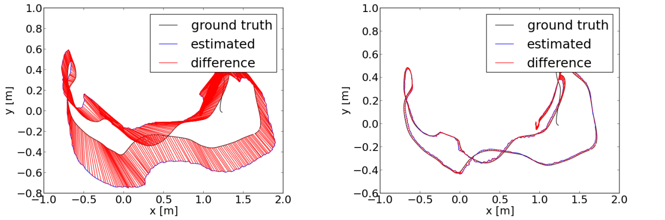
Fast Visual Odometry using Intensity assisted Iterative Closest Point

Shile Li[1] and Dongheui Lee[1]

***Abstract*—This paper presents a novel method for visual odometry estimation from a RGB-D camera. The camera motion is estimated by aligning a source to a target RGB-D frame using an intensity assisted Iterative Closest Point (ICP) algorithm. The proposed method differs from the conventional ICP in following aspects. (i) To reduce the computational cost, salient point selection is performed on the source frame, where only points that contain valuable information for registration are used. (ii) To reduce the influence of outliers and noises, robust weighting function is proposed to weight corresponding pairs based on statistics of their spatial distances and intensity differences. (iii) The obtained robust weighting function from (ii) is used for correspondence estimation of the following ICP iteration. The proposed method runs in real-time with a single core CPU thread, hence it is suitable for robots with limited computation resources. The evaluation on TUM RGB-D benchmark shows that in the majority of the tested sequences, our proposed method outperforms state-of-the-art accuracy in terms of translational drift per second with a computation speed of 78 Hz.**

***Index Terms*—Visual Tracking; RGB-D Perception; Visual-Based Navigation**



(a) Visual odometry estimated from the conventional ICP method [12]

(b) Visual odometry estimated from our ICP method

Fig. 1. Comparison of estimated camera trajectories on ”fr1/desk” sequence from TUM RGB-D benchmark [13]. Our method improves the conventional ICP greatly.

I. INTRODUCTION

SIX degrees of freedom (DOF) odometry estimation from visual data, estimation of camera's position and orientation from images, is one of the most active research area in the last decade [1][2][3][4][5][6][7]. Visual odometry is important for a wide range of robotic applications such as localization and mapping tasks. Also understanding ego motion from visual odometry can provide additional sensor information for tasks such as obstacle avoidance [8], object pose estimation [9], scene flow estimation [10] and robot walking [11]. Especially visual odometry provides 3D pose, whereas wheel odometry or GPS navigation only provide 2D pose.

In recent years, the availability of lightweight RGB-D sensors such as Asus Xtion raised the popularity of visual odometry estimation from both color and depth images or depth images alone. The recent proposed depth based visual odometry algorithms can be categorized into two groups.

The first group formulates the task as an energy minimization problem [14][1][15][16]. The energy function is devised from pixelwise photometric and/or depth residual error between the target image and the warped source image, where ”warping” is performed by rigid transforming the source image and projection onto the target image. This problem is iteratively solved by numerical optimization such as gradient descent method, where linearized Jacobian matrix with respect to the 6 DOF motion is required for each optimization iteration. These methods are efficient to solve, however it is based on a strong assumption that the energy function is locally smooth with respect to the 6 DOF pose, which is often not true due to the non-linear nature of image and sensor noises. Therefore a coarse-to-fine strategy is used in most methods, where image pyramid is built to make the energy function smoother in coarser levels, but image pyramid building and image gradient estimation for each level require extra computation.

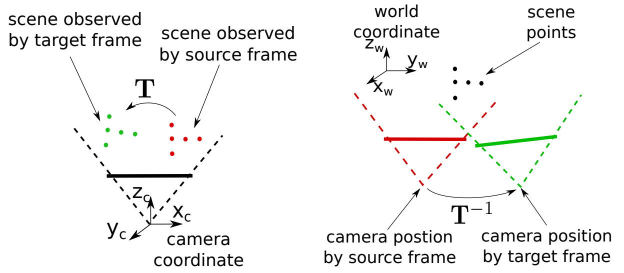
The second group [2][17] relies on the classic registration method: Iterative Closest Point (ICP) [12]. Corresponding points between the source and the target frames are first estimated based on a certain metric. Then the relative transformation is estimated with a closed-form solution to minimize the distances between correspondences. The above two steps are performed iteratively until a convergence criterion or the maximum iteration number is reached. However ICP suffers from the risk to be trapped in a local minimum, especially in cases of large camera motion or lack of 3D structure in the observed scene. One option to avoid local minimum is to use a feature (such as SIFT, SURF) based alignment first and use ICP only as a refinement step [17][18]. The feature based method improves the probability of convergence in the global minimum, however detection, description and matching of sophisticated keypoint require substantial computation time that hinders the performance to keep up the camera frame rate. Another option is to establish correspondences using normal projection instead of finding nearest points [2][19][6], which helps ICP to avoid some local minimum. However surface normal estimation requires even more computation than a

Manuscript received: August 13, 2015; Revised December 3, 2015; Accepted February 9, 2016.

This paper was recommended for publication by Editor Francois Chaumette upon evaluation of the Associate Editor and Reviewers' comments.

[1] The authors are with Chair of Automatic Control Engineering, Department of Electrical Engineering and Computer Engineering, Technical University of Munich li.shile@mytum.de, dhlee@tum.de

Digital Object Identifier (DOI): see top of this page.


scene observed by target frame
scene observed by source frame
T
z_c
x_c
y_c
camera coordinate
world coordinate
z_w
y_w
x_w
scene points
T^{-1}
camera postion by source frame
camera position by target frame

(a) static scene observed by camera in two different frames

(b) corresponding camera motion in the world coordinate

Fig. 2. Our method estimates the camera motion by aligning the scene points observed from different frames. (a): The scene points are rigidly transformed with $\mathbf{T}$ in the camera coordinate. (b): The corresponding camera motion in the world coordinate is the inverse of $\mathbf{T}$.

feature based method, because the surface normal is densely needed for each pixel. Meilland et. al. [16] choose 3D pixels which best condition the 6 degrees of freedom of the camera, but their method still needs to compute the image Jacobian.

