2017 IEEE International Conference on Robotics and Automation (ICRA)
Singapore, May 29 - June 3, 2017

Illumination Change Robustness in Direct Visual SLAM

Seonwook Park[1] Thomas Schöps[1] Marc Pollefeys[1,2]
spark@student.ethz.ch {thomas.schoeps,marc.pollefeys}@inf.ethz.ch

***Abstract*— Direct visual odometry and Simultaneous Localization and Mapping (SLAM) methods determine camera poses by means of direct image alignment. This optimizes a photometric cost term based on the Lucas-Kanade method. Many recent works use the brightness constancy assumption in the alignment cost formulation and therefore cannot cope with significant illumination changes. Such changes are especially likely to occur for loop closures in SLAM. Alternatives exist which attempt to match images more robustly. In our paper, we perform a systematic evaluation of real-time capable methods. We determine their accuracy and robustness in the context of odometry and of loop closures, both on real images as well as synthetic datasets with simulated lighting changes. We find that for real images, a Census-based method outperforms the others. We make our new datasets available online[3].**

I. Introduction

Direct methods for camera pose estimation based on the Lucas-Kanade method [1] have become popular recently in visual odometry and SLAM (*c.f.* [2]–[5]). They can use more image information than feature-based methods which are limited to certain feature types. Feature-based methods compute descriptors such as SIFT [6] to gain invariance against appearance changes. In contrast, many recent works that use direct methods [2]–[5], [7]–[15] compare pixels based on the brightness constancy assumption and treat appearance changes due to lighting variations as outliers. As our results show (*c.f.* Fig. 5), this fails as soon as larger illumination changes occur. Since the illumination cannot always be controlled in real-world applications, there is a need for direct image alignment methods which are both real-time capable and robust against light changes. In this paper, we aim to evaluate such methods with respect to their accuracy and robustness, determining the trade-off between invariance and matching ambiguity that has to be made.

We evaluate this both in the context of visual odometry, where illumination changes between subsequent frames are usually small, and in scenarios such as loop closures, where significant illumination changes are common. In particular, we make the following contributions: i) We provide an extensive evaluation of fast, lighting change robust direct pose tracking methods. ii) We introduce an extension to the ICL-NUIM dataset [16], [17] (see Fig. 1) and provide RGB-D sequences recorded by a Kinect which allow to evaluate robustness against global and local lighting changes.

[1]Department of Computer Science, ETH Zürich, Zürich, Switzerland.
[2]Microsoft, Redmond, United States of America.
This work was supported by Google and by Qualcomm. We thank Torsten Sattler for helpful comments.
[3]http://cvg.ethz.ch/research/illumination-change-robust-dslam

Fig. 1. Example images from our extension to the ICL-NUIM dataset [16] exhibiting high temporal variation in illumination. Clockwise from the top left: static lighting, global variation, flashlight, local variation.

II. Related Work

We first survey methods for illumination invariance used in recent works utilizing direct pose estimation. Then we discuss related works on evaluating robust image matching.

Methods. A large number of recent works (*e.g.*, [2]–[5], [7]–[15]) rely on the brightness constancy assumption and do not directly account for illumination changes. Small image regions affected by local light changes are treated as outliers. We describe this baseline approach in Sec. III.

[18]–[20] gain robustness against global intensity biases by subtracting the median value from the pixel residuals. Local illumination changes are downweighted by the Huber function. We detail this method in Sec. IV-A.

Several works [21], [22] jointly optimize for the relative pose as well as an affine brightness transfer function between the images. In DSO [23], in addition a photometric calibration of the camera is used to explicitly account for its response function, vignetting and exposure time changes, if known. Jin *et al.* [24] estimate affine models for local patches, handling more fine-grained illumination changes. We evaluate the approach of [21] as described in Sec. IV-B.

Mutual Information (MI) is a metric which is well suited for aligning images with vastly different appearance. While it has not been used for visual odometry yet, efficient optimization methods for it have been proposed. Dame and Marchand [25] present an inverse compositional approach in the context of real-time template tracking. In contrast to earlier works (*e.g.* [26], [27]), they take second order terms of the cost function's Hessian into account and show that

this is important for the optimization. However, as noted by Fraissinet-Tachet *et al*. [28], their derivative terms are incorrect. We evaluate the use of MI for tracking based on [25], [28] as described in Sec. IV-D.

Another approach is to compare gradients (instead of comparing raw image intensities) as used by Dai *et al*. [29]. We describe this approach in Sec. IV-F and also evaluate using gradient magnitudes in Sec. IV-E.

In the context of planar template tracking, the use of Zero-Mean Normalized Cross Correlation (ZNCC), which is invariant against affine intensity changes, has also been explored. Scandaroli *et al*. [30] present an efficient Newton-style optimization for ZNCC-based tracking and propose to increase its robustness against local outliers by subdivision and weighting of the template region. Irani and Anandan [31] propose to transform the images into high-pass energy images and align those by optimizing a patch-based ZNCC cost. We evaluate ZNCC applied to whole images using the formulation in [30], described in Sec. IV-C.

Crivellaro and Lepetit [32] present an approach based on dense descriptor computation. They specifically aim to obtain a clear global optimum of the cost function, allowing for a wide basin of convergence on low-pass filtered images. We evaluate their first-order descriptor, defined in Sec. IV-H.

Recently, the Census transform [33] has been proposed to be used for direct camera tracking by Alismail *et al*. [34]. We describe this approach in Sec. IV-I.

Other related methods include that of Silveira and Malis [35], who present a template tracking approach which explicitly models illumination changes on the planar template, assuming smooth changes. It does not directly apply to general scenes, since the smoothness assumption is violated at depth discontinuities. Meilland *et al*. [36] model the environment in high dynamic range by estimating the exposure time jointly with the pose, which allows to globally account for camera-induced brightness changes. However, we are also interested in accounting for externally introduced illumination changes. Furthermore, Bartoli [37] describes how to extend the inverse compositional approach for pose updates to also handle photometric differences while keeping the ability for pre-computing the inverse of the Hessian matrix.

Cross-Cumulative Residual Entropy [38] and Sum of Conditional Variance [39] are further metrics for robust image alignment that are used in template tracking, which are out of the scope of this paper.

