Combining Edge Images and Depth Maps for Robust Visual Odometry

Fabian Schenk
schenk@icg.tugraz.at
Friedrich Fraundorfer
fraundorfer@icg.tugraz.at

Institute for Computer Graphics and Vision
Graz University of Technology
Graz, Austria

Abstract

In this work, we propose a robust visual odometry system for RGBD sensors. The core of our method is a combination of edge images and depth maps for joint camera pose estimation. Edges are more stable under varying lighting conditions than raw intensity values and depth maps further add stability in poorly textured environments. This leads to higher accuracy and robustness in scenes, where feature- or photoconsistency-based approaches often fail. We demonstrate the robustness of our method under challenging conditions on various real-world scenarios recorded with our own RGBD sensor. Further, we evaluate on several sequences from standard benchmark datasets covering a wide variety of scenes and camera motions. The results show that our method performs best in terms of trajectory accuracy for most of the sequences indicating that the chosen combination of edge and depth terms in the cost function is suitable for a multitude of scenes.

1 Introduction

Visual odometry (VO) [11, 12] is the task of estimating the ego-motion of a camera from a temporally ordered sequence of images. It is a key requirement for many applications in computer vision and robotics such as autonomous driving, navigation, 3D reconstruction, augmented and virtual reality. While monocular and stereo camera setups have been extensively used in the past, the recent introduction of cheap RGBD sensors has opened many new possibilities for VO research. RGBD sensors like the Microsoft Kinect, Asus Xtion and Orbbec Astra can simultaneously record a scene's texture as an RGB image and its geometry as a depth map. The processing of the sensor input data is the main difference between current VO systems and divides them roughly into three categories. (i) Feature-based methods [9, 11] extract and match features, thereby discarding most of the image content. Thus, they typically do not work very well in scenes with insufficient texture. In contrast, (ii) direct methods work better in poorly textured environments as they do not rely on feature extraction and matching but process image information directly. Dense or semi-dense approaches based on the photoconsistency assumption [4, 7, 14] and edge-based variants [8, 13] are used. (iii) Iterative closest point (ICP) methods [10, 18] directly align 3D point clouds but require sufficient 3D structure and a costly correspondence matching step. Since texture- and structure-less surfaces like walls, floors or ceilings as well as illumination changes even

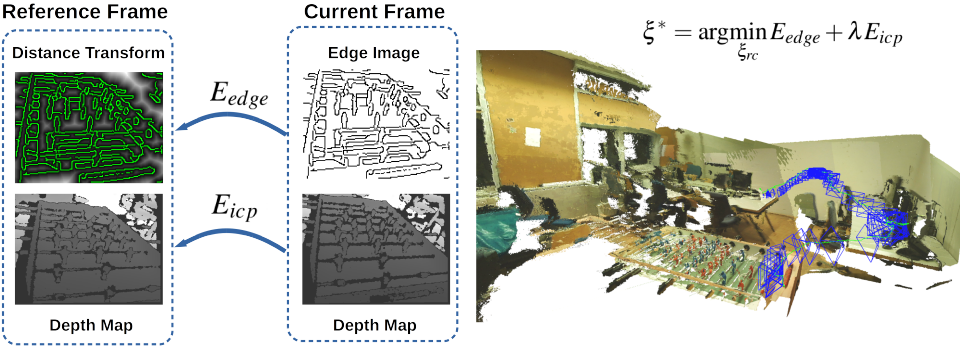

Figure 1: We propose a method that combines edge images and depth maps to jointly minimize edge distance and point-to-plane error for camera motion estimation.

occur in common office or indoor scenes, various combinations of these categories have been proposed to tackle these challenges [7, 17, 18].

In this work, we propose a robust RGBD VO system that addresses the challenges of typical indoor scenes. We present a method that uses a combination of edge images and depth maps and jointly minimizes edge distance and point-to-plane error to exploit the benefits of both (see Fig. 1). While the accuracy of edges and their robustness to illumination variations increase tracking quality compared to photoconsistency-based approaches, the point-to-plane term adds stability in poorly textured regions and imposes additional geometric constraints on the camera pose estimation. The main contributions of our paper can be summarized as:

- A combination of edge distance and point-to-plane error for camera pose estimation
- An optimization on spatially close reference frames to increase trajectory accuracy
- An extensive evaluation on challenging real-world and standard benchmark datasets
- A VO framework that runs in real-time on a CPU[1]

2 Related Work

Classical indirect feature-based methods extract point features, establish correspondences and estimate the camera motion between images [9, 10]. All point feature-based methods discard valuable image information during keypoint extraction, limiting them to texture-rich scenes. The recent ORB-SLAM2 [9] utilizes ORB features for tracking, mapping and loop closing and shows very promising results. Nevertheless, a known problem of feature-based approaches is, that the spatial distribution of the extracted sparse features influences the pose estimation, which is a significant limitation in practice. Indoor scenes for example consist of areas, where many features are detected, e.g. posters, keyboards, and parts that are nearly feature free, e.g. ceilings or walls, which creates challenging situations.

