Proceedings of the 2016 IEEE
International Conference on Robotics and Biomimetics
Qingdao, China, December 3-7, 2016

Robust Dense Visual Odometry with Boundary Pixel Suppression

Yijia He, Yue Guo, Aixue Ye, Feng Wen, and Kui Yuan

Abstract— **Pose estimation and 3D environment reconstruction are crucial for autonomous navigation in mobile robotics. Robust dense visual odometry based on a RGB-D sensor uses all pixels to estimate frame-to-frame motion by minimizing the photometric and geometric error. 3D coordinates of each pixel are calculated necessarily with its corresponding depth. However, depths of some pixels near object boundaries from RGB-D sensors are not accurate. The general robust dense visual odometry does not consider depth noise impact for photometric error and geometric error. In this paper, we construct uncertainties of photometric error and geometric error with depth noise and point out depth noise near object boundaries can significantly affect the result of motion estimation. We present a modified robust dense visual odometry with boundary pixel suppression. Publicly available benchmark datasets are employed to evaluate our system, and results showed that our method achieves higher accuracy compared with the state-of-the-art Dense Visual Odometry (DVO).**

I. INTRODUCTION

Pose estimation and 3D environment reconstruction are popular for researchers and very important for autonomous mobile robot applications. A visual odometry can estimate the trajectory for the robot with onboard cameras. Recently, some significant relevant approaches have been proposed [1], [2], in which a single RGB camera are employed to estimate the rigid body motion and create a sparse map in the environment. However, they compute the pose and environment structure up to a scale factor instead of the true metric [3].

To tackle the scale problem, multi-source systems, such as monocular vision system combined with Inertial Measurement Units (IMUs) [4], or stereo visual system [5] can be concerned. We can get absolute scale factors in systems above, while depths in poorly textured areas are still unmeasurable.

RGB-D sensors, typically, such as Asus Xtion Pro Live, can simultaneously provide dense depths and color information in the environment. Benefit from this, recently RGB-D sensors have been widely used in visual odometry [6], [7], [8]. Two mainstream methods exist for motion estimation with RGB-D sensors: feature-based method and dense method. The feature-based method aligns sparse visual features to calculate the rigid body motion between frames [9]. However, it only uses a few points in an image and excludes a set of available environment information. In contrast, many dense tracking and mapping methods based on RGB-D sensors have emerged recently. Newcombe et al. [10] proposed a variant of the Iterative Closest Points (ICP) approach only using depth values to estimate the camera pose and build 3D dense map. Instead of aligning 3D point clouds, reference [7] minimized the photometric error among consecutive RGB-D frames to align intensity images. To deal with outliers in the photometric error, Kerl et al. [8] compared several different weight functions and further improved the system accuracy using the student's t-distribution to weight the photometric error. However, photometric error based methods are unavailable on poorly textured areas, and geometric error based methods perform poorly on structureless scenes. Therefore, only the geometric error among 3D points or the photometric error among intensity images limits previous relevant works. Several approaches based on joint photometric and geometric error minimization over all pixels were proposed [11], [12]. To balance these two error cost terms, heuristical scale parameters are used to weight photometric and geometric cost functions. Kerl et al. [13] use an automatically adapted covariance matrix based on the two error terms to scale photometric and geometric cost functions. Minimization of the joint cost function achieved better generalization performance than both photometric error based methods and geometric error based methods in different type of scenes [13].

Iterative optimization for the rigid body motion is employed to minimize cost functions in all the dense visual odometry methods described above. Most researchers model the cost function only related to the rigid body motion while ignoring the depth noise. Unfortunately, as shown in Fig. 1, the depth noise can directly affect the photometric error. Engel et al. [14] considered the depth noise and compute variance-normalized photometric error by taking into account the inverse depth variance in their monocular SLAM system. Although depth noise from RGB-D sensor is usually small than those estimated by monocular SLAM, the depth noise near the object edges cannot be neglected [15]. Fig. 2 shows a RGB image and its corresponding depth map. Depth measurement noise nearby object boundaries is displayed in Fig. 2(b), where edges of the computer monitor and door are very uneven. Some pixels near these edges get wrong depth values. Nguyen et al. [16] derived a noise model for the RGB-D sensor and improved tracking resluts by using filter to smooth depth maps, but they did not analyze how the depth noise affect the geometric error and the photometric error.