Evaluations. Antonakos *et al*. [40] look at the alignment quality achieved by different descriptors in dense descriptor based face alignment and fitting. They find that HOG [41] and SIFT [6] work particularly well. For their case, warping descriptors computed on the original images outperforms computing descriptors on the warped images in each iteration. In contrast to this our results indicate that, by adopting good depth maps which can be used for reprojection, re-computing descriptors on warped images significantly outperforms warping descriptor images. We consider dense computation of expensive descriptors like SIFT for whole images as currently unsuitable for real-time tracking and therefore do not evaluate this.

For the problem of optical flow computation, an evaluation of matching costs has been carried out by Vogel *et al*. [42]. They find that the Census transform, and an approximate variant of it, have a slight advantage over other methods. Optical flow differs from direct image alignment in that much more unknowns are optimized compared to the residual count, and regularization is necessary to constrain the optimization.

Hirschmüller and Scharstein [43] evaluate different matching metrics in stereo depth estimation for images with radiometric differences. They conclude that the Census transform showed the best overall performance. Similar to optical flow, stereo matching aims to determine much more unknowns per residual than direct image alignment, thus potentially requiring more discriminative cost metrics per pixel.

III. Direct Alignment for Camera Pose Tracking

In this section, we describe the basic cost formulation that other approaches presented later build on, as well as the optimization approach we use. This basic formulation is in this or a similar form used in many existing works (*e.g*., [2]–[5], [7]–[15]), therefore allowing for wide applicability of our results. We aim to look at the robustness of the photometric residual in isolation and thus do not use depth residuals. We refer to [13] for a comparison between using photometric residuals only, depth residuals only, and both. In this section, we also present two ways of using descriptors for alignment.

Cost. We model images as functions mapping a pixel coordinate to an image intensity value obtained using bilinear interpolation, *e.g*. $I : \mathbb{R}^2 \to \mathbb{R}$, respectively a depth value computed using nearest-neighbor interpolation for depth maps. The input images are a template T with a depth map D, and an image I which shall be aligned to the template, for which no depth information is needed. D contains depth values for a subset Ω_D of all pixels in T. A warp function $W(\mathbf{x}, d, \mathbf{M})$ re-projects a pixel with coordinates $\mathbf{x}$ and depth d from one image to another given the rigid transformation matrix $\mathbf{M} \in \mathbb{R}^{3\times4}$ for their relative pose:

$$W(\mathbf{x}, d, \mathbf{M}) = \pi_{I_1}(\mathbf{M} \cdot \pi_{I_2}^{-1}(\mathbf{x}, d)) \tag{1}$$

Here, π_{I_1} denotes the projection of a 3D point into image I_1, and $\pi_{I_2}^{-1}$ analogously denotes un-projection of a pixel with given depth from image I_2 to a 3D point. The image intrinsics for π_{I_1}, $\pi_{I_2}^{-1}$ are given by the context of the re-projection. The multiplication with $\mathbf{M}$ uses homogeneous coordinates. Fig. 2 (left) illustrates the warp. In the following, we abbreviate the notation as $\mathbf{x}_\mathbf{M} := W(\mathbf{x}, D(\mathbf{x}), \mathbf{M})$.

Starting from an initial guess for the pose, the image alignment can then be formulated as an optimization problem. The basic cost term is:

$$C(\mathbf{M}) = \frac{1}{|\Omega_D|} \sum_{\mathbf{x}\in\Omega_D} \rho\left(I(\mathbf{x}_\mathbf{M}) - T(\mathbf{x})\right) \tag{2}$$

Here, ρ is a robust weighting function which downweights outliers such as moving objects and specular reflections, for

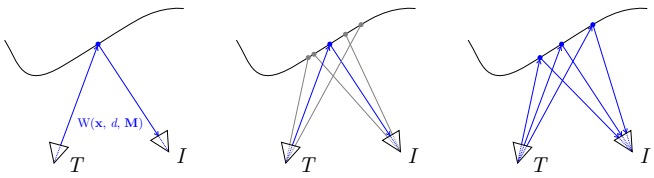

Fig. 2. **Left**: Sketch of the warping function for pixel-wise cost calculation. Each pixel $\mathbf{x}$ with a depth estimate d is unprojected from the template T, and reprojected into the image I for the intensity lookup. **Middle**: If looking up pre-computed patch descriptor vectors based on the warped center pixel (blue), patch pixels (gray) used for descriptor computation do not necessarily correspond to the same surface points, even at the correct pose estimate. We depict interpolated pixel rays here for clarity. **Right**: If the depth of all patch pixels is known, warping each pixel separately allows using patch-based metrics while all patch pixels correspond in the case of correct alignment.

which we choose the Huber norm [44]:

$$\rho_k(r) = \begin{cases} \frac{r^2}{2}, & \text{if } |r| \leq k \\ k(|r| - \frac{k}{2}), & \text{otherwise} \end{cases} \tag{3}$$

In this work, we use the *inverse compositional* (IC) formulation [1] of the cost which allows to precompute some terms for higher runtime performance. In this formulation, the roles of the image and template are swapped, and the cost is optimized with respect to a transformation increment $\Delta\mathbf{M}$ over the current estimate $\mathbf{M}$:

$$C(\Delta\mathbf{M}) = \frac{1}{|\Omega_D|} \sum_{\mathbf{x}\in\Omega_D} \rho(I(\mathbf{x}_\mathbf{M}) - T(\mathbf{x}_{\Delta\mathbf{M}})) \tag{4}$$

After determining $\Delta\mathbf{M}$, $\mathbf{M}$ is then updated as $\mathbf{M} := \mathbf{M}(\Delta\mathbf{M})^{-1}$ and $\Delta\mathbf{M}$ reset to $\mathbf{0}$. For evaluations of the IC formulation compared to the forwards compositional and efficient second order method, we refer to [21], [32].

Optimization. Eq. (4) is typically minimized using the Gauss-Newton or Levenberg-Marquardt algorithm. In this work we use the latter, and account for the robust cost functions with Iterative Re-weighted Least Squares. For a minimal parametrization of pose updates, we represent them as $\mathfrak{se}(3)$ Lie algebra elements in minimal notation and use the exponential map to obtain pose updates in SE(3) [45]. $\Delta\mathbf{M}$ in Eq. (4) is therefore replaced by $\exp(\hat{\epsilon})$ with $\hat{\epsilon} \in \mathfrak{se}(3)$, where the hat operator $\hat{(\cdot)}$ transforms the minimal representation to a Lie Algebra element. As is also common, the optimization is embedded into a multi-resolution scheme to achieve a larger basin of convergence.

