# WSDF: Weighting of Signed Distance Function for Camera Motion Estimation in RGB-D Data

Pham Minh Hoang, Vo Hoai Viet, Ly Quoc Ngoc
Department of Computer Vision and Robotics, University Of Science, VNU-HCM, Viet Nam

***Abstract*—With the recent advent of the cost-effective Kinect, which can capture real-time high-resolution RGB and visual depth information, has opened an opportunity to significantly increase the capabilities of many automated vision based recognition including object/action classification, 3D reconstruction, etc… In this work, we address the camera motion estimation which is an important phase in 3D object reconstruction system based on RGB-D data. We segment objects by thresholding algorithm based on depth data and propose the weighting function for SDF that is called WSDF. The problem of minimizing of this function is solved by Gauss-Newton methods. We systematically evaluate our method on TUM dataset. The experimental results are measured by ATE and RPE that evaluate both global and local consistency of camera motion estimation algorithm. We demonstrate large improvements over the state-of-the-art methods on both plant and teddy3 objects and achieve the best ATE as 0.00564 and 0.0182 and the best RPE as 0.00719 and 0.00104, respectively. These experiments show that the proposed method significantly outperforms state-of-the-art techniques.**

***Keywords*—*RGB-D data; 3D Reconstruction; SDF; Camera Motion Estimation***

## I. Introduction

Reconstructing 3D object is an interesting and challenging problem in computer vision. It has attracted many research efforts from the computer vision community in recent decades for its high potential applications such as game, SLAM, medical technology, virtual reality, and robotics. Due to its wide range of applications, 3D object reconstruction has attracted much attention in recent years [2]. Generally speaking, 3D object reconstruction framework contains three main steps namely object segmentation, camera motion estimation, and surface reconstruction (see in Fig. 2). Object segmentation is to identity the object region in images that can achieve by using the algorithms such as kmean, mean shift, ostu ... Camera motion estimation aims to represent the movement of object over frames. The result of this phase is point cloud that describe object in 3D space. Surface reconstruction focus on reconstructing the surface mesh… In this work, we only focus the problem of the camera motion estimation phase. We use the Ostu and thresholding algorithm for object segmentation.

The advent of affordable RGB-D sensors has opened up a whole new range of applications based on the 3d perception of the environment by computers, which includes the creation of a virtual 3d representation of real objects. Compared with conventional color data, depth maps provide several advantages, such as the ability of reflecting pure geometry and shape cues, or insensitive to changes in lighting conditions. Moreover, the range sensor provides 3D structural information of the scene and objects. These characteristics will be helpful for object segmentation and camera motion estimation.

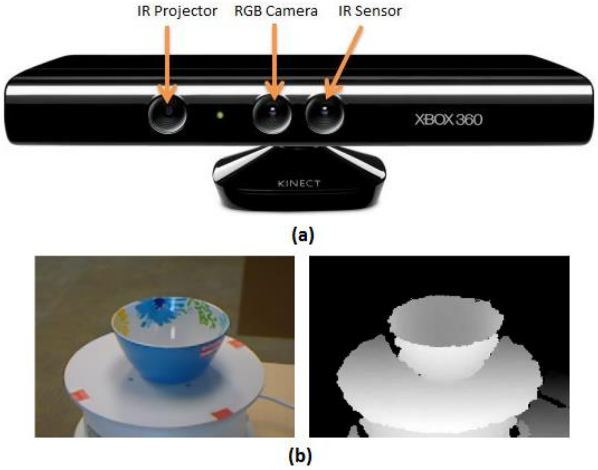


Fig. 1. Illustration of 3D camera and RGB-D data: a) Microsoft Kinect Device; b) an object example of RGB-D data is captured by Kinect

In this manuscript, we proposed the weighting parameters for SDF that was proposed at [4, 5] to improve the performance of camera motion of 3D reconstruction system based on RGB-D data. The main contributions of this paper are summarized as follows: Firstly, we apply the weighting approach for SDF for camera motion estimation based on RGB-D data. Secondly, we systematically evaluate our WSDF on four challenging datasets.