[1]Code available: https://www.tugraz.at/index.php?id=22399

Recently, direct methods that completely avoid feature extraction and correspondence matching have become very popular. Direct dense methods [7, 14] always process the complete image information, thereby avoiding a bias due to spatial distribution. Steinbrücker et al. [14] utilize depth information and optimize a rigid body motion such that the photoconsistency between two images is maximized. Kerl et al. [7] propose a probabilistic formulation to jointly minimize intensity and depth errors for pose estimation. In contrast to [7, 14], LSD-SLAM [9] is a semi-dense method that only relies on high-gradient regions to minimize the photoconsistency error. However, the quality of photoconsistency-based pose estimation suffers under motion blur and as demonstrated in [8, 13] is limited to small inter-frame motions. Instead of high-gradient regions, many methods use the more robust edges. Tarrio and Pedre [16] detect Canny edges [2] and try to match them between images by searching along the normal direction, which is computationally expensive and error prone. In their direct edge-alignment (D-EA), Kuse and Shen [8] instead pre-compute the distance to the closest edge at each pixel position with a distance transform (DT) [5] and optimize with a subgradient method. Wang et al. [17] jointly minimize edge distance and a photometric error at high-gradient pixels. In [13], Schenk and Fraundorfer study the influence of different edge detectors and demonstrate how to efficiently remove outliers to increase accuracy and robustness. Similar to feature-based methods, the distribution of edges can influence pose estimation.

Instead of images, various works directly align 3D point clouds with the iterative closest point (ICP) [1, 3, 10] algorithm. The standard versions of ICP have several issues, such as missing data and outliers and are usually not real-time capable as they require a computationally expensive nearest-neighbor search in 3D space in each iteration. KinectFusion [10] exclusively processes depth maps and reduces the search to 2D space by warping onto the current depth map and establishing correspondences according to the 2D pixel coordinates. This strategy and the intensive use of a GPU make KinectFusion one of the first real-time capable VO systems. Due to the requirement of 3D structure, depth only methods often fail in room size scenes. Whelan et al. [18] tackle this problem by jointly optimizing a point-to-plane and a photoconsistency error. However, their method requires a very strong GPU to be real-time capable.

In this work, we present a VO method for RGBD data, which combines edge distance with point-to-plane error [3]. We demonstrate that our system performs best on a wide variety of scenes and compare to feature-based [9], direct [7, 8] and point-to-plane ICP-based [18] approaches. Our joint camera pose estimation alleviates many of the limitations of previous works. In contrast to photoconsistency- [14] or feature-based methods [9, 11], the dense point-to-plane ICP term stabilizes optimization in poorly textured scenes and reduces the influence of spatial feature distribution. Environments with little 3D structure are problematic for ICP approaches [1, 3, 10] but edge features prevent misalignment in such cases. Additionally, our system runs in real-time on a CPU, while many approaches require a GPU [10, 18].

3 Relative Camera Motion Estimation

At each time step t, we receive a frame F_t that comprises an RGB image I_t and a depth map Z_t. We assume I_t and Z_t to be synchronized and aligned such that at a pixel position $p = (p_x, p_y)$, the intensity is given as $I(p)$ and the corresponding depth as $Z(p)$. The 3D point $P = (x, y, z)$ can be computed in the respective camera coordinate system from p and

the corresponding depth $Z(p)$ with the inverse projection function π^{-1}:

$$P = \pi^{-1}(p, Z(p)) = \left(\frac{p_x - c_x}{f_x} z, \frac{p_y - c_y}{f_y} z, z \right), \tag{1}$$

where c_x, c_y are the principal offsets, f_x, f_y the focal lengths as defined by a standard pinhole camera model and $z = Z(p)$. Similarly, the projection function π is given as:

$$p = \pi(P) = \left(\frac{x f_x}{z} + c_x, \frac{y f_y}{z} + c_y \right). \tag{2}$$

