# Dense RGB-D Visual Odometry using Inverse Depth*

Daniel Gutiérrez-Gómez†, Walterio Mayol-Cuevas‡, and J.J. Guerrero†

April 8, 2016

*Abstract* – **In this paper we present a dense visual odometry system for RGB-D cameras performing both photometric and geometric error minimisation to estimate the camera motion between frames. Contrary to most works in the literature, we parametrise the geometric error by the inverse depth instead of the depth, which translates into a better fit of the distribution of the geometric error to the used robust cost functions. To improve the accuracy we propose to use a keyframe switching strategy based on a visibility criteria between frames. For the comparison of our approach with state-of-the-art approaches we use the popular datasets from the TUM for RGB-D benchmarking as well as two synthetic datasets. Our approach shows to be competitive with state-of-the-art methods in terms of drift in meters per second, even compared to methods performing loop closure too. When comparing to approaches performing pure odometry like ours, our method outperforms them in the majority of the tested datasets. Additionally we show that our approach is able to work in real time and we provide a qualitative evaluation on our own sequences showing a low drift in the 3D reconstructions. We have implemented this method within the scope of PCL (Point Cloud Library) as a branch of the code for large scale KinectFusion, where the original ICP system for odometry estimation has been completely substituted by our method. A PCL fork including the modified method is available for download .**

## 1 Introduction

Visual localisation and mapping has become one of the most active robotics research topics in the last decade. Traditional monocular vision systems track sparse features detected in the images, to both estimate the camera pose and build a map either using filtering [7] or bundle adjustment techniques [34] [24]. However, due to their purely projective nature, monocular vision systems do not directly provide depth measurements of the observed environment. This implies the existence of an unknown scale parameter for the camera poses and map estimates. In a SLAM context, this scale ambiguity results in an increased odometry drift and in initialisation issues.

One straightforward way to address the scale problem is to use stereo vision systems [37, 38, 30] in which a fixed baseline between two cameras allows for depth estimation. However with respect to monocular systems they are much more expensive, bigger and more difficult to calibrate, and also they cannot accurately measure the depth of distant scene points or poorly textured areas.

For this reason the recent advent of new RGB-D sensors has aroused great interest in the development of visual odometry and SLAM systems. Their cheapness and their ability to provide dense depth measurements of the environment in contrast to traditional stereo cameras makes them quite appealing to address not only localisation and mapping but also many other problems for which monocular systems are typically used. The main limitation is their use being limited to indoor environments.

Some of the first systems using RGB-D sensors for SLAM [14], [19], [9] tended to adapt the sparse feature based approaches from monocular vision, using the depth information to straightforwardly lift the features to 3D points and occasionally to apply the iterative closest point (ICP) algorithm for refinement of the pose estimate. Preference for systems using sparse features for localisation might be caused by the fact that in monocular systems, direct odometry estimation from raw frames, *i.e.*, without extracting sparse features, inherently forced to estimate the dense depth or optical flow map between frames simultaneously or prior to the camera motion. These are ill posed problems with more unknowns than constraints and requiring from the use of regularisation and variational methods for their resolution [36]. In addition to this, dense methods require high computational power for real time computation of pixel-wise operations. In this sense, advent of new generation CPUs and high performance GPUs almost simultaneously to RGB-D sensors allowed for a significant cost reduction of dense algorithms due

*This work was supported by spanish project DPI2012-31781, FEDER and FPU scholarship AP-2012-5507.

†Daniel Gutiérrez-Gómez and Josechu Guerrero are with the Departamento de Informática e Ingeniería de Sistemas (DIIS) y el Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Spain. {danielgg,josechu.guerrero}@unizar.es

‡Walterio Mayol-Cuevas is with the Department of Computer Science, University of Bristol, UK. wmayol@cs.bris.ac.uk

to new programming paradigms which allowed for high paralellisation of per pixel operations.

One of the first and maybe most known approaches for direct odometry estimation is KinectFusion [35], which using only the depth channel, is able to estimate the odometry and a dense map by using only the ICP algorithm. Almost alongside with KinectFusion came more direct approaches for odometry estimation either minimising the pixel-wise photometric error [41] or both photometric and geometric errors [46] between pairs of close enough frames.

This paper extends on our previous conference paper [12] which presented a direct visual odometry method minimising both types of error. Our main contribution is the novelty of using the inverse depth to parametrise the geometric error instead of the depth as most works do. In monocular vision, the ability of inverse depth (if measured along the camera optical axis) or inverse distance (if measured along the projected ray) to easily deal with points at long distances [5] has been shown to lead to an improvement in performance [40]. In depth range systems, though there is no need to deal with points at distances greater than the maximum camera range, inverse depth has still the theoretical benefit of fitting better to the depth error model of a RGB-D camera. This potential benefit of using the inverse depth is experimentally validated in the Technische Universität München (TUM) benchmarking datasets [44] showing the better performance of the geometric error based on inverse depth. As additional contribution, though equivalent in essence, the problem formulation is slightly varied with respect to related works, first linearising the flow equations, obtaining generic linear 3D flow equations and then applying the assumption of small rigid scene motion between frames to get the linear constraints just on camera motion parameters. We also evaluate the performance under different robust cost functions (Huber, Tukey biweight and Student's t distribution-based estimator) and two different methods in the state-of-the-art to compute the uncertainty-based scaling parameters of an error distribution: one with the Median Absolute Deviation (MAD), and one with its Maximum Likelihood (ML) estimator given the cost function used for robust optimisation. We have implemented this method within the scope of PCL (Point Cloud Library) [39] as a branch of the code for large scale KinectFusion [3], where the original ICP system for odometry estimation has been completely substituted by our method, while algorithms for dense volumetric mapping and volume shifting have been kept unchanged. A fork of PCL including our modification is available for download [1].

In addition to this, the novel contributions w.r.t. our previous conference paper are:

- A method for statistically based selection of a reduced number of pixel samples for the computation of the scaling parameters of the error distributions
- The use of a mutual visibility criteria for reference frame switching, as an alternative to alignment of consecutive frames
- A more exhaustive evaluation of our method comparing against an extended set of state-of-the art approaches for RGB-D localisation and mapping, not only in terms Relative Pose Error (RPE), measured as drift per time unit, but also in terms of the Absolute Trajectory Error (ATE).
- Evaluation of our method in the synthetic datasets for RGB-D benchmarking presented in [13]
- An evaluation of the Hessian ill-posedness in structurally and texturally poor datasets

[1] http://webdiis.unizar.es/~danielgg/code.html

## 2 Related Work

We classify the related work in RGB-D visual odometry into two categories: methods which rely at some point on the extraction and matching of sparse RGB features and those which are completely dense performing pixel-wise minimisation of photometric and/or geometric constraints from the intensity and depth maps.