The rest of this paper is organized as follows: Section II gives a concise review of existing works on camera motion estimation for 3D reconstruction. Section III presents signed distance function for camera motion estimation. Section IV introduces our improvement for camera motion estimation. Section V presents action classification. Section V shows the experiment results on relevant benchmarks. Finally, section VI draws conclusions of our work and indicates future studies.

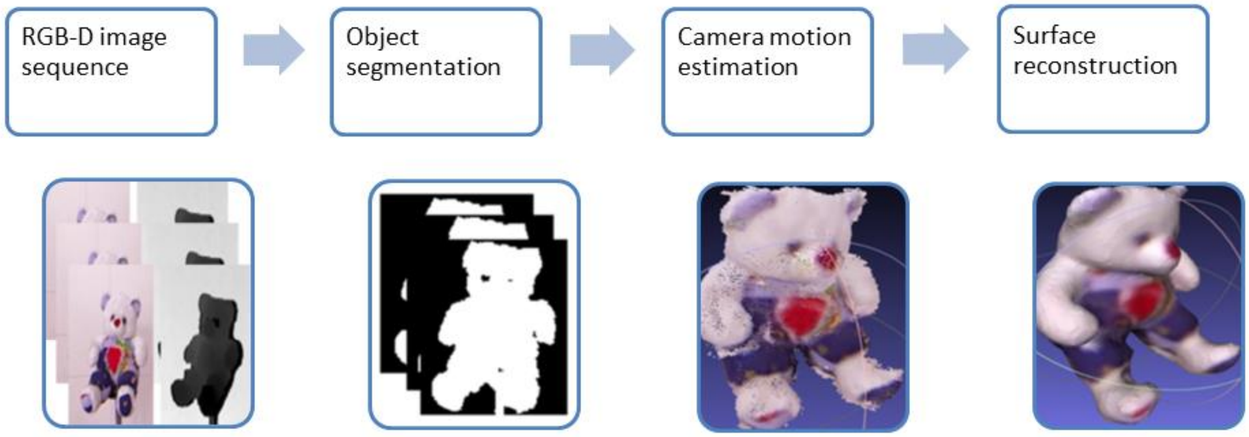


Fig. 2. Flowchart of 3D object reconstruction system in RGB-D data

## II. Literature Review

Comprehensive reviews of the previous studies can be found in [2]. Our discussion in this section is restricted to a few influential and relevant parts of literature, with a focus on camera motion estimation based on RGB-D data.

The camera motion estimation aims to find the affine transformations to convert point clouds in local frames into global coordination and integrate them into a final point cloud for object representation. These transformations represent the movement of camera from the first frame to the last frame. The earliest approaches focus on finding the affine transformation between two consecutive frames. In [13], the author use ICP algorithm to find affine between two consecutive frames based on the features are extracted from them. Another famous method are called Kinect Fusion [10, 11], the method build the Signed Distance Function (SDF) and use the function for initializing the point cloud for each frame. Then, ICP algorithm is used to find affine transform in the next frame. However, the integration of affine transformations between two consecutive frames makes the errors that accumulated to misleading in the following frame is greater. The difference from Kinect Fusion, these methods in [7, 10] estimate directly the affine transformation by minimizing the RSME of SDF, then updating SDF based on the computed transformation. In [8], the authors build SDF based on Octree to reduce memory and computational cost. These methods that use ICP algorithm focus on minimizing the point cloud, some methods [3 , 4 , 5, 6, 10] minimize the RGB-D of SDF between two consecutive frames. In [9], the method finds corresponding points between two consecutive frames and minimizes the total of the distance of these corresponding points.

In this paper, we propose the camera motion estimation based on SDF in [5, 6]. However, we improve SDF by adding the weighting function in [3] that is called WSDF. And, the problem of minimizing for this function is solved by Gauss-Newton method.