The main contribution of our work is that we analyze how the depth noise affects the photometric and geometric error, point out the two error of pixels near object boundaries are

*This work was supported by National Natural Science Foundation of China under Grant No.61203328

Yijia He, Yue Guo, Feng Wen, and Kui Yuan are with Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China(email: { heyijia2013, guoyue2013, aixue.ye, feng.wen, kui.yuan }@ia.ac.cn)

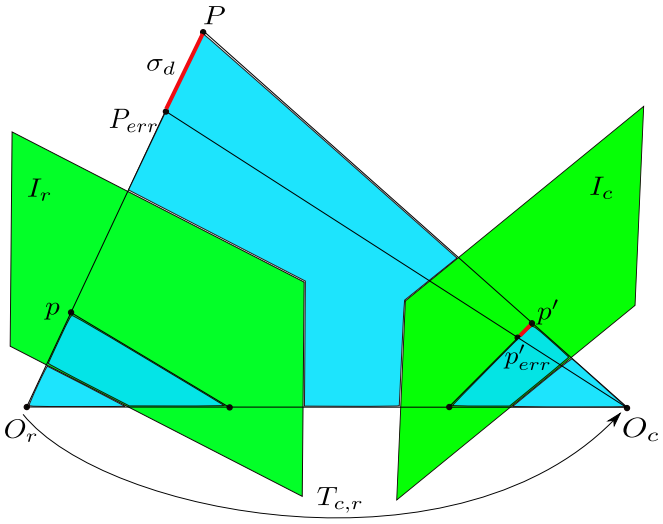


Fig. 1. In order to calculate the photometric error, we need to reconstruct a space point P from the image point p in image I_r through inverse projection. Then we project P to a point p' in image I_c. The photometric error is related to the intensity values of the two points p and p' and may be sensitve to the depth noise σ_d.

senstived to depth noise, and we present a modified robust dense visual odometry with boundary pixel suppression. The Dense Visual Odeomtry (DVO) method proposed by Kerl et al. [13] is adopted in our system as comparison, since it demonstrates high accuracy and robustness. We implement our system on TUM benchmark datasets [17]. Experimental results show that our system have a better performance than the DVO method.

II. ROBUST DENSE VISUAL ODOMETRY

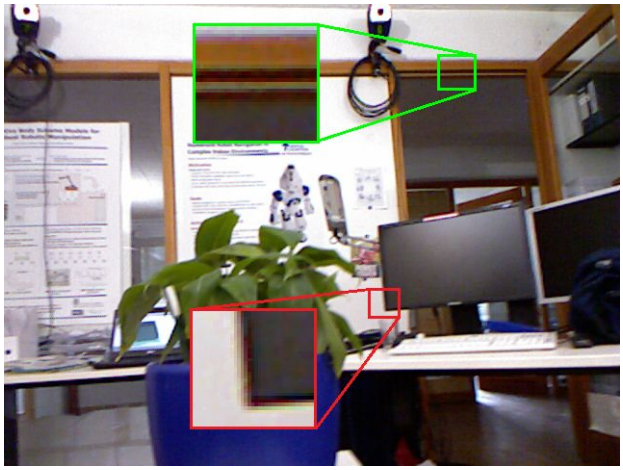
In this section, we introduce a general robust dense method whose main idea is to optimize a photometric cost function or a geometric cost function to calculate the camera pose.

As show in Fig. 1, assuming that we know the camera intrinsic matrix K, given an image point p in the reference frame image plane I_r and the correct rigid body transformation $T_{c,r}$ from the reference frame to the current frame, we can predict its corresponding pixel coordinates p' in the current frame image plane I_c. The two image points should have the same brightness based on the photo-consistency assumption, i.e. $I_r(p) = I_c(p')$. However, the rigid boy transformation is unknown and our goal is to estimate the correct motion $T_{c,r}$. It can be solved by minimizing the photometric cost function:

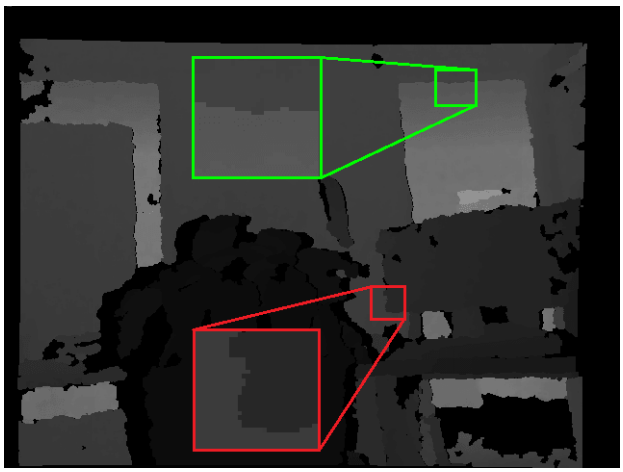
$$T_{c,r} = \arg\min_{T_{c,r}} \sum_{i}^{n} w_i r_{I,i}^2 \tag{1}$$

$$r_{I,i} = I_c(p'_i) - I_r(p_i) \tag{2}$$

where $r_{I,i}$ is the photometric error for the pixel p_i and w_i is the scale factor computed by M-estimator based on the photometric error. An image point p in the reference frame image plane I_r can be transformed to the current frame I_c with a warp function $p' = W(T_{c,r}, p, d)$. The warp function is constructed as follows:



(a) Rgb image



(b) Corresponding depth map

Fig. 2. A RGB-D frame. The depth measurements of object edges highlighted by color (red, green) squares are uneven.

Firstly, an image point $p = (u, v)^T$ projects back to a 3D space defined by the reference frame using inverse projection function π^{-1}:

$$\begin{aligned} P_r &= \pi^{-1}(p, d) \\ &= (\frac{u - c_x}{f_x} d, \frac{v - c_y}{f_y} d, d)^T \end{aligned} \tag{3}$$

where f_x, f_y are the focal lengths on x axis and y axis, and $(c_x, c_y)^T$ is the camera center coordinate. d is the depth value of pixel p read from depth map.

Then, transform the 3D point $P_r = (X_r, Y_r, Z_r)^T$ into the current frame with the rigid body motion $T_{c,r}$, i.e.

$$P_c = \mathbf{R}_{c,r} P_r + \mathbf{t}_{c,r} \tag{4}$$

$$T_{c,r} = \begin{bmatrix} \mathbf{R}_{c,r} & \mathbf{t}_{c,r} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{5}$$

with $\mathbf{R}_{c,r}$ is a 3×3 rotation matrix and $\mathbf{t}_{c,r} = (t_x, t_y, t_z)^T$ is a 3×1 vector represent a translation from reference frame to current frame.

Lastly, project the 3D point $P_c = (X_c, Y_c, Z_c)^T$ to the

image plane I_c using project function π:

$$\begin{aligned} p' &= \pi(P_c) \\ &= (\frac{X_c f_x}{Z_c} + c_x, \frac{Y_c f_y}{Z_c} + c_y)^T \end{aligned} \tag{6}$$

We get the warp function by (3), (4), (5), (6):

$$W(T_{c,r}, p, d) = \pi(\mathbf{R}_{c,r}\pi^{-1}(p, d) + \mathbf{t}_{c,r}) \tag{7}$$

Similarly, the geometric error between pixels projected by a same 3D point should be zero. We can estimate the camera motion by minimizing the geometric cost function [13], i.e.

$$T_{c,r} = \arg\min_{T_{c,r}} \sum_i^n w_i r_{d,i}^2 \tag{8}$$

$$r_{d,i} = D_c(p_i') - \lfloor \mathbf{R}_{c,r}\pi^{-1}(p_i, d_i) + \mathbf{t}_{c,r} \rfloor_3 \tag{9}$$

where D_c is the depth map provided by the current frame and $\lfloor \cdot \rfloor_3$ is the Z component of a 3D point.

Obviously, minimizing the joint photometric and geometric cost function is another way to estimate camera motion. DVO method provide a formulation of the joint cost function [13]:

$$T_{c,r} = \arg\min_{T_{c,r}} \sum_i^n w_i \mathbf{r}_i^T \mathbf{\Sigma}^{-1} \mathbf{r}_i \tag{10}$$

where $\mathbf{r}_i = (r_{I,i}, r_{d,i})^T$ is a vector combining photometric error and geometric error. The covariance matrix $\mathbf{\Sigma}$ is used to balance the photometric cost term and the geometric cost term. The weights w_i is a M-estimator, and Kerl et.al [13] uses a t-distribution:

$$w_i = \frac{v+1}{v + \mathbf{r}_i^T \mathbf{\Sigma}^{-1} \mathbf{r}_i} \tag{11}$$

