2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Efficient Compositional Approaches for Real-Time Robust Direct Visual Odometry from RGB-D Data

Sebastian Klose[1], Philipp Heise[1] and Alois Knoll[1]

Abstract— **In this paper we give an evaluation of different methods for computing frame-to-frame motion estimates for a moving RGB-D sensor, by means of aligning two images using photometric error minimization. These kind of algorithms have recently shown to be very accurate and robust and therefore provide an attractive solution for robot ego-motion estimation and navigation. We demonstrate three different alignment strategies, namely the Forward-Compositional, the Inverse-Compositional and the Efficient Second-Order Minimization approach, in a general robust estimation framework. We further show how estimating global affine illumination changes, in general improves the performance of the algorithms. We compare our results with recently published work, considered as state-of-the art in this field, and show that our solutions are in general more precise and can perform in real-time on standard hardware.**

I. INTRODUCTION

Computing relative motion between consecutive image frames of a moving camera is called visual odometry [1]. These class of algorithms, usually form the basis for full Simultaneous Localization and Mapping Systems (VSLAM). In the last few years, the development of low cost RGB-D sensors, such as the Microsoft Kinect or the Asus Xtion Pro Live, have shifted the focus from feature-based systems to dense direct systems. In this work, we also concentrate on RGB-D data. These sensors deliver a standard RGB color image and a more or less dense depth map at the same time. Feature-based visual odometry systems, perform motion estimation by tracking salient feature points over time, to build up point-correspondences between different camera views, which are in turn used usually within a RANSAC and Pose-from-Point (PnP) scheme. Recently, there have been several works which compute the relative motion between two frames by so called direct methods. In this case, the motion estimation is formulated as an image registration problem depending *directly* on the intensity values of two or more images. The general setup for this is depicted in Figure 1. It has been shown, that this approach provides good precision and can be computed in real-time not only by using graphics hardware. The basic idea of these approaches, adopts the formulation of the well known Lucas-Kanade approach for image registration [2]. A nice and extensive review of this framework can be found in the series of Baker and Matthews [3]. While most methods rely on the Forward Compositional approach for image alignment, we will evaluate the performance of three different alignment algorithms in this work: Forward Compositional (FC), Inverse Compositional (IC) and Efficient Second order Minimization (ESM). Furthermore, we demonstrate that estimating global affine illumination changes provides improved precision and how to add a regularization term to the formulation. We also adopt the idea of selecting the most useful information from the reference image, in a similar way as described in [4] and [5]. In the evaluation Section V, we will give an extensive comparison of different combinations of our methods and evaluate the performance of our implementation using the publicly available datasets presented in [6], [7]. The same dataset has also been used for evaluation within several related work [8], [9], which enables us to give a comparison of the results. The contributions of this paper are as follows:

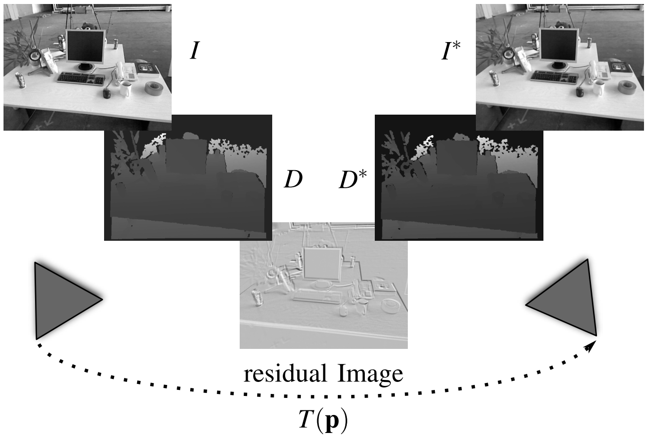

Fig. 1. Relative motion between two time frames: For each instance, we obtain a grayscale image and a corresponding depth map. The goal is to estimate the relative 6-DOF transformation $T(\mathbf{p})$ between a reference frame I^* and the current frame I by aligning the images, such that the error in the residual image is minimized.

- a unifying modular framework for direct visual odometry algorithms
- performance comparison of different alignment formulations
- influence of affine illumination estimation
- inclusion of a regularization term
- a robust real-time open-source system[1], runnable on standard PC hardware

[1] S. Klose, P. Heise and A. Knoll are with the Chair for Robotics and Embedded Systems, Department of Informatics, Technische Universität München, 85748 Garching, Germany {kloses, heise, knoll} at in.tum.de

[1] http://www6.in.tum.de/Main/ResearchDirectVO

978-1-4673-6357-0/13/$31.00 ©2013 IEEE

II. RELATED WORK

Visual motion estimation is a very active field and strongly driven by the needs of autonomous robotics. The appearance of affordable RGB-D sensors, like the Asus Xtion Pro Live or the Microsoft Kinect, has lead to new types of algorithms, efficiently making use of the dense depth data available from these sensors. Before, feature-based algorithms have been mostly applied in SLAM systems. In this scenario, PTAM [10] has been widely adopted and successfully applied in several robotic systems for robot motion estimation and mapping with a single camera. Feature-based systems using stereo are for example described by Konolige et al. [11] and Mei et al. [12]. RGB-D sensors in combination with features and ICP for the motion estimation are used by Henry et. al [13] and Engelhard et al. [14]. Endres et al. [15] evaluated the performance of different feature detectors and descriptors within the RGB-D SLAM system. For comparison of the different approaches they use the public dataset of Sturm et al. [6], [7]. This dataset incorporates precise ground truth pose estimates at 100Hz, obtained by an external infrared tracking system. Furthermore, the dataset also includes python based evaluation code.

