Camera Tracking in Lighting Adaptable Maps of Indoor Environments

Tim Caselitz[1] Michael Krawez[1] Jugesh Sundram[2] Mark Van Loock[2] Wolfram Burgard[1,3]

***Abstract*— Tracking the pose of a camera is at the core of visual localization methods used in many applications. As the observations of a camera are inherently affected by lighting, it has always been a challenge for these methods to cope with varying lighting conditions. Thus far, this issue has mainly been approached with the intent to increase robustness by choosing lighting invariant map representations. In contrast, our work aims at explicitly exploiting lighting effects for camera tracking. To achieve this, we propose a lighting adaptable map representation for indoor environments that allows real-time rendering of the scene illuminated by an arbitrary subset of the lamps contained in the model. Our method for estimating the light setting from the current camera observation enables us to adapt the model according to the lighting conditions present in the scene. As a result, lighting effects like cast shadows do no longer act as disturbances that demand robustness but rather as beneficial features when matching observations against the map. We leverage these capabilities in a direct dense camera tracking approach and demonstrate its performance in real-world experiments in scenes with varying lighting conditions.**

I. Introduction

Cameras are popular sensors for egomotion estimation in various applications including autonomous driving, service robotics, and augmented reality. Like the latter two, many of these applications target indoor environments where lighting conditions can change rapidly, e.g., when lights are switched on or off. In contrast to methods that rely on other modalities, actively project light into the scene, or operate outside the visible spectrum, egomotion estimation with a common camera can be significantly affected by lighting changes as they have a direct impact on the camera observations. How crucial this impact is for camera tracking depends on the image gradients introduced by lighting in relation to the gradients caused by changes in reflectance. In highly textured parts of the environment, i.e., areas with frequently changing reflectance, the multitude of reflectance gradients might dominate those caused by lighting. However, shadows cast onto texture-less areas, e.g., floors with uniform carpet or walls with uniform paint, introduce the only and therefore extremely valuable gradients. This even remains true when using an RGB-D camera, which additionally relies on active depth measurements, as the mentioned areas are often not only texture-less but also planar and therefore do not provide geometric features either. Especially in indoor environments, such areas are omnipresent and lighting can provide valuable information that we do not want to ignore but instead exploit explicitly for visual localization.

[1]Autonomous Intelligent Systems, University of Freiburg, Germany
[2]Toyota Motor Europe, R&D - Advanced Technology, Brussels, Belgium
[3]Toyota Research Institute, Los Altos, USA

Fig. 1: Our method performs camera pose tracking in varying lighting conditions. We estimate which lamps in the scene are currently on, adapt the map accordingly, and match the camera observation (bottom right) against the rendering of the lighting adapted map (bottom left).

The term *visual localization* describes the task of using images to estimate the camera pose w.r.t. an entire map that has been built previously. This is in contrast to *visual odometry* that only uses the most recent frame(s) as the reference. Visual localization contains the subtasks of finding a coarse global pose estimate, often called *global localization* or *relocalization*, and the subtask of *tracking*, which means accurately estimating the camera pose over time given an initial estimate. While we suppose that the proposed map representation can also be beneficial for relocalization, the focus of this paper is to use it for camera tracking.

The proposed map representation builds on our method for building dense reflectance maps of indoor environments [1]. In this paper we extend the reflectance map representation to so-called *lighting adaptable* maps, which, in addition to the surface reflectance model, contain the global lighting contributions of the lamps present in the scene. The model parameters allow to switch the contributions of individual lamps on and off. Even though these parameters could potentially be provided by an external system (e.g., home automation), we propose a method to estimate them from a single color image. Adapting the model parameters to predict the effects of lighting in real-time can subsequently be exploited when matching against real-world observations. We leverage these capabilities for camera tracking in lighting adaptable maps using a direct dense approach.

II. Related Work

Methods for visual localization typically aim to achieve robustness to changing lighting conditions by relying on map representations that target illumination invariance. This includes approaches using local feature descriptors which are invariant to affine changes in illumination. A prominent example is SIFT [2] that uses normalization and is based on gradients, i.e., is invariant to changes which multiply or add a constant to the image intensities. Other local features, e.g., ORB [3], which builds on a variant of BRIEF [4], are generally also based on intensity differences and perform similar w.r.t. illumination changes. As they are only designed to be invariant to affine variations, these descriptors cannot cope with more complex lighting changes. An idea is to improve this by learning invariances from data. Stavens and Thrun [5] learn domain-optimized versions of SIFT and HOG [6] whereas Carlevaris-Bianco and Eustice [7] directly learn a descriptor embedding for varying lighting conditions from raw pixel input. Ranganathan *et al.* [8] learn a fine vocabulary [9] and model variations in lighting using a probability distribution over descriptor space to achieve illumination invariance for visual localization.