Descriptors. In Sec. IV, variations of Eq. (4) will be introduced which compute the cost based on patches of pixels instead of operating on single pixels only. We denote the descriptor vector computed from a patch $P = \{\mathbf{x}_1, \mathbf{x}_2, \ldots\}$ in image I as $\mathbf{d}(I, P)$.

We evaluate two different ways of using descriptors. The first is to compute dense descriptor images for T and I, $\mathbf{D}_T$ and $\mathbf{D}_I$. For each pixel in those images, a descriptor is computed based on its local neighborhood $N(\mathbf{x})$: $\mathbf{D}_I(\mathbf{x}) = \mathbf{d}(I, N(\mathbf{x}))$. Then, descriptor images are aligned by minimizing:

$$C(\Delta\mathbf{M}) = \frac{1}{|\Omega_D|} \sum_{\mathbf{x}\in\Omega_D} \rho(\|\mathbf{D}_I(\mathbf{x}_\mathbf{M}) - \mathbf{D}_T(\mathbf{x}_{\Delta\mathbf{M}})\|_2) \tag{5}$$

This is fast to compute as the descriptor images can be precomputed, but does not take the warp into account for descriptor computation. As a result, in general corresponding patches are not computed from the same surface points, as illustrated in Fig. 2 (middle), which will degrade the results.

An alternative is to compute descriptors from warped images, as shown in Fig. 2 (right):

$$C(\Delta\mathbf{M}) = \frac{1}{|\Omega_D|} \sum_{\mathbf{x}\in\Omega_D} \rho(\|\mathbf{d}(I, N(\mathbf{x})_\mathbf{M}) - \mathbf{d}(T, N(\mathbf{x})_{\Delta\mathbf{M}})\|_2) \tag{6}$$

Here, we extend the notation $\mathbf{x}_\mathbf{M}$ for a reprojected pixel to sets of pixels where each pixel is reprojected separately. We drop any pixels $\mathbf{x}$ for which at least one depth value in $N(\mathbf{x})$ is missing in this formulation, as the warped image values for computing the descriptor cannot be obtained in this case. An exception to this is the method presented in Sec. IV-G, which uses each pixel having a depth estimate and at least one additional depth estimate in $N(\mathbf{x})$.

While the latter variant of using descriptors is slower, it correctly takes the warp into account for comparing patches. Note that the derivatives of this variant can be slow to compute or uninformative depending on the descriptor. We therefore use an approximation which is also employed in [40] and use the descriptor image gradient instead, assuming

$$\frac{\partial}{\partial\epsilon}\mathbf{d}(T, N(\mathbf{x})_{\Delta\mathbf{M}}) \approx \nabla\mathbf{D}_T \frac{\partial}{\partial\epsilon}\mathbf{x}_{\Delta\mathbf{M}} \tag{7}$$

IV. Illumination Change Robust Formulations

In this section, we present variations to the brightness constancy assumption (BCA) based formulation of Sec. III which aim to improve its robustness against illumination changes. The selection of methods is intended to cover real-time capable methods suited for direct SLAM. The methods described in Sec. IV-A - IV-D use global models for intensity changes between the images. The following ones described in Sec. IV-E - IV-G are based on image gradients which leads to invariance against local intensity bias changes. Finally, Sec. IV-H, IV-I present patch-based methods similar to matching cost metrics in 3D reconstruction, respectively descriptors in feature-based SLAM. Tab. I lists the theoretical invariance properties of the presented methods, and (qualitative) actual robustness against global and local changes as derived from our results (Fig. 5).

A. Global median bias normalization (GMedian)

[18], [19] suggest to normalize for global intensity bias using the median of the residuals. Before each optimization step, the bias is estimated as:

$$\beta = \text{median}_\mathbf{x}(I(\mathbf{x}_\mathbf{M}) - T(\mathbf{x}_{\Delta\mathbf{M}})) \tag{8}$$

Eq. (4) then becomes:

$$C(\Delta\mathbf{M}) = \frac{1}{|\Omega_D|} \sum_{\mathbf{x}\in\Omega_D} \rho(I(\mathbf{x}_\mathbf{M}) - T(\mathbf{x}_{\Delta\mathbf{M}}) - \beta) \tag{9}$$

TABLE I

ILLUMINATION INVARIANCE PROPERTIES OF THE EVALUATED METHODS. +, O, AND - DENOTE HIGH, MEDIUM, AND LOW INVARIANCE.

	BCA (III)	GMedian (IV-A)	GAffine (IV-B)	ZNCC (IV-C)	MI (IV-D)	GradM (IV-E)	Grad (IV-F)	LMean (IV-G)	DF (IV-H)	Census (IV-I)
Invariance domain	none	global	global	global	global	local	local	local	local	local
Invariance type	none	bias	affine	affine	mutual inf.	bias	bias	bias	bias	order-preserving
Global changes	-	o	o	+	+	+	+	+	+	+
Local changes	-	o	o	o	o	+	+	+	+	+

B. Global affine model (GAffine)

In this formulation (*e.g.*, used in [21], [22]), the intensities of one of the two images being aligned are transformed by an affine function to model global light or exposure changes. $I(\mathbf{x_M})$ is thus replaced by $(1+\alpha)I(\mathbf{x_M})+\beta$ in Eq. (4). By using an inverse update rule, Eq. (4) becomes:

$$C(\Delta\mathbf{M},\delta\alpha,\delta\beta)=\frac{1}{|\Omega_D|}\sum_{\mathbf{x}\in\Omega_D}\rho((1+\alpha)I(\mathbf{x_M})+\beta-(1+\delta\alpha)T(\mathbf{x}_{\Delta\mathbf{M}})-\delta\beta) \tag{10}$$

Each iteration jointly optimizes for both $\Delta\mathbf{M}$ and $\delta\alpha$, $\delta\beta$. The coefficients α and β are then updated as follows after each iteration, and $\delta\alpha$ and $\delta\beta$ are reset to zero afterwards:

$$\alpha:=\frac{\alpha-\delta\alpha}{1+\delta\alpha}\qquad\beta:=\frac{\beta-\delta\beta}{1+\delta\alpha} \tag{11}$$

C. Zero-mean normalized cross correlation (ZNCC)

ZNCC is a correlation metric which is invariant against affine intensity changes. The ZNCC of the template image with the other image can be computed as:

$$\mathrm{ZNCC}(\Delta\mathbf{M})=\frac{\sum_{\mathbf{x}\in\Omega_D}i_\mathbf{x}t_\mathbf{x}}{\sqrt{\sum_{\mathbf{x}\in\Omega_D}i_\mathbf{x}^2}\sqrt{\sum_{\mathbf{x}\in\Omega_D}t_\mathbf{x}^2}} \tag{12}$$

with $i_\mathbf{x}=(I(\mathbf{x_M})-\bar{I})$, $t_\mathbf{x}=(T(\mathbf{x}_{\Delta\mathbf{M}})-\bar{T})$ where $\bar{I}=\frac{1}{|\Omega_D|}\sum_{\mathbf{x}\in\Omega_D}I(\mathbf{x})$. For image alignment, Eq. (12) is maximized. We refer to [30] for details of the maximization.