In this work, so called direct pixel-based methods are for visual odometry estimation. Similar work has been done by Newcombe et al. [16], where a forward-compositional formulation for the motion of a single camera is used. In their work a dense 3D model of the world is computed from monocular camera input. After a bootstrapping phase applying a feature-based method adopted from PTAM [10], the following camera motions are estimated by aligning the current frame with a virtual reference frame generated by re-projection of the reconstructed dense 3D scene model. The KinectFusion [17], [18] algorithm uses a similar approach for the scene representation, but is designed to work with RGB-D sensors. In KinectFusion, the camera pose is computed using ICP based on the depth maps. The Kintinuous algorithm described by Whelan et al. [19], describes a method to spatially extend the operation space of KinectFusion, wich was originally limited to small workspaces. These systems show impressive performance in terms of accuracy and robustness. A drawback of these approaches is their dependence on graphics hardware to enable real-time processing, which is currently not yet available on many robotic systems. Meilland et al. [20], use an offline created map of spherical reference frames. The motion is also estimated by minimizing the photometric error of the projection of the current image sphere onto the active (closest) reference sphere. In another system [21], Meilland also shows how to incorporate illumination changes within a direct motion estimation system based on stereo input. A hybrid camera tracking approach, mixing relative motion estimation against an offline model, with online visual odometry using the geometry of the model and the intensities of the previous online image has been described by Comport et al. [22], as an extension of their previous work. Another system proposed by the same authors [23], uses constraints of quadrifocal geometry within a stereo-setup in conjunction with direct motion estimation. The system described by Audras et al.[5] also operates on RGB-D images. Their alignment approach uses an Inverse-Compositional formulation and uses robust estimation techniques to improve the robustness of the system against outliers. A further extension of this work is given by Tykkala et al [24], where additionally consistency of the depth-map is enforced by formulating a bi-objective cost-function. The system of Steinbruecker [25], also uses a forward compositional approach to direct visual odometry from a RGB-D sensor. Their system is able to run in real-time at 15Hz on a standard CPU. An extension of their system is shown by Kerl et al.[8] where improved performance is achieved by incorporating a robust sensor model learned from real data. Furthermore, the authors give a formulation of how to use motion priors in their system and show evaluations against the benchmark dataset of Sturm et al.[7]. Moreover their system is available as open-source implementation. Different combinations of approaches based on their previous Kintinuous work [19], a GPU implementation of the work of Steinbruecker [25] and the FOVIS system of Huang et al.[26] has been developed by Whelan et al.[9]. Again, for performance evaluation, the same benchmark dataset has been used. Our system is in principle very close to the ones described by Kerl et al.[8], Comport et al. [23] and Meilland et al. [21]. We evaluate different compositional formulations, show how to model global affine illumination changes and how to incorporate a regularization term. Since our system is targeted towards RGB-D sensors, we use same benchmark dataset to give a comparison of our system with the results shown by Whelan [9] and Kerl [8].

III. DIRECT MOTION ESTIMATION

In this section we will describe the direct motion estimation framework including the three different formulations we use within this paper.

Similar as in [8], [9], [5], we formulate the motion estimation process as a minimization problem of a photometric cost function. The general problem is shown in Figure 1. We want to estimate the 6-dof motion $T(\mathbf{p})$ between a reference frame and the current frame, each given in terms of an intensity I^* and depth image D^*. As we only use the intensity information in this work, we convert the RGB images to grayscale first. In our nomenclature, we refer to I^* as the reference frame and I as the current frame and use the sum of squared pixel differences as similarity metric to model the costs:

$$\underset{\mathbf{p}}{\operatorname{argmin}} \| I^* - I(\omega(\mathbf{p}, \mathbf{X})) \|^2 \tag{1}$$

Here we denote by I^*, the vector of pixel values corresponding to the reference frame and $I(\omega(\mathbf{p}, \mathbf{X}))$ the vector of pixel values, computed by re-projecting the 3D-points $\mathbf{X}$ from the reference frame onto the current frame using the pose estimate $\mathbf{p}$. This re-projection is modeled by a function $\omega(\mathbf{p}, \mathbf{X})$. The 3D points for the reference frame can be computed from the depth information stored in D^* of the reference frame and the calibration matrix $\mathbf{K}$ by