We further define a relative rigid body motion $g \in SE(3)$, which is typical represented as transformation matrix T comprising a 3×3 orthogonal rotation matrix $R \in SO(3)$ and a 3×1 translation vector $t \in \mathbb{R}^3$:

$$T_{4\times4} = \begin{bmatrix} R_{3\times3} & t_{3\times1} \\ 0 & 1 \end{bmatrix}. \tag{3}$$

Since g only has 6 degrees of freedom, T is over-parametrized. We use the more compact representation as twist coordinates ξ defined by the Lie algebra $se(3)$ associated with the group $SE(3)$. ξ is then a 6×1 vector given as:

$$\xi = (v_1, v_2, v_3, \omega_1, \omega_2, \omega_3)^T \in \mathbb{R}^6, \tag{4}$$

where v_1, v_2, v_3 is the linear velocity and $\omega_1, \omega_2, \omega_3$ the angular velocity. T can be retrieved from ξ with the matrix exponential as $T = exp(\xi)$. The full warping function τ reprojects a pixel p from frame F_j with depth $Z_j(p)$ to p' in F_i under the transformation T_{ij} and is defined as:

$$p' = \tau(\xi_{ij}, p, Z_j(p)) = \pi(T_{ij} \pi^{-1}(p, Z_j(p))). \tag{5}$$

3.1 Edge- and ICP-based Image Alignment

In our method, we estimate the relative motion ξ_{rc} from a current frame F_c to a reference frame F_r by jointly minimizing an edge distance error E_{edge} and an ICP-based point-to-plane error E_{icp} (see Fig. 1). The optimization is given as:

$$\xi^* = \underset{\xi_{rc}}{\operatorname{argmin}} E_{edge} + \lambda E_{icp}, \tag{6}$$

where ξ^* denotes the optimal relative motion and λ is a balancing factor. We minimize Eq. (6) in a coarse-to-fine scheme using an iteratively re-weighted Levenberg-Marquardt method in a left compositional formulation similar to [4]. Further, we initialize the optimization according to a constant motion assumption to start close to the minimum. In our optimization formulation, cost evaluation solely depends on the reference frame, thus we have to recompute all frame-dependent structures only when a new reference frame is added. We follow the strategy proposed in [13] and insert a new reference frame, when the overlap between reprojected edges from 3 previous frames and the edges in the current frame is lower than the number of non-overlapping edges.

Edge Distance Error E_{edge}: We detect Canny edges [2] in each frame and on all pyramid levels and reproject only edge pixels with a valid depth to the reference frame. Finding the distance to the closest edge by searching through the image in each iteration is usually very slow. Instead, whenever a new reference frame is added, we compute the Euclidean distance to the closest edge at each pixel position using the distance transform (DT) [5]. The edge distance residual r_e can then be obtained by evaluating the DT at the reprojected pixel position (see Fig. 1). We define our edge distance error as:

$$E_{edge} = \sum_{p_e \in \Omega_{E_c}} \delta_H(r_e) r_e^2, \quad r_e = DT_r(\tau(\xi_{rc}, p_e, Z_c(p_e))), \quad \delta_H(r_e) = \begin{cases} 1 & r_e \leq \Theta_H \\ \frac{\Theta_H}{r_e} & r_e > \Theta_H \end{cases}, \quad (7)$$

where DT_r denotes the DT of the reference frame, Ω_{E_c} the set of edge pixels with valid depth in the current frame, p_e the pixel position (p_x, p_y) of an edge and $\delta_H(r_e)$ a Huber weight function that reduces the influence of large residuals. Edge detections can differ between frames and to remove potential outliers, we filter residuals r_e above a threshold Θ_e at each pyramid level.

Point-to-Plane Error E_{icp}: We also perform geometric pose optimization with an ICP-based point-to-plane error [3]. Instead of the costly nearest-neighbor search in 3D space, we perform projective association, where we reproject all pixels p_i with valid depth from the current to the reference frame with the full warping function $p' = \tau(\xi_{rc}, p_i, Z_c(p_i))$ and establish correspondences according to the pixel coordinates [10]. The point-to-plane error projects the distance vector between two 3D points onto the surface using the corresponding normal vector, which allows flat regions to slide along each other [3]. This requires the computation of a surface normal map N_r whenever a new reference frame is added. We define the point-to-plane error as:

$$E_{icp} = \sum_{p_i \in \Omega_{Z_c}} \delta_I(r_i) r_i^2; \quad r_i = \left\langle (P_r - T_{rc}\pi^{-1}(p_i, Z_c(p_i))); n_r \right\rangle, \quad \delta_I(r_i) = \frac{1.5}{1.5 + r_i^2} \quad (8)$$

where Ω_{Z_c} denotes the set of valid depth values in the current frame, r_i the residual and p' the reprojected pixel position. The normal vector is given as $n_r = N_r(p')$ and the 3D point as $P_r = \pi^{-1}(p', Z_r(p'))$. A weight function $\delta_I(r_i)$ similar to [7] reduces the influence of large residuals.

Balancing Factor λ: When jointly optimizing two error terms, a balancing factor λ is typically introduced. Finding the optimal λ is especially challenging when optimizing errors with completely different metrics, e.g. edge distance in pixels and point-to-plane error in meters, or abundance, e.g. rather sparse edges compared to the dense ICP term. Our experiments indicate that the ideal value of λ also depends on the sequence and can also vary between iterations. However, it is possible to obtain very accurate trajectories even without the optimal value. Our results show that the edge-based term is more important for accuracy than the ICP-based term (see Tab. 1), suggesting to give a higher weight to the edge-based term. We can impose a higher influence of the edge residuals by setting $\lambda = 1$[2]. When only few edges are detected, the influence of the ICP residuals gets higher as intended.

[2] This is mainly due to different metrics of the residuals

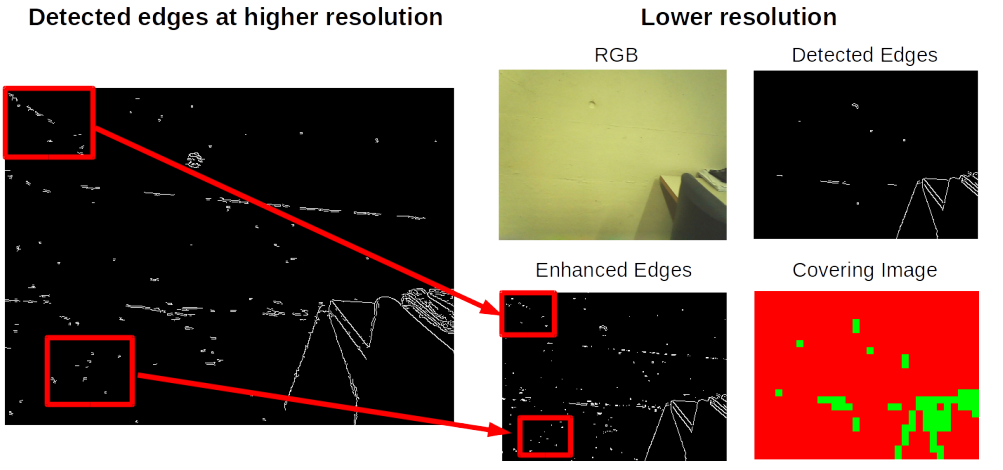

Figure 2: We improve the spatial distribution of the detected edges by computing a covering image to find parts that are insufficiently covered (depicted in red). We then transfer edge detections from a higher to the lower resolution level.

Improving Spatial Distribution of Edges: A common challenge for edge-based methods is the spatial distribution of the edges, which can bias camera pose estimation due to cluttered or edge-less areas in the image. While typically an abundance of edges is detected at the highest level of the image pyramid, fine-grained parts are smoothed over in lower levels and do not produce edge responses anymore. This is especially problematic in areas with little intensity difference (e.g. walls), where edge detectors rely on fine texture. We find areas that are insufficiently covered by computing the spatial distribution of the detected edges in the lower pyramid levels. We count the number of edge pixels in each $N \times N$ patch and if the number is below Θ_E, we mark the patch as insufficiently covered. In this way, we compute a covering image showing sufficiently (green) and insufficiently (red) covered patches (see Fig. 2). If a patch is insufficiently covered, we enhance the edge image by transferring detections from a higher resolution image into the lower resolution image.

3.2 Optimization on Spatially Close Reference Frames:

To reduce drift and close small loops, we additionally optimize the current reference frame's pose with respect to up to M previous reference frames. Instead of a standard sliding window approach that takes the temporally last M frames into account and therefore is only able to reduce drift in a limited temporal window, our method is not temporally constrained. From a relative transformation T_{ij} between camera centers comprising a rotation R_{ij} and a translation t_{ij}, we compute the distance between camera centers as $d = ||t_{ij}||_2$ and the view angle as $\alpha = \arccos(\frac{1}{2}(trace(R_{ij}) - 1))$. We define a frame as spatially close if the $d < \Theta_d$ and $\alpha < \Theta_\alpha$, i.e. both views see similar parts of the scene.