D. Mutual information (MI)

Mutual information is a measure of the dependence of two random variables. In the context of matching images, it can be defined as:

$$\mathrm{MI}(\Delta\mathbf{M})=\sum_{r,t\in\Omega_I}p_{IT}(r,t,\Delta\mathbf{M})\log\left(\frac{p_{IT}(r,t,\Delta\mathbf{M})}{p_I(r)p_T(t,\Delta\mathbf{M})}\right) \tag{13}$$

where Ω_I is the domain of a histogram of image intensities, and p denotes the occurrence probability of the intensities associated with a bin. $p_I(r)$ is the probability of intensity bin r in image I. p_T and p_{IT} are the template bin and joint bin probabilities, given $\Delta\mathbf{M}$. For calculation and optimization of this metric, we largely follow the real-time capable formulation of [25], with improvements as noted in [28]: We use Levenberg-Marquardt as the optimization algorithm, and use the derivatives of MI of [28]. In addition, since in our case parts of the template may be unobserved in the other image, we re-compute the Hessian at the start of every scale in the multi-scale optimization to only take the currently overlapping pixels into account.

E. Gradient magnitude (GradM)

Instead of aligning raw image intensities, this formulation aligns gradient magnitudes and is therefore invariant against local intensity bias changes. [32] includes this formulation in its evaluation. Here, we view gradient computation for a pixel in image I as a function depending on the pixel's neighborhood: $\nabla(I,N(\mathbf{x}))$. We thus use cost function (5) or (6) for this method, with the single-component descriptor defined as the gradient magnitude: $\mathbf{d}(I,N(\mathbf{x}))=\|\nabla(I,N(\mathbf{x}))\|_2$. This allows to compute the gradient from warped neighbor pixel coordinates, *c.f.* Eq. (6) and Fig. 2 (right).

F. Gradient (Grad)

Similar to the previous formulation, this variant uses gradients to be invariant against local bias changes, but matches them directly in vector form. This has been used in [29]. The descriptor for this method is $\mathbf{d}(I,N(\mathbf{x}))=\nabla(I,N(\mathbf{x}))$.

G. Local mean bias normalization (LMean)

This formulation locally normalizes pixel intensities by subtracting from them the mean intensity of a patch around them and is therefore also invariant against local bias changes. This normalization has for example been applied to stereo depth estimation in [43]. The cost can be formulated using a single-component descriptor:

$$d(I,N(\mathbf{x}))=I(\mathbf{x})-\frac{1}{|N(\mathbf{x})|}\sum_{\mathbf{y}\in N(\mathbf{x})}I(\mathbf{y}) \tag{14}$$

H. Descriptor fields (DF)

This descriptor-based method [32] defines $\mathbf{d}$ as follows:

$$\mathbf{d}(I,N(\mathbf{x}))=\left[[(\mathbf{f}_1*I)(\mathbf{x})]^+,[(\mathbf{f}_1*I)(\mathbf{x})]^-,\ldots,[(\mathbf{f}_n*I)(\mathbf{x})]^+,[(\mathbf{f}_n*I)(\mathbf{x})]^-\right]^T, \tag{15}$$

where the $\mathbf{f}_i$ are convolution kernels applied to the image I, and the $[\cdot]^+$ and $[\cdot]^-$ operators keep only the absolute value of positive respectively negative values:

$$[x]^+=\begin{cases}x, & \text{if } x\geq 0\\ 0, & \text{otherwise}\end{cases},\ \text{and}\quad [x]^-=[-x]^+\ . \tag{16}$$

We implemented the first-order variant included in the open source code of [32], which is well-performing according to [32] and uses the derivatives of a Gaussian with standard deviation 1 as convolution kernels: $\mathbf{f}_1=G_x$, $\mathbf{f}_2=G_y$.

I. Census transform (Census)

The Census transform [33] is popular in stereo depth estimation. This local descriptor is invariant against all intensity transformations that preserve the intensity ordering, but may

provide less precise information about the exact alignment than other metrics. Recently, it has also been applied to visual odometry as the 'bit-planes' descriptor [34]. To compute the Census transform for a pixel $\mathbf{x}$, the intensity of each pixel in its local neighborhood $N(\mathbf{x})$ is compared to that of pixel $\mathbf{x}$. Each comparison results in one bit of information indicating whether the central pixel is brighter or darker than the other one. The choice of comparison operator ($<$, $\leq$, $\geq$, $>$) must be consistent, but the exact choice is irrelevant. All bits obtained by this procedure are concatenated to form the descriptor. Following [34], we treat them as individual components of a descriptor vector $\mathbf{d}(I, N(\mathbf{x}))$ here. We use a 3×3 neighborhood in our implementation. The i-th component of the descriptor is thus defined as follows, depending on the i-th neighbor pixel $\mathbf{x}_i \in N(\mathbf{x}) \setminus \{\mathbf{x}\}$:

$$\mathbf{d}(I, N(\mathbf{x}))_i = \begin{cases} 1, & \text{if } I(\mathbf{x}) < I(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

Gradient-based optimization is in principle not suited to optimize an alignment using this descriptor, since the gradient is zero or undefined everywhere. The gradient approximation in Eq. (7) makes it possible to obtain results. One can also approximate the Census transform to avoid this issue [42].

V. Evaluation

We first describe the datasets used in our evaluations in Sec. V-A. We evaluate different parameter settings for our methods in Sec. V-B. We then test the methods using the best parameters we determined both in the context of high frame rate visual odometry where changes between frames tend to be small (Sec. V-C), as well as by aligning images with different appearance to simulate loop closures (Sec. V-D). We list the timings of our implementations in Sec. V-E.