The domain of metrical visual odometry and SLAM can be divided into feature-based (indirect) [10], [11], semi-direct [12], and direct approaches [13], [14], [15]. The former inherit the invariance to affine illumination changes from the discussed local feature descriptors. In comparison, direct approaches are by default much more sensitive to lighting changes as they directly compare image intensities and assume brightness constancy [16]. However, they can also gain robustness to changes in illumination, e.g., by correcting biases [17] or using affine illumination models [18], [19], [20]. Park *et al.* [21] provide a survey of methods for illumination invariance in direct visual SLAM. Compared to visual odometry, visual SLAM methods can be expected to experience more severe lighting changes as the map is built from data that is older than the last frame(s). This applies even more to localization in previously built maps where illumination changes are often too complex to be modeled with affine transformations. Clement and Kelly [22] address this issue by learning from data and propose an approach to direct visual localization that transforms input images into a canonical representation using a deep convolutional encoder-decoder network. In contrast, Corke *et al.* [23] adopt a camera-based model [24] and use illumination invariant imaging [25] for visual localization.

An approach for illumination invariant 3D reconstruction is presented by Kerl *et al.* [26]. Using an RGB-D camera, it infers color albedo (diffuse reflectance) by transfer from the infrared domain. Our method [1] avoids the assumption that invariance can be transfered from a different spectrum and recovers the diffuse reflectance based on the transport of visible light. However, directly using this lighting invariant reflectance map representation for visual localization would require to transform the camera images into reflectance space as well, i.e., to perform intrinsic image decomposition [27]. Even though approaches for real-time reflectance estimation exist [28], [29], they do not provide the spatial resolution required for localization. More importantly, matching in illumination invariant space ignores the effects of lighting and is thus not our intent. Instead, we employ the reflectance map to predict and explicitly exploit these effects. We use a dense camera tracking approach as non-dense methods might neglect low magnitude gradients induced by lighting.

Relighting a scene based on a recovered reflectance model has already been described by Yu *et al.* [30] two decades ago. Still, reflectance and lighting estimation remain active research topics [31], [32], [33]. From an application point of view, relighting has mainly been used for augmented reality. For instance, Zhang *et al.* [34] relight indoor spaces for realistic refurnishing and Meilland *et al.* [35] apply dense visual SLAM to relight virtual objects. Exploiting relighting for visual localization has received less attention in the literature. One approach is presented by Whelan *et al.* [36] who introduce a method for light source detection into their SLAM system. They discuss the benefits for pose estimation but eventually improve it by masking out specular reflections instead of explicitly exploiting the lighting predictions.

Kim *et al.* [37] present an approach to visual localization that builds multiple maps for various lighting conditions. The approach recognizes the illumination level and selects an appropriate map for localization. In contrast to our method, it uses a sparse feature-based map representation, stores multiple maps instead of adapting one parametric model, and assumes that illumination causes uniform image changes, i.e., does not consider complex lighting effects.

III. Proposed Method

This paper proposes a method to exploit lighting effects for camera tracking in indoor environments. The three main contributions are described in the following subsections. First, we present a lighting adaptable map representation for indoor environments. Second, we propose a method to estimate the light setting present in the scene from the current camera observation. Third, we leverage these components to exploit lighting effects in a direct dense camera tracking approach. Figure 1 illustrates the principle of our method.

A. Lighting Adaptable Maps

The proposed map representation builds on our approach for reflectance mapping [1] which reconstructs the surface geometry of the scene as a triangular mesh defined on a set of vertices $v_i \in V$. Based on the measured radiosity $B(v_i)$ of the scene, we compute the irradiance $H(v_i)$ and obtain the diffuse reflectance $\rho(v_i) = B(v_i)/H(v_i)$. The quantities are related by the radiosity equation [38]

$$B(v_i) = \rho(v_i) \sum_{j \neq i} B(v_j) F(v_i, v_j) G(v_i, v_j) \tag{1}$$

where $F(v_i, v_j)$ is the form factor describing the geometrical relationship between v_i and v_j based on distance and surface normals. $G(v_i, v_j) \in \{0, 1\}$ determines whether the line of sight between v_i and v_j is blocked (by using ray tracing).

Lighting adaptable maps represent the global illumination contributions of the lamps in the scene. To segment and index the lamps $l \in \mathcal{L} = \{0, 1, \ldots, L-1\}$, we employ the method proposed in [1] which clusters subsets of vertices $V_l \subset V$ with high radiosity. Therefore, a lamp must be on during data acquisition to be segmented. However, as we assume static geometry, lamp positions do not change and missing lamps can be added to the map when switched on later.

The essential capability of our map representation is to predict the scene radiosity $\hat{B}_{\mathcal{L}_{on}}(v_i)$ for a given light setting $\mathcal{L}_{on} \subseteq \mathcal{L}$ that defines which lamps are on. We precompute the radiosity contributions of individual lamps $\hat{B}_l(v_i)$, store them in the map, and exploit the linearity of the radiosity equation to later compose them in real-time:

$$\hat{B}_{\mathcal{L}_{on}}(v_i) = \sum_{l \in \mathcal{L}_{on}} \hat{B}_l(v_i) \tag{2}$$