### 2.1 Sparse feature-based methods

One of the first SLAM methods with RGB-D cameras was presented by Henry *et al.* [14]. They perform the visual odometry estimation between two frames in two steps. In the first step SIFT features are matched and lifted to 3D points using the available depth information. Then RANSAC-based 3D alignment between the features in both frames is applied to find the initial estimate of the relative rigid transformation. In the second step the motion estimate is refined by joint minimisation of the euclidean 3D distance between inlier correspondences from previous step and the point to plane distance from the ICP alignment between point clouds. Both error contributions have to be weighted by a parameter $\alpha$ which is empirically estimated. This method is further developed in [15], where main changes in the visual odometry estimation process are the substitution of SIFT features by FAST features, and the substitution of the 3D euclidean error between features by the image reprojection error.

In [9], Endres *et al.* propose a RGB-D SLAM system, which similarly to [14] estimate the initial transformation by RANSAC-based 3D alignment of sparse features. However, in the refinement step they only minimise the point cloud alignment error from the ICP algorithm. The method is improved in [10], including an Environment

Measurement Model to prune wrong motion estimates which passed undetected in the RANSAC and ICP steps.

Maybe one weakness of both methods is that both seem to rely on the detection of loop closures to provide accurate map and trajectories estimates. The accuracy of the methods for visual odometry, *i.e.*, computing the motion estimate only between temporally close frames, can not be assessed since quantitative evaluation is performed after the loop closure step. Also they tend to directly extend the algorithm from visual odometry estimation for loop closure, rather than making first an appearance-based selection of loop candidates. In [15] authors attempt to align each new created keyframe with all the previous keyframes by RANSAC, which would lead to prohibitive computational costs for large datasets. In [10] in turn authors attempt to close loops in a random sample from a set of keyframes and each frame-to-frame motion calculation between candidates for loop closure is parallelised, thus keeping bounded the computational cost and close to the frame rate of the camera (between 5 and 15 Hz). This has proven to be a successful approach in the RGB-D TUM datasets, usually comprising short datasets with frequent loop closures. However in the case for large datasets with few loop closures, it could be argued that as the pool of selectionable keyframes steadily grew, the chances of randomly picking keyframes with successful loop closures would be increasingly reduced.

In [8], Dong *et al.* propose the combination of 3D RANSAC alignment and large scale KinectFusion for RGB-D dense mapping.

## 2.2 Direct methods

In the last years, approaches which estimate the camera motion directly from the images without a previous extraction and matching of sparse features have become more and more popular. The main characteristic of these methods is that they compute the motion estimate between frames from pixel-wise constraints instead of sparse feature correspondences. Since direct methods make use of the whole dense information contained in the image, they are likely to offer better accuracy for camera tracking than methods based on sparse features. The main restriction of direct methods is that inter-frame motion must be small, producing pixel disparities between frames of a few pixels. Though this restriction can be relaxed up to some degree with coarse-to-fine approaches applying image downsampling; the main consequence is that while direct methods can be used with video sequences, where the temporal proximity between frames generally guarantees small pixel disparities, for temporally unrelated pairs of frames, as occurs in loop closure, sparse feature methods perform better.

Maybe one of the first dense approaches with RGB-D cameras is KinectFusion by Newcombe *et al.* [35]. KinectFusion is composed by two different modules, one for camera tracking and one for dense volumetric mapping. For each new frame first the motion is estimated by frame-to-model ICP alignment of depth maps, *i.e.*, current depth map is aligned with the depth map raycasted from a voxelized 3D model. Then, current depth map is integrated in the 3D volume using a truncated signed distance function. One drawback of the original KinectFusion is its limitation to small workspaces, which was nevertheless solved in latter works by using a cyclical buffer to shift the volume as the camera explores the environment [3], [47].

Bylow *et al.* [4] proposed a method which, as KinectFusion, uses only the depth fusion, but rather than raycasting a depth map from the model for posterior ICP-alignment, the camera is tracked by directly minimising the signed distance function between the current warped depth map and the model surface defined in the voxelised volume. This results in a better accuracy and similar real-time computational performance compared to KinectFusion.

In contrast to direct approaches using only geometric information, Steinbrücke *et al.* [41] presented a method for visual odometry estimation based on the pixel-wise minimisation of the photometric error between consecutive frames, showing that if interframe motion is small enough their approach is more accurate and computationally efficient than ICP alignment. Audras *et al.* [1] propose a similar method, but instead of standard least squares they propose the Huber robust cost function in order to gain robustness to outliers, *e.g.*, moving objects or occlusions. An information selection scheme is used to prune pixels in homogeneous regions and gain computational performance and motion is estimated by aligning the current and a reference frame, switching the reference frame when the Median Absolute Deviation (MAD) of the error of the aligned frames is above a given threshold.

In [22], Kerl *et al.* extend the method described [41] by modelling the photometric error by a Student's t-distribution. This leads to a cost function which shows to be robust to outliers and performs better than other widely used estimators like Huber's or Tukey's.

Klose *et al.* [25] optimise the Tukey-robustified photometric error by performing Efficient Second Order Minimisation (ESM) between a reference and current frame, using the accumulated camera motion, to decide when to switch the reference frame. Also, to gain robustness to ilumination changes they include parameters modelling the variation in contrast and brightness in the optimised variables.

Following the paradigm of working on 3D models [35], Stuckler and Behnke propose in [42] and [43] converting the RGB and depth images into multiresolution surfel maps by using a voxel octree representation. Each surfel maintains a shape-texture descriptor, which guide

data association between surfels in different maps during camera pose estimation. To alleviate the odometry drift they register the current frame with respect to the latest keyframe. A new keyframe is inserted when camera motion w.r.t. last keyframe is large enough. They also propose a loop closure technique where loop closure candidates are randomly sampled from a probability density function which positively weights the selection of spatially closer keyframes.

Direct motion estimation minimising both the geometric and photometric residuals simultaneously was proposed first by Tykkala *et al.* [46]. To solve the problematic of mixing residuals in different magnitudes they propose the heuristic of weighting the depth residuals by the quotient of the medians of the intensity and depth maps. In [6], Damen *et al.* propose an ESM approach to minimise both residuals, weighting their contributions by an empirically set parameter. In [49], Whelan *et al.* propose to compute the visual odometry by mixing the costs functions from [35] and [41], also weighting them by an empirically set parameter. Based on this work Whelan et al. proposed in [48] a RGBD-D SLAM system with volumetric fusion performing appearance-based loop closing to improve the systme accuracy. The use of heuristics to weight both error contributions can be risky, since a tuning which works well for some dataset could not do so in other ones. For this reason it is advisable to reduce their use as much as possible. In this sense Kerl *et al.* [21] propose the computation of an automatic scaling matrix based on the covariance of the photometric and geometric pixel residuals, which produces a rigorous normalisation of both residuals. In addition to this automatic scaling they propose the estimation of the camera motion with respect to keyframes, which are switched following an entropy-based criteria, and the inclusion of a simple but effective loop closure method based on keyframes spatial proximity to further refine the final odometry estimation.

In all of the described approaches the use of a constant scaling parameter, either heuristic or automatic, for all the geometric residuals is prone to be a source of inaccuracies in the estimation process due to the quadratic grow of the depth uncertainty in RGB-D sensors [23]. Meilland *et al.* [33] take this fact into account, weighting the depth residuals by the inverse squared depth, but still use additional heuristic parameters to weight photometric and depth residuals. Also, under the frequent assumption of a symmetric, generally Gaussian, distribution of the disparity error; the depth which is inversely proportional to the image disparity, is not symmetric (Fig. 1), and thus inaccurately modeled by the robust cost functions frequently employed in the literature.