## III. Background of Camera Motion Estimation

In this session, we present the camera motion estimation over frames from RGB-D sequences. The inputs of this phase are local point clouds are extracted from RGB and depth of each frame $P_i = \{x_j\}$ with $x_j$ is 3D vertex of point cloud $P_i$. The problem is to find affine transformation to transfer the local point cloud at i-th frame from local coordinate to global coordinate. The affine transformation also describes motion of camera over frames, so this phase is called camera motion estimation. In [4, 5], Bylow et al. introduced the method of camera motion estimation based on signed distance function (SDF).

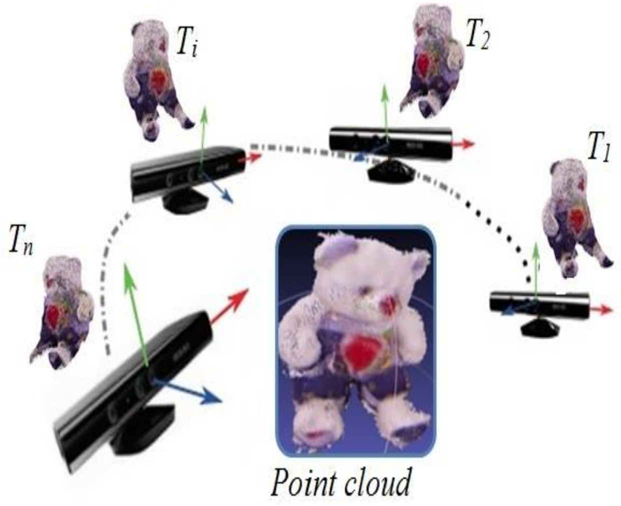


Fig. 3. An example of camera motion estimation

### A. Signed Distance Function

The SDF of given surface $\theta(\mathrm{x}): \mathrm{R}^3 \rightarrow \mathrm{R}$. This function returns for any point $\mathrm{x} \in \mathrm{R}^3$ the signed distance from x to the surface. The SDF have four properties as follows:

- If $x$ is outside the surface then $\theta(x) > 0$.
- If $x$ is inside the surface then $\theta(x) < 0$.
- If $x$ is on the surface then $\theta(x) = 0$.
- If $x$ is nearer the surface then $\theta(x)$ is smaller.

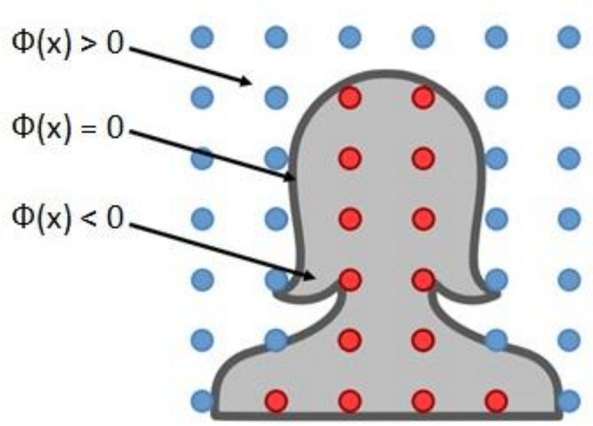


Fig. 4. Illustration of SDF for object's surface

### B. Affine Transformation

An affine transformation consists of two components: a three-dimension square matrix $R_i$ and a three-dimension translation vector $t_i$. We assume that we already have the surface of object represented by a signed distance function (SDF). For each vertex of point cloud in local coordinate, our goal is the transformed point lies as close as possible to the object surface. It means $[\theta(R_i x_j + t_i)]^2$ is as smaller as possible. We must find $R_i$ and $t_i$ such that the function $E(R_i, t_i) = \sum_i [\theta(R_i x_j + t_i)]^2$ is minimized.

Considering the function $T_i = [R_i|t_i]$ consists of 12 parameters. However, the limitation of problem only needs the rotation and translation that can be solved by 6 parameters with three parameters for rotation $(\omega_1, \omega_2, \omega_3)$ and three parameters for translation $(t_1, t_2, t_3)$. Therefore, $T_i$ can be written as a vector of 6 dimensions $\xi_i = (\omega_1, \omega_2, \omega_3, t_1, t_2, t_3)$ and $E(R_i, t_i)$ is also written as $E(\xi_i) = \sum_i [\theta_j(\xi_i)]^2$. To minimize this function, Bylow et al. [4, 5] used Gauss-Newton algorithm.

