement. Importantly however, such methods fail in featureless environments. Finally, we emphasis DTAM [8] performs batch operations on a set of images taken from a narrow field of view, and henceforth introduces a fixed lag before depth is perceived by the system. As this is often unacceptable for robotics setups, our method updates depth incrementally after *each* frame.

2 Proposed Energy for Monocular Depth Estimation

We formulate our energy function for poses and depth w.r.t. the photometric error over time. Similar to LSD-SLAM, we employ a keyframe-to-frame comparison to estimate camera displacement and each pixels' depth in the reference image. Let us denote the keyframe as I and its immediately succeeding images as $(I_t)_{t=1}^{T}$. The tuple of valid pixel locations on the keyframe's plane is represented by $\mathcal{X} = (\mathbf{x}_i)_{i=1}^{|\mathcal{X}|}$ in *normalized* homogeneous coordinates (*i.e.* $z_i = 1$),

and their corresponding *inverse* depth values are expressed by $\mathcal{D} = (d_i)_{i=1}^{|\mathcal{X}|}$. Since we aim to model planar surfaces, we use an over-parametrization given by $\mathcal{S} = (\mathbf{s}_i^\intercal)_{i=1}^{|\mathcal{X}|} \cong \mathbb{R}^{3|\mathcal{X}|}$, where $\mathbf{s}_i = (u_i, v_i, w_i)^\intercal$ are planes with disparity gradients u_i, v_i, and inverse depth at 0, w_i. Hence, the relation $d_i = \mathbf{s}_i^\intercal \mathbf{x}_i$ holds.

Tuple $\Xi = (\xi_t)_{t=1}^T$ denotes the changes in camera pose, where $\xi_t \in SE(3)$ is composed of rotation $\mathbf{R}_t \in SO(3) \subset \mathbb{R}^{3\times 3}$ and translation $\mathbf{t}_t \in \mathbb{R}^3$ between the keyframe I and frame I_t. In principle, the complete cost function should incorporate all available images associated with the current keyframe and optimize over the depth and all poses jointly,

$$\hat{E}_{Total}\left(\mathcal{S}, \Xi\right) = \sum_{t=1}^{T} E_{Match}^{(t)}(\mathcal{S}, \xi_t) + E_{Smooth}(\mathcal{S}). \tag{1}$$

Here $E_{Match}^{(t)}$ and E_{Smooth} are energy terms related to image-based matching costs and spatial smoothing assumptions, respectively. Before we describe these terms in more detail in subsequent sections, we modify $\hat{E}_{Total}$ to be more suitable for an incremental online approach. This is advisable since, the objective $\hat{E}_{Total}$ involves the complete history of all frames I_t mapped to the current keyframe I. Intuitively the optimization of the poses $(\xi_t)_{t=1}^{T-1}$ is no longer relevant at time T, as only the current pose ξ_T and $\mathcal{S}$ is required. Analytically, we introduce

$$E_{History}^{(T)}\left(\mathcal{S}\right) := \min_{(\xi_t)_{t=1}^{T-1}} \sum_{t=1}^{T-1} E_{Match}^{(t)}(\mathcal{S}, \xi_t) \tag{2}$$

where $(\xi_t)_{t=1}^{T-1}$ is the tuple of poses, minimized in previous frames. By splitting the first term in (1), the energy becomes

$$\hat{E}_{Total}\left(\mathcal{S}, \Xi\right) = E_{History}^{(T)}\left(\mathcal{S}\right) + E_{Match}^{(T)}(\mathcal{S}, \xi_T) + E_{Smooth}(\mathcal{S}). \tag{3}$$

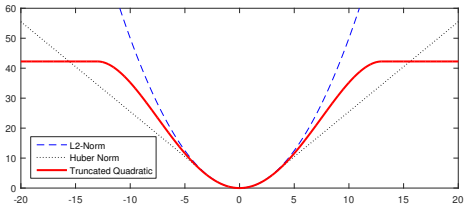
Now we replace $E_{History}^{(T)}$ with its second order expansion around

$$\left(\mathcal{S}^*, \xi_1^*, \ldots, \xi_{T-1}^*\right) = \operatorname*{argmin}_{\mathcal{S}, (\xi_t)_{t=1}^{T-1}} \sum_{t=1}^{T-1} E_{Match}^{(t)}(\mathcal{S}, \xi_t), \tag{4}$$

and thus we obtain an approximation of $E_{History}^{(T)}(\mathcal{S})$, denoted $E_{Temporal}^{(T)}(\mathcal{S})$:

$$\begin{aligned} E_{Temporal}^{(T)}\left(\mathcal{S}\right) &:= E_{History}^{(T)}\left(\mathcal{S}^*\right) + \left(\nabla_{\mathcal{S}} E_{History}^{(T)}\left(\mathcal{S}^*\right)\right)^\intercal \left(\mathcal{S} - \mathcal{S}^*\right) \\ &\quad + \frac{1}{2}(\mathcal{S} - \mathcal{S}^*)^\intercal \left(\nabla_{\mathcal{S}}^2 E_{History}^{(T)}\left(\mathcal{S}^*\right)\right)\left(\mathcal{S} - \mathcal{S}^*\right) \\ &= E_{History}^{(T)}\left(\mathcal{S}^*\right) + \frac{1}{2}(\mathcal{S} - \mathcal{S}^*)^\intercal \left(\nabla_{\mathcal{S}}^2 E_{History}^{(T)}\left(\mathcal{S}^*\right)\right)\left(\mathcal{S} - \mathcal{S}^*\right) \end{aligned} \tag{5}$$