Inverse depth, in turn, depends linearly on the disparity, which means that it follows the same error distribution and thus its uncertainty is constant for any depth. In spite of this, to the best of our knowledge, the use of inverse depth for dense visual odometry with RGB-D cameras has been only proposed in by Lui *et al.* [28], by extending the ICP algorithm from KinectFusion. However its performance is only tested on short sequences, lacking from a thorough evaluation on larger RGB-D sequences and a comparison against state-of-the-art methods for dense RGB-D odometry estimation.

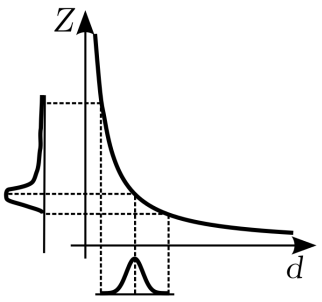



Figure 1: Assuming that the disparity ($d$) error follows a Gaussian or, more generally, a symmetric distribution, the depth ($Z \propto \frac{1}{d}$) error distribution is not Gaussian, not even symmetric. The asymmetry is more pronounced for higher $Z$.

# 3 Linear Visual Odometry Constraints from Optical Flow

In this section we derive the visual odometry pixel-wise constraints through the flow equations obtained from the photometric and geometric constraints between two camera positions.

## 3.1 Optical flow equations

Let us denote two camera frames as $A$ and $B$, at instants $t$ and $t + \Delta t$ respectively. Given the intensity images $\mathcal{I}_A$ and $\mathcal{I}_B$, and inverse depth maps $\mathcal{W}_A$ and $\mathcal{W}_B$ defined over the image domain $\Omega \subset \mathbb{P}^2$, for an image point $\mathbf{p} = (u\ v\ 1)^T \in \Omega$ in frame $A$, the following constraints hold:

$$\mathcal{I}_B(\mathbf{p} + \mathbf{\Delta p}) = \mathcal{I}_A(\mathbf{p}) \tag{1}$$

$$\mathcal{W}_B(\mathbf{p} + \mathbf{\Delta p}) = \frac{1}{\mathbf{e_z^T} \mathbf{X}_B}, \tag{2}$$

where $\mathbf{X}_B$ is the 3D point lifted from pixel $\mathbf{p} + \mathbf{\Delta p}$ in frame $B$, $\mathbf{\Delta p} = (\Delta u\ \Delta v\ 0)^T$ is the displacement of one point from frame $A$ to $B$, and $\mathbf{e_z}^T = (0\ 0\ 1)$. The constraint in intensity assumes constant illumination of one scene point. The second constraint is the measurement model of the depth sensor at frame $B$.

Assuming small pixel displacements between frames we compute the flow equations from (1) and (2):

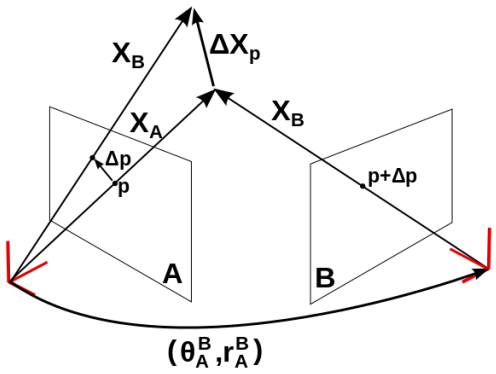



Figure 2: Schematic representation of optical and scene flow between two frames A and B.

$$\nabla\mathcal{I}_A(\mathbf{p})\mathbf{\Delta p} + \mathcal{I}_B(\mathbf{p}) = \mathcal{I}_A(\mathbf{p}) \tag{3}$$

$$\nabla\mathcal{W}_A(\mathbf{p})\mathbf{\Delta p} + \mathcal{W}_B(\mathbf{p}) = \frac{1}{\mathbf{e_z}^T\mathbf{X}_B}, \tag{4}$$

where the gradient operators $\nabla\mathcal{I} = \left(\frac{\partial\mathcal{I}}{\partial u}\ \frac{\partial\mathcal{I}}{\partial v}\ 0\right)$ and $\nabla\mathcal{W} = \left(\frac{\partial\mathcal{W}}{\partial u}\ \frac{\partial\mathcal{W}}{\partial v}\ 0\right)$.

### 3.2 Projection model

A world point $\mathbf{X}$ is projected in the image point $\mathbf{p}$ by:

$$\mathbf{p} = \pi\left(\mathbf{X}\right) = \mathbf{K}\frac{\mathbf{X}}{\mathbf{e_z}^T\mathbf{X}} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}\frac{\mathbf{X}}{\mathbf{e_z}^T\mathbf{X}}, \tag{5}$$

where $\mathbf{K}$ is the conventional calibration matrix, including the camera intrinsic parameters.

Inverse depth measurements $\mathcal{W}(\mathbf{p}) = \frac{1}{\mathbf{e_z}^T\mathbf{X}}$ allow to lift 2D points from the image to 3D coordinates by the inverse projection function:

$$\mathbf{X} = \pi^{-1}(\mathbf{p}) = \frac{1}{\mathcal{W}(\mathbf{p})}\mathbf{K}^{-1}\mathbf{p}. \tag{6}$$

### 3.3 3D flow equations

Flow constraints (3) and (4), can be manipulated to get constraints on the 3D flow at one pixel, which is denoted as $\mathbf{\Delta X_p} \doteq \mathbf{X}_B - \mathbf{X}_A$ (see Fig. 2). Using this relation, first we compute the first order Taylor expansion of the inverse depth of one point at frame $B$:

$$\frac{1}{\mathbf{e_z}^T\mathbf{X}_B} = \frac{1}{\mathbf{e_z}^T\mathbf{X}_A} - \frac{1}{(\mathbf{e_z}^T\mathbf{X}_A)^2}\mathbf{e_z}^T\mathbf{\Delta X_p} + \mathcal{O}\left(\left|\left|\mathbf{e_z}^T\mathbf{\Delta X_p}\right|\right|^2\right) \tag{7}$$

$$\approx \mathcal{W}_A(\mathbf{p}) - \mathcal{W}_A^2(\mathbf{p})\mathbf{e_z}^T\mathbf{\Delta X_p}. \tag{8}$$