Since (10) is nonlinear, we can tackle the problem using Gauss-Newton or Levenberg-Marquard algorithms. However, the rotation components of rigid body motion are non-Euclidean and the transformation matrix $T_{c,r}$ is an over-parameterized representation of rigid body motion. To avoid this problem, we use twist coordinates ξ in Lie algebra $se(3)$ during the iterative optimization. The transformation matrix $T_{c,r}$ can be represented with the exponential map:

$$T_{c,r} = \exp(\hat{\xi}) \tag{12}$$

with $\xi = (\mathbf{w}, \mathbf{v})^T \in R^6$, where $\mathbf{w}$ is angular velocity and $\mathbf{v}$ is linear velocity. $\hat{\xi}$ is skew symmetric matrix of ξ. Therefore, to calculate the optimal estimation according (10), we can obtain the normal equation [13]:

$$\sum_i^n w_i \mathbf{J}_i^T \mathbf{\Sigma}^{-1} \mathbf{J}_i \delta\xi = -\sum_i^n w_i \mathbf{J}_i^T \mathbf{\Sigma}^{-1} \mathbf{r}_i \tag{13}$$

with

$$\mathbf{J}_i = (\frac{\partial r_{I,i}}{\partial \xi}, \frac{\partial r_{d,i}}{\partial \xi})^T \tag{14}$$

The estimation $T_{c,r}$ can be updated by iteratively solve the normal equation (13):

$$T_{c,r}^t = \exp(\hat{\delta\xi}) T_{c,r}^{t-1} \tag{15}$$

As the robust dense visual odometry method described above, when calculating the error using pixels with valid depths, the photometric cost function and the geometric cost function are thought only related to the rigid body motion. However, it is worth noting that the depth noise especially the depth outlier may directly affect the photometric and geometric error. The cost term for some pixels with depth noise may not decrease by iterative optimizing the camera pose.

III. ROBUST DENSE VISUAL ODOMETRY WITH BOUNDARY PIXEL SUPPRESSION

Although depth noise can affect the photometric and geometric error, the influence can be different from different pixels with same depth noise. In this section, firstly, we analyze how the depth noise affect the photometric and geometric error. Then, we point out which pixels their depth noise can significantly affect optimization. At last, we propose DVO with boundary pixel suppression.

A. Impact of Depth Noise on Photometric and Geometric Error

Photometric error can be seen as a function about depth measurements based on (2) and (7), i.e., $r_{I,i} = f(d_i)$. As depth noise fit Gaussian distributions $n_d \in \mathcal{N}(0, \sigma_d^2)$ [16], the uncertainty of the photometric error contributed by depth measurements uncertainty σ_d can be computed using covariance propagation:

$$\sigma_{r_{I,i},d_i}^2 = (J_{r_{I,i},d_i})^2 \sigma_{d_i}^2 \tag{16}$$

where $J_{r_{I,i},d_i}$ is the Jacobian value $\frac{\partial r_{I,i}}{\partial d_i}$. It can be calculated by the chain rule as

$$J_{r_{I,i},d_i} = \frac{\partial(I_c(p') - I_r(p))}{\partial d_i} \tag{17}$$

$$= \left.\frac{\partial I_c(p')}{\partial p'}\right|_{p'=W(T_{c,r},p_i,d_i)} \frac{\partial W(T_{c,r}, p_i, d_i)}{\partial d_i} \tag{18}$$

The first part of (18) is the derivative of the current image I_c at the warped position p'. The second part is the derivative of the warped pixel point at the depth measurement. The Jacobian (18) can be evaluated as:

$$\begin{aligned} J_{r_{I,i},d_i} = \frac{1}{Z_r Z_c^2}(&\bigtriangledown I_{c,u} \cdot f_x \cdot (X_c t_z - Z_c t_x) \\ &+ \bigtriangledown I_{c,v} \cdot f_y \cdot (Y_c t_z - Z_c t_y)) \end{aligned} \tag{19}$$

Similarly, from (9), the uncertainty of the geometric error $\sigma_{r_{d,i},d_i}^2$ contributed by depth measurements uncertainty $\sigma_{d_i}^2$ can be computed as

$$\sigma_{r_{d,i},d_i}^2 = (J_{r_{d,i},d_i})^2 \sigma_{d_i}^2 \tag{20}$$

with

$$\begin{aligned} J_{r_{I,i},d_i} = \frac{1}{Z_r Z_c^2}(&\bigtriangledown D_{c,u} \cdot f_x \cdot (X_c t_z - Z_c t_x) \\ &+ \bigtriangledown D_{c,v} \cdot f_y \cdot (Y_c t_z - Z_c t_y)) + \frac{1}{Z_r}(t_z - Z_c) \end{aligned} \tag{21}$$