The individual lamp contribution $\hat{B}_l(v_i)$ can be computed by solving Equation 1 using $\rho(v_i)$ from the reflectance map and the measured radiosity $B(v_i)$ for vertices of lamp l. Although it is theoretically possible to solve the remaining linear equation system analytically, it is infeasible in practice due to the high number of vertices. Therefore, we compute an approximate solution by iteratively evaluating Equation 1

$$\hat{B}_l^{k+1}(v_i) = \rho(v_i) \sum_{j \neq i} \hat{B}_l^k(v_j) F(v_i, v_j) G(v_i, v_j) \tag{3}$$

where $\hat{B}_l^k(v_i)$ is the radiosity after k iterations (bounces). We initialize this iterative evaluation with

$$\hat{B}_l^0(v_i) = \begin{cases} B(v_i) & v_i \in V_l \\ 0 & \text{otherwise.} \end{cases}$$

In our experiments we found that $K = 10$ iterations provide a sufficiently accurate approximation. To simplify notation, we imply $k = K$ if the dependency on k is omitted. Figure 2 illustrates how the predicted radiosity for different light settings $\hat{B}_{\mathcal{L}_{on}}^k$ depends on the number of bounces and compares to the measured reference radiosity $B_{\mathcal{L}_{on}}$.

In order to point out the benefits of our map representation, consider a naive approach to lighting adaptable mapping that creates a new radiosity map for each encountered light setting. For L lamps, such an approach would require to store 2^L radiosity values for each vertex and need at least the same number of mapping runs to observe all light settings. In contrast, we store only L radiosity components per vertex and just need a single mapping run if all lamps are on.

B. Light Setting Estimation

A lighting adaptable map is a parametric model that allows to predict the scene radiosity depending on its parameters called a light setting. To compare observations of the real-world to the map, it is required to determine whether lamps are on or off in reality and to adapt the map accordingly. In the following we propose a method for light setting estimation based on a single color image.

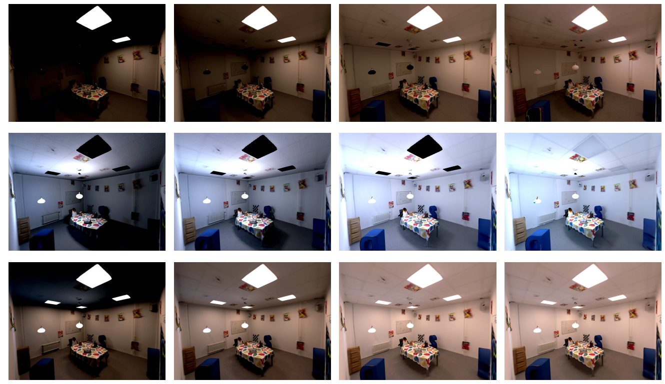

Fig. 2: Predicted radiosity $\hat{B}_{\mathcal{L}_{on}}^k$ for $k = 0, 1, 10$ bounces compared to the measured radiosity $B_{\mathcal{L}_{on}}$ (left to right) for the light settings LS03, LS48, LS63 (top to bottom) rendered with a virtual exposure of 30 ms, 300 ms, 30 ms, respectively.

As all parameters of our model are binary variables, a light setting $\mathcal{L}_{on}$ can also be expressed as a vector $\boldsymbol{x} \in \{0, 1\}^L$ with

$$x_l = \begin{cases} 1 & l \in \mathcal{L}_{on} \\ 0 & \text{otherwise.} \end{cases}$$

To find the light setting $\boldsymbol{x}^*$ that best explains the current color image $\boldsymbol{I}_C$, we want to exploit the superposition in Equation 2 and thus transform $\boldsymbol{I}_C$ to its corresponding radiosity image $\boldsymbol{I}_B = f^{-1}(\boldsymbol{I}_C)/(c \cdot \Delta t)$ using the inverse camera response function f^{-1} obtained by radiometric calibration [39], [40] as well as the gain c and exposure time Δt used to capture the image. Given the current camera pose estimate, we can obtain a rendering of the map $\hat{\boldsymbol{I}}_{\hat{B}_{\boldsymbol{x}}}$ illuminated by an arbitrary light setting $\boldsymbol{x}$ and compare it with the real image $\boldsymbol{I}_B$. More specifically, we find the light setting $\boldsymbol{x}$ that minimizes the sum of squared errors between the measured radiosity B and its predicted counterpart $\hat{B}_{\boldsymbol{x}}$ normalized on the map reflectance ρ over all pixels $\boldsymbol{u}$ in the image domain Ω:

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \sum_{\boldsymbol{u} \in \Omega} \| \big(\boldsymbol{I}_B(\boldsymbol{u}) - \hat{\boldsymbol{I}}_{\hat{B}_{\boldsymbol{x}}}(\boldsymbol{u})\big) / \hat{\boldsymbol{I}}_\rho(\boldsymbol{u}) \|_2 \tag{4}$$