$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{K}^{-1} D^*(\mathbf{u}) \begin{bmatrix} \mathbf{u} \\ 1 \end{bmatrix} \quad \text{with} \quad \mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

with focal lengths f_x, f_y and the principal point of the camera at $[c_x, c_y]^T$. Here $\mathbf{u}$ is the pixel position in the reference frame corresponding to $\mathbf{X}$. The warp function ω is a combination of the underlying motion model and the projective transformation. The transformation between the two views is modeled by a rigid body motion in 3D and parametrized using the Lie algebra $\mathfrak{se}3$ of the special euclidean group $\mathbb{SE}3$. Therefore we have, $\mathbf{p} \in \mathfrak{se}3$, a six dimensional vector. For a given parameter vector $\mathbf{p}$, the corresponding 4×4-Transformation Matrix can be computed via the exponential map:

$$T(\mathbf{p}) = \exp\left(\sum_i p_i \mathbf{G}_i\right) \tag{3}$$

where p_i is the i-th component of the parameter-vector and $\mathbf{G}_i$ is the i-th generator of the $\mathbb{SE}3$. For the $\mathbb{SE}3$, there exists a closed form solution to compute the matrix exponential. For more information about Lie-algebras, the interested reader is referred to the book of Ma et. al [27]. The overall warp function is defined by

$$\omega(\mathbf{p}, \mathbf{X}) = \pi\left(T(\mathbf{p}) \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}\right) \quad \text{with } \pi(\mathbf{X}) = \begin{pmatrix} \frac{f_x \cdot X}{Z} - c_x \\ \frac{f_y \cdot Y}{Z} - c_y \end{pmatrix} \tag{4}$$

A. Forward Compositional Alignment

In the Forward Compositional formulation, the current estimate for the transformation is composed with an incremental transformation. Iterative updates are computed for those incremental parameters, until convergence is achieved. Therefore equation (1) becomes

$$\underset{\delta}{\operatorname{argmin}} \underbrace{\|I^* - I(\omega(\mathbf{p}, \omega(\delta, \mathbf{X})))\|^2}_{C(\delta)} \tag{5}$$

To compute an update step for the current costs, a linearization of equation (5) around $\delta = \mathbf{0}$ is computed by using a first order Taylor-Expansion:

$$C(\mathbf{0} \circ \delta) \approx C(\mathbf{0}) + \underbrace{\frac{\partial C(\delta)}{\partial \delta}}_{\mathbf{J}(\delta)}\bigg|_{\delta=\mathbf{0}} \delta \tag{6}$$

By deriving equation (6) with respect to δ and equating to zero, we can solve for the optimum update step δ for the current iteration as

$$\delta = -\left(\mathbf{J}(\mathbf{0})^T \mathbf{J}(\mathbf{0})\right)^{-1} \mathbf{J}(\mathbf{0})^T \left(I(\mathbf{p}) - I^*\right) \tag{7}$$

The Jacobian of the warp function in Equation (6) is computed via the chain rule as

$$J(\mathbf{0}) = \frac{\partial I(\omega(\mathbf{p}, \omega(\delta, \mathbf{X}))}{\partial \delta}\bigg|_{\delta=\mathbf{0}} \tag{8}$$

$$= \frac{\partial I(\mathbf{x})}{\partial \mathbf{x}}\bigg|_{\mathbf{x}=\omega(\mathbf{p},\mathbf{X})} \frac{\partial \omega(\delta, \mathbf{X})}{\partial \delta}\bigg|_{\delta=\mathbf{0}} \tag{9}$$

As the second part of equation (9), does not change, it can be precomputed for the reference frame. The first part of the Jacobian is the gradient of the current image, evaluated at the warped positions using the current parameters $\mathbf{p}$. After each iteration, the incremental transformation is composed onto the current one by applying the update rule

$$T(\mathbf{p}) \leftarrow T(\mathbf{p}) T(\delta) \tag{10}$$

B. Inverse Compositional Alignment

The inverse compositional approach uses incremental updates $T(\delta)$ in terms of the reference frame I^*. The formulation changes as follows:

$$\underset{\delta}{\operatorname{argmin}} \|I^*(\omega(\delta, \mathbf{X})) - I(\omega(\mathbf{p}, \mathbf{X}))\|^2 \tag{11}$$

By proceeding in a similar way as for the forward alignment, the minimizing step is computed as:

$$\delta = \left(\mathbf{J}_{\text{ic}}(\mathbf{0})^T \mathbf{J}_{\text{ic}}(\mathbf{0})\right)^{-1} \mathbf{J}_{\text{ic}}(\mathbf{0})^T \left(I(\mathbf{p}) - I^*\right) \tag{12}$$

with the Jacobian

$$\mathbf{J}_{\text{ic}}(\mathbf{0}) = \frac{\partial I^*(\mathbf{x})}{\partial \mathbf{x}}\bigg|_{\mathbf{x}=\omega(\mathbf{0},\mathbf{X})} \frac{\partial \omega(\delta, \mathbf{X})}{\partial \delta}\bigg|_{\delta=\mathbf{0}} \tag{13}$$

The advantage of the inverse compositional approach, is that the Jacobian $\mathbf{J}(\mathbf{0})_{\text{ic}}$ can be completely evaluated offline. The update rule for the current parameters now uses the inverse parameters by:

$$T(\mathbf{p}) \leftarrow T(-\delta) T(\mathbf{p}) \tag{14}$$

by using $T(-\delta) = T(\delta)^{-1}$ of our pose parametrization.