### C. Update the SDF and the colors

The SDF is not traditional formula function due to it is formed by dividing the space into grids in 3D. Each node in 3D grid is called voxel. If a point does not match to voxel, SDF value of x is obtained based SDF value of the nearest neighbor voxels. So, the objective in this step is to compute SDF for each voxel.

Assume that $v^G$ is global coordinate of each voxel. Based on the estimated pose $T_i$, we can transfer to local coordinate of frame i as $v^L = R^T(v^G - t)$. According to camera model, with the focal lengths $f_x$ and $f_y$ and principal point $(c_x, c_y)$,we can project 3D point $v^L = (v_x^L, v_y^L, v_z^L)$ to image plane by projection

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

Let (i, j) be pixel coordinate of projected point $v^L$ in image and I(d) be the corresponding depth value at $(i, j)$. We can compute distance $d(v^L)$ of the depth of voxel and the depth value at $(i, j)$.

$$d(v^L) = z - I_d(i, j)$$

Since the distance $d(v^L)$ is a rough approximation which can get arbitrary wrong, we follow the standard approach to reduce the impact of bad measurements by truncating the measured distance if $|d| > \delta$ for some threshold $\delta$ as follows:

$$d = \begin{cases} -\delta & if \quad d < -\delta \\ d & if \quad |d| \le \delta \\ \delta & if \quad d > \delta \end{cases}$$

For each frame, we can compute the distance $d_i$ of each voxel at frame $i^{th}$. The SDF value of a voxel can be obtained by weighted average of these distances as follows:

$$\theta(v) = \frac{\sum_i w_i d_i}{w_i}$$

However, this is not enough to decrease the impact of bad measurements. We do also have a higher uncertainty when the voxel lies behind the surface. To handle this, we weight the measurements using the following weight function as follows:

$$w(d) = \begin{cases} 1 & if \quad d < \varepsilon \\ e^{-\sigma(d-\varepsilon)^2} & if \quad d \ge \varepsilon \ and \ d \le \delta \\ 0 & if \quad d > \delta \end{cases}$$

Therefore, we can update SDF of each voxel as follows:

$$\theta = \frac{W\theta + w_i d_i}{W + w_i}$$

$$W = W + w_i$$

From the RGB image and each voxel the color is estimated as the formula as follows:

$$R \to \frac{W^c R + w_i^c r}{W^c + w_i^c}$$

$$G \to \frac{W^c G + w_i^c g}{W^c + w_i^c}$$

$$B \to \frac{W^c B + w_i^c b}{W^c + w_i^c}$$

Where $w_i^c$ the weight of color for new measurement, $w_i^c$ is used as $w_i^c = w_i cos\alpha$ where $\alpha$ is the angle between the ray and the principal axis to give more weight to pixels whose normal is pointing towards the key frame.

## IV. Weighting of Signed Distance Function

To increase the accuracy for the problem of minimize $E(\xi_i)$, we propose the weighting function $w(r_i)$ for SDF that is called WSDF where $r_i(\xi) = \theta_j(\xi_i)$. According to [3], the weighting function $w(r_i)$ is defined as follows:

$$w(r_i) = \frac{\nu + 1}{\nu + \left(\frac{r_i}{\sigma}\right)^2}$$

The points are near surface can more accurately describe the shape of the object than the points are far from surface. So, the w($r_i$) will increase when $r_i$ increase, this means the weights of the points are near the surface will be higher than the weight of the points are far from the surface. Meanwhile, we have to find $\xi_i$ by solving the optimization of the non-linear function $\xi_i = \sum_i \operatorname{argmin}(w(r_i)(r_i(\xi))^2)$. We apply the Gauss-Newton method to solve the problem. The initialization for $\xi = \xi^{(0)}$, and $\xi$ at each loop is computed by the following formula: $\xi^{(k+1)} = \xi^{(k)} - (J^T W J)^{-1} J^T W r(\xi^{(k)})$ where J is Jacobian matrix $J = \left[\frac{\partial r}{\partial \omega_1} \frac{\partial r}{\partial \omega_2} \frac{\partial r}{\partial \omega_3} \frac{\partial r}{\partial t_1} \frac{\partial r}{\partial t_2} \frac{\partial r}{\partial t_3}\right]$ and $W$ is matrix that is created by main diagonal of $w(r_i)$. The loop will end when $\| \xi^{(k+1)} - \xi^{(k)} \|_\infty$ is enough small or the number of loop achieve the limitation. We adopt $\nu = 5$ based on the experiment, $\sigma^2$ at each loop is computed as follows:

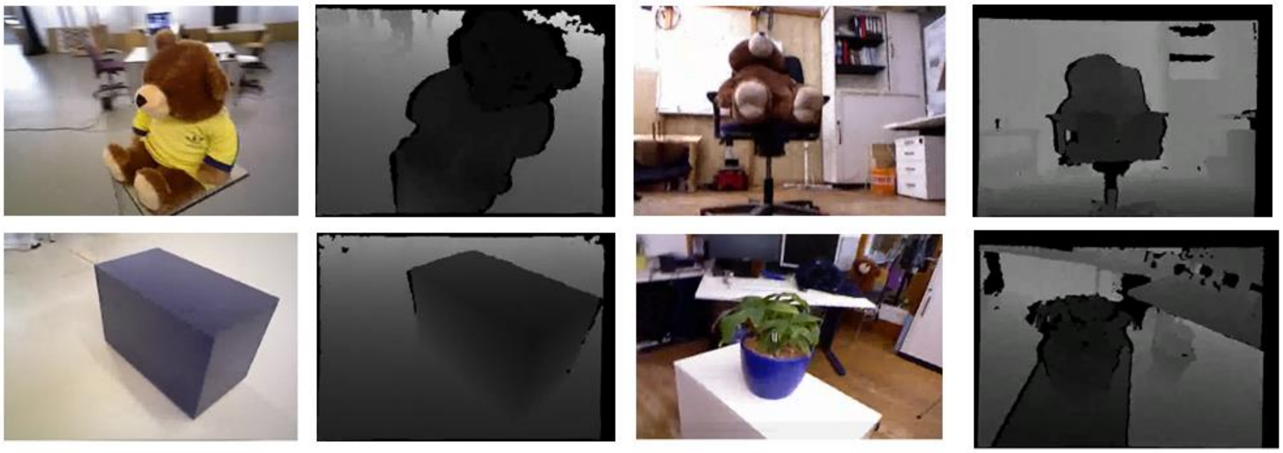


Fig. 5. Some RGB and depth frames from TUM dataset

$$\sigma^2 = \frac{1}{n} \sum_i r_i^2 \frac{\nu + 1}{\nu + \left(\frac{r_i}{\sigma}\right)^2}$$

The end of the process, we have $T_i$ is computed by a vector of 6 dimensions of $\xi_i$. Then, we update SDF to compute for the next frame.

## V. Expriment Results

### A. Dataset

We also evaluated our approach on the TUM 3D object reconstruction RGB-D benchmark dataset [12]. In this wok, we use plant and teddy 3 to measure the errors of our approach. Fig. 5 shows some examples of the TUM dataset.

### B. Measurement Evaluation

#### 1) Relative pose error (RPE)

The relative pose error [8] measures the local accuracy of the trajectory over a fixed time interval Δ. Therefore, the relative pose error corresponds to the drift of the trajectory which is in particular useful for the evaluation of visual odometry systems. We define the relative pose error at time step i as follow:

$$E_i = (Q_i^{-1} Q_{i+\Delta})^{-1} (P_i^{-1} P_{i+\Delta})$$

From a sequence of n camera poses, we obtain in this way m = n − Δ individual relative pose errors along the sequence. From these errors, we propose to compute the root mean squared error (RMSE) over all-time indices of the translational component as follows:

$$RMSE(E_{1:n}, \Delta) = \left( \frac{1}{m} \sum_{i=1}^{m} \| trans(E_i) \|^2 \right)^{1/2}$$

where $trans(E_i)$ refers to the translational components of the relative pose error $E_i$.