A. Datasets

We perform visual odometry testing on datasets from three different sources: first, we evaluate on sequences from the commonly used TUM RGB-D dataset [46] to allow comparisons between our results, and those of previously published works. However, these datasets only contain minor illumination variations. Thus, we generated synthetic sequences based on the ICL-NUIM dataset [16] to specifically test for robustness against such variations. Synthetic data was used to have perfect ground truth and control over the sequence setup. We apply noise to the depth maps as in [17]. Each sequence (**`s1`**, **`s2`**) is generated in five variations: first, with its default, fixed lighting configuration (**`const`**). Second, with gradual temporal global and local lighting changes both individually (**`global`**, **`local`**) and jointly (**`loc+glo`**). The last variant (**`flash`**) simulates a flashlight attached to the camera. Finally, to test illumination invariance on real data, we acquired RGB-D sequences using a 1st-generation Kinect with ground truth provided by a VICON system. We recorded 3 sequences with variations similar to our synthetic datasets: **`global`**, **`local`**, **`flash`**.

For loop closure testing, we additionally generated synthetic image pairs with a resolution of 320×240 by rendering 3D reconstructions created with a Google Tango device [47].

TABLE II
Selected Huber parameters

BCA	GMedian	GAffine	GradM	Grad	LMean	DF
10.0	6.5	5.2	16.9	7.3	2.9	300.0

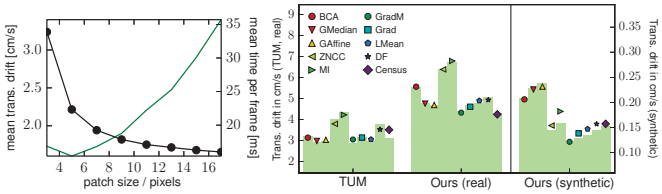

Fig. 3. **Left:** Translational drift (black) and runtime (green) for varying patch sizes in *LMean*. **Right:** Mean translational drift of visual odometry for three groups of sequences: TUM, synthetic (ICL-NUIM) and real. The green bars qualitatively show rotational drift, scaled to the same mean value.

B. Parameter evaluation

We tune the methods' parameters by running them in a visual odometry setting on a set of training sequences. We use the tools provided in [46] to evaluate the accuracy. Trajectory estimation is done frame-by-frame where each frame is aligned against the previous one. We report mean values over multiple datasets for the translational root mean square error (RMSE) of relative pose change.

For the multi-resolution optimization, we compute image pyramids down to a size of 40×30 pixels. The Levenberg-Marquardt algorithm is run on each pyramid level until the mean pixel position update (*c.f.* [28]) is less than 10^{-2}, or 20 iterations are reached. We generally compute gradients with centralized finite differences, which worked best over multiple methods. The Sobel filter is used for the *GradM* method, and for pixel selection for depth density variation.

The Huber parameter k in Eq. (3) is tuned for each method by minimizing the mean translational drift over the TUM RGB-D datasets **`fr1/desk`**, **`desk2`**, **`plant`** and **`floor`** via a parameter sweep. The results are shown in Tab. II. The Huber function is not used by *ZNCC*, *MI*, and *Census*.

To select a patch size for *LMean*, we run a parameter sweep on the synthetic datasets {**`s1`**, **`s2`**}**`/const`** and {**`s1`**, **`s2`**}**`/loc+glo`**. The results are shown in Fig. 3 (left). Increasing the patch size reduces the drift, but makes the algorithm slower. We select a size of 11×11 as a tradeoff.

If dense depth maps are available as input, a sub-selection of depth pixels may be helpful to improve runtime performance. Other ways of acquiring depth maps, such as multi-view stereo, may only estimate depth values for high-gradient pixels. We therefore evaluate the effect of varying depth map density by discarding all pixels having a gradient magnitude below a threshold. The relative change in translational drift and time taken per frame induced by this is shown in Fig. 4. Generally, sparser depth maps lead to higher drift while reducing the runtime. However, in particular for *Census*, discarding low-gradient pixels reduces the drift because descriptors for those pixels are noisy. We therefore choose the threshold minimizing the drift for each method.

As discussed in Sec. III, descriptors can be used by *pre-computing* descriptor images (Eq. (5)), or by *re-computing*

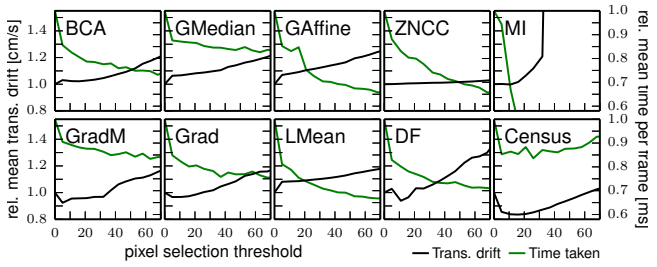

Fig. 4. Relative change in translational drift (black) and time per frame (green) when varying depth image density via a gradient magnitude threshold. *MI* fails for high thresholds. Evaluated on {**s1, s2**}**/const** and {**s1, s2**}**/loc+glo**.

TABLE III

TRANSLAT. DRIFT (cm / second) FOR DESCRIPTOR-BASED METHODS[1]

	Pre-computation	Re-computation
GradM	3.96	**3.53**
Grad	4.15	**3.69**
LMean	5.77	**3.74**
DF	7.17	**3.78**
Census	4.82	**4.05**

[1] Evaluated on **TUM** and **real** sequences.

descriptors in every optimization iteration (Eq. (6)). We show the translational drift for descriptor-based methods over all real RGB-D sequences used in this evaluation for both variants in Tab. III. Re-computation always results in significantly lower drift. Notably, this is true also for gradient-based methods.

C. Visual odometry

The results of all methods on the visual odometry datasets described in Sec. V-A are shown in Tab. IV, with mean values plotted in Fig. 3 (right). We generally only state translational drift since we observed the rotational drift to behave similarly (*c.f.* Fig. 3 (right)). As the TUM RGB-D datasets do not exhibit strong illumination changes, the basic *BCA* performs well on them. *GMedian*, *GAffine*, *GradM*, and *LMean* deal with the slight lighting changes very well and outperform more complex methods, with *GMedian* performing best overall on these datasets by a slight margin. However, many methods perform similarly well on these datasets. The results on the synthetic datasets show that while all methods can cope well with gradual **global** and **local** lighting changes, the **flash** variant poses a harder challenge. *GradM* yields the best results for these synthetic sequences. However, on our real datasets, *Census* performs better than all other methods. *GradM* still performs well, yielding the second best results on average. *GMedian*, *GAffine*, and *LMean* do not work as well on these datasets as they do on the TUM RGB-D datasets.