Using this relation and the camera projection model we get also:

$$\mathbf{\Delta p} = \mathbf{K}\frac{\mathbf{X}_B}{\mathbf{e_z}^T\mathbf{X}_B} - \mathbf{K}\frac{\mathbf{X}_A}{\mathbf{e_z}^T\mathbf{X}_A} \tag{9}$$

$$\overset{(8)}{=} \mathbf{K}\mathbf{X}_B\left(\mathcal{W}_A(\mathbf{p}) - \mathcal{W}_A^2(\mathbf{p})\mathbf{e_z}^T\mathbf{\Delta X_p}\right) - \mathbf{K}\mathbf{X}_A\mathcal{W}_A(\mathbf{p}) \tag{10}$$

$$\overset{(5)}{=} \mathcal{W}_A(\mathbf{p})\left(\mathbf{K} - \mathbf{p}\mathbf{e_z}^T\right)\mathbf{\Delta X_p} - \mathbf{K}\mathcal{W}_A^2(\mathbf{p})\mathbf{\Delta X_p}\mathbf{e_z}^T\mathbf{\Delta X_p} \tag{11}$$

$$\approx \mathcal{W}_A(\mathbf{p})\left(\mathbf{K} - \mathbf{p}\mathbf{e_z}^T\right)\mathbf{\Delta X_p}. \tag{12}$$

And substituting in (3) and (4) we get:

$$\mathcal{W}_A(\mathbf{p})\nabla\mathcal{I}_A(\mathbf{p})\left(\mathbf{K} - \mathbf{p}\mathbf{e_z}^T\right)\mathbf{\Delta X_p} + \mathcal{I}_B(\mathbf{p}) - \mathcal{I}_A(\mathbf{p}) = 0 \tag{13}$$

$$\mathcal{W}_A(\mathbf{p})\left(\nabla\mathcal{W}_A(\mathbf{p})\left(\mathbf{K} - \mathbf{p}\mathbf{e_z}^T\right) + \mathcal{W}_A(\mathbf{p})\mathbf{e_z}^T\right)\mathbf{\Delta X_p} + \\ + \mathcal{W}_B(\mathbf{p}) - \mathcal{W}_A(\mathbf{p}) = 0. \tag{14}$$

### 3.4 Rigid motion

We have obtained general flow equations taking small pixel displacement as the only assumption. Only (3) presents a dense optical 2D flow estimation problem [18], while (13) and (14) involve a dense scene 3D flow problem [16]. Both are ill posed problems which require regularisation and variational methods to reach a solution.

We focus instead on RGB-D visual odometry estimation. This implies the assumption of a rigid scene, i.e., the displacements $\mathbf{\Delta X_p}$ of each of the $W_{im}H_{im}$ points projected on the image frame are due only to the motion of the camera, which has 6 DoF. Assuming a small motion described by the rotation and translation pair $({}_A\mathbf{R}^B, \mathbf{r}_B^A) \in \mathbb{SE}(3)$ we have:

$$\begin{aligned}\mathbf{\Delta X_p} &= {}_B\mathbf{R}^A\mathbf{X}_A + \mathbf{r}_B^A - \mathbf{X}_A \\ &= \left(\mathbf{I} + [\boldsymbol{\theta}_B^A]_\times\right)\mathbf{X}_A + \mathbf{r}_B^A - \mathbf{X}_A + \mathcal{O}\left(\left|\left|[\boldsymbol{\theta}_B^A]_\times^2\mathbf{X}_A\right|\right|\right) \\ &\approx \mathbf{r}_B^A - [\pi^{-1}(\mathbf{p})]_\times\boldsymbol{\theta}_B^A = \mathbf{M}(\mathbf{p})\boldsymbol{\xi}_B^A.\end{aligned} \tag{15}$$

where $[\cdot]_\times$ denotes the antisymmetric matrix from a vector and $\boldsymbol{\theta}_B^A)$ is the logarithmic map of ${}_A\mathbf{R}^B$. Note that $\boldsymbol{\xi}_B^A = (\mathbf{r}_B^A;\ \boldsymbol{\theta}_B^A)$ is not a twist, *i.e.*, $\boldsymbol{\xi}_B^A \notin \mathfrak{se}(3)$, since $\mathbf{r}_B^A$ is yet the translation part of the rigid motion. Eq. (15) leads to a well-posed problem with 6 unknowns for nearly $W_{im}H_{im}$ constraints, excluding pixels without depth measurements, with the following residuals:

$$r_\mathcal{I}(\mathbf{p}, \boldsymbol{\xi}) = \mathcal{W}_A(\mathbf{p})\nabla\mathcal{I}_A(\mathbf{p})(\mathbf{K} - \mathbf{p}\mathbf{e_z}^T)\mathbf{M}(\mathbf{p})\boldsymbol{\xi} + \\ + \mathcal{I}_B(\mathbf{p}) - \mathcal{I}_A(\mathbf{p}) \tag{16}$$

$$r_\mathcal{W}(\mathbf{p}, \boldsymbol{\xi}) = \mathcal{W}_A(\mathbf{p})\left(\nabla\mathcal{W}_A(\mathbf{p})(\mathbf{K} - \mathbf{p}\mathbf{e_z}^T) + \mathcal{W}_A(\mathbf{p})\mathbf{e_z}^T\right)\mathbf{M}(\mathbf{p})\boldsymbol{\xi} + \\ + \mathcal{W}_B(\mathbf{p}) - \mathcal{W}_A(\mathbf{p}), \tag{17}$$

which can be straightforwardly minimised by standard Gauss-Newton least squares.

Note that in the monocular RGB case, no depth is provided and only the constraint (16) would be used. Thus $\mathcal{W}_A(\mathbf{p})$ becomes an unknown yielding an ill-posed problem with $W_{im}H_{im}$ constraints for $W_{im}H_{im}+6$ unknowns, which is solved using optical flow variational methods [36], or by performing variable baseline stereo matching [11].

# 4 Visual Odometry Estimation by Iterative Optimisation

With the proposed residuals, $\boldsymbol{\xi}_B^A$ is computed as the solution to the following optimisation problem:

$$\boldsymbol{\xi}_B^A = \underset{\boldsymbol{\xi}}{\operatorname{argmin}} \sum_{\mathbf{p}\in\Omega} \rho\left(\frac{r_{\mathcal{I}}(\mathbf{p},\boldsymbol{\xi})}{\sigma_{r_{\mathcal{I}}}}\right) + \rho\left(\frac{r_{\mathcal{W}}(\mathbf{p},\boldsymbol{\xi})}{\sigma_{r_{\mathcal{W}}}}\right), \quad (18)$$

where $\rho(x)$ is a generic cost function which must be symmetric, definite positive and $\rho(0) = 0$. $\sigma_{r_{\mathcal{I}}}$ and $\sigma_{r_{\mathcal{W}}}$ are scaling parameters which capture the uncertainty in intensity and inverse depth residuals, and allow for normalisation of residuals in different magnitudes. The choice $\rho(x) = \frac{x^2}{2}$ results in standard least-squares linear optimisation. Nevertheless to gain robustness against outliers, *e.g.*, pixels belonging to non-static elements, robust M-estimators are usually employed. Optimisation with robust cost functions is addressed by the Iteratively Reweighted Least Squares algorithm (IRLS) [17], which results in a linear least-squares problem to be solved at each iteration:

$$\boldsymbol{\xi}_B^A = \underset{\boldsymbol{\xi}}{\operatorname{argmin}} \sum_{\mathbf{p}\in\Omega} \omega\left(\frac{\breve{r}_{\mathcal{I}}(\mathbf{p})}{\sigma_{r_{\mathcal{I}}}}\right)\frac{r_{\mathcal{I}}^2(\mathbf{p},\boldsymbol{\xi})}{\sigma_{r_{\mathcal{I}}}^2} + \omega\left(\frac{\breve{r}_{\mathcal{W}}(\mathbf{p})}{\sigma_{r_{\mathcal{W}}}}\right)\frac{r_{\mathcal{W}}^2(\mathbf{p},\boldsymbol{\xi})}{\sigma_{r_{\mathcal{W}}}^2}, \quad (19)$$

where $\breve{r}_{\mathcal{I}}(\mathbf{p})$ and $\breve{r}_{\mathcal{W}}(\mathbf{p})$ denote the initial residuals computed after updating the camera motion at previous iteration, and the weighting function $\omega(x)$ depends on the used M-estimator. Cost and weight functions for different M-estimators can be found in [50].