The error for a specific $\boldsymbol{x}$ can be written as

$$\begin{aligned} e_{\boldsymbol{x}} &= \boldsymbol{x}^T \boldsymbol{A}^T \boldsymbol{A} \boldsymbol{x} - 2 \boldsymbol{x}^T \boldsymbol{A}^T \boldsymbol{b} + \boldsymbol{b}^T \boldsymbol{b} \\ \boldsymbol{A} &= \Big[\text{stack}(\hat{\boldsymbol{I}}_{\hat{B}_0}) \ldots \text{stack}(\hat{\boldsymbol{I}}_{\hat{B}_{L-1}}) \Big] * \boldsymbol{w} \\ \boldsymbol{b} &= \text{stack}(\boldsymbol{I}_B) * \boldsymbol{w} \\ \boldsymbol{w} &= 1/\text{stack}(\hat{\boldsymbol{I}}_\rho) \end{aligned} \tag{5}$$

where the stack() operator stacks all image pixels, in this case for each color channel, into a vector. The multiplication of $\boldsymbol{w}$ is meant row-wise, the devision component-wise. We efficiently build the components $\boldsymbol{A}^T\boldsymbol{A}$ $(L \times L)$, $\boldsymbol{A}^T\boldsymbol{b}$ $(L \times 1)$, and $\boldsymbol{b}^T\boldsymbol{b}$ (1×1) in parallel on the GPU and only perform the evaluation of $e_{\boldsymbol{x}}$ on the CPU. As this evaluation is extremely fast (approximately 1 ns), we can afford to solve Equation 4 brute-force. For up to $L = 20$ lamps ($2^L \approx 10^6$) the required computation time is less than 1 ms and dominated by the time needed to build the components.

C. Camera Tracking

Camera tracking is the problem of estimating the camera pose $\boldsymbol{T} \in \mathrm{SE}(3)$ for every time t the camera provides a frame $\boldsymbol{I}_t$. Using the terms from [41], *frame-to-frame* tracking yields an odometry solution prone to drift for the global pose $\boldsymbol{T}_{t,0} = \boldsymbol{T}_{t,t-1} \ldots \boldsymbol{T}_{1,0}$, while *frame-to-model* tracking, at least in a previously built model, does not suffer from these effects due to its reference to the global map. As the names suggest, the former finds $\boldsymbol{T}_{t,t-1}$ by comparing $\boldsymbol{I}_t$ to $\boldsymbol{I}_{t-1}$ whereas the latter compares $\boldsymbol{I}_t$ to a rendering of the model $\hat{\boldsymbol{I}}$. Given two input frames, there is no difference between both tracking variants, so in the following we only describe the terms for frame-to-model tracking as the ones for frame-to-frame tracking can easily be obtained by replacing the rendered quantities with the ones from the last frame.

Our direct dense camera tracking approach can work on color (or gray-scale) images by utilizing the depth from the model to apply projective data association

$$\tilde{\boldsymbol{u}} = \pi(\boldsymbol{K}\boldsymbol{T}_{t,t-1}\hat{\boldsymbol{V}}(\boldsymbol{u})) \tag{6}$$

where π is the perspective projection function, $\boldsymbol{K}$ the camera matrix, and $\hat{\boldsymbol{V}}$ the vertex map created using the depth image $\hat{\boldsymbol{I}}_D$ rendered from the model. As proposed by [41], we embed the data association optimization loop in a coarse-to-fine approach using three image pyramid levels.

In case a measured depth image $\boldsymbol{I}_D$ is provided, we can use its geometric information in addition to the radiometric information contained in the color image $\boldsymbol{I}_C$. We estimate the camera pose

$$\boldsymbol{T}^*_{t,t-1} = \underset{\boldsymbol{T}_{t,t-1}}{\arg\min}\, E_C + w_G E_G \tag{7}$$

with $w_G = 0$ if only the color error is used and $w_G = 10$ if the geometric error is added. The weighting is realized as described by [36], which we also follow to efficiently solve the pose estimation problem on the GPU. The geometric error term

$$E_G = \sum_{\boldsymbol{u}\in\Omega} \|\left(\boldsymbol{T}_{t-1,t}\boldsymbol{V}(\tilde{\boldsymbol{u}}) - \hat{\boldsymbol{V}}(\boldsymbol{u})\right)^T \hat{\boldsymbol{N}}(\boldsymbol{u})\|_2 \tag{8}$$

uses a point-to-plane metric. Its Jacobians are left out for brevity here as they can be found in [41]. The color error term

$$E_C = \sum_{\boldsymbol{u}\in\Omega} \|\boldsymbol{I}_C(\tilde{\boldsymbol{u}}) - \hat{\boldsymbol{I}}_{\hat{C}}(\boldsymbol{u})\|_2 \tag{9}$$

uses image warping as described by [42].

The core idea to leverage our map representation and light setting estimation for direct dense camera tracking is to adapt the rendered color image $\hat{\boldsymbol{I}}_{\hat{C}} = f(\hat{\boldsymbol{I}}_{\hat{B}_{\boldsymbol{x}^*}} \cdot c \cdot \Delta t)$ to the lighting conditions currently present in the scene. We perform a single frame-to-frame tracking step and use the previous global pose to obtain an approximate pose estimate $\boldsymbol{T}^{\approx}_{t,0} = \boldsymbol{T}_{t,t-1}\boldsymbol{T}_{t-1,0}$ for the light setting estimation and as initial estimate for the model tracking of the global pose $\boldsymbol{T}_{t,0}$.