As $\mathcal{S}^*$ is a local minimizer of $E_{History}^{(T)}$, $\nabla_{\mathcal{S}} E_{History}^{(T)}(\mathcal{S}^*) = 0$. Furthermore, as our choice of terms leads to a nonlinear least-squares formulation, $\nabla_{\mathcal{S}}^2 E_{History}^{(T)}(\mathcal{S}^*)$ is



$$\|r\|_\tau = \begin{cases} \frac{1}{2}\|r\|^2 \left(1 - \frac{\|r\|^2}{2\tau^2}\right) & \text{if } \|r\|^2 \leq \tau^2 \\ \frac{\tau^2}{4} & \text{otherwise} \end{cases}$$

Fig. 2. The smooth truncated quadratic compared to the squared L_2-norm and Huber cost (left), and the smooth truncated quadratic's mathematical representation (right).

computed using the Gauss-Newton approximation. Finally, since $E^{(T)}_{History}$ jointly optimizes the inverse depths (in terms of its over-parametrization $\mathcal{S}$) and (internally) the poses, but $E^{(T)}_{Temporal}$ is solely a function of $\mathcal{S}$, we employ the Schur complement to factor out the poses $(\xi_t)_{t=1}^{T-1}$. However, as the poses link the entire depth map, the Schur complement matrix will be dense. We obtain a tractable approximation by using its block-diagonal consisting of 3×3 blocks (corresponding to $\mathbf{s}_i = (u_i, v_i, w_i)^\intercal$).[1] The resulting objective at time T is therefore

$$E^{(T)}_{Total}(\mathcal{S}, \xi_T) = E^{(T)}_{Temporal}(\mathcal{S}) + E^{(T)}_{Match}(\mathcal{S}, \xi_T) + E_{Smooth}(\mathcal{S}). \tag{6}$$

There is a clear connection between $E^{(T)}_{Total}$, extended Kalman filtering and maximum likelihood estimation. If $E^{(T)}_{History}$ is interpreted as log-likelihood, then $\left(\mathcal{S}^*, (\xi^*_t)_{t=1}^{T-1}\right)$ is an asymptotically normal maximum likelihood estimate with the Hessian as (approximate) inverse covariance (*i.e.* precision) matrix. The Schur complement to factor out the poses (in the energy-minimization perspective) corresponds to marginalizing over the poses according to their uncertainty. $E^{(T)}_{Total}$ can be read as probabilistic fusion of past and current observation, but this correspondence is limited, since we are searching for MAP estimates and not posteriors. In the following section we discuss the remaining terms in $E^{(T)}_{Total}$.

2.1 Photometric Energy

The matching cost $E^{(T)}_{Match}(\mathcal{S}, \xi_T)$ is derived from an appearance (*e.g.* brightness) consistency assumption commonly employed in literature, *e.g.* [28]. Let us define the monocular warping function $W(\mathbf{x}_i, d_i, \xi_t)$ which maps point $\mathbf{x}_i$ in the keyframe to its representation $\mathbf{x}'_i$ in frame t by

$$\mathbf{x}'_i = W(\mathbf{x}_i, d_i, \xi_t) = \hom\left(\mathbf{R}_t^\intercal \left(\mathbf{x}_i - \mathbf{t}_t d_i\right)\right), \tag{7}$$

under camera rotation $\mathbf{R}_t$ and translation $\mathbf{t}_t$, where $\hom(\cdot)$ normalizes the homogeneous coordinate. Now we express the matching energy as

$$E^{(T)}_{Match}(\mathcal{S}, \xi_T) = \sum_{\mathbf{x}_i \in \mathcal{X}} \|I(\mathbf{x}_i) - I_T\left(W(\mathbf{x}_i, d_i, \xi_T)\right)\|_{\tau_{Match}}, \tag{8}$$

[1] The block-diagonal is an overconfident approximation of the precision. As compensation, we employ a forgetting factor $\lambda_{Temporal}$ in our implementation (see §3.2).

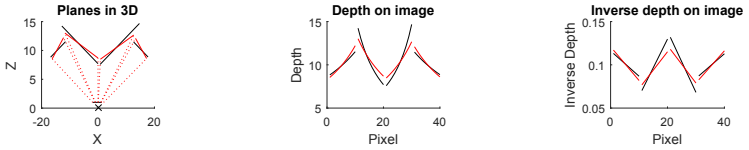


Fig. 3. Planes in 3D space are aligned via smoothing in the inverse depth image (black represent original planes, red represents the smoothed versions).

where $I(\mathbf{x})$ and $I_T(\mathbf{x})$ are descriptors extracted around pixel $\mathbf{x}$ from keyframe and current frame respectively. We use image intensity values (*i.e.* a descriptor at pixel only), so that the disparity gradients do not need to be taken into account during warping. Robustness is achieved by employing a smooth truncated quadratic error [29] (visualized in fig. 2) in the implementation of $\|\cdot\|_{\tau_{Match}}$.