We only optimize the pose of the current reference frame ξ_{cw} with respect to the world and keep the other poses fixed. Our world coordinate system has its center at $(0,0,0)$ and coincides with the center of the first camera. By extending Eq. (6) to multiple frames, we get:

$$\xi^* = \underset{\xi_{cw}}{\operatorname{argmin}} \sum_{f \in \Omega_M} E_{edge}(\xi_{cf}) + \lambda E_{icp}(\xi_{cf}), \quad \xi_{cf} = \xi_{cw}\xi_{wf}, \tag{9}$$

where Ω_M is the set of spatially close reference frames, ξ_{cf} is the transformation between a previous and the current frame, ξ_{wf} is the transformation between a previous frame and the world and we again set the balancing factor $\lambda = 1$.

4 Results and Discussion

We implemented our system in C++ using OpenCV for edge detection, distance transform computation and image in- and output. The complete VO system runs in real-time on an Intel i7-4790 desktop computer with 32 GB of RAM. The Levenberg-Marquardt optimization scheme uses a 3 level coarse-to-fine scheme with maximum resolution of 640×480 px. For edge distance optimization, we apply a Huber weight function δ_H with a value of $\Theta_H = 0.3$ and remove edges with a distance greater than $\Theta_e = 10$, 20 and 30 px at each respective pyramid level. We optimize on a maximum number of $M = 4$ spatially close reference frames and set the search parameters $\Theta_d = 0.1$ m and $\Theta_\alpha = 30°$. We compute the spatial distribution for patches with $N = 5$, 10 and 20 px for each pyramid level and mark a patch as insufficiently covered if $\Theta_E < 0.05N^2$. To compensate for differences between sensors, we adapt the parameters for Canny [2] but keep the values constant for all sequences in a dataset. In our experiments, we found that depth measurements at borders with strong depth jumps are often noisy and can give near or far values. If only edges at strong depth jumps are detected this might be problematic. However, in practice this is not an issue because most of the edges are detected on planar surfaces (walls, tables,..) or at very small depth jumps, where the measurements are typically correct. Further, the influence of edges with wrong depth is down-weighted during the optimization.

We demonstrate the robustness of our system by example of many challenging real-world recorded with our own sensor that gives RGBD sequences at 30 fps and a maximum resolution of 640×480. To show the quantitative performance of our method, we extensively evaluate on a large variety of camera motions and scenes. We choose the two standard RGBD benchmarks datasets TUM RGBD [15] and ICL-NUIM [6]. TUM RGBD comprises a large number of sequences recorded with a Microsoft Kinect at 30 Hz with highly accurate ground truth poses from a motion capture system. We select 7 different sequences including also scenes with little texture such as *fr3/str-ntex-far* and *fr3/large-cabinet*. ICL-NUIM offers synthetically generated RGBD data that is completely noise-free along with a perfect ground truth. From the ICL-NUIM we choose three typical indoor sequences that also contain poorly textured walls. The ICL-NUIM dataset is evaluated in the same way as the TUM RGBD dataset and we apply the same metrics for both.

Evaluation metrics: Sturm et al. [15] proposed the relative pose error (RPE) to measure drift over a fixed time interval Δt between a set of poses Q from the ground truth trajectory and a set of poses P from the estimated trajectory. The RPE at time step i is defined as:

$$RPE_i = (Q_i^{-1} Q_{i+\Delta t})^{-1} (P_i^{-1} P_{i+\Delta t}), \tag{10}$$

Another common measure to evaluate the performance of a system is the absolute trajectory error (ATE) [15], which can be computed at time step i as:

$$ATE_i = Q_i^{-1} S P_i, \tag{11}$$

where S is a rigid body transformation that aligns Q and P. We evaluate the root mean squared error (RMSE) of the translational component of the RPE in $[\frac{m}{s}]$ and the ATE in $[m]$.

4.1 Results on Real-World Datasets

To demonstrate the robustness of our method in practice, we recorded many different RGBD sequences with our Orbbec Astra Pro sensor. Figure 3 shows an excerpt from three of these sequences including the estimated trajectory and a dense reconstruction directly generated by our viewer.