Dealing with a large amount of color and depth data, achieving real-time performance that keeps up the camera frame rate (30 Hz) becomes an issue. In order to perform online visual odometry, some approaches that only use single core CPU [14][1][20], need to perform their algorithms on lower resolution images than the original sensor data to compromise between accuracy and processing time. Some other approaches that directly perform on the original resolution [2][4][6] require state-of-the art Graphics Processing Unit (GPU) to parallelize their algorithms, however not every mobile platform is equipped with a GPU.

In this paper, we present a fast and robust ICP based visual odometry method that uses both intensity and depth data. As the example shown in Figure 1, the proposed method improves the conventional ICP significantly. The contributions of this paper are as follows:

- An intelligent salient point selection method for the source frame is proposed, where points that provide valuable information for ICP are selected. With the reduced point number, substantial computation time is saved.
- With robust statistics on the real data [21], intensity values are integrated into correspondence estimation and correspondence weighting stages of ICP.
- The proposed method runs with a single CPU thread in real-time (78 Hz) with state-of-the-art accuracy on the TUM dataset [13].

This paper is organized as follows. Preliminaries about camera model and the conventional ICP method [12] are described in section II. The proposed intensity assisted ICP method is explained in section III. The performance of our methods is shown with evaluation on TUM RGB-D benchmark in section IV. Finally, a conclusion is given in section V. The code of our method is available at www.hri.ei.tum.de/download.

II. Preliminaries

A. Camera model

Given a 3D point $\mathbf{p} = (x, y, z, 1)^T$ in homogeneous coordinate relative to the camera, the image pixel coordinate $\mathbf{x} = (u, v)^T$ ($u \in [0, width-1], v \in [0, height-1]$) of $\mathbf{p}$ is calculated with the camera projection function π:

$$\mathbf{x} = \pi(\mathbf{p}) = (\frac{xf_x}{z} + o_x, \frac{yf_y}{z} + o_y)^T, \tag{1}$$

where *height* and *width* are the pixel number in image's x- and y- direction, f_x, f_y are the camera focal lengths and o_x, o_y are the camera center coordinates.

As shown in Figure 2, in the camera coordinate, if the scene points are rigidly transformed with a transformation matrix $\mathbf{T} \in \mathbb{R}^{4\times4}$, then a camera motion $\mathbf{T^{-1}}$ in the world coordinate is implied. In the camera coordinate, the point $\mathbf{p}$ is then changed to: $\mathbf{p}' = \mathbf{Tp}$, and its pixel coordinate becomes $\mathbf{x}' = \pi(\mathbf{p}') = \pi(\mathbf{Tp})$.

At time step t, an intensity image I_t and an organized point cloud P_t with the resolution $width \times height$ are obtained, where $P_t(i)$ indicates the ith point in P_t. Intensity value of pixel $\mathbf{x}$ is $I_t(\mathbf{x}) \in [0, 255]$, the pixel $\mathbf{x}$'s corresponding 3D point $\mathbf{p}$ is indicated as $P_t(ind(\mathbf{x}))$, where $ind()$ is a function that maps a image coordinate to a point index of a one-dimensional list of organized point cloud:

$$ind(\mathbf{x}) = ind((u, v)^T) = v \times width + u, \tag{2}$$

To retrieve the image coordinate $\mathbf{x}$ of a point index i, the inverse function is:

$$\mathbf{x} = ind^{-1}(i) = (i - \lfloor \frac{i}{width} \rfloor width, \lfloor \frac{i}{width} \rfloor)^T. \tag{3}$$

B. Conventional ICP

Let us consider two subsequent frames that need to be aligned as $< I_1, P_1 >$ for the source frame and $< I_2, P_2 >$ for the target frame. The conventional ICP method [12] uses the point cloud pair P_1, P_2 to iteratively find the optimal relative rigid transformation matrix $\mathbf{T}^*$, such that after transforming P_1 with $\mathbf{T}^*$, the source observation will be aligned with the target observation (Figure 2(a)). ICP is an iterative method, the matrix $\mathbf{T}^*$ is initialized as an identity matrix, the kth iteration of the ICP algorithm can be summarized as follows:

1) Search for each point of the source cloud a closest point in the target cloud as correspondence. For the ith point in P_1, the index of the corresponding point in P_2 is denoted as $c(i)$, where

$$c(i) = \underset{j}{\operatorname{argmin}} \|\mathbf{T}^* P_1(i) - P_2(j)\|. \tag{4}$$

2) Compute the optimal incremental transformation $\mathbf{T}_k$ that minimizes the distances between the established correspondences:

$$\mathbf{T}_k = \underset{\mathbf{T}}{\operatorname{argmin}} \sum_i \|\mathbf{T}\mathbf{T}^* P_1(i) - P_2(c(i))\| \tag{5}$$

This is usually solved with a closed-form solution [22] such as Singular Value Decomposition [23].

3) Update $\mathbf{T}^*$ as:

$$\mathbf{T}^* \leftarrow \mathbf{T}_k \mathbf{T}^*. \tag{6}$$

The above steps are performed iteratively until the incremental transformation is smaller than a threshold or the maximum allowable iteration number has reached.