2.2 Local Spatial Plane Regularizer

The smoothness constraint $E_{Smooth}(\mathcal{S})$ is based on a planar assumption often found in stereo setups [13, 23, 24], which we adapt in this work to support monocular video data. Surface $\mathbf{s}_i$ induces a linear extrapolation of inverse depth via $\hat{d}_i(\mathbf{x}) = \mathbf{s}_i^{\mathsf{T}}\mathbf{x}$. Plugging this into the homographic transformation yields

$$W(\mathbf{x}, \hat{d}_i(\mathbf{x}), \xi_t) = \hom\left(\mathbf{R}_t^{\mathsf{T}}\left(\mathbf{x} - \mathbf{t}_t\mathbf{s}_i^{\mathsf{T}}\mathbf{x}\right)\right) = \hom\left(\mathbf{R}_t^{\mathsf{T}}\left(\mathbf{x}_i - \mathbf{t}_t\frac{\mathbf{n}_i^{\mathsf{T}}}{r_i}\mathbf{x}_i\right)\right), \quad (9)$$

where $\mathbf{n}_i$ is the plane normal and r_i is the point-plane distance to the camera center. Hence we can identify $\mathbf{s}_i \propto \mathbf{n}_i$ and therefore smoothing planes in inverse depth parametrization also smoothes the alignment in 3D space (fig. 3).

With λ_{Smooth} as balancing term, we define the spatial smoothness energy as

$$\begin{aligned} E_{Smooth}(\mathcal{S}) &= \lambda_{Smooth}\sum_{\mathbf{x}_i\in\mathcal{X}}\sum_{\mathbf{x}_j\in\mathcal{N}_i}\|\mathbf{s}_i^{\mathsf{T}}\mathbf{x}_i - \mathbf{s}_j^{\mathsf{T}}\mathbf{x}_i\|_{\tau_{Smooth}} \\ &= \lambda_{Smooth}\sum_{\mathbf{x}_i\in\mathcal{X}}\sum_{\mathbf{x}_j\in\mathcal{N}_i}\|d_i - (d_j + \mathbf{s}_j^{\mathsf{T}}(\mathbf{x}_i - \mathbf{x}_j))\|_{\tau_{Smooth}}, \end{aligned} \quad (10)$$

where $\mathcal{N}_i$ denotes the 4-neighborhood of $\mathbf{x}_i$. Thus, E_{Smooth} penalizes deviations between linearly extrapolated depth at $\mathbf{x}_j$ and its actual depth. Although some methods try to introduce robustness by appearance-based edge detection, *e.g.* [30], we again simply employ the smooth version of the truncated quadratic for $\|\cdot\|_{\tau_{Smooth}}$. Hence, our method is inherently robust without arbitrary color constraints. Unfortunately, (10) is not scale invariant, and scaling the baseline $\mathbf{t}_t$ scales the contribution of E_{Smooth}. This is a potential issue only for the first pair of frames (I, I_1), since subsequent frames have their scale determined by preceding frames. It is common usage to fix the initial scale by setting $\|\mathbf{t}_1\| = 1$, but this is a suboptimal choice, since the same 3D scene geometry is regularized differently depending on the initial baseline. A more sensible choice is to fix *e.g.* the average depth (or inverse depth) to make E_{Smooth} invariant w.r.t. baselines. For our reconstruction we constrain the average inverse depth to one.

Algorithm 1 Dense Incremental Planar Depth Estimation

Input: Keyframe I and images $(I_t)_{t=1}^{T}$.
Output: Final pose ξ and depth hypothesis $\mathcal{S}$.

$\mathbf{s}_i \leftarrow [0\ 0\ 1]^{\mathrm{T}}$ and $\Lambda_i \leftarrow \mathbf{0}$ for all $\mathbf{x}_i \in \mathcal{X}$.
compute resolution pyramid for the keyframe I.
$\xi \leftarrow (\mathbf{I} \in \mathbb{R}^{3\times 3}, [0\ 0\ 0]^{\mathrm{T}})$
for each frame I_t **do**
  compute resolution pyramid for the frame I_t.
  for each pyramid level **do**
    optimize ξ via lie algebra $\mathfrak{se}(3)$ through Levenberg-Marquardt.
    repeat
      update ξ (and $\mathbf{s}_i \leftarrow \mathbf{s}_i + \mathbb{I}_c(\mathbf{x}_i)\Delta_c$ if applicable).
      introduce new component Δ_c.
      estimate $\mathbb{I}_c(\mathbf{x}_i)$ via eigenvector of $\sum_{\mathbf{x}_i \in \mathcal{X}} \nabla_{\mathbf{s}_i} \nabla_{\mathbf{s}_i}^{\mathrm{T}}$.
      optimize ξ and Δ_c through Levenberg-Marquardt.
    until improvement below $\epsilon_{Complex}$ or maximum components reached
  end for
  update precision Λ_i and depth $\mathbf{s}_i^*$ for temporal constraint.
end for

3 Optimization Strategy

In this section we detail our optimization strategy for the energy in (6). We assume small changes between consecutive frames, as video data is used. Therefore we use a similar approach as in standard differential tracking and optical flow by locally linearizing the image intensities I_T in the matching term $E_{Match}^{(T)}$. The pseudocode of the proposed method is given in alg. 1. The underlying idea is to optimize the energy incrementally with increased complexity using the scale-space pyramid representation and our restricted depth map update which we detail below. The aim of doing this is two-fold: Firstly it substantially reduces the number of unknowns in the main objective and therefore makes the optimization much more efficient, and secondly it provides an additional level of regularization within the algorithm and combines naturally with a scale-space framework to avoid poor local minima. We discuss this constrained depth map update in the following, and then introduce our optimization which exploits this update to allow for truly simultaneous pose and depth estimation. Finally we present a strategy for realtime performance on CPU.