Overall, we note that *GradM* consistently performs very well over all datasets. However, on the real-world datasets we captured, it is outperformed by *Census*, potentially indicating that the appearance variations found in real-world data are too complex for gradients to provide invariance against them. Nevertheless, the results on the synthetic and real data are mostly consistent, showing that valid insights can be derived from the synthetic data.

D. Alignment under image degradations

In this experiment, we evaluate the methods' convergence probabilities. We generated many synthetic image pairs with given degradations applied, and evaluated the rate of successful alignments achieved by each method depending on the degradation. We treat poses which differ from the ground truth by less than 2% in translation relative to the average scene depth, and by less than 1° in shortest-path rotation, as correctly aligned. Slightly increasing these parameters does not significantly change the results since diverged poses are usually far away. Results are shown in Fig. 5.

We test for robustness against translation, rotation, global and local illumination changes, intensity and depth noise, blur, and occlusion. The translation amount is defined by the mean optical flow in pixels induced by only camera translation, to have a measure independent of scene depth. Rotation is given in degrees. Affine global illumination changes are simulated by transforming the image to be aligned (I) with gain and bias parameters such that $I' = \alpha \cdot I + \beta$. We choose $\alpha = 1 - \frac{d}{2}, \beta = \frac{255}{2}d$ with $d \in [0;1]$ given in the plots. We simulate flashlight-like local illumination changes by multiplying each pixel's intensity in I by $1 - \frac{r}{r_{\max}}d$. r is the distance from the image center relative to the distance from a corner to the center, $r_{\max}$, while $d \in [0;1]$ is the degradation amount given in the plots. Intensity noise is added to both the template and I by for each pixel independently sampling from a zero-mean normal distribution and adding the sample to each color channel. We specify the noise intensity as the standard deviation σ. Similarly, for depth noise we add independent samples from a normal distribution to each pixel's depth in the template, with the intensity specified by σ. For blur, we apply a Gaussian blurring kernel to I with the standard deviation given in the plots. For occlusion we paint a dot in the center of I and specify the distortion intensity as the amount of the image area covered. We evaluate each degradation type together with translation since degraded but aligned images often do not offer a challenge.

Fig. 5 shows the convergence rates (mapped to a color gradient shown at the bottom) for each method, depending on the translation between the images (x axis of each plot) and the other degradation given by the table row (y axis). The leftmost column illustrates the maximum distortion for each type by example. We used identity as the initial estimate for each alignment. The results are not transferable in an absolute sense, as they depend on *e.g.* the scene's texturedness. However, they offer a relative comparison.

The most important observations from this experiment are: *BCA* fails even for relatively small, abrupt global or local light changes. *GAffine*, due to the joint optimization process, only shows partial robustness against global affine changes. *MI* does not perform well. We believe that the optimization easily gets stuck in local optima for this method. The results of all gradient-based methods are very similar. They show good robustness against the simulated light changes, but have a smaller convergence basin compared to the previous methods. For comparison, we also include results obtained by

TABLE IV

EVALUATION RESULTS FOR VISUAL ODOMETRY (TRANSLATIONAL DRIFT IN cm / second)

	TUM RGB-D datasets						ICL-NUIM datasets with illumination changes[1]						Real data with illum. changes			
	fr1/desk	**fr1/desk2**	**fr1/plant**	**fr1/room**	**fr2/desk**	mean	**const**	**global**	**local**	**loc+glo**	**flash**	mean	**global**	**local**	**flash**	mean
BCA	3.02	4.74	2.45	4.25	1.22	3.14	**0.11**	0.13	0.12	0.12	0.55	0.21	7.03	5.08	4.57	5.56
GMedian	2.93	4.34	2.42	**4.00**	**1.21**	**2.98**	**0.11**	0.12	0.12	0.12	0.67	0.23	5.25	4.78	4.23	4.76
GAffine	2.89	4.49	2.51	4.07	**1.21**	3.03	0.12	0.12	0.12	0.12	0.69	0.23	5.20	4.70	4.17	4.69
ZNCC	3.94	4.93	3.93	4.64	1.52	3.79	0.13	0.13	0.14	0.13	0.24	0.15	7.67	6.13	5.35	6.39
MI	4.01	5.40	3.99	5.76	1.96	4.22	0.16	0.16	0.17	0.16	0.26	0.18	7.54	6.59	6.23	6.79
GradM	2.88	4.35	2.56	4.10	1.34	3.05	**0.11**	**0.11**	**0.11**	**0.11**	**0.17**	**0.12**	**4.77**	4.51	**3.71**	4.33
Grad	3.13	4.78	**2.15**	4.19	1.47	3.14	0.13	0.13	0.13	0.13	0.18	0.14	5.14	4.62	4.07	4.61
LMean	**2.84**	**4.29**	2.47	4.30	1.36	3.05	0.13	0.14	0.14	0.14	0.19	0.15	5.47	4.90	4.32	4.90
DF	3.56	5.27	2.48	4.89	1.41	3.52	0.14	0.14	0.14	0.14	0.21	0.16	5.21	5.06	4.52	4.93
Census	2.94	4.33	2.85	4.69	2.72	3.50	0.14	0.15	0.14	0.15	0.20	0.16	4.79	**4.19**	3.72	**4.23**

[1] Evaluated on two trajectories per variant: **s1** and **s2**.