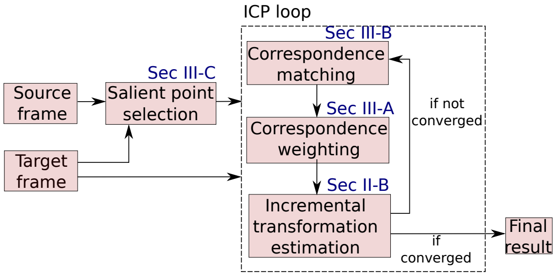


Fig. 3. Intensity assisted ICP overview

III. Intensity assisted Iterative Closest Point

Rusinkiewicz et. al. discussed in [24] about ICP variants, where the variants differ in the following stages: selection, matching, weighting, rejection, error metric and minimizing. Based on the categorization from [24], our proposed ICP method differs from the conventional ICP in the following stages.

1) **Selection** - Salient points selection is performed on the source frame, where points that provide valuable information for ICP are selected. For the target frame, the original image resolution is kept without any sampling. See details in section III-C.
2) **Matching** - The search of correspondences is performed by examining nearby points in the image coordinate, where the matching point is determined by considering both intensity and geometric distance. See details in section III-B.
3) **Weighting** - Weighting of corresponding pairs is performed based on robust statistic [21]. This improves the robustness of ICP against false correspondences. Additionally, the sensor noise model of the depth camera is also considered for the weighting. See details in section III-A.

An overview of our method is illustrated in Figure 3.

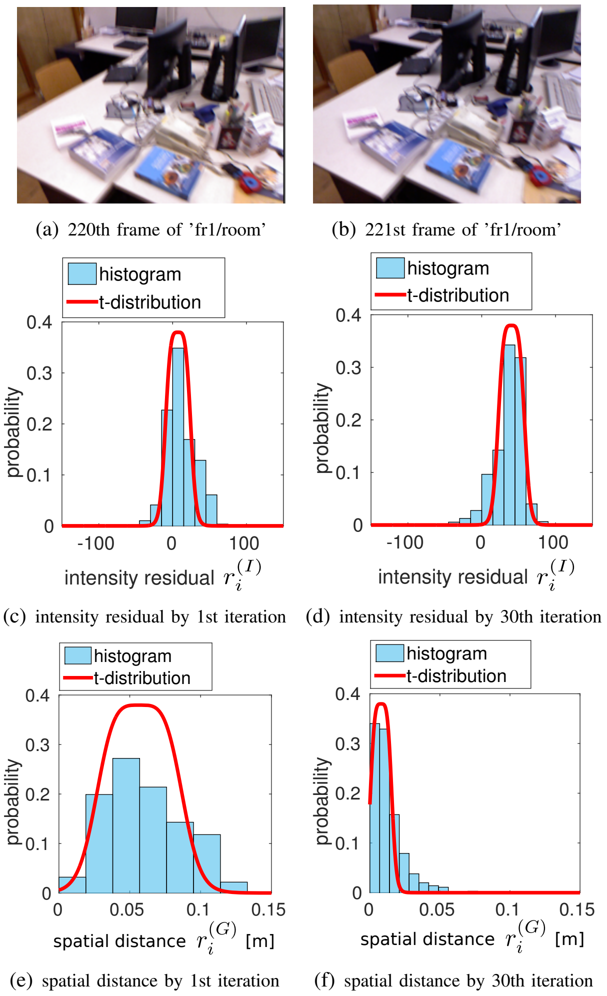
A. Robust correspondence weighting

With the method in section III-B, correspondences between source and target frames are established, where corresponding point of the ith point in source frame $P_1(i)$ is $P_2(c(i))$. In practice, not every correspondence is determined correctly. The resulting outliers have a bad influence on the accuracy of the estimated transformation. Moreover the precision of depth value depends on the distance to the camera, where a more distanced point has lower precision for depth value. In our method, robust weighting function is applied to reduce the influence of outliers and to adapt to the sensor noise model, where the ith corresponding pair a weight $w(i)$, thus Equation (5) changes to:

$$\mathbf{T}_k = \operatorname*{argmin}_{\mathbf{T}} \sum_i w(i) \|\mathbf{T}\mathbf{T}^* P_1(i) - P_2(c(i))\|. \tag{7}$$

Intensity residuals between the correspondences are considered for weighting, the intensity residual $r_i^{(I)}$ of the ith pair is calculated as:

$$r_i^{(I)} = I_1(ind^{-1}(i)) - I_2(ind^{-1}(c(i))). \tag{8}$$



(a) 220th frame of 'fr1/room'

(b) 221st frame of 'fr1/room'

(c) intensity residual by 1st iteration

(d) intensity residual by 30th iteration

(e) spatial distance by 1st iteration

(f) spatial distance by 30th iteration

Fig. 4. (a)(b): The illumination of the 221st frame is lower than the 220th frame due to auto-exposure, the average intensity has dropped by 43 in 221st frame. (c)(d): The intensity residual distribution changes towards the true illumination model as iteration number grows. (e)(f): The distribution of spatial distance between correspondences moves from larger value toward zero as iteration number grows.

Inspired by [14], the Student's t-distribution, a M-Estimator function, is used to weight each intensity residual. Following [25], the derived weighting function based on the t-distribution is:

$$w_I(r_i^{(I)}) = \frac{\nu + 1}{\nu + ((r_i^{(I)} - \mu^{(I)})/\sigma^{(I)})^2}, \tag{9}$$

where ν is the degree of freedom of t-distribution, $\nu = 5$ is used as the same in [14]. The mean $\mu^{(I)}$ of all intensity residuals, is estimated as the median value:

$$\mu^{(I)} = Med(\{r_i^{(I)}\}_{i=1}^{N}), \tag{10}$$

where N is the number of correspondences. The standard deviation $\sigma^{(I)}$ is estimated using the median absolute deviation:

$$\sigma^{(I)} = 1.4826\, Med(\{|r_i^{(I)} - \mu^{(I)}|\}_{i=1}^{N}). \tag{11}$$

The t-distribution fits the real data nicely as shown in Figure 4(c)(d), which gives outliers very small weights. An important reason to weight the correspondences based on the statistics over all residuals is to consider the changing illumination caused by auto-exposure of the camera. Figure 4 gives an example from 'fr1/room' sequence of TUM dataset, the average pixel intensity value has dropped by 43 from Figure 4(a) to Figure 4(b). By first iteration, the mean value of intensity residuals is still almost zero (Figure 4(c)), as iteration number grows, the intensity residual distribution changes towards the correct illumination difference value (Figure 4(d)). Then the statistic based weighting ensures larger weights for the correct intensity residuals.