3.1 Constrained Depth Map Updates

If we consider the current frame at time T and optimize E_{Total} (recall (6)) w.r.t. ξ_T and $\mathcal{S}$, then our algorithmic design choice is to restrict the update $\mathcal{S} - \mathcal{S}^*$ to have low complexity in the following sense:

$$\mathbf{s}_i = \mathbf{s}_i^* + \sum_{c=1}^{C} \mathbb{I}_c(\mathbf{x}_i)\Delta_c, \tag{11}$$

where $\mathbb{I}_c : \mathcal{X} \rightarrow \{+1, -1\}$ is an indicator function, splitting the set of pixels into positive or negative parts. This means that a depth update at each pixel $\mathbf{x}_i$ is constrained to take one of 2^C values. With increasing cardinality C, the complexity of the depth map increases.

The optimization is performed greedily by adding a single component Δ_c at a time. Notice, if ξ_T and $\mathcal{S}$ were to be optimized simultaneously, an equation with $6 + 3|\mathcal{X}|$ unknowns had to be solved inside a nonlinear least squares solver (*i.e.* 6 parameters for an element in the lie algebra $\mathfrak{se}(3)$ and 3 for the over-parameterized depth values at each pixel). By using the constrained shape for the updates and by using a greedy framework, we reduce the optimization to $6+3$ variables at a time (*i.e.* $\mathfrak{se}(3)$ and the 3 vector Δ_c), improving the execution cost and robustness significantly.

Our methodology can be seen in analogy to multi-resolution pyramids which spatially increase the quantization of the image plane, but in addition to spatial resolution we also incrementally increase the quantization level of inverse depths. Specifically, we exploit the representation of a pixel's plane $\mathbf{s}_i$ as summed components Δ_c, given in (11). These values correspond to the inverse depth resolution which increases when new components are introduced.

This coarse-to-fine depth estimation is inspired by the human vision [31], which perceives depth in relation to other areas in the scene, rather than absolute values. Specifically, we perform the introduction of new distance values in a relational setting, splitting the data points based on their desired depth value direction. The advantages of this approach are three-fold: (1) we introduce depth by enforcing a regularization across all pixels, (2) our splitting function separates the image data into multiple planes, which naturally encode the image hierarchically from coarse to fine, and (3) the incremental introduction of depth enables fast computation whilst optimizing transformation and depth simultaneously. Moreover, we emphasize while our approach is greedy, it is not final since corrections can be made through further splitting.

Our design choice to regularize the updates of $\mathcal{S}$ requires to determine the binary function $\mathbb{I}_c : \mathcal{X} \rightarrow \{+1, -1\}$. Essentially, if Δ_c is given, $\mathbb{I}_c(\mathbf{x}_i)$ corresponds to the sign of the correlation $\Delta_c^{\mathsf{T}} \nabla_{\mathbf{s}_i} E_{Total}$ between the depth update direction Δ_c and the gradient of the objective with respect to $\mathbf{s}_i$. Since Δ_c is subject to subsequent optimization, we determine an initial estimate $\tilde{\Delta}_c$ as follows: given the current gradients $\nabla_{\mathbf{s}_i} E_{Total}$ (which we abbreviate to $\nabla_{\mathbf{s}_i}$), it is sensible to obtain $\tilde{\Delta}_c$ as principal direction of the set $\{\nabla_{\mathbf{s}_i}\}_{i=1}^{|\mathcal{X}|}$, due to the symmetric range in $\mathbb{I}_c$:

$$\tilde{\Delta}_c \leftarrow \underset{u:\|u\|=1}{\operatorname{argmax}} \left\{ u^{\mathsf{T}} \sum\nolimits_{\mathbf{x}_i \in \mathcal{X}} \nabla_{\mathbf{s}_i} \nabla_{\mathbf{s}_i}^{\mathsf{T}} \, u \right\}. \tag{12}$$

This can be obtained by eigenvalue or singular value decomposition of the 3×3 scatter matrix $\sum_{\mathbf{x}_i \in \mathcal{X}} \nabla_{\mathbf{s}_i} \nabla_{\mathbf{s}_i}^{\mathsf{T}}$. Finally, the indicator function is given by

$$\mathbb{I}_c(\mathbf{x}_i) = \begin{cases} 1 & \text{if } \tilde{\Delta}_c^{\mathsf{T}} \nabla_{\mathbf{s}_i} \geq 0 \\ -1 & \text{otherwise} \end{cases} = \operatorname{sgn}\left(\tilde{\Delta}_c^{\mathsf{T}} \nabla_{\mathbf{s}_i} \right). \tag{13}$$

3.2 Simultaneous Pose and Depth Estimation

Let us assume we have an initial estimate for ξ_T and $\mathcal{S}$ available (*e.g.* $\xi_T \leftarrow \xi_{T-1}$ and $\mathcal{S} \leftarrow \mathcal{S}^*$, which is equivalent to $C = 0$ in (11)). Since our objective is an instance of nonlinear least-squares problems we utilize the Levenberg-Marquardt (LM) algorithm for robust and fast second order minimization. The robust kernels $\|\cdot\|_{\tau_{Match}}$ and $\|\cdot\|_{\tau_{Smooth}}$ are handled by an iteratively reweighed least square (IRLS) strategy. Potentially enlarging the convergence basin via a lifted representation of the robust kernel [32] is a topic for future work.