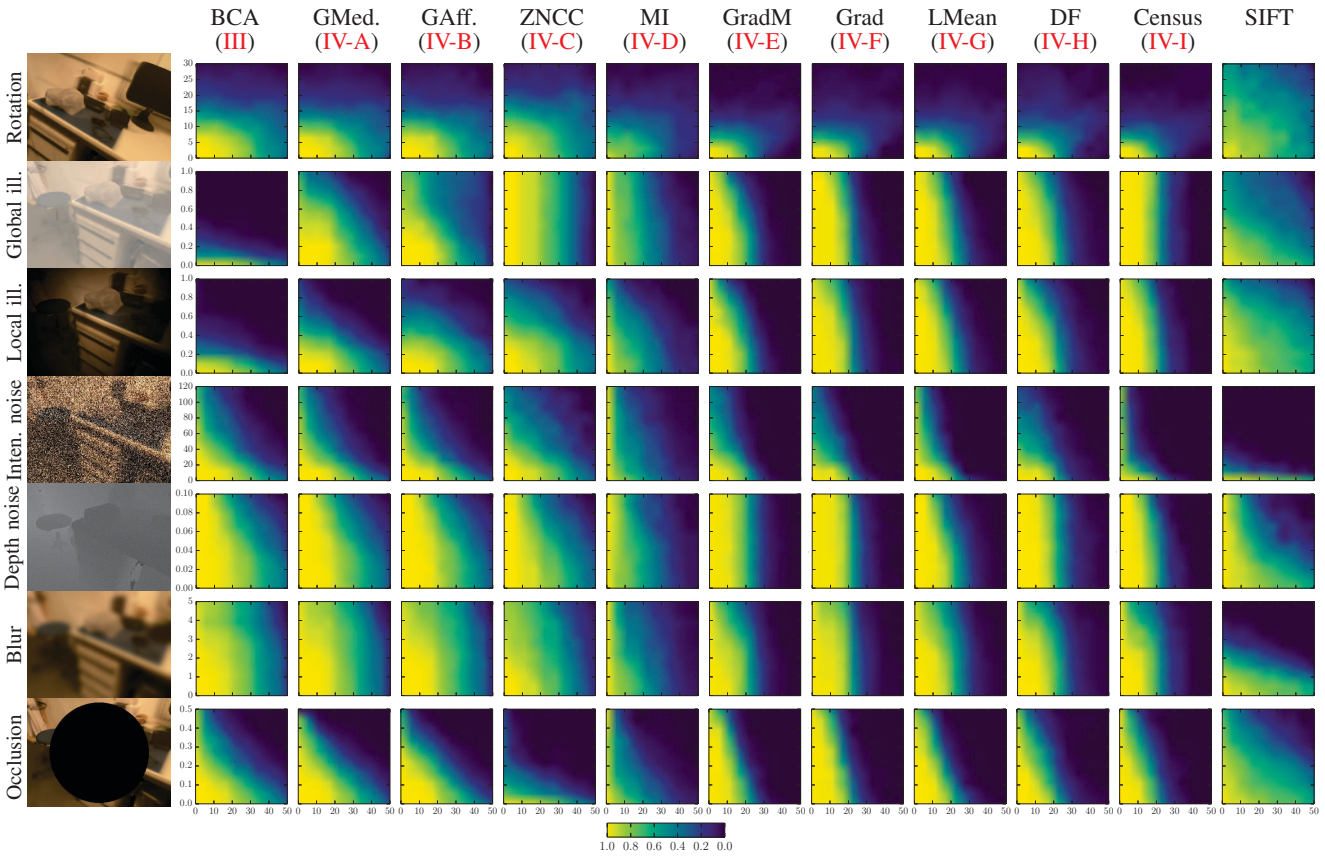

Fig. 5. Convergence evaluation on image pairs with different image degradations. X axis of each plot: amount of translation between the images as mean optical flow in pixels. Y axis of each plot: amount of degradation, given by the row. The color indicates the fraction of successful alignments as given by the scale on the bottom: Yellow means all alignments succeeded, blue means that all failed. See Sec. V-D for a detailed explanation and analysis.

SIFT descriptor matching using COLMAP [48]. As expected, this performs very well in this convergence evaluation since it finds matches in descriptor space instead of only locally optimizing the pose. It however performs poorly for intensity noise and blur, since those degradation types corrupt the image gradients. Furthermore, we also evaluated the resulting mean alignment accuracy for all pairs on which all methods converge, which showed that SIFT consistently resulted in significantly less accurate alignments than the direct methods. Among the direct methods, *GAffine* and similar methods have the largest convergence basin.

E. Runtime performance

We report the timings and average iteration counts of all methods for the visual odometry scenario in Tab. V. Evaluations are performed on a PC with an Intel Core i7 950 (3.07GHz) CPU and Nvidia GTX 950 GPU. All methods apart from *MI* are implemented on the GPU using CUDA. Our implementations are unoptimized. Furthermore, if good initial estimates are available, *e.g.*, from integrating data of an inertial measurement unit, convergence will be faster.

VI. CONCLUSIONS

We evaluated real-time capable direct image alignment methods for their accuracy and robustness under challenging lighting conditions. Using the brightness constancy assumption, as done by many recent works, fails in cases of abrupt illumination changes. The *GradM* method performs well in visual odometry accuracy evaluations while also being fast. Analogously to other methods which compute residuals from

TABLE V

AVERAGE TIME (ms) AND NUMBER OF ITERATIONS PER FRAME[1]

	Time	Iter.		Pre-compute Time	Pre-compute Iter.	Re-compute Time	Re-compute Iter.
BCA	**28** ± 13	25	GradM	35 ± 14	32	37 ± 15	32
GMedian	51 ± 21	26	Grad	67 ± 42	30	66 ± 40	29
GAffine	31 ± **11**	25	LMean	100 ± 14	73	72 ± 30	31
ZNCC	58 ± 39	**24**	Census	489 ± 195	72	355 ± 114	49
MI[2]	2374 ± 412	44	DF	114 ± 81	28	124 ± 86	28

[1] Evaluated on **`TUM`** and **`real`** sequences.
[2] CPU implementation. All others are GPU implementations.

pixel patches, it benefits from computing gradients based on warped pixel coordinates. However, for real-world datasets with significant illumination changes, we observed *Census* to give the most accurate results. While it allows for some interesting observations, care should therefore be taken when using synthetic data only, and one possible direction of future work is to increase its realism. For loop closures, as expected SIFT descriptor matching is in many scenarios able to align images with larger pose changes than the direct methods, while the latter provide better accuracy. This suggests to first use feature matching and then refine the pose with a direct method (as in [29]).