Another component of the weighting function is based on the spatial distance between correspondences. It follows the same procedure as intensity based weighting, where spatial distance is considered as residual:

$$r_i^{(G)} = \|\mathbf{T}^* P_1(i) - P_2(c(i))\|. \tag{12}$$

The computation of $w_G(r_i^{(G)})$ follows the same as the weighting for intensity residual.

$$w_G(r_i^{(G)}) = \frac{\nu + 1}{\nu + ((r_i^{(G)} - \mu^{(G)})/\sigma^{(G)})^2}. \tag{13}$$

The t-distribution also fits the distribution of spatial residuals nicely as shown in Figure 4(e)(f). By the first iteration (Figure 4(e)), the spatial distance is relative large due to the camera motion, the variance is also large due to the rotation component of the camera motion. By later iteration (Figure 4(f)), the current transformation estimate $\mathbf{T}^*$ gets closer to the true transformation, thus the mean spatial residual value is closer to zero and the variance is also much smaller.

To compensate the noises caused by depth sensor model, another weight $w_S(i)$ is assigned. The sensor noise model devised from [26] is used without considering the influence of surface normal. The weight for the ith corresponding pair is computed based on the average depth value of $P_1(i)$ and $P_2(c(i))$:

$$w_S(i) = \frac{1}{0.0012 + 0.0019(\frac{\mathbf{e}_z^T P_1(i) + \mathbf{e}_z^T P_2(c(i))}{2})^2}, \tag{14}$$

where $\mathbf{e}_z^T = (0,0,1,0)$.

The total weight $w(i)$ for ith corresponding pair combines the three weights from intensity difference, spatial distance and sensor noise model:

$$w(i) = w_I(r_i^{(I)}) w_G(r_i^{(G)}) w_S(i), \tag{15}$$

where the three weight components are complementary to downweight outlying correspondences and correspondences with higher sensor noise. The multiplication in Equation (15) ensures that only corresponding pairs that have relative larger weight in all three components, are assigned with large total weight. With weighting stage using robust estimation from statistic, the outliers are intuitively downweighted.

B. Intensity assisted point matching

The conventional ICP method establishes correspondences only based on spatial distance (Equation (4)). In this paper, intensity difference and spatial distance are both used to determine "closest" point by considering the robust statistic obtained from section III-A. Given the query point pair $< P_1(i), P_2(j) >$, the score function for this query pair is:

$$\begin{aligned} s(i,j) \quad &= w_I(I_1(ind^{-1}(i)) - I_2(ind^{-1}(j))) \\ &\quad w_G^{(0)}(\|\mathbf{T}^* P_1(i) - P_2(j)\|), \end{aligned} \tag{16}$$

$w_I()$ and $w_G^{(0)}()$ are the robust function derived from last ICP iteration (Equation (9) (13)). [1] $w_G^{(0)}()$ has the form:

$$w_G^{(0)}(r) = \frac{\nu + 1}{\nu + ((r - 0)/\sigma^{(G)})^2}, \tag{17}$$

where the mean value is set to zero, because we adopt the assumption from the conventional ICP [12], where closer point is more probable to be the correspondence. In case of a large camera motion, where closest point assumption does not necessarily hold, the intensity weight term in Equation (16) provides an additional constrain compared to the conventional ICP. For the first iteration, where the robust statistic is not available, $\mu^{(I)}$ is set to zero, and $\sigma^{(I)}, \sigma^{(G)}$ are initialized with 10 and 0.04m in our implementation. The matching point for $P_1(i)$ is then the point that results in the highest score:

$$c(i) = \underset{j}{\operatorname{argmax}}\, s(i,j). \tag{18}$$

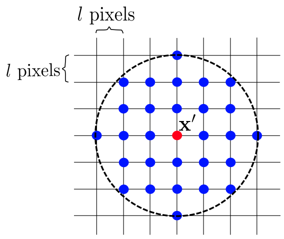


Fig. 5. The search pattern for matching point: the blue points within the circle of the radius $3l$ in the image coordinate are examined

Taking advantage of the organized point cloud, the search of matching point is performed in the image coordinate. Based on the current known transformation T^* from last iteration, $P_1(i)$ is warped onto target frame to determine the area of search, where $P_1(i)$'s image coordinate in the target frame is $\mathbf{x}' = \pi(\mathbf{T}^* P_1(i))$. With $\mathbf{x}'$ as the center, the search pattern is illustrated in Figure 5, where the blue points within the circle of the radius $3l$ are examined.

The offset parameter $l \in \mathbb{Z}^+$ (Figure 5) is used to control the size of the search area. To cope with large camera motion, the search area should be larger in the initial iterations, thus l is set with larger values. As the iteration number grows and ICP converges, l decreases accordingly until 1, which brings a decreasing search area and increasing precision as ICP proceeds.

C. Salient point selection

Due to 3D motion between two frames and projective nature of image, some points in P_1 might be occluded in P_2. If

[1] Notice that for each iteration, the parameters (mean, variance) of weighting functions for correspondence 'matching' (Eq. (18)), and for correspondence 'weighting' (Eq. (15)) are different. Parameters for correspondence 'matching' are based on the statistic from the last iteration (for the 1st iteration, default values are used). Parameters for correspondence 'weighting' are derived from the statistic over newly established correspondences of the current iteration.