Algorithm 1 Boundary Pixel Detection

Input: reference depth map D
Output: indicator vector $\mathcal{I}$

for each depth image pixel $u \in D$ **do**
  if $D(u) \neq 0$ **then**
    $\triangle d_x \leftarrow$ the derivative x of depth map at pixel u
    $\triangle d_y \leftarrow$ the derivative y of depth map at pixel u
    $G \leftarrow \sqrt{(\triangle d_x^2 + \triangle d_y^2)}$
  end if
  if $G > threshold$ **then**
    $\mathcal{I}_u \leftarrow 1$
  else
    $\mathcal{I}_u \leftarrow 0$
  end if
end for

From (16) and (20), we model the uncertainty of the photometric error and the geometric error with the uncertainty of depth measurement. From (19) and (21), variances of the photometric error and the geometric one are not only related to the depth variance, but also related to the image gradients. It means that the same depth measurement with same noise may influent cost function quite differently. If depth noise remarkably affects the photometric and geometric cost function, then we need to remove it during the optimization.

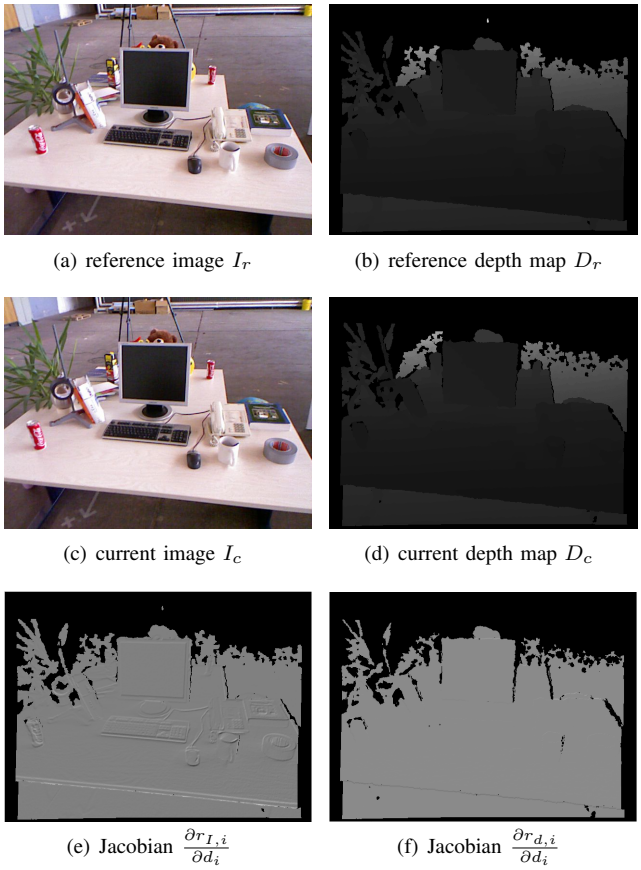
B. DVO With Boundary Pixel Suppression

Due to the gradients of RGB image and depth image near their edges are large than those in smooth regions. Variances of photometric and geometric error near the object edges affect by depth noise may be large. Jacobians for depth values d are shown in Fig. 3(e) and Fig. 3(f), the absolute Jacobian value of pixels near edges are larger than the value of smooth regions. These errors resulted from depth noise will reduce the precision of the rigid transformation estimated by DVO. However, in the smooth region of image, gradients are close to zero and depth noise are also small, their influence to cost function can be neglected. As a result, we can suppress pixels near object boundaries where depth may have large noise. For every pixel with a valid depth, we use an indicator function to decide whether it should be suppressed or not during optimization. To suppress pixels near object boundaries that significantly affect the cost function in (10), we add the indication factor $\mathcal{I}_i$ in the cost function :

$$T_{c,r} = \arg\min_{T_{c,r}} \sum_{i}^{n} \mathcal{I}_i w_i \mathbf{r}_i^T \mathbf{\Sigma}^{-1} \mathbf{r}_i \qquad (22)$$

The (22) indicates that instead of using all pixels to compute the rigid transformation, we only use the pixels less affected by depth noise.