As outlined in §3.1 the complexity of depth map updates is increased greedily, which means that new components Δ_c are successively introduced. We start with $C = 0$ and iteratively increase C by adding new components. After introduction of a new component Δ_c (and having an estimate for $\mathbb{I}_c$), minimizing E_{Total} with respect to Δ_c and ξ_T amounts to solving

$$\begin{aligned}\underset{\xi_T,\Delta_c}{\operatorname{argmin}}\Big\{ &\sum_{\mathbf{x}_i\in\mathcal{X}} \|I(\mathbf{x}_i) - I_T\left(W(\mathbf{x}_i, (\mathbf{s}_i + \mathbb{I}_c(\mathbf{x}_i)\Delta_c)^\intercal \mathbf{x}_i, \xi_T)\right)\|_{\tau_{Match}} \\ &+ \lambda_{Smooth} \sum_{\mathbf{x}_i\in\mathcal{X}} \sum_{\mathbf{x}_j\in\mathcal{N}_i} \|\left(\mathbf{s}_i + \mathbb{I}_c(\mathbf{x}_i)\Delta_c\right)^\intercal \mathbf{x}_i - \left(\mathbf{s}_j + \mathbb{I}_c(\mathbf{x}_j)\Delta_c\right)^\intercal \mathbf{x}_i\|_{\tau_{Smooth}} \\ &+ \sum_{\mathbf{x}_i\in\mathcal{X}} \|\mathbf{s}_i^* - (\mathbf{s}_i + \mathbb{I}_c(\mathbf{x}_i)\Delta_c)\|_{\Lambda_i}\Big\} \end{aligned} \tag{14}$$

(via LM), followed by the update $\mathbf{s}_i \leftarrow \mathbf{s}_i + \mathbb{I}_c(\mathbf{x}_i)\Delta_c$. We emphasize, as Δ_c is shared between all pixels, this problem is unlikely to be rank deficient. Further components Δ_c are introduced as long as E_{Total} is reduced sufficiently (*i.e.* an improvement larger than $\epsilon_{Complex}$). Notice, while our algorithm iteratively introduces new components Δ_c, it optimizes pose and depth simultaneously. Analogous to the resolution-based scale-space pyramid, the indicator function acts as surrogate for increased resolution in depth,

For the first frame I_1 matched with the keyframe I we need to enforce that the average inverse depth is 1 (recall Section 2.2), which implies that

$$\sum_{\mathbf{x}_i} \left(\mathbf{s}_i + \mathbb{I}_c(\mathbf{x}_i)\Delta_c\right)^\intercal \mathbf{x}_i = \sum_{\mathbf{x}_i} \left(d_i + \mathbb{I}_c(\mathbf{x}_i)\Delta_c^\intercal \mathbf{x}_i\right) = 1 \tag{15}$$

must hold. If d_i already satisfies $\sum_{\mathbf{x}_i} d_i = 1$, then the above reduces to

$$\sum_{\mathbf{x}_i} \mathbb{I}_c(\mathbf{x}_i)\mathbf{x}_i^\intercal \Delta_c = 0. \tag{16}$$

We chose a projected gradient approach by projecting the gradient w.r.t. Δ_c to the feasible subspace defined by (16) inside the LM optimizer. Note that the planes are initialized to $\mathbf{s}_i = (0, 0, 1)^\intercal$ in the beginning of the algorithm, and by induction $\sum_{\mathbf{x}_i} \mathbf{s}_i^\intercal \mathbf{s}_i = \sum_{\mathbf{x}_i} d_i = 1$ is always satisfied for the first frame. In subsequent frames the constraint in (16) is not active.

Finally, to determine the precision matrices $\Lambda_i \in \mathbb{R}^{3\times3}$ needed for $E^{(T+1)}_{Temporal}$, we employ the approximate Hessian via the Jacobian $\mathbf{J}_{Match}$ of $E^{(T)}_{Match}$:

$$\begin{pmatrix} \tilde{H}_{\mathcal{S},\mathcal{S}} & \tilde{H}^{\mathsf{T}}_{\mathcal{S},\xi_T} \\ \tilde{H}_{\mathcal{S},\xi_T} & \tilde{H}_{\xi_T,\xi_T} \end{pmatrix} := \mathbf{J}^{\mathsf{T}}_{Match}\mathbf{J}_{Match}, \tag{17}$$

and the 3×3-diagonal block of the Schur complement $\tilde{H}_{\mathcal{S},\mathcal{S}} - \tilde{H}^{\mathsf{T}}_{\mathcal{S},\xi_T}\tilde{H}^{-1}_{\xi_T,\xi_T}\tilde{H}_{\mathcal{S},\xi_T}$ (denoted Λ_{Match}). We employ a forgetting factor $\lambda_{Temporal}$ to reduce the over-confident precision matrix, and update $\Lambda_i \leftarrow \lambda_{Temporal}\Lambda_i + \Lambda_{Match}$. Recall that $\tilde{H}_{\xi_T,\xi_T} \in \mathbb{R}^{6\times6}$ and $\tilde{H}_{\mathcal{S},\xi_T}$ are very sparse.

3.3 CPU Computation in Realtime

Thus far, we present our energy for each pixel in the input video stream. While this is generally useful for dense depth estimation, we may adopt our approach to semi-dense computation to reduce running time. Similar to LSD-SLAM, we can represent the image by its significant gradient values. By only computing on these gradients, execution is significantly reduced. In fact, in comparison to LSD-SLAM, we only need one additional LM iteration per split to introduce depth on top of pose estimation. Finally, we can limit the number of introduced depth components per resolution level to achieve constant running time.

4 Results

We perform our experiments on 13 video sequences in total, using 6 TUM [33] image streams and 7 sequences recoded ourselves. The TUM dataset comprises a number of video sequences with groundtruth pose, as recorded by a Vicon system, and approximate depth through depth sensors [33]. We select a subset of the handheld SLAM videos to measure system performance (*i.e.* fr1-desk, fr1-desk2, fr1-floor, fr1-room, fr2-xyz and fr3-office). As we are interested in the local aspect of SLAM (operating with single keyframe), we further divide these into smaller sequences. Notice, as we perform keyframe-to-frame comparison, the videos need to contain enough overlap with the reference image. Additionally, we record 7 videos, using a GoPro Hero 3 with a wide angle lens at 30 fps.