Fig. 3: Maps for the datasets DS1 (left) and DS2 (right) used in the experimental evaluation. The black indices and red segmentations illustrate the lamps contained in the models.

Decimal	Binary (LSB $= x_0$, MSB $= x_5$)						Set
LS	x_0	x_1	x_2	x_3	x_4	x_5	$\mathcal{L}_{on}$
00	0	0	0	0	0	0	{}
01	1	0	0	0	0	0	{0}
02	0	1	0	0	0	0	{1}
03	1	1	0	0	0	0	{0, 1}
...			...				...
12	0	0	1	1	0	0	{2, 3}
15	1	1	1	1	0	0	{0, 1, 2, 3}
16	0	0	0	0	1	0	{4}
28	0	0	1	1	1	0	{2, 3, 4}
32	0	0	0	0	0	1	{5}
35	1	1	0	0	0	1	{0, 1, 5}
48	0	0	0	0	1	1	{4, 5}
51	1	1	0	0	1	1	{0, 1, 4, 5}
60	0	0	1	1	1	1	{2, 3, 4, 5}
...			...				...
63	1	1	1	1	1	1	{0, 1, 2, 3, 4, 5}

TABLE I: Light Setting Notations

IV. Experimental Evaluation

In this section we present experiments to evaluate the three main contributions of this paper. After introducing the datasets, we show the lighting prediction capabilities of the proposed map representation, discuss the accuracy of our light setting estimation method, and finally investigate the benefits of those for a direct dense camera tracking approach.

A. Datasets

We recorded two datasets in a conference room of our lab.[1] As can be seen in Figure 3, the geometry in the room ($4.8\,\mathrm{m} \times 4.5\,\mathrm{m} \times 2.9\,\mathrm{m}$) alters only slightly between the datasets but DS1 has significantly more texture than DS2. The reconstructed meshes have a resolution of $5\,\mathrm{mm}$, which results in 3,786,298 vertices and 7,570,727 faces for DS1 and 4,287,363 vertices and 8,568,480 faces for DS2. For a fair comparison in the pose tracking experiments, we intended to use poses from ORB-SLAM2 [11] to build these maps. However, due to its performance, this was only feasible for the feature-rich dataset DS1 and we had to use ground-truth poses for DS2. To obtain ground-truth poses, we employed a VIVE tracking system with four base stations placed in the room ceiling corners and rigidly attached an extrinsically calibrated VIVE tracker to our ASUS Xtion Pro Live RGB-D camera using a 3D-printed mounting.

[1] The datasets used for the experimental evaluation of our method are available at: `http://tracklam.informatik.uni-freiburg.de`

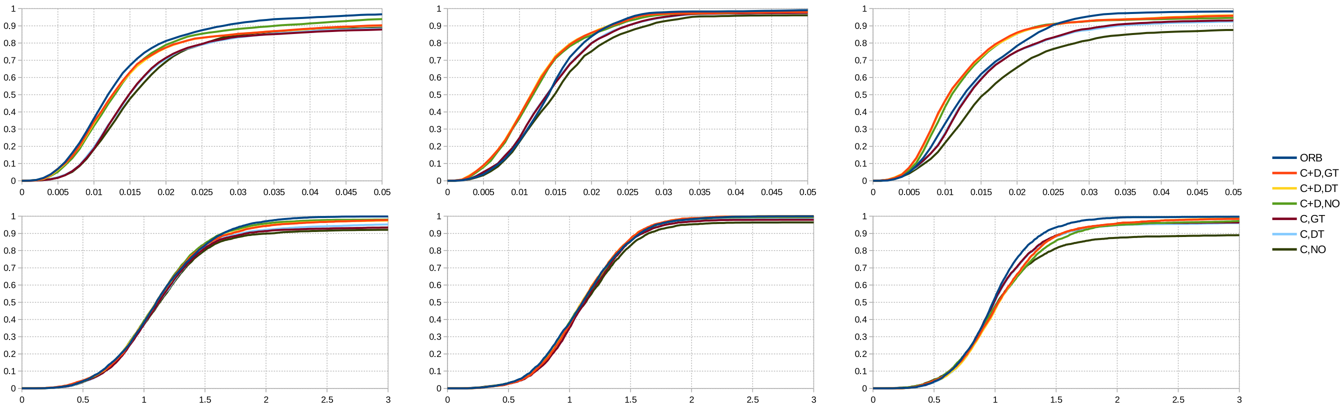

Fig. 4: Fraction of poses as a function of the translational $[\mathrm{m}]$ (top) and the rotational $[°]$ (bottom) tracking error for the trajectories DS1.T1, DS1.T2, and DS1.T3 (left to right). All methods show similar performance on the texture-rich DS1.

The conference room contains four area ceiling lights by default and we added two additional light bulbs to increase the complexity of the datasets. Figure 3 shows how these lamps got segmented and indexed in our lighting adaptable map representation. In reality, the ceiling lamps 0 and 1 as well as 2 and 3 can only be switched on and off as pairs while the light bulbs 4 and 5 can be controlled individually. As we do not provide this information to our method, it has to consider $2^L = 64$ possible light settings even though not all are actually contained in the datasets. Light setting notations are provided for convenient look-up in Table I.