The first sequence is a recording of a small room enclosed by non-textured walls, where our method never looses track and shows a consistent trajectory and reconstruction. The second sequence comprises movements in front of a white wall, where our method accurately reconstructs the wall. Sequence 3 is a typical university room containing a soccer table and many computers. The challenging part is the turn, where the camera is facing the poorly textured wall. In all of these sequences, the addition of the ICP term is essential due to nearly texture-less surfaces. We also tried ORB-SLAM2 [9] in VO mode on our sequences and it repeatedly failed in the poorly textured parts of the first two sequences even after trying many different threshold settings. Surprisingly, the third sequence worked quite well after some minor threshold adjustments.

4.2 Results on Standard Benchmark Datasets

We compare our system to four approaches that can handle RGBD data: (i) the feature-based ORB-SLAM2 [9] as comparison to a different category of VO, two direct methods, (ii) the dense DVO [7] and the (iii) edge-based D-EA [8] and finally, (iv) ICPCUPA [18] that minimizes the point-to-plane error similarly to ours. Note that, ICPCUDA processes only depth and is the ICP part of the combined optimization presented in [18]. For ORB-SLAM2 [9], we set *mbOnlyTracking* = *true* such that it only performs VO instead of the full SLAM pipeline. We run DVO [7] in the standard weighted configuration with 4 pyramid levels. For D-EA [8] we utilize the code available online without any modifications except some threshold adaptations for difficult sequences. We use the ICPCUDA [18] version that is provided online[3].

Table 1 shows the quantitative evaluation on the TUM RGBD [15] and the ICL-NUIM [6] sequences. We present four different version of our method to demonstrate the influence of each part: (i) edges only, (ii) depth only, (iii) edges and depth (E+D) and (iv) edges + depth + optimization on spatially close frames (E+D+Opt).

The results on TUM RGBD suggest that all the methods can work quite well on specific scenes but do not generalize to large variety of scenes. Our method however, demonstrates very accurate results regarding the absolute trajectory error on all the sequences, which indicates that our combination is well-suited for a multitude of scenes. While our edges-only approach performs very well, the results clearly show that adding the point-to-plane error improves accuracy. We attribute this mostly to the added stability in poorly textured areas and the geometric constraints imposed on camera pose estimation. Additionally, the optimization on spatially close reference frames reduces drift and closes small loops, which drastically lowers the ATE of short sequences such as *fr1/rpy*, *fr1/desk* and *fr1/desk2*. In *fr2/desk* and *fr3/large_cabinet* an object is in the middle and the camera is moving around it with start and end point being close together. Both sequences are very long and the accumulated drift cannot be overcome by our optimization.

When looking at the RPE, please note that the results are in $[\frac{m}{s}]$ and often the differences between the systems are in the $[mm]$ range, where the results can be considered equal due