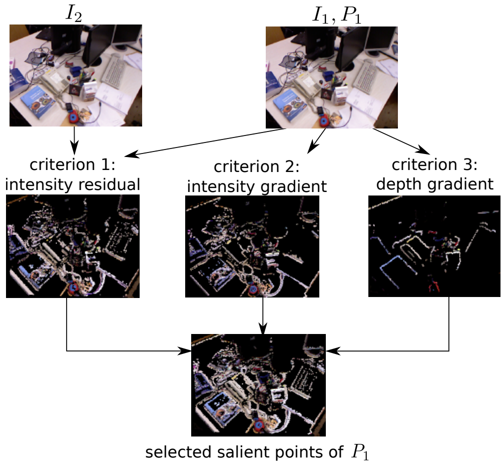


Fig. 6. Salient point selection. Top: the original source and target frame. Middle: selected salient points based on three different criteria. Down: all salient points combining three criteria.

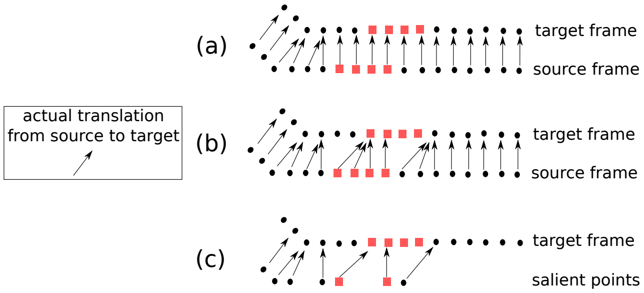


Fig. 7. Point pair matching for a translated scene. (a) Conventional ICP, (b) Conventional ICP + matching score with Eq. (16), (c) Conventional ICP + matching score with Eq. (16) + salient point selection

occluded points from P_1 are used for correspondence estimation, the established correspondences is definitely wrong, which influences the accuracy of estimated pose (Equation (5)). Therefore, by correspondence estimation, the points from P_1 which have high probability to be occluded in P_2, are discarded. These points are background points that are near the edges of foreground region, where a subtle motion might cause the occlusion. The point $P_1(ind(\mathbf{x}))$ is considered to have high occlusion probability, and is discarded if it satisfies:

$$\begin{array}{rll} & e_Z^T P_1(ind(\mathbf{x})) - e_Z^T P_1(ind(\mathbf{x}+(0,5)^T)) & > \tau_r, \\ or & e_Z^T P_1(ind(\mathbf{x})) - e_Z^T P_1(ind(\mathbf{x}+(0,-5)^T)) & > \tau_r, \\ or & e_Z^T P_1(ind(\mathbf{x})) - e_Z^T P_1(ind(\mathbf{x}+(5,0)^T)) & > \tau_r, \\ or & e_Z^T P_1(ind(\mathbf{x})) - e_Z^T P_1(ind(\mathbf{x}+(-5,0)^T)) & > \tau_r, \end{array} \tag{19}$$

which implies that $P_1(ind(\mathbf{x}))$ is discarded if it has a much larger depth value than a nearby point.

After background point rejection, a lot of the remaining points in the source frame still might fail to find their correct correspondences in the target frame. In particular, the points which lie in homogeneous regions of intensity image or depth image have higher chances to be matched to false correspondences. Figure 7 illustrates example cases of correspondence matching result for a translated scene[2]. For a source frame point in homogeneous region of depth image, the target frame points near the correct match can be closer to the query point, thus many false matches can occur (Figure 7(a)). Even with an intensity assisted matching term (Eq. (18)), a lot of source frame points in homogeneous region of intensity image can still be matched to wrong points (Figure 7(b)). Figure 7(c) shows that by neglecting homogeneous regions, the ratio of correct matches can be increased and this leads to more accurate incremental transformation result (Eq. (5)). To avoid homogeneous regions, three selection criteria are used to determine whether a source frame point should be considered.

1) Intensity residual based: The first criterion is intensity residual. By computing intensity residual of same pixel between two frames, pixels in homogeneous regions of intensity image do not result in large intensity difference. In contrast, pixels in textured regions or border of homogeneous regions, might result in large intensity residual from camera motion. Therefore the first criterion for salient points is:

$$|I_1(\mathbf{x}) - I_2(\mathbf{x})| > \tau_1. \tag{20}$$

2) Intensity gradient based: Points inside homogeneous region have low intensity gradient, therefore points with high intensity gradient are selected:

$$\begin{array}{rll} & |I_1(\mathbf{x}+(0,2)^T) - I_1(\mathbf{x}+(0,-2)^T)| & > \tau_2, \\ or & |I_1(\mathbf{x}+(2,0)^T) - I_1(\mathbf{x}+(-2,0)^T)| & > \tau_2, \end{array} \tag{21}$$

where either gradient in x-direction or y-direction of the image coordinate should be greater than the gradient threshold τ_2. Based on the intensity term from Equation (16), these points have higher probability to distinguish their correct correspondence.

3) Depth gradient based: Similar to intensity, points inside a homogeneous depth space also have ambiguity to find its correct correspondence. Therefore the third criterion is depth gradient, where points that contains variation in the depth space are selected:

$$\begin{array}{rll} & \frac{|e_Z^T P_1(ind(\mathbf{x}+(0,2)^T)) - e_Z^T P_1(ind(\mathbf{x}+(0,-2)^T))|}{e_Z^T P_1(ind(\mathbf{x}))} & > \tau_3, \\ or & \frac{|e_Z^T P_1(ind(\mathbf{x}+(2,0)^T)) - e_Z^T P_1(ind(\mathbf{x}+(-2,0)^T))|}{e_Z^T P_1(ind(\mathbf{x}))} & > \tau_3. \end{array} \tag{22}$$

In summary, for salient point selection, one rejection criterion and three acceptance criteria are used. The algorithm to retrieve the salient points is described in Algorithm 1. The salient point selection is efficient, because a salient point only needs to meet one of the three criteria. As soon as the point fits one criterion, the algorithm skips the rest of criteria and proceeds to check next point. The salient points selected based on different criteria are illustrated in Figure 6.