For both datasets we recorded three trajectories under varying lighting conditions as can be seen in Table II. All sequences use fixed camera exposure times and gains suited for the light settings contained in the trajectory. Equal trajectory numbers contain similar lighting in both datasets. To provide ground-truth light settings, we noted down lamp switching events during recording and later labeled the exact point in time by looking at the trajectories frame by frame.

Trajectory	Frames	c	Δt	Light Settings
DS1.T1	5248	3	25 ms	12, 15, 03, 15, 12
DS1.T2	3078	4	25 ms	12, 28, 60, 48
DS1.T3	3595	10	30 ms	48, 32, 48, 16, 48, 32, 48, 16, 48
DS2.T1	4541	2	20 ms	60, 28, 12, 15, 03, 35, 51
DS2.T2	6458	4	25 ms	51, 48, 60, 28, 12, 28, 60, 48, 51
DS2.T3	6058	10	30 ms	48, 16, 48, 32, 48, 16

TABLE II: Trajectories with Varying Lighting Conditions

B. Lighting Adaptable Maps

To demonstrate the advantages of our map representation, we recorded the maps when all lamps in the scene were on (LS63). In this case, we can directly build a lighting adaptable map that allows to predict the scene radiosity for arbitrary light settings $\mathcal{L}_{on}$. To evaluate the accuracy of this prediction, we additionally captured radiosity maps of DS1 for the light settings LS03 and LS48 as a reference. We compare the radiosity predicted by our model $\hat{B}_{\mathcal{L}_{on}}$ with the

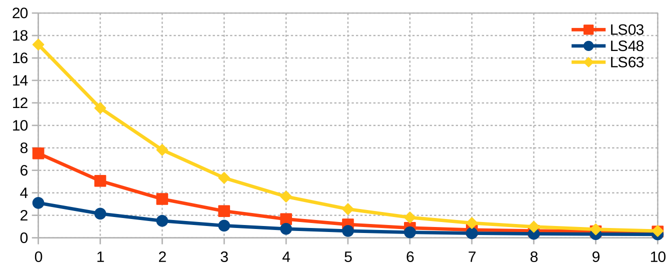

Fig. 5: Radiosity error $E^k_{\mathcal{L}_{on}}$ as a function of the number of bounces k evaluated for different light settings $\mathcal{L}_{on}$ on DS1.

measured reference $B_{\mathcal{L}_{on}}$ and compute the error

$$E^k_{\mathcal{L}_{on}} = \frac{1}{|V|} \sum_{v_i \in V} |\hat{B}^k_{\mathcal{L}_{on}}(v_i) - B_{\mathcal{L}_{on}}(v_i)| \qquad (10)$$

depending on the number of bounces k. As the measured reference map might slightly vary from our model, we do not use vertex samples that differ more than $10\,\%$ in reflectance. As can be seen in Figure 5, $k = 10$ bounces are sufficient to propagate the light in the scene. Figure 2 illustrates the dependency on the bounces and shows how well the radiosity is predicted by our map representation compared to the measured reference.

C. Light Setting Estimation

The light setting estimation is a crucial part of our method if no other means to obtain this information are available. As light setting estimation is a binary classification problem for each lamp, we evaluate it using the accuracy

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \qquad (11)$$

and report results in Table III. In order to obtain independent light setting estimates for this evaluation, we do not use the estimates for camera tracking and instead obtain the required poses using the ground-truth light settings (C+D,GT).

When interpreting the results, it should be considered that the estimation problem might not be sufficiently constrained in certain situations. The results show that the accuracies