As a monocular approach, our method does not fix the scale. Hence, we employ a scale corrected error (SCE) for translation:

$$e(\mathbf{t}_t, \hat{\mathbf{t}}_t) = \left\| \mathbf{t}_t \frac{\|\hat{\mathbf{t}}_t\|}{\|\mathbf{t}_t\|} - \hat{\mathbf{t}} \right\|, \tag{18}$$

where $\mathbf{t}_t$ is the translational displacement of the pose ξ_t, and $\hat{\mathbf{t}}_t$ is the groundtruth with respect to the keyframe (or initial frame). An error in rotation is indirectly captured, as it effects the translation of future frames. We now introduce a scale invariant measure to evaluate the depth's completeness. Given true inverse depth at the keyframe $\hat{\mathcal{D}} = (\hat{d}_i)_{i=1}^{|\mathcal{X}|}$ we define the completeness as the proportion

Table 1. Median Scale Corrected Error (in mm) for the compared methods after the listed frame number for different TUM-Dataset sequences. (Note, different characteristics of camera motion in each video lead to different length of keyframe overlaps.)

		LSD-SLAM	LSD-Key	Disjoint	SIP	DIP
fr1-desk	frame 5	34	34	33	**25**	27
	frame 10	44	62	55	43	**30**
	frame 30	106	130	119	135	**46**
fr1-desk2	frame 5	68	68	53	23	**18**
	frame 10	103	115	87	**41**	44
	frame 20	207	-	162	163	**64**
fr1-floor	frame 5	30	30	36	**30**	34
	frame 10	**55**	58	76	58	60
	frame 15	85	88	111	**79**	86
fr1-room	frame 5	13	13	19	**10**	16
	frame 10	40	40	52	**39**	42
	frame 25	**9**	79	117	-	53
fr2-xyz	frame 10	15	15	10	**9**	**9**
	frame 30	54	68	28	**18**	23
	frame 100	121	88	**45**	**45**	47
fr3-office	frame 10	**29**	30	41	32	33
	frame 50	90	121	182	**53**	100
	frame 150	206	-	265	-	**123**

of depth values, satisfying a given accuracy ϵ:

$$c\left(\hat{\mathcal{D}}, \mathcal{D}\right) = \max_{\alpha} \sum_{i}^{|\mathcal{X}|} \frac{n_\alpha(\hat{d}_i, d_i)}{|\mathcal{X}|}, \text{where } n_\alpha(\hat{d}_i, d_i) = \begin{cases} 1 \text{ if } \|\frac{1}{\hat{d}_i} - \frac{\alpha}{d_i}\| < \epsilon \\ 0 \text{ otherwise} \end{cases}. \quad (19)$$

Parameter α represents scale and is found via grid search and refined through gradient decent. In our work, $\epsilon = 0.05$ which corresponds to $\pm$5cm.

4.1 Quantitative Evaluation on the TUM Dataset

We compare the proposed dense and semi-dense incremental planar system (DIP and SIP respectively) to two versions of LSD-SLAM: (1) we carefully implement a LSD-SLAM version that only uses a single keyframe (LSD-Key), and (2) the original LSD-SLAM as provided by authors of [9], without loop closures or other constraints (LSD-SLAM). We further ensure that mapping is guaranteed to run after every tracking step in both LSD-SLAM systems. Finally, we include our method as disjoint optimization for pose and depth separately and sequentially. Table 1 shows the median SCE for different numbers of frames. The median is calculated over all snippets taken from the individual TUM sequences.

The sequences fr1-desk and fr1-desk2 show an office environment with high camera motion and little overlap towards keyframes. Here, the trajectories are quickly lost when a single keyframe is used. SIP performs best at early stages,

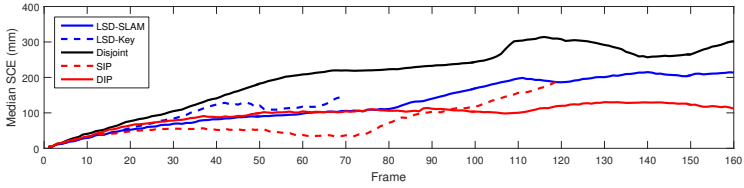


Fig. 4. Median SCE for videos of fr3-office. LSD-SLAM and DIP track long-term, while SIP is more accurate early on. LSD-Key loses track quickly, and the disjoint optimization (Disjoint) is consistently worse.

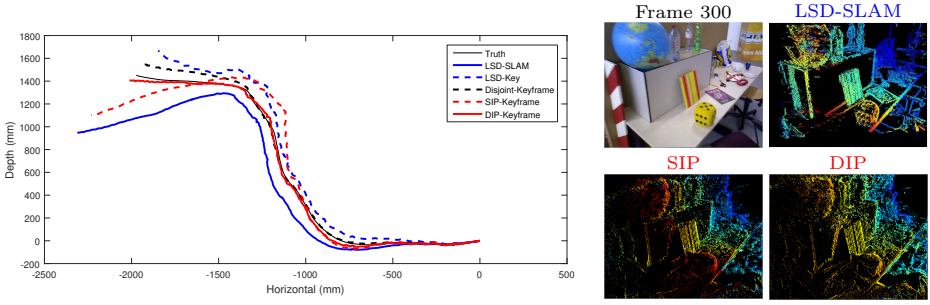


Fig. 5. Trajectories (left) and inverse depth maps (right) of LSD-SLAM, SIP and DIP for the initial 300 images in fr3-office. LSD-SLAM is inaccurate due to scale drift. DIP uses a single keyframe and hence does not drift as significantly. For depth, SIP and DIP benefit from larger keyframe-to-frame baseline, resulting in qualitative better depth.

while DIP is more suitable for longer tracking. The sequences fr1-floor and fr1-room also have little keyframe overlap, but with slower motion. Here LSD-SLAM performs competitively, as it benefits from keyframe generation.