REFERENCES

[1] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *IJCV*, vol. 56, no. 3, pp. 221–255, 2004. 1, 3

[2] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison, "ElasticFusion: Dense SLAM without a pose graph," in *RSS*, 2015. 1, 2

[3] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *ECCV*, 2014. 1, 2

[4] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *ICCV*, 2011. 1, 2

[5] P. Ondrúška, P. Kohli, and S. Izadi, "MobileFusion: Real-time volumetric surface reconstruction and dense tracking on mobile phones," *ISMAR*, 2015. 1, 2

[6] D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, 1999. 1, 2

[7] A. I. Comport, E. Malis, and P. Rives, "Real-time quadrifocal visual odometry," *IJRR*, vol. 29, no. 2-3, pp. 245–266, 2010. 1, 2

[8] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *ICRA*, 2014. 1, 2

[9] M. Meilland, T. Drummond, and A. I. Comport, "A unified rolling shutter and motion blur model for 3D visual registration," in *ICCV*, 2013. 1, 2

[10] C. Kerl, J. Stueckler, and D. Cremers, "Dense continuous-time tracking and mapping with rolling shutter RGB-D cameras," in *ICCV*, 2015. 1, 2

[11] P. Henry, D. Fox, A. Bhowmik, and R. Mongia, "Patch volumes: Segmentation-based consistent mapping with RGB-D cameras," in *3DV*, 2013. 1, 2

[12] D. Caruso, J. Engel, and D. Cremers, "Large-scale direct SLAM for omnidirectional cameras," in *IROS*, 2015. 1, 2

[13] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *IROS*, 2013. 1, 2

[14] L. Ma, C. Kerl, J. Stueckler, and D. Cremers, "CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM," in *ICRA*, 2016. 1, 2

[15] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. Leonard, and J. McDonald, "Real-time large scale dense RGB-D SLAM with volumetric fusion," *IJRR*, 2014. 1, 2

[16] A. Handa, T. Whelan, J. McDonald, and A. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *ICRA*, 2014. 1, 5

[17] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *CVPR*, 2015. 1, 5

[18] M. Meilland, A. Comport, P. Rives, and I. S. A. Méditerranée, "Real-time dense visual tracking under large lighting variations," in *BMVC*, 2011. 1, 3

[19] T. Gonçalves and A. I. Comport, "Real-time direct tracking of color images in the presence of illumination variation," in *ICRA*, 2011. 1, 3

[20] W. N. Greene, K. Ok, P. Lommel, and N. Roy, "Multi-level mapping: Real-time dense monocular SLAM," in *ICRA*, 2016. 1

[21] S. Klose, P. Heise, and A. Knoll, "Efficient compositional approaches for real-time robust direct visual odometry from RGB-D data," in *IROS*, 2013. 1, 3, 4

[22] J. Engel, J. Stueckler, and D. Cremers, "Large-scale direct SLAM with stereo cameras," in *IROS*, 2015. 1, 4

[23] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *arXiv:1607.02565*, 2016. 1

[24] H. Jin, P. Favaro, and S. Soatto, "A semi-direct approach to structure from motion," *The Visual Computer*, vol. 19, no. 6, pp. 377–394, 2003. 1

[25] A. Dame and E. Marchand, "Second-order optimization of mutual information for real-time image registration," *ITIP*, vol. 21, no. 9, pp. 4190–4203, 2012. 1, 2, 4

[26] N. Dowson and R. Bowden, "Mutual information for Lucas-Kanade tracking (MILK): An inverse compositional formulation," *PAMI*, vol. 30, no. 1, pp. 180–185, 2008. 1

[27] N. Dowson and R. Bowden, "A unifying framework for mutual information methods for use in non-linear optimisation," in *ECCV*, 2006. 1

[28] M. Fraissinet-Tachet, M. Schmitt, Z. Wen, and A. Kuijper, "Multi-camera piecewise planar object tracking with mutual information," *Journal of Mathematical Imaging and Vision*, pp. 1–12, 2016. 2, 4, 5

[29] A. Dai, M. Nießner, M. Zollöfer, S. Izadi, and C. Theobalt, "BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface re-integration," *ACM Trans. Graph.*, 2017. 2, 4, 8

[30] G. G. Scandaroli, M. Meilland, and R. Richa, "Improving NCC-Based Direct Visual Tracking," in *ECCV*, 2012. 2, 4

[31] M. Irani and P. Anandan, "Robust multi-sensor image alignment," in *ICCV*. IEEE, 1998, pp. 959–966. 2

[32] A. Crivellaro and V. Lepetit, "Robust 3D tracking with descriptor fields," in *CVPR*, 2014. 2, 3, 4

[33] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *ECCV*, 1994. 2, 4

[34] H. Alismail, B. Browning, and S. Lucey, "Direct visual odometry using bit-planes," *arXiv:1604.00990*, 2016. 2, 5

[35] G. Silveira and E. Malis, "Real-time visual tracking under arbitrary illumination changes," in *CVPR*, 2007. 2

[36] M. Meilland, C. Barat, and A. Comport, "3D high dynamic range dense visual SLAM and its application to real-time object re-lighting," in *ISMAR*, 2013. 2

[37] A. Bartoli, "Groupwise geometric and photometric direct image registration," *PAMI*, vol. 30, no. 12, pp. 2098–2108, 2008. 2

[38] F. Wang and B. C. Vemuri, "Non-rigid multi-modal image registration using cross-cumulative residual entropy," *IJCV*, vol. 74, no. 2, pp. 201–215, 2007. 2

[39] R. Richa, M. Souza, G. Scandaroli, E. Comunello, and A. von Wangenheim, "Direct visual tracking under extreme illumination variations using the sum of conditional variance," in *ICIP*, 2014. 2

[40] E. Antonakos, J. Alabort-i Medina, G. Tzimiropoulos, and S. P. Zafeiriou, "Feature-based Lucas-Kanade and active appearance models," *ITIP*, vol. 24, no. 9, pp. 2617–2632, 2015. 2, 3

[41] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005. 2

[42] C. Vogel, S. Roth, and K. Schindler, "An evaluation of data costs for optical flow," in *GCPR*, 2013. 2, 5

[43] H. Hirschmüller and D. Scharstein, "Evaluation of stereo matching costs on images with radiometric differences," *PAMI*, vol. 31, no. 9, pp. 1582–1599, 2009. 2, 4

[44] P. J. Huber, *Robust statistical procedures*. SIAM, 1996. 3

[45] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D vision: From Images to Geometric Models*. Springer Science & Business Media, 2012, vol. 26. 3

[46] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems," in *IROS*, 2012. 5

[47] M. Klingensmith, I. Dryanovski, S. Srinivasa, and J. Xiao, "Chisel: Real time large scale 3D reconstruction onboard a mobile device," in *RSS*, 2015. 5

[48] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *CVPR*, 2016. 7