Rigid motion between frames is computed in a coarse-to-fine manner using image pyramids, performing a number of iterations at each pyramid level. Let us have the intensity and inverse depth image pairs $\{\mathcal{I}_k,\, \mathcal{W}_k\}$ and $\{\mathcal{I}_{k+1},\, \mathcal{W}_{k+1}\}$, between consecutive frames $k$ and $k+1$. At the start and every time we step down to the next pyramid level, we set $\{\mathcal{I}_A,\, \mathcal{W}_A\} = \left\{\mathcal{I}_k^{(pyr)},\, \mathcal{W}_k^{(pyr)}\right\}$, and compute $\{\nabla\mathcal{I}_A,\, \nabla\mathcal{W}_A\}$. Initial camera motion, expressed by the transform ${}_k\mathbf{T}_{(0)}^{k+1}$ is initialised assuming a constant velocity, *i.e.*, ${}_k\mathbf{T}_{(0)}^{k+1} = {}_{k-1}\mathbf{T}^k$.

After initialisation, the following steps are performed at each iteration $\gamma$: image warping, scaling parameters computation, optimisation and pose composition.

## 4.1 Image Warping

Image warping is performed at the start of every iteration in order to reset the incremental motion estimate to $\boldsymbol{\xi}_B^{A(\gamma)} = 0$, instead of accumulating it. This is done to avoid unrealistic intensity or inverse depth estimates in frame $B$ beyond the sensor measurement limits, as it can be verified if we take a look back to the left members of (3) and (4). At each iteration, $\{\mathcal{I}_{k+1},\, \mathcal{W}_{k+1}\}$ are warped towards frame $k$ using the current motion estimate ${}_k\mathbf{T}_{(\gamma)}^{k+1}$, resulting in the warped images $\{\mathcal{I}_B^{(\gamma)},\, \mathcal{W}_B^{(\gamma)}\}$. This is done by reverse warping in the following steps:

- Given a pixel $\mathbf{p}$ in the destination warped image, the corresponding pixel $\mathbf{p}_{k+1}^{(\gamma)}$ in the source image is obtained as:

$$\mathbf{X}_{k+1}^{(\gamma)} = {}_{k+1}\mathbf{R}_{(\gamma)}^k\mathbf{K}^{-1}\mathbf{p}\frac{1}{\mathcal{W}_k(\mathbf{p})} + \mathbf{r}_{k+1}^{k(\gamma)}, \quad (20)$$

$$\mathbf{p}_{k+1}^{(\gamma)} = \mathbf{p} + \boldsymbol{\Delta}\mathbf{p}^{(\gamma)} = \mathbf{K}\frac{\mathbf{X}_{k+1}^{(\gamma)}}{\mathbf{e}_{\mathbf{z}}^T\mathbf{X}_{k+1}^{(\gamma)}}, \quad (21)$$

- By using (1) and (2) and resetting $\mathbf{X}_{k+1}^{(\gamma)} = {}_{k+1}\mathbf{R}_{(\gamma)}^k\mathbf{K}^{-1}\mathbf{p}\frac{1}{\mathcal{W}_B^{(\gamma)}(\mathbf{p})} + \mathbf{r}_{k+1}^{k(\gamma)}$, compute the warped intensity and inverse depth maps, $\mathcal{I}_B^{(\gamma)}$ and $\mathcal{W}_B^{(\gamma)}$ as:

$$\mathcal{I}_B^{(\gamma)}(\mathbf{p}) = \mathcal{I}_{k+1}(\mathbf{p}_{k+1}^{(\gamma)}) \quad (22)$$

$$\mathcal{W}_B^{(\gamma)}(\mathbf{p}) = \frac{\mathbf{e}_{\mathbf{z}}^T\,{}_{k+1}\mathbf{R}_{(\gamma)}^k\mathbf{K}^{-1}\mathbf{p}}{1 - \mathbf{e}_{\mathbf{z}}^T\mathbf{r}_{k+1}^{k(\gamma)}\mathcal{W}_{k+1}(\mathbf{p}_{k+1}^{(\gamma)})}\mathcal{W}_{k+1}(\mathbf{p}_{k+1}^{(\gamma)}), \quad (23)$$

  where $\mathcal{I}_{k+1}(\mathbf{p}_{k+1}^{(\gamma)})$ and $\mathcal{W}_{k+1}(\mathbf{p}_{k+1}^{(\gamma)})$ are obtained by bilinear interpolation, which is efficiently computed with CUDA capable NVIDIA GPUs using *texture memory*.

  Warping is performed at the level with highest resolution. Once the warping is done, $\{\mathcal{I}_B^{(\gamma)},\, \mathcal{W}_B^{(\gamma)}\}$ are downsampled to the pyramid level where current optimisation step is taking place.

## 4.2 Scaling parameters

In a proper minimisation problem, specially when mixing residuals in different magnitudes, the scaling parameters related with the covariance of the residuals need to be provided. In some cases these scaling parameters are known or can be estimated before the optimisation and thus they can be introduced as constants. However in other cases they are difficult to know and they have to be computed prior to

every optimisation step from the current estimates of the residuals. In principle we assume that these parameters are not known and to obtain them we first compute the initial residuals at $\boldsymbol{\xi}_B^{A(\gamma)} = 0$ :

$$\breve{r}_{\{\mathcal{I},\mathcal{W}\}}(\mathbf{p}) = \{\mathcal{I}_B^{(\gamma)}(\mathbf{p}),\ \mathcal{W}_B^{(\gamma)}(\mathbf{p})\} - \{\mathcal{I}_A(\mathbf{p}),\ \mathcal{W}_A(\mathbf{p})\}. \tag{24}$$