As in Algorithm 1, every 16th source frame point (every 4th row and 4th column) is checked for its saliency. This reduces the computation time for salient point selection stage by a factor of 16, but the lost of accuracy is not too much because no subsampling is performed on the target frame $< I_2, P_2 >$. In contrast, energy minimization based methods [1][20] perform subsampling on both source and target frames, which results in a relative higher accuracy lost.

For each ICP iteration, a different subset of salient points is randomly selected and used for registration. The reason of

[2] Notice that Figure 7 is only an illustrative example, the ratio of homogeneous region in real scene is much larger.

the random selection for each iteration is: firstly, it further reduces the computation cost by using less points per iteration; secondly, all selected salient points should have equal possibility to contribute in registration. The number of this randomly selected subset per iteration is set as ca. $100-200$, which is found empirically based on trade-off between the computational cost and the accuracy. Experimental results show the accurate 6 DOF pose estimation from this amount of correspondences.

Algorithm 1 Salient point selection

```
Input: - Source frame $<I_1, P_1>$ and target frame $<I_2>$
Output: - List $list$ that contains indexes of salient points from $P_1$.
  - $list \leftarrow \emptyset$
  for $u = 1:4:width$ do
    for $v = 1:4:height$ do
      - get the point to be checked: $P(ind((u,v)^T))$ with intensity $I((u,v)^T)$
      if $P(ind((u,v)^T))$ satisfies rejection criterion (Equation (19)) then
        continue;
      end if
      for c=1:3 do
        if $P(ind((u,v)^T))$ satisfies $c$th criterion (Equation (20) or (21) or (22)) then
          - $list.append(ind((u,v)^T))$;
          - continue;
        end if
      end for
    end for
  end for
```

IV. EXPERIMENT

The proposed method is evaluated using the TUM RGB-D benchmark [13]. The benchmark contains 89 RGB-D video sequences, for each video sequence, accurate ground truth for camera motion is provided by a motion capture system. We evaluated our method on 14 video sequences, which are commonly used in the previous publications. For evaluation of visual odometry accuracy, the root mean square error (RMSE) of translational drift in m/s is used, which is a standard metric to measure accuracy of visual odometry method [13].

In this section, our method is first compared with other RGB-D based visual odometry methods [1][6][4] that use both intensity and depth data. Then our method is compared with depth only based methods [27][20] by turning off all intensity related process in our method. The computation performance is also evaluated with different parameter settings. Finally we show some qualitative result of reconstructed scenes based on our visual odometry method.

Our experiments are performed on a desktop computer with Ubuntu 12.04, equipped with Intel Core i7-4790K CPU (4 GHz) and 16GB RAM. Notice that our implementation only runs on a single CPU thread.

TABLE I
ON THE TWO 'FR1' DESK SEQUENCES, EACH OF THE THREE ALGORITHM COMPONENTS PROVIDES INCREMENTAL CONTRIBUTION TO THE ACCURACY.

Method	RMSE of translational drift[m/s]		Average improvement
	fr1/desk	fr2/desk	
ICP	0.175	0.182	0%
ICP+s	0.045	0.056	71.7%
ICP+s+w	0.044	0.051	73.4%
ICP+s+m	0.029	0.041	80.39%
ICP+s+m+w	**0.021**	**0.038**	**83.25%**

TABLE III
RMSE OF TRANSLATIONAL DRIFT (M/S): COMPARISON WITH OTHER DEPTH ALONE BASED METHODS

	fr1/desk	fr1/desk2	fr1/room	fr2/desk
Our method (depth only)	**0.0297**	**0.384**	**0.0484**	0.0330
Sparse depth [27]	0.058	0.073	0.073	**0.028**
Fast 3-D [20]	0.0366	0.0528	0.0489	0.0313

A. Accuracy

For each ICP iteration, a subset of 100 salient points are used (section III-C). In total we run 30 iterations: 3 levels with 10 iteration per level. For the first 10 iterations, the offset l for correspondence search is set to 6, for the second 10 iterations l is set to 3 and for the last 10 iterations l is set to 1. Every 5th frame is used as the source frame ($<I_1, P_1>$), and the current frame is used as the target frame ($<I_2, P_2>$). The values of the thresholds $<\tau_r, \tau_1, \tau_2, \tau_3>$ are empirically set as $<0.02, 30, 30, 0.03>$ in all experiments.

To prove the contribution of each component in our method, 'fr1/desk' and 'fr1/desk2' sequences are used. Selection (s), matching (m) and weighting (w) component of our method are incrementally added to the conventional ICP method, Table I shows that each component provides an incremental improvement on the accuracy. Among the three components, the selection component contributes the most by neglecting the source frame points that have higher probability to be incorrectly matched. The matching component provides the second largest contribution by combining intensity term for more accurate correspondence matching process. Finally the weighting component reduces the influences from outliers and sensor noises.

Table II shows the comparison of our method and other state-of-the-art visual odometry methods based on both depth and intensity, where in Table II, '-' means that result is not provided in the corresponding literature.[3] Our method provides the best accuracy in 11 of the 14 tested sequences, and even outperforms the method RGB+D+KF+Opt from [1] that uses loop closure detection and global pose optimization. This is probably because our method can avoid some local minimum, which energy minimization based methods cannot avoid due to the non-linear nature of images.