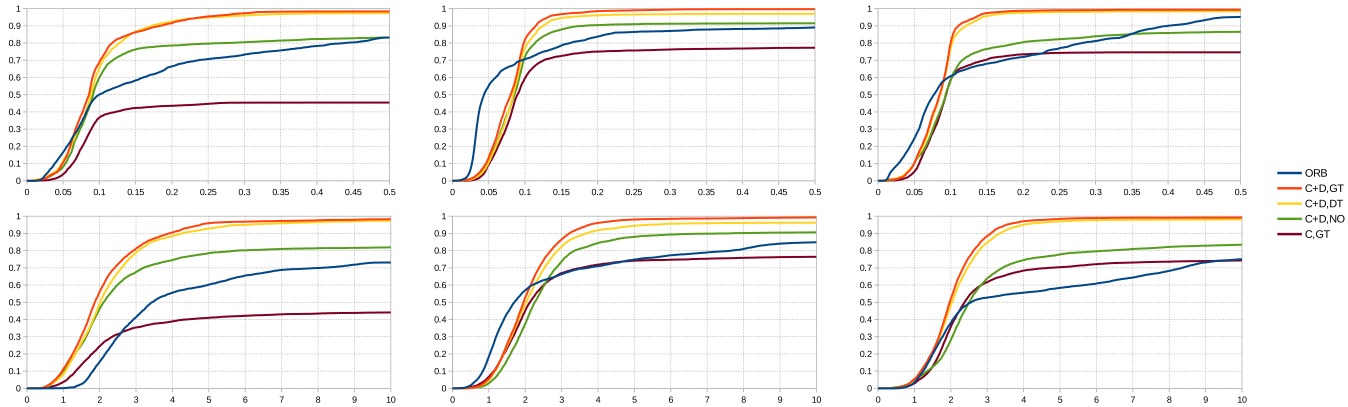

Fig. 6: Fraction of poses as a function of the translational $[\mathrm{m}]$ (top) and the rotational $[°]$ (bottom) tracking error for the trajectories DS2.T1, DS2.T2, and DS2.T3 (left to right). On the texture-poor DS2, exploiting gradients caused by lighting improves camera tracking as shown by the superior performance of C+D,GT and C+D,DT compared to C+D,NO.

	Lamp 0	Lamp 1	Lamp 2	Lamp 3	Lamp 4	Lamp 5	All
DS1.T1	97.80 %	97.25 %	93.08 %	94.93 %	64.67 %	32.40 %	76.47 %
DS1.T2	91.90 %	94.08 %	93.40 %	93.33 %	97.46 %	62.59 %	88.17 %
DS1.T3	100.00 %	100.00 %	100.00 %	99.33 %	99.36 %	94.38 %	98.61 %
DS2.T1	71.41 %	71.88 %	83.61 %	81.51 %	60.34 %	70.85 %	73.64 %
DS2.T2	87.33 %	87.78 %	91.31 %	87.75 %	85.96 %	62.45 %	83.05 %
DS2.T3	99.30 %	99.50 %	99.92 %	99.59 %	97.34 %	90.80 %	97.43 %
All	91.32 %	91.67 %	93.54 %	92.59 %	83.39 %	68.21 %	85.88 %

TABLE III: Accuracy of Light Setting Estimation

for the brighter ceiling lamps 0 to 3 are significantly higher than those for the medium bright light bulb 4 and the even less bright light bulb 5. This is not surprising, as the color images are dominated by the illumination of the ceiling lamps, especially in the trajectories T1 and to a lesser extent in T2. In contrast, the accuracy for the bulbs is comparatively high in the T3 trajectories where the ceiling lamps remain off. On the bright side, estimation errors due to missing constraints will likely not cause severe problems for camera tracking, as the falsely predicted light setting will still result in a rendering that is close to the one for the correct setting. We claim that the achieved accuracy is sufficient to employ our light setting estimation for camera tracking.

D. Camera Tracking

As the goal is to evaluate camera tracking performance and our approach does not include any means for camera relocalization, we (re)initialize the camera with ground-truth poses. To compute the tracking errors we align poses to the respective map coordinate system using the method of Horn [43] as proposed by [44]. We evaluate the translational and rotational tracking errors for variants of our method that use the color error term only (C) and variants that additionally use the geometric error term (C+D). Moreover, we distinguish between variants that use ground-truth light settings (GT), estimate the light settings (DT), and do no lighting adaption (NO). NO uses a rendered color image for LS63 whose mean radiosity is equalized to fit the one of the measured color image. We also provide a comparison to the RGB-D ORB-SLAM2 localization mode (ORB) which performs pose tracking on a map built using LS63.

Figure 4 shows the tracking errors for dataset DS1. All methods perform well since the map is rich in reflectance-based gradients. ORB slightly outperforms our approaches in terms of robustness, i.e., reaches closer to 1 as it tracks more poses successfully as does C+D compared to C. There is also a tendency that methods that do no lighting adaption (NO and ORB) perform better on lower trajectory numbers which contain light settings more similar to LS63.

Figure 6 reports the results for DS2. We do not use all variants on this dataset as it is too challenging for color only tracking as can be seen from the poor performances of C,GT. Also ORB performs significantly worse than on DS1 as it has to rely on fewer reflectance gradients that can provide robust features. One of the interesting insights from Figure 6 is the similar performance of C+D,GT and C+D,DT which shows that using our light setting estimation method leads to camera tracking results comparable to those when using ground-truth labels. Even more importantly, they clearly outperform C+D,NO showing that exploiting lighting-based gradients is beneficial for accurate and robust camera tracking.

E. Runtime Performance

We implemented our method efficiently to achieve real-time performance on modern hardware. Camera tracking including light setting estimation runs with approximately 60 fps using an i7-4790K CPU and a GTX 2080Ti GPU.

V. Conclusions

In this paper, we presented a lighting adaptable map representation for indoor environments and a light setting estimation method that uses a single color image to determine which lamps in the scene are currently on. We leverage these capabilities in a direct dense camera tracking approach by matching the camera observations against renderings of the correspondingly adapted map. We evaluated the proposed approach in real-world experiments in scenes with varying lighting conditions and showed that our method exploits the effects of lighting, which is especially beneficial for camera tracking in environments with few texture-based gradients.