Scaling parameters $\sigma_{r_\mathcal{I}}$ and $\sigma_{r_\mathcal{W}}$ can then be computed by the Median Absolute Deviation (MAD):

$$\sigma^{MAD}_{r_{\{\mathcal{I},\mathcal{W}\}}} = 1.4286 \operatorname*{med}_{\mathbf{p}} |\breve{r}_{\{\mathcal{I},\mathcal{W}\}}(\mathbf{p}) - \operatorname*{med}_{\mathbf{p}} \left(\breve{r}_{\{\mathcal{I},\mathcal{W}\}}(\mathbf{p})\right)|, \tag{25}$$

or alternatively they can be computed by their Maximum Likelihood (ML) estimator, noticing that for any cost function $\rho(\frac{r-\mu}{\sigma})$ we can obtain the associated likelihood function as:

$$f_\rho(r|\mu,\sigma) = \frac{K_\rho}{\sigma} \exp\left(-\rho\left(\frac{r-\mu}{\sigma}\right)\right), \tag{26}$$

where $K_\rho$ is a scaling constant for $\int_{-\infty}^{\infty} f_\rho(r|\mu,\sigma)dr = 1$, and $\mu$ and $\sigma$ the location and the scaling parameter of the residuals respectively. The scaling parameters would be computed by iteratively solving:

$$\left[\mu^{ML}_{r_{\{\mathcal{I},\mathcal{W}\}}}, \sigma^{ML}_{r_{\{\mathcal{I},\mathcal{W}\}}}\right] = \operatorname*{argmin}_{\mu,\,\sigma} \sum_{\mathbf{p}\in\Omega} \log\sigma + \rho\left(\frac{\breve{r}_{\{\mathcal{I},\mathcal{W}\}}(\mathbf{p}) - \mu}{\sigma}\right), \tag{27}$$

taking $\rho(x) = \frac{x^2}{2}$ in the first iteration to compute the initial seed and then switching to our selected cost function. Though not explicitly required, location parameters $\mu^{ML}_{r_{\{\mathcal{I},\mathcal{W}\}}}$ are also calculated since the scaling parameters depend on their estimate.

Computing the scaling parameters using all the pixels in the image can involve a high computational cost. As an alternative, we propose computing the scale parameter taking only a sample from all the pixel residuals. Following a similar reasoning to the typically followed to compute the minimum number of iterations in a RANSAC scheme, we can determine statistically the minimum sample size $N$ to reach a relative precision $\epsilon$ in the scale parameter estimation with a confidence $1-\alpha$. Assuming that the sum of the weighted squared normalised residuals follows a chi squared distribution, we have that:

$$\frac{(N-1)\hat{\sigma}^2}{\sigma^2} \sim \chi^2_{N-1}, \tag{28}$$

where $\sigma$ is the true scaling parameter and $\hat{\sigma}$ its estimate. We can define then a confidence interval with the desired relative precision in our estimate.

$$P\left((1-\epsilon)\sigma^2 < \hat{\sigma}^2 < (1+\epsilon)\sigma^2\right) = \\ = P\left((1-\epsilon)(N-1) < \frac{(N-1)\hat{\sigma}^2}{\sigma^2} < (1+\epsilon)(N-1)\right) = 1-\alpha \tag{29}$$

Since $N > 50$, we can approximate the chi squared distribution by a Gaussian distribution, $\chi^2_{N-1} \sim \mathcal{N}(N-1, 2(N-1))$, and then we have:

$$P\left(-\frac{\epsilon(N-1)}{\sqrt{2(N-1)}} < z < \frac{\epsilon(N-1)}{\sqrt{2(N-1)}}\right) = 1-\alpha, \tag{30}$$

where $z$ is a standard normally distributed random variable. From the previous expression we obtain the minimum required number of samples to obtain a desired precision in the estimated scaling parameter with a given confidence level.

$$N = \frac{2z^2_{1-\frac{\alpha}{2}}}{\epsilon^2} + 1. \tag{31}$$

Taking a sample of $N = 10000$ pixels we obtain a relative precision $\epsilon = 0.05$ for the scaling parameter with a confidence level greater than $99.9\%$.

### 4.3 Robust optimisation

We consider for comparison 3 different cost functions frequently used in the related literature: Huber [20], Tukey biweight [2] and a Student's t-distribution-based estimator [27], which will be denoted as Student in advance. The constants for Huber and Tukey estimators are set up to $1.345$ and $4.685$ respectively, which yield an asymptotic relative efficiency (ARE) of $95\%$ in case the error followed a Gaussian distribution. The number of degrees of freedom of the Student estimator is set up to $\nu = 5$ as in [22]. We have verified numerically that the ARE of the Student's estimator with this setup is near to $94\%$ for a normally distributed error.

Given an M-estimator, once we have the initial residuals and the scaling parameters the computation of the weights for IRLS is straightforward. This is followed by the computation of $\boldsymbol{\xi}_B^{A(\gamma)}$ by solving the linear optimisation in (19).

### 4.4 Motion update

After each iteration the motion estimate between frames $k$ and $k+1$ is updated by the current incremental estimate:

$${}_k\mathbf{T}^{k+1}_{(\gamma+1)} = \begin{pmatrix} \exp([\boldsymbol{\theta}_B^{A(\gamma)}]_\times) & \mathbf{r}_B^A \\ 0 & 1 \end{pmatrix}^{-1} {}_k\mathbf{T}^{k+1}_{(\gamma)}. \tag{32}$$

### 4.5 Keyframe selection by mutual covisibility ratios

Taking a reference frame for estimation of the visual odometry requires a switching strategy. Some works use the rotational and translational distance between frames to take a decision. However we find that this might be a poor criteria since first, how the captured environment changes when the camera translates depends also on the depth of the elements in the scene, and secondly, camera motion does not necessarily result in a variation of the captured scene (consider for example the case of simultaneously translating the camera along its horizontal axis and rotating it around its vertical axis)

Other works use a statistically based criteria for reference frame switching. In [32] authors use the MAD estimator of the standard deviation of the final residuals which is compared with a reference value to make the decision. In [21] instead of the variance of the residuals, authors use the covariance matrix of the computed motion. They take as reference value the covariance of the motion estimate between the last inserted keyframe and its consecutive frame and take a keyframe switching decision based on the ratio between the entropy of the reference covariance and the current estimate's.

Using sparse features a frequently used criteria for keyframe insertion is the visibility of map features in current frame [24], [19]. This criteria has also been used for dense point clouds [31].

In this work we use a mutual covisibility criteria described as follows. Given two frames $A$ and $B$ and the camera motion estimate between them $({}_A\mathbf{R}^B, \mathbf{r}_B^A) \in \mathbb{SE}(3)$, we transfer pixels from $A$ to $B$ using (20) and (21). Then if the following two conditions are met:

- $\mathbf{p}_B \in \Omega$
- $\left| \mathcal{W}_B(\mathbf{p}_B) - \frac{1}{\mathbf{e_z}^T \mathbf{X}_B} \right| < 3\sigma_w$

a pixel is tagged as visible. First condition rejects points out of the image domain, while the second condition rejects occluded pixels and where $\sigma_w$ is computed using (27) after the last iteration of the visual odometry algorithm. After this test we can compute the visibility ratio:

$$vis\_ratio_{A\to B} = \frac{\#visible\_pixels_{A\to B}}{\#nohole\_pixels_{A\to B}}. \qquad (33)$$

This procedure is repeated switching the role of $A$ and $B$, and then we select the minimum visibility ratio. If this ratio es below a threshold, the reference frame is switched. Different thresholds will be tested in the experiments section.