#### 2) Absolute trajectory error (ATE)

The absolute trajectory error [8] measures the global consistency can be evaluated by comparing the absolute distances between the estimated and the ground truth trajectory. As both trajectories can be specified in arbitrary coordinate frames, they first need to be aligned.

This can be achieved in closed form using the method of Horn [1], which finds the rigid-body transformation S corresponding to the least-squares solution that maps the estimated trajectory $P_{1:n}$ onto the ground truth trajectory $Q_{1:n}$. Given this transformation, the absolute trajectory error at time step i can be computed as follows:

$$F_i = Q_i^{-1} S P_i$$

Similar to the relative pose error, we propose to evaluate the root mean squared error over all time indices of the translational components as follows:

$$RMSE(E_{1:n}) = \left( \frac{1}{m} \sum_{i=1}^{m} \|trans(F_i)\|^2 \right)^{1/2}$$

where $trans(F_i)$ refers to the translational components of the relative pose error $F_i$.

### C. Experimental Results

We firstly evaluate our proposed approach on the benchmark objects in TUM dataset. Then we compare our experimental results to the-state-of-the-art methods to prove the effectiveness and robust of the proposed method.

In this research, we focus on camera motion estimation for 3D object reconstruction. Our approach based on object segmentation and SDF in RGB-D data. More specific, we use depth data for segmenting object and proposed the weighting function for SDF and solve the problem of minimizing for this function by using Gauss-Newton method. We evaluate our method by ATE and RPE that evaluate both global and local consistency. Moreover, we also evaluate many different time intervals to have deeper in understanding of the problem of camera motion estimation.  Table I and II give our experimental results on plant and teddy3 objects. However, the same approach has the different result on the different objects. This is the different characteristics of these datasets. The plant object have the slow movement more than teddy3 object. In addition, teddy3 object have structure of surface more complexity than plant object.

Table III, IV, V and VI compare our experimental results with state-of-the-art results on TUM dataset. We achieve better than Bylow's approach on both plant and teddy3 object. Our method is more efficient on both global and local consistency (can see Fig. 6). These results show that our approach is robust for camera motion estimation. To have these promising results based on updating SDF with the weighting function to get more accuracy when estimate the motion between two consecutive frames.

TABLE I. EXPERIMENTIAL RESULTS ON PLANT OBJECT

| **Frames** | **Measurement (m)** | |
|---|---|---|
| | ***ATE*** | ***RPE*** |
| **10** | 0.00654 | 0.0182 |
| **20** | 0.00856 | 0.0209 |
| **30** | 0.00809 | 0.0294 |
| **40** | 0.01024 | 0.0504 |
| **50** | 0.01444 | 0.0673 |

TABLE II. EXPERIMENTIAL RESULTS ON TEDDY3 OBJECT

| **Frames** | **Measurement (m)** | |
|---|---|---|
| | ***ATE*** | ***RPE*** |
| **10** | 0.00719 | 0.00104 |
| **20** | 0.007 | 0.01152 |
| **30** | 0.00813 | 0.01387 |
| **40** | 0.01433 | 0.02149 |
| **50** | 0.0225 | 0.03348 |

TABLE III. COMPARISION WITH THE STATE OF THE ARE METHOD ON PLANT OBJECT USING ATE

| **Frames** | **Methods** | |
|---|---|---|
| | ***Our approach*** | ***Bylow [4]*** |
| **10** | 0.00654 | 0.00937 |
| **20** | 0.00856 | 0.01168 |
| **30** | 0.00809 | 0.01193 |
| **40** | 0.01024 | 0.01605 |
| **50** | 0.01444 | 0.02335 |