C. Efficient Second Order Minimization

The ESM algorithm was originally used for 2D image alignment in [28], [29]. The idea is to use a second order Taylor expansion of the cost function

$$C(\delta) \approx C(\mathbf{0}) + \mathbf{J}(\mathbf{0})\delta + \frac{1}{2}\delta^T \mathbf{H}(\mathbf{0})\delta \tag{15}$$

and apply another first order expansion for the Jacobian:

$$\mathbf{J}(\delta) \approx \mathbf{J}(\mathbf{0}) + \mathbf{H}(\mathbf{0})\delta \tag{16}$$

where $\mathbf{H}(\mathbf{0})$ is the Hessian matrix of the cost function. By plugging Equation 16 into Equation 15, we get

$$C(\delta)_{\text{esm}} \approx C(\mathbf{0}) + \mathbf{J}(\mathbf{0})\delta + \frac{1}{2}\left[\mathbf{J}(\delta) - \mathbf{J}(\mathbf{0})\right]\delta \tag{17}$$

$$= C(\mathbf{0}) + \frac{1}{2}\left[\mathbf{J}(\delta) + \mathbf{J}(\mathbf{0})\right]\delta \tag{18}$$

To evaluate $\mathbf{J}(\delta)$, we would need to know δ, which is the quantity we want to solve for. The trick is to use the assumption, that the gradient of the warped image in the converged case, is equal to the gradient of the reference frame and thus yields the following overall Jacobian for the ESM algorithm:

$$\mathbf{J}_{\text{esm}} = \frac{1}{2}\left(\frac{\partial I}{\partial \mathbf{x}}\bigg|_{\mathbf{x}=\omega(\mathbf{p},\mathbf{X})} + \frac{\partial I^*}{\partial \mathbf{x}}\bigg|_{\mathbf{x}=\omega(\mathbf{0},\mathbf{X})}\right) \frac{\partial \omega}{\partial \delta}\bigg|_{\delta=\mathbf{0}} \tag{19}$$

From equation (19) it can be seen, that the ESM Jacobian uses a mixture of the image gradients of the two images. The computation of the update step is done similarly as in Equation (7).

IV. ROBUSTNESS EXTENSIONS

We now describe several extensions, that improve the robustness of the alignment process.

A. Robust Estimation

In order to cope with outliers due to noise, scene dynamics or occlusions, we incorporate a robust estimation scheme. This is done by means of an influence function, weighting each pixel residual, reflecting its categorization as inlier or outlier. We therefore use robust M-estimation in an iteratively re-weighted least-squares approach based on [30], where the weight $w(r)$ for a residual r is calculated as

$$w(r) = \frac{\psi(r)}{r} \qquad \psi(r) = \frac{\partial \rho(r)}{\partial r} \tag{20}$$

We tested two different M-Estimator weight functions for evaluation, namely the Huber and the Tukey influence functions. The weights for the Huber estimator are defined as

$$w_{\text{hub}}(r) = \begin{cases} 1 & \text{if } |r| < 1.345 \\ 1.345/|r| & \textit{else} \end{cases} \tag{21}$$

and the weights for the Tukey estimator as

$$w_{\text{tuk}}(r) = \begin{cases} 0 & \text{if } |r| > 4.6851 \\ (1-(\frac{r}{c})^2)^2 & \textit{else} \end{cases} \tag{22}$$

The constants for the estimators are chosen to obtain the 95% asymptotic efficiency on the standard normal distribution. Furthermore, before calculating the weights, we normalize each residual by removing the median of all residuals and compute an estimate for the standard deviation $\hat{\sigma}$ of the residuals, by using their median absolute deviation:

$$\hat{r}_i = r_i - \text{Med}_i(r_i) \qquad \hat{\sigma} = 1.485 \cdot \text{Med}_i(|\hat{r}_i|) \tag{23}$$

and finally the weights for each pixel residuals are computed by $w_i(r) = w_{\text{estimator}}(\hat{r}/\hat{\sigma})$ and assembled into a diagonal matrix $\mathbf{W}$. The update step (7) changes to

$$\delta = -\left(\mathbf{J}(\mathbf{0})^T\mathbf{W}\mathbf{J}(\mathbf{0})\right)^{-1}\mathbf{J}^T(\mathbf{0})\mathbf{W} \tag{24}$$

and in a similar way for the other alignment methods.

B. Multi-Resolution Estimation

To increase the convergence region, we apply the algorithms on a scale space in a coarse-to-fine manner. Therefore we run the same optimization method, starting from the coarsest pyramid level, propagating the result up to the next finer scale. Note that for the grayscale images, we compute a gaussian pyramid, while for the depth images we interpolate the depth values always on the original finest scale image to avoid errors due to discontinuities in the depth image.