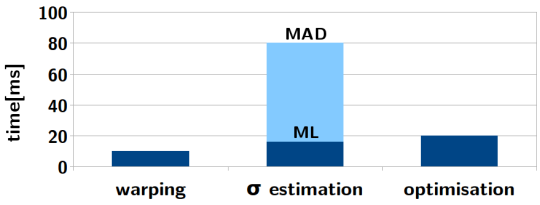



Figure 3: Costs of the processes involved in the computation of the RGB-D visual odometry.

Table 1: Translational drift relative root mean square error (RMSE) in meters per second using different methods for RGB-D visual odometry estimation

| Estimator | Geom. error | $\sigma$ | fr1/desk | fr1/desk2 | fr1/room | fr2/desk |
|---|---|---|---|---|---|---|
| Student | depth | ML | 0.0278 | 0.0425 | 0.0504 | **0.0115** |
| Student | depth | MAD | 0.0271 | 0.0439 | 0.0490 | 0.0121 |
| Student | invDepth | ML | **0.0260** | **0.0387** | 0.0491 | 0.0121 |
| Student | invDepth | MAD | **0.0260** | 0.0396 | 0.0485 | 0.0122 |
| Tukey | depth | ML | 0.0292 | 0.0808 | 0.0502 | 0.0143 |
| Tukey | depth | MAD | 0.0271 | 0.0527 | 0.0483 | 0.0142 |
| Tukey | invDepth | ML | 0.0381 | 0.0720 | 0.0485 | 0.0135 |
| Tukey | invDepth | MAD | 0.0287 | 0.0422 | **0.0471** | 0.0131 |
| Huber | depth | ML | 0.0322 | 0.0495 | 0.0681 | 0.0193 |
| Huber | depth | MAD | 0.0322 | 0.0496 | 0.0662 | 0.0233 |
| Huber | invDepth | ML | 0.0289 | 0.0453 | 0.0640 | 0.0209 |
| Huber | invDepth | MAD | 0.0280 | 0.0435 | 0.0606 | 0.0219 |

### 4.6 Enhancement of the computational performance

Optimisation is performed in a coarse-to-fine scheme at 3 pyramid levels (160x120, 320x240 and 640x480). The naive approach offers the highest precision performing a fixed number of 10 iterations at each level. This results in cost per frame of about 50 ms, which is broke down into the costs of the different processes in Fig. 3. Alternatively, to improve the time performance we consider the following optimisations:

- Skip optimisation on the highest resolution level.
- In Fig. 3 it can be observed that one important fraction of the time is employed in estimating the scale parameters $\sigma_\mathcal{I}$ and $\sigma_\mathcal{W}$. This cost can be completely eliminated if we fix the scaling parameters for the optimisation. We propose taking $\sigma_\mathcal{I} = 5$, with $\mathcal{I}(\mathbf{p}) \in (0,\ 255)$ and $\sigma_\mathcal{W} = 0.0025\ \mathrm{m}^{-1}$. The choice for $\sigma_\mathcal{I}$ is justified by tests with static sequences while $\sigma_\mathcal{W}$ stems from the precision of the disparity measurements [26],[29].
- Instead of warping at the highest resolution and then downsampling at each iteration, a coarser but more efficient alternative would be downsampling $\{\mathcal{I}_{k+1},\ \mathcal{W}_{k+1}\}$ before optimising and warping on the current pyramid level.

# 5 Experiments

We first evaluate the accuracy of our approach with different configurations comparing it to other works in the literature using the RGB-D dataset from Technische Universität München (TUM) [44] and a synthetic dataset with 2 different scenes [13]. Secondly we study with the same datasets, how both the accuracy and the computation speed are affected when applying the options for enhancement of computational performance from section 4.6. Finally we show a qualitative evaluation of our visual odometry method showing the 3D reconstructions obtained in our own sequences.

The experiments were performed on a desktop computer with Ubuntu 12.04 32-bits and equipped with an Intel Core i5-2500 CPU at 3.30 GHz, 8GB and a nVidia GeFroce 660GTX GPU with 2GB of memory. The implementation was done as an extension of the large scale KinectFusion large scale algorithm from the Point Cloud Library (PCL) [39], where the original ICP system for odometry estimation has been completely substituted by our method. The algorithms for dense volumetric mapping and volume shifting are kept unchanged. A fork of PCL including our modified version is available for download in `http://webdiis.unizar.es/~danielgg/code.html`.

## 5.1 Inverse depth vs depth residuals

We evaluated all the possible combinations of robust cost functions (Huber, Tukey and Student), geometric error parametrisation (depth, inverse depth) and residual scale estimators (MAD, ML). Table 1 shows the different values of the RMSE for the translational drift, measured in m/s, using different approaches. Best results are obtained with the Student estimator, with little difference between using MAD or ML estimators for the scaling parameters. Parametrisation of the geometric error with inverse depth yields an improvement over the depth parametrisation in all the datasets except in $fr2/desk$. It can be noted also that the Huber estimator offers in general the lowest accuracy and that the performance of the Tukey estimator is slightly worse if we use the ML estimator for $\sigma$. However its performance is comparable to Student's if the MAD estimator is used. For the rest of the experiments we use the configuration with Student robust cost function, inverse depth parameterisation for the geometric error and the ML estimator for the scale parameters.

For a more detailed evaluation of how the combination of both geometric and photometric errors affect the accuracy of the estimate and better assess the gain of using inverse depth instead of depth for the geometric error, we have performed another set of tests with optimising on intensity error only, geometric only and the combination of both geometric and intensity error. For the cases where geometric error is used we have switched between inverse depth and depth based errors 2. It can be observed that when considering the geometric error only the superiority of using inverse depth residuals is clearer.

Table 2: Translational drift relative root mean square error (RMSE) in meters per second minimising different types and combinations of errors

| Geom. error | Min. errors | fr1/desk | fr1/desk2 | fr1/room | fr2/desk |
|---|---|---|---|---|---|
| none | phot | 0.0312 | 0.0513 | 0.0503 | 0.0119 |
| invDepth | phot+geom | 0.0260 | 0.0387 | 0.0491 | 0.0121 |
| depth | phot+geom | 0.0278 | 0.0425 | 0.0504 | 0.0115 |
| invDepth | geom | 0.0332 | 0.0454 | 0.0495 | 0.0356 |
| depth | geom | 0.0377 | 0.0562 | 0.0555 | 0.0465 |

## 5.2 Performance vs accuracy

We evaluated how the accuracy and computational performance are affected after applying the modifications proposed in Sec. 4.6 to decrease the computational cost of our approach. Following the order in which they are presented, we denote these as $lvl$ followed by the level index at which optimisation is stopped, $sigmaFix$ and $pyrFirst$. Results of using one or a combination of these modifications are shown in Table 3. It can be observed that either stopping optimisation at level 1 or using constant scale we achieve a computation time in the limits of the camera frame rate of 30 Hz with little lost in accuracy, and even we can reach a reduction to 9ms applying all the proposed optimisations.