TABLE IV. COMPARISION WITH THE STATE OF THE ARE METHOD ON PLANT OBJECT USING RPE

| **Frames** | **Methods** | |
|---|---|---|
| | ***Our approach*** | ***Bylow [4]*** |
| **10** | 0.0182 | 0.0278 |
| **20** | 0.0209 | 0.0338 |
| **30** | 0.0294 | 0.0503 |
| **40** | 0.0504 | 0.0847 |
| **50** | 0.0673 | 0.119 |

TABLE V. COMPARISION WITH THE STATE OF THE ARE METHOD ON TEDDY3 OBJECT ON ATE

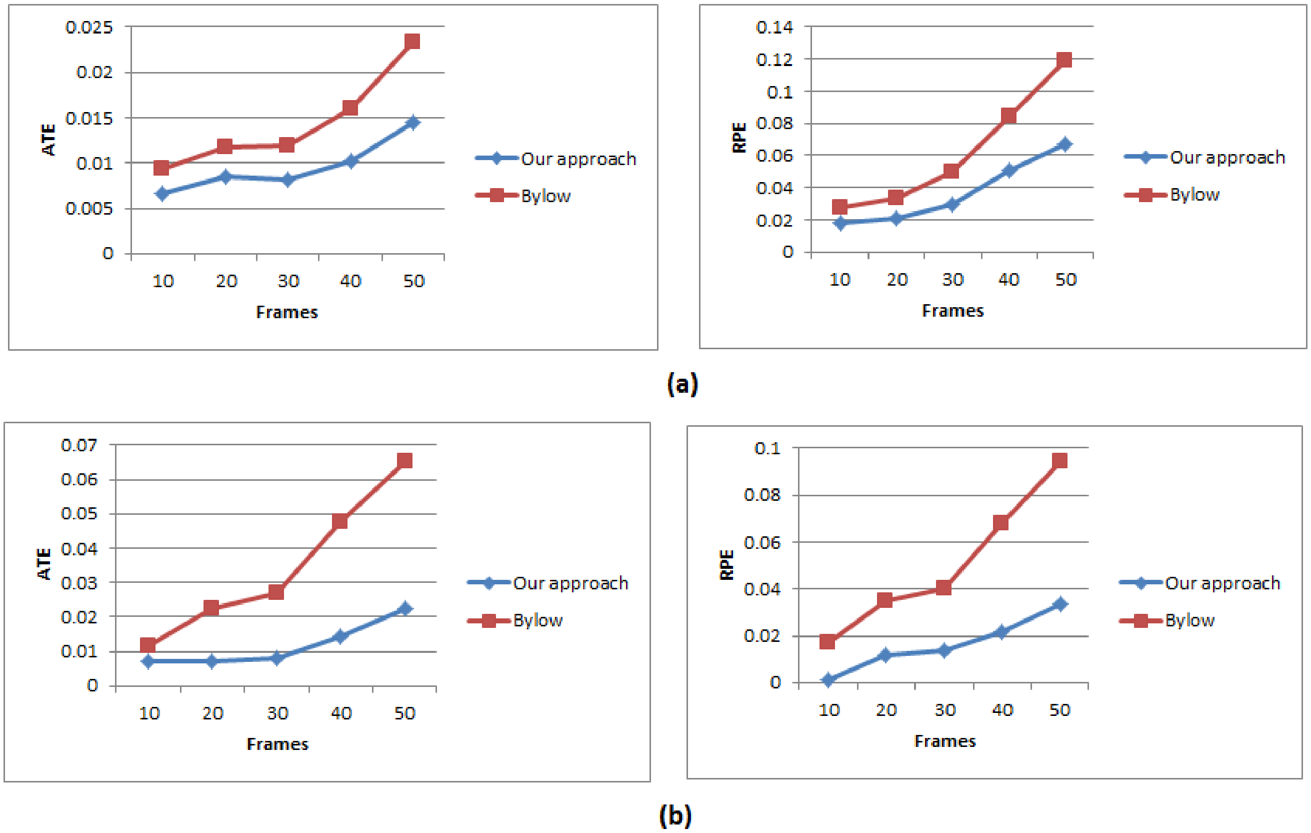
| **Frames** | **Methods** | |
|---|---|---|
| | ***Our approach*** | ***Bylow [4]*** |
| **10** | 0.00719 | 0.0114 |
| **20** | 0.007 | 0.0224 |
| **30** | 0.00813 | 0.027 |
| **40** | 0.01433 | 0.0476 |
| **50** | 0.0225 | 0.0654 |