C. Global Affine Illumination

As the illumination within a scene might change, we add two additional parameters to our warp model, corresponding to a global bias and gain of the pixel values. Similar to [31] we assume the following relation between the two images:

$$I^* = (1+\alpha)I + \beta \tag{25}$$

For these parameters, we always use an inverse update rule, such that

$$(1+\delta\alpha)I^* + \delta\beta = (1+\alpha)I + \beta \tag{26}$$

This adds two additional entries to the parameter vector and thus also to the Jacobian. The entries are easy to compute though: for the gain parameter the Jacobian value equals the intensity of the corresponding reference pixel and for the bias parameter the value of the Jacobian is always 1. The update equation for the parameters after one iteration is given by:

$$\alpha \leftarrow \frac{\alpha - \delta\alpha}{1+\delta\alpha} \qquad \beta \leftarrow \frac{\beta - \delta\beta}{1+\delta\alpha} \tag{27}$$

D. Information Selection

We also implemented a saliency selection scheme, based on the one presented by Audras et al. [5]. In each iteration, we select the most valuable information, based on the strongest Jacobian components. The details of this selection scheme can be reviewed in the original paper. In Section V, we will evaluate its influence on the performance and precision. We have to note, that our implementation is rather straightforward and improvements in terms of performance are still left to our future work.

E. Regularization

In [8] the authors use a probabilistic formulation to incorporate motion priors into their system. We use a quite similar approach, based on regularization, to constrain the optimization towards smaller changes in the pose update. To do this, we add a regularization term to the costs, which depends on the overall computed update parameters Δ relative to the initial point:

$$\underset{\delta}{\text{argmin}}\left(\frac{1}{n}C(\delta) + \lambda\,(\delta+\Delta)^T\,\mathbf{D}\,(\delta+\Delta)\right) \tag{28}$$

Here n are the number of pixels contributing to the costs, such that the pixel costs term in Equation (28) becomes the mean squared pixel error, lambda is an weighting factor, specifying the influence of the regularization term and $\mathbf{D}$ is a diagonal matrix, which can be used to specify certainty about the parameters. $\mathbf{D}$ could be for example the inverse covariance matrix (also called information matrix) of the pose. By adding the regularization term, the update equation (7) changes to

$$\delta = -\left(\frac{1}{n}\mathbf{J}(\mathbf{0})^T\mathbf{J}(\mathbf{0}) + \lambda\mathbf{D}\right)^{-1}\left(\frac{1}{n}\mathbf{J}(\mathbf{0})^T\left(I(\mathbf{p}) - I^*\right) + \lambda\mathbf{D}\Delta\right) \tag{29}$$

where after each iteration we update the current Δ by

$$\Delta \leftarrow \Delta + \delta \tag{30}$$

Note that we keep the current Δ when iterating over the scale-space.

V. EVALUATION

In this section we will demonstrate the influence of the presented methods on the performance of the visual odometry result.

A. Precision

For evaluating the performance of different combinations of the presented methods, we run the algorithms using some of the datasets from the TUM RGB-D Benchmark [32]. We also use the provided tools to evaluate the precision for each run. The tool provides different metrics for evaluation, but we will restrict to the translational RMS error of the relative poses, which measures the drift in meter per second. This is a realistic error metric for the overall precision of an algorithm over the whole trajectory. Furthermore, the same metric has also been used by [8] and [9] and thus we can directly compare our results with theirs, as they provide tables for the same datasets. We evaluate the performance of the following different methods:

- Alignment strategy: Forward-Compositional (FC), Inverse-Compositional (IC), Efficient Second-Order Minimization (ESM)
- M-Estimator: Squared, Huber, Tukey
- Global Affine Illumination Estimation (AI)
- Regularization (REG)
- Information-Selection (SEL)

We tried all combinations of these algorithms, where each run was performed with similar parameters: We used four octaves with a scale factor of 0.5 between two octaves. The reference frame was updated automatically, when the distance to the previous frame or the final SSD pixel error were above certain thresholds (max. 20cm translational distance, 3 degrees relative orientation and mean pixel error above 0.03). All the experiments have been done using normalized floating point images. The number of optimizer iterations was set to a maximum of 10 iterations per octave, although the optimization is stopped earlier, if the average costs per pixel fall below an acceptable threshold (< 0.005). In regularized cases, we used $\mathbf{D} = \mathbf{I}$, as we don't have an underlying motion model and gave equal weights to the regularization term and the pixel costs, by setting $\alpha = 1$. When using the information selection scheme, we selected a total of 12% of the pixels on each octave, resulting in approximately 37000 points on the finest octave. Overall we evaluated 54 possible combinations. We will provide several smaller tables, to show the influence of different combinations on the overall precision of the algorithms. All in all, we found, that the Tukey estimator performed better than Huber for all the methods. Also the estimation of global affine illumination (AI) gave consistently better results for all cases. Therefore we first compare the results of the three align methods when used in combination with these approaches in Table I. The table is ordered by the best average performance on all datasets (last column). It can be seen, that the IC and ESM strategies perform almost equally well, whilst the FC algorithm is worse on average in comparison. We have to note though, that the forward alignment still works well on most datasets, but seems to have problems on the *desk2* sequence. It is also remarkable, that the information selection scheme has almost no influence on the precision for all of the algorithms, which is in agreement with [5]. The entries marked in bold, highlight the best results for the respective dataset column. The influence of estimating global affine illumination is given in Table II. On average the *AI* versions always outperform the non *AI* versions. Especially for the *large_no_loop* dataset, the difference is significant. Table III shows the influence of the regularization term on the precision of the different algorithms. Using our parameters, on average all of the algorithms did perform worse using the regularization than without using the regularization. We think, that this is due to the fact, that we do not have an automatic estimation of the weighting parameters for the regularization term. which should depend on the current velocity of the camera. Nevertheless, for two of the datasets the regularized version performed best, therefore we argue that in combination with a motion model and a filtering scheme, the regularization term might be more effective. Table IV gives a comparison of our best results with the best methods from [8] and [9]. Our methods are again ordered by the average RMSE for all the datasets. Those datasets, which have not been evaluated by the other authors are marked by a ”−”. It can be seen, that our best methods achieve better results on all datasets. Also our average best method, is better for all but one of the datasets in comparison with the state of the art. Figure 2 shows an exemplary plot of the three estimated translational components for the *fr2/desk*