Long-term tracks are achieved in fr2-xyz and fr3-office. We take a more detailed look at the results of fr3-office. Fig. 4 plots the median SCE for each duration. We see that LSD-SLAM and DIP have similar performance early on, but DIP performs better at later stages. Notice, as LSD-SLAM generates new reference images, the baseline is typically small. In contrast DIP benefits from larger baselines. LSD-Key loses track quickly, while SIP performs well in early stages. The trajectory and inverse depth maps for the very first 300 frames are shown in fig. 5. Fig. 6 plots the depth completeness. Here, DIP and SIP reach a peak correctness with increasing baseline, after which they slightly degrades as points are outside the current view, and smoothing takes over their energies.

We remark, similar to many approaches based on gradient decent, our method converges to local minima. However our method relies on graduated optimization which aims to avoid getting trapped in bad minima by optimizing a smoother energy with gradually increased complexity [34]. In contrast to LSD-SLAM, we employ graduated optimization in depth perception as well as traditional scale-space image pyramids leading to superior results. The indicator function is a

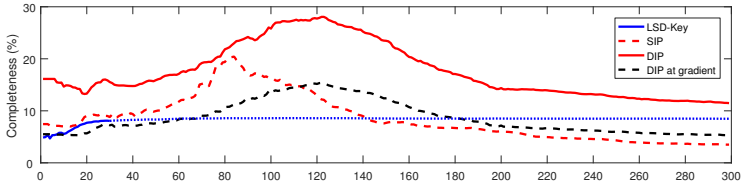


Fig. 6. Depth completeness of LSD-Key, SIP and DIP for initial images in fr3-office. As LSD-Key and SIP only produces depth for high gradient pixels, the results of DIP at gradient only are also shown. Note, LSD-Key remains unchanged after poor tracking.

surrogate for the scale-space pyramid in depth. Finally, we note that the disjoint version is consistently worse in virtually all experiments. The difference is the impact of graduated optimization. For Disjoint, changes in perceived depth are not utilized for pose at the current frame. In contrast, joint optimization finds pose and depth at the same time, yielding improved performance.

In terms of running time, LSD-SLAM and LSD-Key perform tracking and mapping at 14 fps, while SIP performs twice as fast at 30 fps on CPU. DIP is slower on CPU (2 fps), but its GPU implementation runs in realtime (30 fps).

4.2 Qualitative Results

We conclude the experimental with example frames of our 7 additional video sequences (fig. 7). Generally, LSD-SLAM smoothes well in the local neighborhood, while SIP and DIP perform more consistent on the global inverse depth hypothesis. We note, even with non-planar scenes our methods performs well. We argue, that the local planar surface assumption is reasonable in most environments, as was also witnessed by recent stereo systems, *e.g.* [13, 23, 24]. Nonetheless, in non-urban scenes, and in situations where the initial frontal plane assumption is significantly wrong (recall initialization of $\mathbf{s}_i = (0, 0, 1)^\intercal$), the results are less favorable as seen in the last row of fig. 7.

5 Conclusion

We introduced a carefully derived coarse-to-fine planar regularization strategy that optimizes for both, pose and depth simultaneously from monocular streams. Our framework is keyframe-based, and incrementally improves its depth hypothesis at each frame as new data arrives. As semi-dense approach, the proposed method runs in realtime on CPU, while realtime for the dense version can be achieved on GPU. In our evaluation, we improved upon the front-end of LSD-SLAM whilst increasing execution time by a factor of two.

Acknowledgment. O. Miksik is supported by Technicolor. P. Torr wishes to acknowledges the support of ERC grant ERC-2012-AdG 321162-HELIOS, EPSRC/MURI grant ref EP/N019474/1,EPSRC grant EP/M013774/1, EPSRC Programme Grant Seebibyte EP/M013774/1.

Fig. 7. Inverse depth of LSD-SLAM, SIP and DIP for 7 qualitative video sequences (far is blue, near is red). In most scenes, the local planar surface assumption holds and our method performs well. In non-urban environments and where the initialization with frontal planar surfaces does not hold, our method fails (bottom row).