TABLE VI. COMPARISION WITH THE STATE OF THE ARE METHOD ON TEDDY3 OBJECT ON RPE

| **Frames** | **Methods** | |
|---|---|---|
| | ***Our approach*** | ***Bylow [4]*** |
| **10** | 0.00104 | 0.0173 |
| **20** | 0.01152 | 0.0346 |
| **30** | 0.01387 | 0.0401 |
| **40** | 0.02149 | 0.0679 |
| **50** | 0.03348 | 0.0943 |

## VI. CONCLUSION

In this work, we present a novel approach for camera motion estimation based on SDF in 3D object reconstruction using RGB-D data. In order to segment object, we use depth data based on threshold method. To estimate camera motion, we proposed a weighting function is added to SDF function is called WSDF to improve the performance of camera motion estimation phase. And, the WSDF is minimized by Gauss-Newton method. We systematically evaluate our approach on benchmark dataset. The experiments are measured on both ATE and RPE that assess the global and local consistency of the camera motion estimation. The experimental results show that our proposed approach achieves superior performance to the state-of-the-art algorithm on TUM dataset.



(a)

(b)

Fig. 6. Comparison with the baseline method in [4]: a) plant object; b) teddy3 object

In the future, we will consider SIFT or SIFT-flow for camera motion estimation based on RGB data to have better the performance of the system.

## ACKNOWLEDGMENT

This research is funded by Vietnam National University, Ho Chi Minh City (VNU-HCM) under grant number *B2014-18-02*.

## REFERENCES

[1] B. Horn, “Closed-form solution of absolute orientation using unit quaternions,” Journal of the Optical Society of America A, vol. 4, pp. 629–642, 1987.

[2] Berger, M., Tagliasacchi, A., Seversky, L., Alliez, P., Levine, J., Sharf, A.: State of the art in surface reconstruction from point clouds. In: Proceedings of Eurographics—Eurographics star reports, vol. 1, pp. 161–185, 2014.

[3] Christian Kerl, Jurgen Sturm, and Daniel Cremers. "Robust odometry estimation for RGB-D cameras." Robotics and Automation (ICRA), 2013 IEEE International Conference on. IEEE, 2013.

[4] Erik Bylow, et al. "Real-time camera tracking and 3d reconstruction using signed distance functions." Robotics: Science and Systems (RSS) Conference 2013. Vol. 9. Robotics: Science and Systems, 2013.

[5] Erik Bylow, Carl Olsson, and Fredrik Kahl. "Robust Camera Tracking by Combining Color and Depth Measurements." ICPR. 2014.

[6] Fatih Calakli, and Gabriel Taubin. "SSD: Smooth signed distance surface reconstruction." Computer Graphics Forum. Vol. 30. No. 7., 2011.

[7] Fatih Calakli, and Gabriel Taubin. "SSD-C: Smooth signed distance colored surface reconstruction." Expanding the Frontiers of Visual Analytics and Visualization, pp 323-338, 2012.

[8] Jurgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers, A Benchmark for the Evaluation of RGB-D SLAM Systems, International Conference on Intelligent Robots and Systems, 2012.

[9] Qian-Yi Zhou Vladlen Koltun. "Depth camera tracking with contour cues." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[10] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim. "KinectFusion: Real-time dense surface mapping and tracking." Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on. IEEE, 2011.

[11] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, Andrew Fitzgibbon. "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. ACM, 2011.

[12] http://vision.in.tum.de/data/datasets/rgbd-dataset/download#

[13] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, Dieter Fox, “RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments”, International Journal of Robotics Research, Vol. 31, pp 647-663, 2012.