[3] https://github.com/mp3guy/ICPCUDA

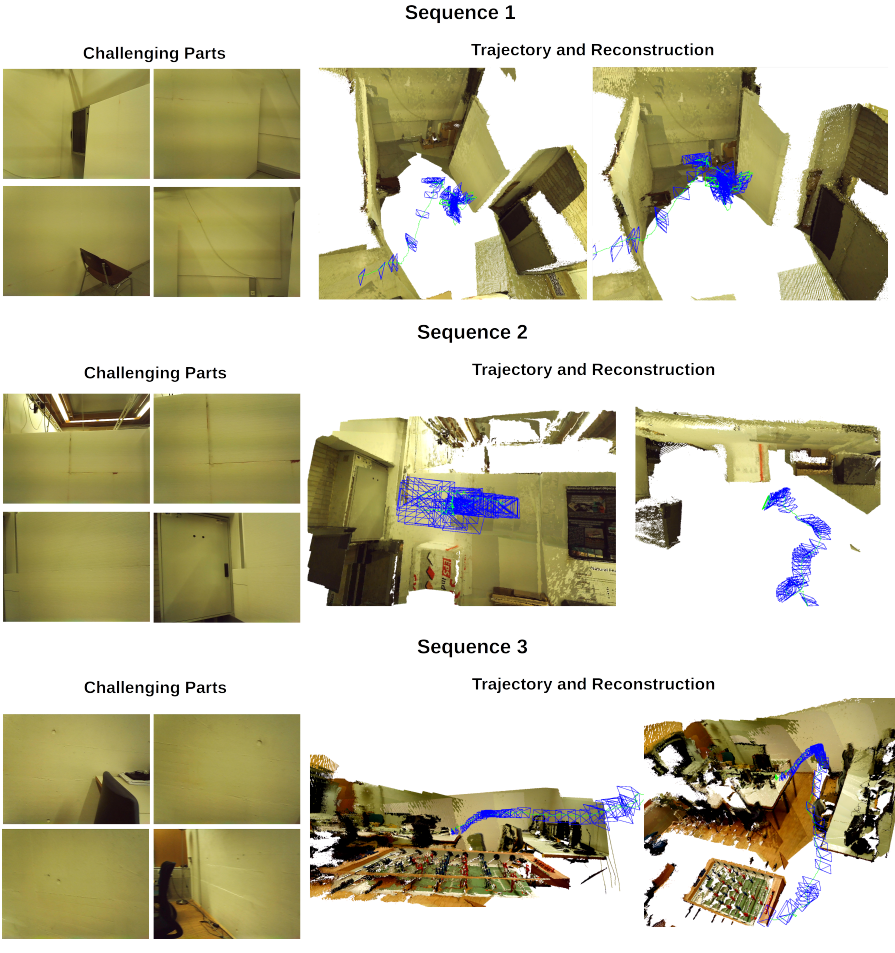

Figure 3: Our method manages to track and reconstruct three challenging indoor sequences recorded with our RGBD sensor. Seq. 1 is a small room enclosed by non-textured walls, Seq. 2 comprises movements in front of a white wall and Seq. 3 is a typical university room.

to ground truth accuracy. We perform comparable or better than the state-of-the-art methods on nearly all the datasets. It is interesting that the optimization on spatially close reference frames does not reduce the RPE. On *fr3/large-cabinet*, DVO performs best due to the favorable setup of the scene, where the black cabinet is clearly distinguishable from a white floor in the intensity image and depth map. Our system also shows the most accurate results on the difficult textureless *fr3/str-ntex-far* sequence, where we had to adapt the threshold for ORB-SLAM2. The results clearly demonstrate that our method generalizes to a larger variety of scenes than the other methods [7, 8, 9, 18].

On the ICL-NUIM dataset we see strong performances of all methods due to the perfect synchronization of RGB and depth images and the noise-free synthetic data. ORB-SLAM2 and DVO work well on all the sequences and their results are quite close together. Interestingly, our depth only approach shows the best results on two sequences, which we attribute to the perfect depth data. This also benefits our combined approach, which again demonstrates strong performance compared to the other methods.

Comparison of the Absolute Trajectory Error (ATE) $[m]$								
	DVO [7]	ORB2(VO) [9]	ICP [15]	D-EA [8]	Our Method			
Seq.	Dense	Features	ICP	Canny	Edges (E)	Depth (D)	E+D	E+D+Opt
fr1/xyz	0.057601	**0.008820**	0.042059	0.130058	0.058097	0.032545	0.048628	0.015516
fr1/rpy	0.163409	0.080904	0.103716	0.148215	0.052101	0.167213	0.055522	**0.022569**
fr1/desk	0.182512	0.090906	0.146944	0.163761	0.060936	0.071993	0.056677	**0.029604**
fr1/desk2	0.188611	0.100898	0.256770	0.448858	0.075279	0.169700	0.076796	**0.059941**
fr2/desk	0.467958	0.386566	1.580273	0.945456	**0.089041**	0.167662	0.092791	0.095057
fr3/large-cab	**0.241312**	0.960640	1.467609	1.334846	0.533166	0.958356	0.355983	0.501608
fr3/cabinet	0.420579	0.311646	0.757028	0.666367	0.275698	0.914652	**0.241852**	0.274828
fr3/str-ntex-far	0.145770	0.101452[1]	0.149810	0.632289	0.072096	0.243308	0.055475	**0.021768**
icl/lr-kt0	0.093531	0.090746	0.697352	0.676783[2]	0.236911	**0.040558**	0.085232	0.054882
icl/lr-kt1	0.199851	0.080261	0.045416	0.514425[2]	0.023476	**0.000762**	0.020264	0.009658
icl/off-kt1	0.114674	0.066268	0.274923	0.450322[2]	0.091087	0.122479	0.023387	**0.015384**
Comparison of the Relative Pose Error (RPE) in [m/s]								
fr1/xyz	0.026610	**0.014700**	0.031164	0.049424	0.028947	0.028644	0.024943	0.022453
fr1/rpy	0.048653	**0.032208**	0.118073	0.161495	0.034887	0.090816	0.035596	0.033032
fr1/desk	0.044288	0.061779	0.102383	0.106539	**0.033794**	0.055813	0.034608	0.034445
fr1/desk2	**0.057216**	0.065347	0.155419	0.201169	0.063362	0.076883	0.060392	0.062999
fr2/desk	0.032475	0.030671	0.108824	0.099676	**0.014821**	0.030511	0.014971	0.018258
fr3/large-cab	**0.076656**	0.331174	0.339208	0.444758	0.219498	0.145515	0.161095	0.152246
fr3/cabinet	0.075096	0.071634	0.171038	0.127961	0.068889	0.113897	**0.057714**	0.070595
fr3/str-ntex-far	0.036536	0.043633[1]	0.104202	0.245941	0.037805	0.073876	0.022864	**0.021468**
icl/lr-kt0	0.024899	0.030423	0.157722	0.203372[2]	0.076550	**0.012057**	0.029750	0.024166
icl/lr-kt1	0.037056	0.021823	0.017719	0.269095[2]	0.011977	**0.000468**	0.008820	0.008558
icl/off-kt1	0.034173	0.031098	0.160583	0.392895[2]	0.045222	0.062383	0.011250	**0.009888**