The boundary pixel detection process described in Algorithm 1. At each iteration, a gradient of the pixel in depth map (if the depth value is non-zero) is computed using Sobel operator. If the gradient is larger than a certain threshold, then this pixel will be identified as a bad pixel to be suppressed during optimization. In our implementation, we take $threshold = 0.2m$. As shown in Fig. 4, the pixels (red) near the object boundaries with large depth changes are should suppressed during optimization, and other pixels such as the keyboard and the mouse are saved to estimate the rigid body motion.



(a) reference image I_r (b) reference depth map D_r
(c) current image I_c (d) current depth map D_c
(e) Jacobian $\frac{\partial r_{I,i}}{\partial d_i}$ (f) Jacobian $\frac{\partial r_{d,i}}{\partial d_i}$

Fig. 3. We compute the photometric error and the geometric error between two frames (a-d). (e), (f) is Jacobians for depth values d. The medium gray in (e) and (f) means the pixel value is 0, the brighter the greater value and the darker the smaller value. Black represents pixels depth value are invalid.

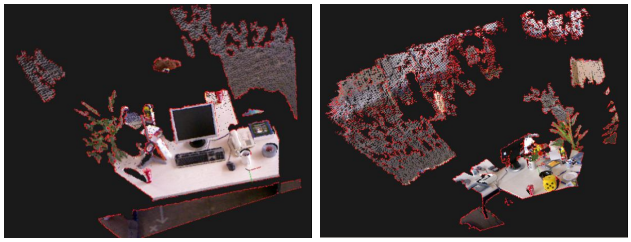


Fig. 4. Examples of pixels suppression (red) in the fr2/desk dataset.

IV. EXPERIMENTAL EVALUATION

In this section, we implement the dense visual odometry with boundary pixel suppression, and compare its accuracies with those of the dense visual odometry (DVO) [13] on the TUM RGB-D datasets [17]. Every dataset contains RGB-D images with 640×480 resolution and a ground truth trajectory.

TABLE I
SOME IMPORTANT PARAMETERS FOR DVO ALGOITHM

Parameters	Values
coarsest level	3
finest level	1
max iterations	100
precision	0.0000005
use weighting	true
weight method	t-distribution

TABLE II
RMSE OF TRANSLATIONAL DRIFT (PRE) IN METERS PER SECONDE USING DIFFERENT METHODS

Dataset	DVO	DVO with boundary pixel suppression
fr1/desk	0.055296	**0.041487**
fr1/desk2	0.068586	**0.051055**
fr1/floor	0.096211	**0.078101**
fr1/plant	0.042330	**0.028905**
fr1/360	0.124606	**0.094262**
fr1/xyz	0.029056	**0.020573**
fr2/desk	0.031591	**0.016062**

In order to get a satisfactory performance, parameters of the DVO algorithm are adopted from [8]. DVO and DVO with boundary pixel suppression share parameters which are in detail shown in Table I. Specifically, values of finest level and coarset level mean that we use three levels of the image pyramid to estimate the rigid body motion, and the finest level image resolution is 320×240, The precision parameter is a threshold used to determine whether the iteration converges during the optimization. As t-distribution is a good choice for the weight calculation than Tukey function and Huber function [8], we compute the weight function based on the t-distribution according (11). We measure the drift using the relative pose error for pose pairs and calculate the RMSE of translation drift in meters per second (m/s). Table II shows the results for the two dense visual odometry system. The DVO with boundary pixel suppression works better on the all five benchmark datasets than DVO algorithm. These results indicate that depth values of pixels near object boundaries can directly affect the dense visual odometry system.

To display more intuitively the improvement by DVO with boundary pixel suppression, we select the experiment results on fr2/desk dataset for a detailed analysis. In this dataset, the intensity of image changes slightly, so it becomes suitable to calculate the photometric error, while the corresponding valid depth measurements are less [18]. Fig. 5 shows the translation drift of the relative pose error and Fig. 6 shows the estimated trajectory with the ground truth. In Fig. 5 there are some time steps can not find the overlap between the estimated pose and the ground truth, such as $t \in (23, 46)s$. However, the error can be calculated from measurements at other time steps, we can find that the DVO algorithm with boundary pixel suppression have a smaller translation drift at almost all the time steps with fewer exceptions. In Fig. 5 near the time $t = 20s$ and $t = 50s$ our algorithm get larger errors than the DVO, which means that when using our boundary pixel detection algorithm, some useful pixels are falsely removed.