There is another group of methods that only uses depth information, our method is compared with two recent ones [27][20]. For comparison, depth only based method is simulated by setting all pixels' color to a same value. As shown in Table III, our method outperforms in most sequences. This indicates that our method can also handle poor lightening condition and textureless scene.

B. Computation time vs. Performance

Real-time capability is crucial for online application, therefore some algorithms require GPU for parallelization and some others operate on lower resolution image and hence loose some

[3] Since the performance varies depending on the parameter settings and the optimal parameters of other methods are unknown, for fair comparison, we only listed the reported values from the original publications[1][6][4].

TABLE II
RMSE OF TRANSLATIONAL DRIFT (M/S): COMPARISON WITH OTHER RBG-D BASED METHOD

Sequence	average translational velocity [m/s]	average angular velocity [degree/s]	**Our method**	RGB+D [1]	RGB+D+ KF [1]	RGB+D+ KF+Opt [1]	ICP+RGB-D [6]	Inverse depth [4]
fr1/desk	0.413	23.327	**0.0217**	0.036	0.030	0.024	0.0393	0.026
fr1/desk2	0.426	29.30	**0.0381**	0.049	0.055	0.050	-	0.0387
fr1/room	0.344	29.882	**0.0416**	0.058	0.048	0.043	0.0622	0.0491
fr2/desk	0.193	6.388	0.0204	-	-	-	0.0208	**0.0121**
fr1/xyz	0.244	8.920	**0.018**	0.026	0.024	0.018	-	-
fr1/rpy	0.062	50.147	**0.031**	0.040	0.043	0.032	-	-
fr1/360	0.210	41.600	**0.072**	0.119	0.119	0.092	-	-
fr1/floor	0.258	15.071	0.10	fail	**0.090**	0.232	-	-
fr1/teddy	0.315	21.320	0.048	0.060	0.067	**0.043**	-	-
fr1/plant	0.365	27.891	**0.018**	0.036	0.036	0.025	-	-
fr3/office	0.249	10.188	**0.016**	-	-	-	-	-
fr3/nostructure_texture_far	0.299	2.890	**0.047**	0.073	-	-	-	-
fr3/structure_notexture_far	0.166	4.000	**0.034**	0.038	-	-	-	-
fr3/structure_texture_far	0.193	4.323	**0.023**	0.039	-	-	-	-

TABLE IV
COMPARISON OF COMPUTATION TIME PER FRAME [MS] VS. HARDWARE SETTING VS. IMAGE RESOLUTION

	Time	CPU	GPU	Resolution
Our method	12	i7-4790K @4.0GHz	-	640×480
RGB+D+KF [1]	32	i7-2600 @3.4GHz	-	320×240
ICP+RGB-D [6]	18	i7-3960X @3.3GHz	NVIDIA GeForce 680GTX	640×480
Inverse depth [4]	47	i5-2500 @3.3GHz	NVIDIA GeForce 660GTX	640×480
Sparse depth [27]	67	i7-2860QM @2.5GHz	-	VoxelGridFilter voxel size: 1cm
Fast 3-D [20]	28	i7-3820 @3.6GHz	-	320×240

TABLE V
DIFFERENT PARAMETERS VS. PRECISION VS. COMPUTATION TIME

Iterations per level	Salient point number per iteration	RMSE of translational drift [m/s]				Time [ms]	
		fr1/desk	fr1/desk2	fr1/room	fr2/desk	mean	max
10	100	0.0217	0.0381	0.0416	0.0204	12.8	17.9
2	100	0.0255	0.0450	0.0425	0.0240	9.3	13.6
5	100	0.0219	0.0369	0.0442	0.0225	10.9	15.0
20	100	0.0218	0.0389	0.0440	0.0184	17.8	22.9
50	100	0.0219	0.0373	0.0455	0.0188	30.4	34.0
100	100	0.0226	0.0389	0.0554	0.0182	58.7	67.6
10	10	0.0274	0.0400	0.0781	0.0312	9.5	12.9
10	20	0.0260	0.0382	0.0467	0.0236	9.9	13.7
10	50	0.0226	0.0384	0.0466	0.0200	11.8	13.6
10	200	0.0219	0.0379	0.0430	0.0195	16.7	22.4
10	500	0.0217	0.0367	0.0415	0.0196	27.3	31.9

accuracy. In contrast, our method uses intelligent salient point sampling method without sacrificing accuracy. Due to sparse sampling in the source image, a high frame rate of 78 Hz using only single CPU thread is achieved. The frame rate is expected to be even higher if GPU programming is used, however 78 Hz is enough for the real-time requirement. The reported computation time of different methods and the hardware settings are compared in Table IV, where the computation time of our method is obtained by using parameter settings from section IV-A. The '-' symbol in Table IV means that GPU programming is not used. Table IV shows that our method is the only one that performs on the original image resolution (640 × 480) without GPU programming which can keep up the camera frame rate.

Furthermore different parameter settings are tested and the influence for accuracy and computation performance are compared. Fixed three levels of offset l for correspondence searching are used, for each level, the number of iterations varies. For each iteration, the number of salient point used for correspondence estimation is also varied.

Table V shows that our method performs well with small number of iteration and small number of salient points. Our method achieved reasonable result even using only 10 salient points per ICP iteration, achieving 100 Hz, which is much higher than the camera frame rate, saving a lot of computational resource for additional tasks such as scene reconstruction. As seen from Table V and Table I, the proposed approach with only 10 correspondences provided a better result than the conventional ICP method with ca. 300000 correspondences on 'fr1/desk' and 'fr1/desk2' sequences. As the number of iteration grows, the drift error does not change much. As the number of salient point per iteration grows, the drift error decreases. This implies that the number of salient point is more important than the number of iteration for visual odometry.