TABLE I

RMSE TRANSLATIONAL DRIFT IN M/S FOR THE DIFFERENT ALIGN METHODS

Method	fr1/desk	fr1/desk2	fr2/desk	fr2/person	fr1/room	fr2/large_no_loop	Avg. RMSE
IC+Tukey+AI	0.030591	0.054388	**0.014538**	0.011780	0.040356	**0.080042**	0.038616
IC+Tukey+AI+SEL	0.030955	0.055407	0.014563	0.011831	0.040300	0.080079	0.038856
ESM+Tukey+AI	0.030187	**0.052622**	0.014719	**0.011676**	0.039696	0.084299	0.038866
ESM+Tukey+AI+SEL	0.030436	0.053953	0.014747	0.011735	**0.039623**	0.083994	0.039081
FC+Tukey+AI+SEL	**0.029644**	0.178340	0.015132	0.012210	0.040177	0.088476	0.060663
FC+Tukey+AI	0.029972	0.181995	0.015057	0.012127	0.040777	0.088336	0.061377

TABLE II

EVALUATION OF THE INFLUENCE OF ESTIMATING GLOBAL AFFINE ILLUMINATION ON THE RMSE DRIFT IN M/S

Method	fr1/desk	fr1/desk2	fr2/desk	fr2/person	fr1/room	fr2/large_no_loop	Avg. RMSE
IC+Tukey+AI	0.030591	0.054388	0.014538	0.011780	0.040356	**0.080042**	0.038616
ESM+Tukey+AI	0.030187	**0.052622**	0.014719	**0.011676**	**0.039696**	0.084299	0.038866
FC+Tukey+AI	**0.029972**	0.181995	0.015057	0.012127	0.040777	0.088336	0.061377
IC+Tukey	0.031820	0.060539	0.014514	0.012480	0.086228	0.223620	0.071534
FC+Tukey	0.031757	0.128766	0.014861	0.012299	0.055483	0.324947	0.094685
ESM+Tukey	0.031984	0.058508	**0.014490**	0.011773	0.052358	0.415930	0.097507

TABLE III

INFLUENCE OF THE REGULARIZATION TERM ON THE RMSE DRIFT IN M/S

Method	fr1/desk	fr1/desk2	fr2/desk	fr2/person	fr1/room	fr2/large_no_loop	Avg. RMSE
IC+Tukey+AI+SEL	0.030955	0.055407	0.014563	0.011831	0.040300	**0.080079**	0.038856
ESM+Tukey+AI+SEL	0.030436	**0.053953**	0.014747	0.011735	**0.039623**	0.083994	0.039081
IC+Tukey+AI+SEL+REG	0.040284	0.081427	**0.013650**	**0.010645**	0.099233	0.096487	0.056954
FC+Tukey+AI+SEL	**0.029644**	0.178340	0.015132	0.012210	0.040177	0.088476	0.060663
ESM+Tukey+AI+SEL+REG	0.040890	0.085713	0.014253	0.010667	0.107752	0.118711	0.062998
FC+Tukey+AI+SEL+REG	0.043579	0.101902	0.014701	0.011143	0.114251	0.130530	0.069351