Table 1: Comparison of the ATE in $[m]$ and the RPE in $[\frac{m}{s}]$ of DVO [7], ORB-SLAM2 [9], ICPCUDA [15], D-EA [8] and our method on the RGBD TUM [15] and ICL-NUIM [6] datasets. [1] Threshold adapted. [2] Sequence could not be completed.

5 Conclusion

In this work, we introduced a robust real-time RGBD VO system that addresses the main challenges of indoor scenes. Our method jointly optimizes edge distance and point-to-plane error to achieve robustness in texture-less areas and under varying lighting conditions. While many other methods only perform well on specific scenes, the results show that the combination of edge images and depth maps is well-suited for a wide variety of scenes. We then extended our camera pose estimation to optimize a reference frame's pose with the respect to spatially close reference frames, which reduces the overall drift and closes small loops. We also demonstrated that transferring edges from higher resolution levels to the lower ones improves the spatial distribution of edge images and increases robustness.

The next step is to extend the VO system to not just optimize on spatially close frames but on all available ones, thereby implicitly closing loops. Afterwards, we want to build a full SLAM system that can be used for robust 3D indoor reconstruction.

Acknowledgements

This work was financed by the KIRAS program (no 850183, CSISmartScan3D) under supervision of the Austrian Research Promotion Agency (FFG).

References

[1] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 14(2):239–256, 1992.

[2] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 8(6):679–698, 1986.

[3] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, 1992.

[4] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 834–849, 2014.

[5] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Distance transforms of sampled functions. *Theory of Computing*, 8:415–428, 2012.

[6] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J. Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1524–1531, 2014.

[7] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Robust odometry estimation for rgb-d cameras. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3748–3754, 2013.

[8] Manohar P. Kuse and Shaojie Shen. Robust camera motion estimation using direct edge alignment and sub-gradient method. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 573–579, 2016.

[9] Raúl Mur-Artal and Juan D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, PP:1–8, 2017.

[10] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 127–136, 2011.

[11] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 652–659, 2004.

[12] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry [tutorial]. *IEEE Robotics & Automation Magazine*, 18(4):80–92, 2011.

[13] Fabian Schenk and Friedrich Fraundorfer. Robust rgbd visual odometry using machine learned edges. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2017.

[14] Frank Steinbrücker, Jürgen Sturm, and Daniel Cremers. Real-time visual odometry from dense rgb-d images. In *Proceedings of the International Conference on Computer Vision Workshops (ICCVW)*, pages 719–722, 2011.

[15] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2012.

[16] Juan J. Tarrio and Sol Pedre. Realtime edge-based visual odometry for a monocular camera. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 702–710, 2015.

[17] Xin Wang, Dong Wei, Mingcai Zhou, Renju Li, Hongbin Zha, and China Beijing. Edge enhanced direct visual odometry. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2016.

[18] Thomas Whelan, Michael Kaess, Hordur Johannsson, Maurice Fallon, John J. Leonard, and John McDonald. Real-time large-scale dense rgb-d slam with volumetric fusion. *The International Journal of Robotics Research (IJRR)*, 34(4-5):598–626, 2015.