C. Qualitative result

To illustrate the performance of our method qualitatively, Figure 8 shows four reconstructed scenes from TUM dataset. The quality of scene reconstruction from video sequence is sensitive to the accuracy of visual odometry, because the error of frame-to-frame registration result can be accumulated to a large drift. Therefore the usual remedy to correct large drift error is to use loop closure detection and global pose graph optimization [1].

In our case, no loop closure detection and no global pose graph optimization are performed by reconstruction. The colored point cloud of each frame is simply added into a global point cloud based on the estimated visual odometry result. The estimated visual odometry result is still accurate enough to reconstruct the scene without large drift.

V. CONCLUSION

In this paper a fast and robust visual odometry estimation method based on intensity assisted ICP is presented. By contributing in the selection, matching and weighting stages,

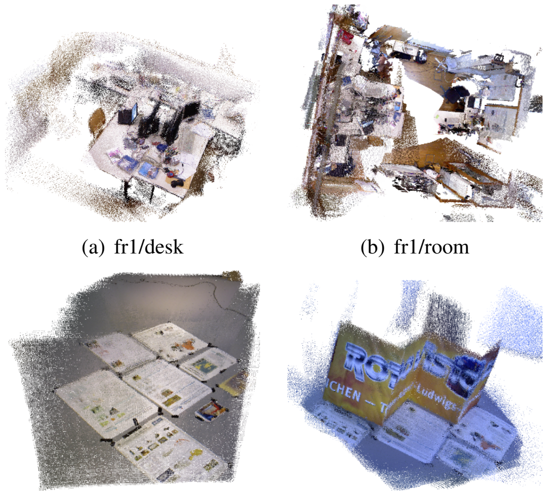

(a) fr1/desk (b) fr1/room (c) fr3/nostructure_texture_far (d) fr3/structure_texture_far

Fig. 8. Reconstructed scene from TUM benchmark sequences based on estimated visual odometry.

our method improved the conventional ICP significantly. Intelligent salient point selection is performed on the source frame thus drastically reduced the computation time. Correspondences are established by searching nearby points in the image coordinate. With weighting function devised from statistics, robustness against outlying correspondences is ensured. The proposed method was evaluated on the TUM Dataset both quantitatively and qualitatively. In terms of translational drift, it outperforms state-of-the-art methods in 11 out of the 14 tested video sequences. Our method runs with an average frame rate of 78 Hz using a single CPU thread. Experimental results showed that our proposed approach achieved overall better accuracy than approaches with GPU parallelization. With changes of parameter settings, our method can even achieve 107 Hz by loosing ca. 12% precision of drift error. With the achieved high frame rate, substantial computation resources can be saved for other online tasks.

References

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

[2] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *IEEE international symposium on Mixed and augmented reality (ISMAR)*, 2011, pp. 127–136.

[3] R. A. Newcombe and A. J. Davison, "Live dense reconstruction with a single moving camera," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 1498–1505.

[4] D. Gutierrez-Gomez, W. Mayol-Cuevas, and J. Guerrero, "Inverse depth for accurate photometric and geometric error minimisation in rgb-d dense visual odometry," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 83–89.

[5] S. Klose, P. Heise, and A. Knoll, "Efficient compositional approaches for real-time robust direct visual odometry from rgb-d data," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 1100–1106.

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

[7] J. Stückler and S. Behnke, "Multi-resolution surfel maps for efficient dense 3d modeling and tracking," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 137–147, 2014.

[8] M. Saveriano and D. Lee, "Distance based dynamical system modulation for reactive avoidance of moving obstacles," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 5618–5623.

[9] S. Koo, D. Lee, and D.-S. Kwon, "Unsupervised object individuation from rgb-d image sequences," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, 2014, pp. 4450–4457.

[10] W. Wang and D. Burschka, "Dense and deformable motion extraction in dynamic scenes based on hierarchical mrf optimization in rgb-d images," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2015, pp. 1115–1122.

[11] K. Hu, C. Ott, and D. Lee, "Online human walking imitation in task and joint space based on quadratic programming," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 3458–3464.

[12] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Robotics-DL tentative*. International Society for Optics and Photonics, 1992, pp. 586–606.

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

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

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

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

[17] 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 *Experimental robotics*. Springer, 2014, pp. 477–491.

[18] 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.

[19] T. Whelan, J. McDonald, M. Kaess, M. Fallon, H. Johannsson, and J. J. Leonard, "Kintinuous: Spatially extended kinectfusion."

[20] M. Jaimez and J. Gonzalez-Jimenez, "Fast visual odometry for 3-d range sensors," *IEEE Transactions on Robotics*, vol. PP, no. 99, pp. 1–14, 2015.

[21] P. J. Huber, *Robust statistics*. Springer, 2011.

[22] D. W. Eggert, A. Lorusso, and R. B. Fisher, "Estimating 3-d rigid body transformations: a comparison of four major algorithms," *Machine Vision and Applications*, vol. 9, no. 5-6, pp. 272–290, 1997.

[23] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 5, pp. 698–700, 1987.

[24] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *Third International Conference on 3-D Digital Imaging and Modeling, 2001. Proceedings*. IEEE, 2001, pp. 145–152.

[25] K. L. Lange, R. J. Little, and J. M. Taylor, "Robust statistical modeling using the t distribution," *Journal of the American Statistical Association*, vol. 84, no. 408, pp. 881–896, 1989.

[26] 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 and Transmission*. IEEE, pp. 524–530.

[27] S. M. Prakhya, L. Bingbing, L. Weisi, and U. Qayyum, "Sparse depth odometry: 3d keypoint based pose estimation from dense depth data," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4216–4223.