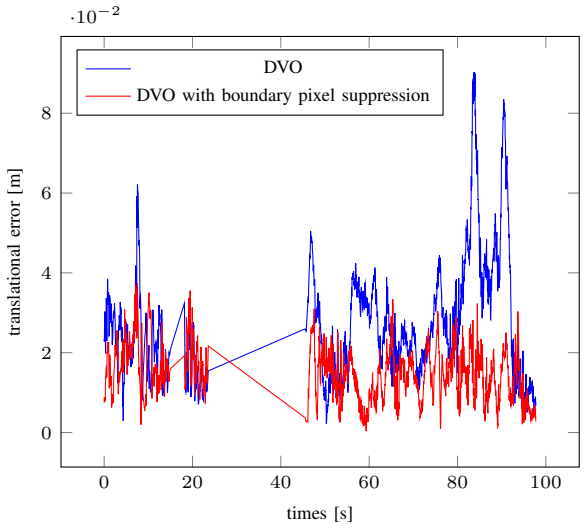


Fig. 5. Translation drift.

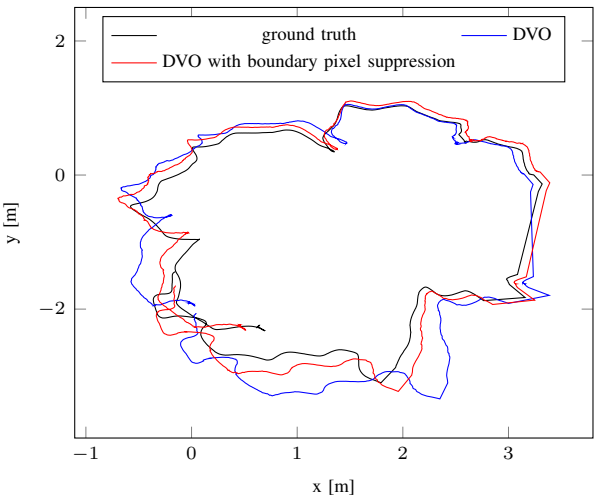


Fig. 6. Camera trajectory.

V. CONCLUSION

In this paper, we propose the dense visual odometry with boundary pixel suppression system. We analyze how the depth noise affects the photometric and geometric error for the rigid body motion estimation, and point out that pixels near object boundaries should be removed if their corresponding depth values reveal large uncertainty. Experimental results evaluated on the TUM benchmark datasets demonstrate that our system significantly improves the accuracy compared with the DVO algorithm. However, we remove pixels roughly so that some good pixels are falsely removed. In the near future, we plan to improve the boundary pixel removing process more reasonably and accurately.

REFERENCES

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

[2] R. Mur-Artal, J. Montiel, and J. D. Tardós, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[3] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part i: The first 30 years and fundamentals," *IEEE Robotics and Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[4] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[5] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme." in *Intelligent Vehicles Symposium*, 2010, pp. 486–492.

[6] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments," in *In the 12th International Symposium on Experimental Robotics (ISER*. Citeseer, 2010.

[7] F. Steinbrücker, 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*. IEEE, 2011, pp. 719–722.

[8] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for rgb-d cameras," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3748–3754.

[9] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-d mapping with an rgb-d camera," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 177–187, 2014.

[10] 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 *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*. IEEE, 2011, pp. 127–136.

[11] T. Tykkälä, C. Audras, and A. I. Comport, "Direct iterative closest point for real-time visual odometry," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2050–2056.

[12] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. McDonald, "Robust real-time visual odometry for dense rgb-d mapping," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 5724–5731.

[13] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 2100–2106.

[14] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision*. Springer, 2014, pp. 834–849.

[15] M. Camplani, T. Mantecón, and L. Salgado, "Depth-color fusion strategy for 3-d scene modeling with kinect," *IEEE transactions on cybernetics*, vol. 43, no. 6, pp. 1560–1571, 2013.

[16] C. V. Nguyen, S. Izadi, and D. Lovell, "Modeling kinect sensor noise for improved 3d reconstruction and tracking," in *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*. IEEE, 2012, pp. 524–530.

[17] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 573–580.

[18] Z. Fang and Y. Zhang, "Experimental evaluation of rgb-d visual odometry methods," *International Journal of Advanced Robotic Systems*, vol. 12, 2015.