TABLE IV

COMPARISON WITH STATE OF THE ART METHODS

Method	fr1/desk	fr1/desk2	fr2/desk	fr2/person	fr1/room	fr2/large_no_loop
IC+Tukey+AI	0.030591	0.054388	0.014538	0.011780	0.040356	0.080042
IC+Tukey+AI+SEL	0.030955	0.055407	0.014563	0.011831	0.040300	0.080079
ESM+Tukey+AI	0.030187	**0.052622**	0.014719	0.011676	0.039696	0.084299
ESM+Tukey+AI+SEL	0.030436	0.053953	0.014747	0.011735	**0.039623**	0.083994
IC+Huber+AI+REG	0.044009	0.078545	**0.012222**	**0.011058**	0.083992	**0.079753**
FC+Tukey+AI+SEL	**0.029644**	0.178340	0.015132	0.012210	0.040177	0.088476
FOVIS (see [9])	0.060400	-	0.013600	-	0.064200	0.096000
ICP+RGB-D (see [9])	0.039300	-	0.020800	-	0.062200	0.179500
t-dist.+weights (see [8])	0.045800	0.070800	0.020300	0.036000	-	-
t-dist.+w.+temp. (see [8])	0.049100	0.068700	0.018800	0.034500	-	-

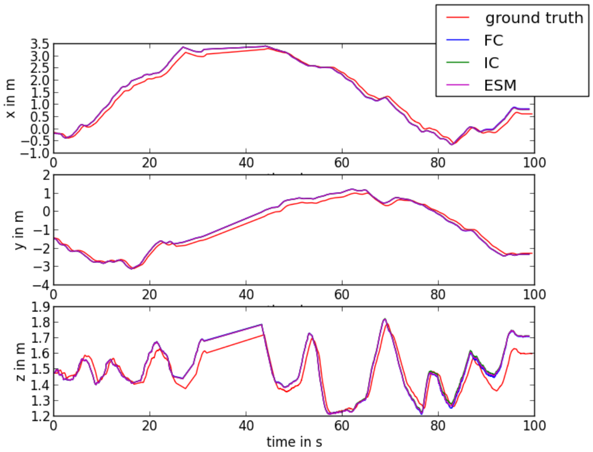

Fig. 2. Superimposed trajectory plot of the *fr2/desk* sequence, using affine illumination estimation, Tukey estimator and Gauss-Newton optimizer for all three methods.

TABLE V

AVERAGE RUNTIME OF THE DIFFERENT ALGORITHMS ON THE *fr2/desk* SEQUENCE

Align Method	Influence Function	Affine Illumination	Information Selection	Avg. runtime/frame
IC	Tukey	true	true	28ms
IC	Tukey	true	false	23ms
IC	Tukey	false	true	30ms
IC	Tukey	false	false	27ms
FC	Tukey	true	true	35ms
FC	Tukey	true	false	31ms
FC	Tukey	false	true	37ms
FC	Tukey	false	false	34ms
ESM	Tukey	true	true	39ms
ESM	Tukey	true	false	35ms
ESM	Tukey	false	true	38ms
ESM	Tukey	false	false	35ms

sequence against the ground truth data for all three alignment methods. The methods where run in combination with affine illumination estimation and the Tukey estimator. For this dataset, all three methods perform more or less equally well and therefore the trajectories are most of the time completely overlayed.

B. Performance

We ran our experiments on an Intel Xeon E31225 3.1GHz Quad-Core desktop machine with 4GB of RAM, using a single core. The average runtimes on the *fr2/desk* sequence for the most precise of our algorithms, are shown in Table V. It can be seen that the information selection scheme is even slower than using all information. The reason for this is that we did not optimize this part of the algorithm and just use a brute force sort the vector of the Jacobians for each degree of freedom. Therefore the increased costs for updating the reference frame outweigh the speedup in the alignment is process. The IC version is the overall fastest and the FC and ESM versions are more or less equally fast. On the one hand, this is due to the fact that for the FC and ESM case, the gradient of the image pyramid has to be evaluated for each frame and not only for the reference image as in the IC case. On the other hand, additional costs due to the bilinear interpolation of the gradients are introduced. At the moment we are interpolating three times for ESM and FC: once for the gray values, and once for each gradient direction. In the inverse compositional case, we only have to interpolate the gray values in each step. Nevertheless, in each case the dominant costs ($\approx 50-60\%$ of the overall processing time) of our system are given by building up the linearized system of equation. As this has to be done in every iteration of the optimization, we target future implementation optimizations towards this end first. We note that the best performing algorithm is also the fastest one, namely the combination of the Inverse Compositional approach with Affine Illumination estimation. Anyhow, all of the algorithms are real-time capable, running in between $25-40$ frames per second.

VI. CONCLUSIONS

In this paper we gave an evaluation of different types of algorithms for computing direct visual odometry from RGB-D data. We have shown in our results, that the Inverse-Compositional and Efficient Second Order methods perform at least equally well as the Forward Compositional formulation and usually even better. Another contribution, of this work, is the inclusion global affine illumination parameters to the formulation. We have shown in our evaluations that our best combinations of methods, generally outperform the current state of the art in this field. Moreover our implementation is available as open-source[2]. In our future work, we want to further reduce the runtime, especially for the information selection part and for the construction of the linearized system. Additionally we want to include more robust similarity metrics into our framework. Another direction

[2] http://www6.in.tum.de/Main/ResearchDirectVO

of future work is the integration of our algorithm with a real-robot for online motion estimation and navigation. To make better use of the introduced regularization part, we also want to combine our system with a Kalman Filtering scheme, to give useful estimates for the regularization parameters based on the system covariances.