References

1. Barfield, W.: Fundamentals of Wearable Computers and Augmented Reality, Second Edition. CRC Press (2016)
2. Engel, J., Sturm, J., Cremers, D.: Scale-Aware Navigation of a Low-Cost Quadrocopter with a Monocular Camera. **62**(11) (Nov 2014) 1646–1656
3. Forster, C., Pizzoli, M., Scaramuzza, D.: SVO: Fast Semi-Direct Monocular Visual Odometry. In: ICRA'14. (2014) 15 – 22
4. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: CVPR'12. (2012) 3354 – 3361
5. Miksik, O., Vineet, V., Lidegaard, M., Prasaath, R., Nießner, M., Golodetz, S., Hicks, S., Pérez, P., Izadi, S., Torr, P.: The Semantic Paintbrush: Interactive 3D Mapping and Recognition in Large Outdoor Spaces. In: ACM Conf. Human Factors in Computing, CHI'15. (2015) 3317 – 3326
6. Vineet, V., Miksik, O., Lidegaard, M., Nießner, M., Golodetz, S., Prisacariu, V., Kähler, O., Murray, D., Izadi, S., Pérez, P., Torr, P.: Incremental Dense Semantic Stereo Fusion for Large-Scale Semantic Scene Reconstruction. In: ICRA'15. (2015)
7. Schöps, T., Engel, J., Cremers, D.: Semi-Dense Visual Odometry for AR on a Smartphone. In: ISMAR'14. (2014) 145 – 150
8. Newcombe, R., Lovegrove, S., Davison, A.: DTAM: Dense tracking and mapping in real-time. In: IEEE Int. Conf. Computer Vision, ICCV'11. (2011) 2320 – 2327
9. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: Large-Scale Direct Monocular SLAM. In: ECCV'14. (2014) 834 – 849
10. Miksik, O., Amar, Y., Vineet, V., Pérez, P., Torr, P.: Incremental Dense Multi-Modal 3D Scene Reconstruction. In: IROS'15. (2015)
11. Newcombe, R., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.: KinectFusion: Real-Time Dense Surface Mapping and Tracking. In: ISMAR'11. (2011) 127–136
12. Salas-Moreno, R., Glocker, B., Kelly, P., Davison, A.: Dense Planar SLAM. In: ISMAR'14. (2014) 157 – 164
13. Yamaguchi, K., McAllester, D., Urtasun, R.: Efficient Joint Segmentation, Occlusion Labeling, Stereo and Flow Estimation. In: ECCV'14. (2014) 756 – 771
14. Nister, D., Naroditsky, O., Bergen, J.: Indoor Positioning Using Multi-Frequency RSS with Foot-Mounted INS. In: CVPR'04. (2004) 652 – 659
15. Davison, A.: Real-Time Simultaneous Localisation and Mapping with a Single Camera. In: CVPR'03. (2003) 1403 – 1410
16. Davison, A., Reid, I., Molton, N., Stasse, O.: MonoSLAM: Real-Time Single Camera SLAM. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6) (Jun 2007) 1052 – 1067
17. Klein, G., Murray, D.: Parallel Tracking and Mapping for Small AR Workspaces. In: ISMAR'07. (2007)
18. Wendel, A., Maurer, M., Graber, G., Pock, T., Bischof, H.: Dense Reconstruction On-the-fly. In: CVPR'12. (2012) 1450 – 1457
19. Pradeep, V., Rhemann, C., Izadi, S., Zach, C., Bleyer, M., Bathiche, S.: Monofusion: Real-time 3d reconstruction of small scenes with a single web camera. In: ISMAR', IEEE (2013) 83–88
20. Concha, A., Civera, J.: DPPTAM: Dense Piecewise Planar Tracking and Mapping from a Monocular Sequence . In: IROS'15. (2015)
21. Tarrio, J., Pedre, S.: Realtime edge-based visual odometry for a monocular camera. In: IEEE Int. Conf. Computer Vision, ICCV'15. (2015) 702 – 710

22. Geiger, A., Roser, M., Urtasun, R.: Efficient Large-Scale Stereo Matching. In: Asian Conf. Computer Vision, ACCV'10. (2010) 25 – 38
23. Sinha, S., Scharstein, D., Szeliski, S.: Efficient High-Resolution Stereo Matching Using Local Plane Sweeps. In: CVPR'14. (2014) 1582 – 1589
24. Zhang, C., Li, Z., Cheng, Y., Cai, R., Chao, H., Rui, Y.: MeshStereo: A Global Stereo Model with Mesh Alignment Regularization for View Interpolation. In: IEEE Int. Conf. Computer Vision, ICCV'15. (2015) 2057–2065
25. Becker, F., Lenzen, F., Kappes, J., Schnörr, C.: Variational Recursive Joint Estimation of Dense Scene Structure and Camera Motion from Monocular High Speed Traffic Sequences. In: IEEE Int. Conf. Computer Vision, ICCV'11. (2011) 1692 – 1699
26. Concha, A., Hussain, W., Montano, L., Civera, J.: Incorporating Scene Priors to Dense Monocular Mapping. Autonomous Robots **39**(3) (Oct 2015) 279–292
27. Salas, M., Hussain, W., Concha, A., Montano, L., Civera, J., Montiel, J.: Layout Aware Visual Tracking and Mapping. In: IROS'15. (2015)
28. Lucas, B., Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In: Int. Joint Conf. Artifical Intelligence, IJCAI'81. (1981) 674 – 679
29. Li, H., Summer, R., Pauly, M.: Global Correspondence Optimization for Non-Rigid Registration of Depth Scans. **27**(5) (2008) 1421–1430
30. Yang, J., Li, H.: Dense, Accurate Optical Flow Estimation with Piecewise Parametric Model. In: ECCV'15. (2015) 1019 – 1027
31. Westheimer, G.: Cooperative Neural Processes Involved in Stereoscopic Acuity. **36** (1979) 585–597
32. Zach, C.: Robust Bundle Adjustment Revisited. In: ECCV'14. (2014) 772–787
33. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A Benchmark for the Evaluation of RGB-D SLAM Systems. In: IROS'12. (2012)
34. Mobahi, H., Fisher, J.: On the Link between Gaussian Homotopy Continuation and Convex Envelopes. In: Int. Conf. Energy Minimization Methethods Computer Vision and Pattern Recognition, EMMCVPR'15. (2015) 43–56