References

[1] M. Krawez, T. Caselitz, D. Büscher, M. Van Loock, and W. Burgard, "Building dense reflectance maps of indoor environments using an rgb-d camera," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.

[3] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *International Conference on Computer Vision (ICCV)*, 2011.

[4] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *European Conference on Computer Vision (ECCV)*, 2010.

[5] D. Stavens and S. Thrun, "Unsupervised learning of invariant features using video," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[7] N. Carlevaris-Bianco and R. M. Eustice, "Learning visual feature descriptors for dynamic lighting conditions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[8] A. Ranganathan, S. Matsumoto, and D. Ilstrup, "Towards illumination invariance for visual localization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[9] A. Mikulík, M. Perdoch, O. Chum, and J. Matas, "Learning a fine vocabulary," in *European Conference on Computer Vision (ECCV)*, 2010.

[10] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.

[11] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics (T-RO)*, vol. 33, no. 5, pp. 1255–1262, 2017.

[12] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[13] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 40, no. 3, pp. 611–625, 2017.

[14] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision (ECCV)*, 2014.

[15] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *International Conference on Computer Vision (ICCV)*, 2011.

[16] M. Irani and P. Anandan, "About direct methods," in *International Workshop on Vision Algorithms (IWVA)*, 1999.

[17] T. Gonçalves and A. I. Comport, "Real-time direct tracking of color images in the presence of illumination variation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[18] S. Klose, P. Heise, and A. Knoll, "Efficient compositional approaches for real-time robust direct visual odometry from rgb-d data," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[19] J. Engel, J. Stückler, and D. Cremers, "Large-scale direct slam with stereo cameras," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[20] P. Kim, H. Lim, and H. J. Kim, "Robust visual odometry to irregular illumination changes with rgb-d camera," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[21] S. Park, T. Schöps, and M. Pollefeys, "Illumination change robustness in direct visual slam," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[22] L. Clement and J. Kelly, "How to train a cat: Learning canonical appearance transformations for direct visual localization under illumination change," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 3, pp. 2447–2454, 2018.

[23] P. Corke, R. Paul, W. Churchill, and P. Newman, "Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localisation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[24] G. D. Finlayson, S. D. Hordley, C. Lu, and M. S. Drew, "On the removal of shadows from images," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 28, no. 1, pp. 59–68, 2005.

[25] W. Maddern, A. Stewart, C. McManus, B. Upcroft, W. Churchill, and P. Newman, "Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles," in *Visual Place Recognition in Changing Environments Workshop, IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[26] C. Kerl, M. Souiai, J. Sturm, and D. Cremers, "Towards illumination-invariant 3d reconstruction using tof rgb-d cameras," in *International Conference on 3D Vision (3DV)*, 2014.

[27] H. Barrow, J. Tenenbaum, A. Hanson, and E. Riseman, "Recovering intrinsic scene characteristics," *Computer Vision Systems*, vol. 2, no. 3-26, p. 2, 1978.

[28] K. Kim, J. Gu, S. Tyree, P. Molchanov, M. Nießner, and J. Kautz, "A lightweight approach for on-the-fly reflectance estimation," in *International Conference on Computer Vision (ICCV)*, 2017.

[29] A. Meka, M. Maximov, M. Zollhoefer, A. Chatterjee, H.-P. Seidel, C. Richardt, and C. Theobalt, "Lime: Live intrinsic material estimation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[30] Y. Yu, P. Debevec, J. Malik, and T. Hawkins, "Inverse global illumination: Recovering reflectance models of real scenes from photographs," in *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1999.

[31] D. Azinovic, T.-M. Li, A. Kaplanyan, and M. Nießner, "Inverse path tracing for joint material and lighting estimation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[32] M. Kasper and C. Heckman, "Multiple point light estimation from low-quality 3d reconstructions," in *International Conference on 3D Vision (3DV)*, 2019.

[33] S. Song and T. Funkhouser, "Neural illumination: Lighting prediction for indoor environments," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[34] E. Zhang, M. F. Cohen, and B. Curless, "Emptying, refurnishing, and relighting indoor spaces," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, pp. 1–14, 2016.

[35] M. Meilland, C. Barat, and A. Comport, "3d high dynamic range dense visual slam and its application to real-time object re-lighting," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2013.

[36] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "Elasticfusion: Real-time dense slam and light source estimation," *International Journal of Robotics Research (IJRR)*, vol. 35, no. 14, pp. 1697–1716, 2016.

[37] P. Kim, B. Coltin, O. Alexandrov, and H. J. Kim, "Robust visual localization in changing lighting conditions," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[38] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile, "Modeling the interaction of light between diffuse surfaces," *ACM SIGGRAPH Computer Graphics*, vol. 18, no. 3, pp. 213–222, 1984.

[39] P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," in *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1997.

[40] S. V. Alexandrov, J. Prankl, M. Zillich, and M. Vincze, "Calibration and correction of vignetting effects with an application to 3d mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[41] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[42] F. Steinbrücker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense rgb-d images," in *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011.

[43] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A (JOSA A)*, vol. 4, no. 4, pp. 629–642, 1987.

[44] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.