Acknowledgements

The authors gratefully acknowledge the support by the International Graduate School of Science and Engineering (IGSSE), Technische Universiät München, project 4.03 Image Aided Flight Control and by the German BMBF projects ECU (grant number:13N11936) and Car2X (grant number: 13N11933).

References

[1] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2004, pp. I–652–I–659 Vol.1.

[2] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proceedings of the 7th international joint conference on Artificial intelligence*, 1981.

[3] S. Baker and I. Matthews, "Lucas-Kanade 20 Years On: A Unifying Framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[4] M. Meilland, A. I. Comport, and P. Rives, "A spherical robot-centered representation for urban navigation," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 5196–5201.

[5] C. Audras, A. Comport, M. Meilland, and P. Rives, "Real-time dense appearance-based SLAM for RGB-D sensors," *Australasian Conf. on Robotics and Automation*, 2011.

[6] J. Sturm, S. Magnenat, N. Engelhard, F. Pomerleau, F. Colas, W. Burgard, D. Cremers, and R. Siegwart, "Towards a benchmark for RGB-D SLAM evaluation," *Proc. of the RGB-D Workshop on Advanced Reasoning with Depth Cameras at Robotics: Science and Systems Conf.(RSS), Los Angeles, USA*, vol. 2, p. 3, 2011.

[7] J. Sturm, W. Burgard, and D. Cremers, "Evaluating Egomotion and Structure-from-Motion Approaches Using the TUM RGB-D Benchmark," *Proc. of the Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RJS International Conference on Intelligent Robot Systems (IROS)*, 2012.

[8] C. Kerl, J. Sturm, and D. Cremers, "Robust Odometry Estimation for RGB-D Cameras," *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2013.

[9] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. B. McDonald, "Robust Real-Time Visual Odometry for Dense RGB-D Mapping," in *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Karlsruhe, Germany, May 2013.

[10] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pp. 225–234, 2007.

[11] K. Konolige, J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua, "View-based Maps," *International Journal of Robotics Research*, vol. 29, no. 8, Jul. 2010.

[12] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, "A constant time efficient stereo SLAM system," *Proceedings of the British Machine Vision Conference (BMVC)*, 2009.

[13] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments," *International Journal of Robotics Research*, 2012.

[14] N. Engelhard, F. Endres, J. Hess, J. Sturm, and W. Burgard, "Real-time 3D visual SLAM with a hand-held RGB-D camera," *Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, Vasteras, Sweden*, vol. 2011, 2011.

[15] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the RGB-D SLAM system," *IEEE International Conference on Robotics and Automation (ICRA), 2012*, pp. 1691–1696, 2012.

[16] R. Newcombe, S. Lovegrove, and A. Davison, "DTAM: Dense tracking and mapping in real-time," *Proc. of the Intl. Conf. on Computer Vision (ICCV), Barcelona, Spain*, vol. 1, 2011.

[17] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pp. 127–136, 2011.

[18] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, and A. Davison, "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera," *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 559–568, 2011.

[19] T. Whelan, M. Kaess, M. F. Fallon, H. Johannsson, J. J. Leonard, and J. B. McDonald, "Kintinuous: Spatially Extended KinectFusion," in *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Sydney, Australia, Jul. 2012.

[20] M. Meilland, A. Comport, and P. Rives, "Dense visual mapping of large scale environments for real-time localisation," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 2011, pp. 4242–4248.

[21] *Real-Time Dense Visual Tracking Under Large Lighting Variations*, 2011.

[22] A. Comport, M. Meilland, and P. Rives, "An asymmetric real-time dense visual localisation and mapping system," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 2011, pp. 700–703.

[23] A. Comport, E. Malis, and P. Rives, "Real-time Quadrifocal Visual Odometry," *International Journal of Robotics Research*, vol. 29, no. 2-3, pp. 245–266, Mar. 2010.

[24] T. Tykkala, C. Audras, and A. Comport, "Direct Iterative Closest Point for real-time visual odometry," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 2011, pp. 2050–2056.

[25] F. Steinbrucker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense RGB-D images," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 2011, pp. 719–722.

[26] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an RGB-D camera," *Proc of the 15th International Symposium on Robotics Research (ISRR2011)*, 2011.

[27] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*, 1st ed. Springer New York, 2004.

[28] S. Benhimane and E. Malis, "Real-time image-based tracking of planes using efficient second-order minimization," in *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on.*

[29] E. Malis, "Improving vision-based control using efficient second-order minimization techniques," *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 2, pp. 1843–1848, 2004.

[30] P. J. Huber, *Robust statistical procedures*. SIAM, 1996, vol. 68.

[31] M. Hwangbo, J. S. Kim, and T. Kanade, "Gyro-aided feature tracking for a moving camera: fusion, auto-calibration and GPU implementation," *International Journal of Robotics Research*, vol. 30, no. 14, pp. 1755–1774, 2011.

[32] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," *Proc. of the IEEE Int. Conf. on Intelligent Robot Systems (IROS)*, 2012.