## 5.3 State-of-the-art comparative

In this section we first compare our method to state-of-the-art RGB-D visual odometry and SLAM approaches. For the comparison we use 2 publicly available benchmarking datasets: the TUM benchmarking dataset [44], consisting of real image sequences, and a RGB-D benchmarking dataset generated from 2 synthetic scenes, one office and one living room [13]. For our approach we have considered different thresholds for reference frame switching. The evaluation in the TUM datasets has been carried out in the two error metrics proposed by the authors of the TUM datasets: the Relative Pose Error (RPE) in meters per second and the Absolute Trajectory Error (ATE) in meters. For the synthetic datasets we only evaluated the ATE since RPE is not compared in the literature.

In the RPE evaluation (Table 4), we observe that there does not exist a clear difference in accuracy between taking consecutive frames or a reference frame for odometry estimation. With respect to the state-of-the-art, our frame-to-frame approach has the lowest error in the $fr1/desk2$ and $fr2/desk$ datasets. In the $fr1/desk$ dataset our approach is not the best but the results are close to [31]

Table 3: Translational drift and average and maximum computation time per frame for different options to enhance the computational performance

| Approach | RMSE[m/s] | | | | time | |
|---|---|---|---|---|---|---|
| | fr1/desk | fr1/desk2 | fr1/room | fr2/desk | mean[ms] | max[ms] |
| naïve | 0.0260 | 0.0387 | 0.0491 | 0.0121 | 47 | 50 |
| lvl1 | 0.0268 | 0.0407 | 0.0492 | 0.0120 | 26 | 28 |
| pyrFirst | 0.0270 | 0.0401 | 0.0491 | 0.0127 | 40 | 42 |
| pyrFirst+lvl1 | 0.0282 | 0.0416 | 0.0498 | 0.0138 | 18 | 20 |
| $\sigma$Fix | 0.0260 | 0.0389 | 0.0498 | 0.0112 | 34 | 35 |
| $\sigma$Fix+lvl1 | 0.0271 | 0.0399 | 0.0500 | 0.0121 | 17 | 17 |
| $\sigma$Fix+pyrFirst | 0.0272 | 0.0397 | 0.0498 | 0.0118 | 26 | 27 |
| $\sigma$Fix+pyrFirst+lvl1 | 0.0287 | 0.0409 | 0.0500 | 0.0143 | 9 | 10 |

Table 4: Translational drift relative root mean square error (RMSE) in meters per second using different visibility ratio thresholds for keyframe switching and comparison with state-of-the-art approaches

| Visibility ratio | fr1/desk | fr1/desk2 | fr1/room | fr2/desk | fr3/office |
|---|---|---|---|---|---|
| No KF | 0.0260 | 0.0387 | 0.0491 | 0.0121 | 0.0168 |
| $vr_{th} = 0.9$ | 0.0255 | 0.0384 | 0.0473 | **0.0115** | **0.0118** |
| $vr_{th} = 0.8$ | 0.0253 | **0.0382** | 0.0472 | 0.0124 | 0.0120 |
| $vr_{th} = 0.7$ | **0.0246** | 0.0385 | 0.0441 | 0.0131 | 0.0121 |
| $vr_{th} = 0.6$ | 0.0264 | 0.0413 | **0.0423** | 0.0258 | 0.0812 |
| FOVIS ([19], comp. in [47]) | 0.0604 | - | 0.0642 | **0.0136** | - |
| ICP+RGB-D [47] | 0.0393 | - | 0.0622 | 0.0208 | - |
| VP [33] | 0.0259 | - | **0.0351** | 0.0147 | - |
| ESM + Tukey + Aff. Il. [25] | 0.0302 | 0.0526 | 0.0397 | 0.0147 | - |
| RGB+D [21] | 0.036 | **0.049** | 0.058 | - | - |
| RGB+D+KF [21] | 0.030 | 0.055 | 0.048 | - | - |
| RGB+D+KF+Opt [21] | **0.024** | 0.050 | 0.043 | - | - |

and RGB-D+KF+Opt [21], even ours not using reference frames. In $fr1/room$, the better accuracy of [31] is clear.

In terms of ATE (Table 5), we found that though our approach is not the best in the tested real TUM datasets, it shows competitive results considering that many of the methods considered in the comparison include some kind of loop closure method, which significantly helps into reducing this error. As in the RPE comparative it can be observed that taking a reference frame has a moderate or almost unnoticeable effect on the accuracy except for the $fr2/desk$ and $fr3/office$ datasets, where the ATE metrics are twice larger when not taking reference frames. Given that these datasets were acquired with a slow moving camera, this greater error is likely to be caused by a motion between consecutive frames producing in some parts an optical flow beyond the limits of the pixel accuracy. Fixing the visibility ratio threshold to a high value ($0.8$ or $0.9$) seems to be enough to prevent this problem from happening in sequences with slow camera motion and has no negative effects on sequences with faster motion. A visual qualitative evaluation of the trajectory error is shown in Fig. 4.

In the synthetic datasets the comparison was performed against the RGB-D odometry methods originally evaluated by the authors of the benchmark. In Table 6 we show the ATE for different values of the visibility ratio threshold which have been compared against the lowest ATE of the approaches evaluated in [13], which generally corresponds to the one resulting from using a model-to-frame ICP algorithm originally used in KinectFusion. Surprisingly, though our approach computes the odometry in a frame-to-frame fashion we obtain the best accuracy in most of the sequences with simulated noise and some of the noiseless ones. A qualitative evaluation of the trajectories for all the sequences using different visibility ratio thresholds is shown in Figs. 5 and 6. As occurs with the $fr2/desk$ dataset, quantitative and qualitative results show that using a keyframe switching strategy provide a better accuracy.

## 5.4 Odometry covariance and image filtering

In the proposed approach we initially avoided to apply any filtering to the intensity and inverse depth maps in order to conserve the raw sensor measurements and we dealt with the sensor noise by computing the scaling parameters of photometric and geometric residuals. As shown in the experiments on structurally and texturally rich scenes, the effect of not filtering the noise showed not to be harmful, obtaining accurate motion estimates.

However it must be noted that uncertainty in the odometry estimate is modeled by its covariance matrix which is computed as the inverse of the Hessian at the last iteration of the minimisation problem. The Hessian depends on the intensity and inverse depth gradients computed at the reference frame. This means that if the captured environment is poorly textured and highly planar, *e.g.*, a white wall, the Hessian must be nearly singular and thus the problem is ill conditioned.

We noted nevertheless that in some tests with planar and poorly textured scenes this hypothesis was not verified, due to the noise making the Hessian well conditioned without actually providing useful information. For this reason we noticed that, though the motion estimate would be wrong as well, applying a bilateral filter [45] only as a prior step to the computation of the gradients of the intensity and depth maps, would likely produce ill-posed Hessians when they are meant to arise, and thus help to detect poor odometry estimates.

To evaluate how the noise filtering affects the estimate of the odometry covariance we propose to evaluate the condition number of its inverse, the Hessian, for the case of minimising only on the geometric error. Looking at (17) it can be verified that when viewing a plane perpendicular to the camera optical axis, $\nabla \mathcal{W}_A(\mathbf{p}) = (0,\ 0,\ 0)$ for every pixel and thus the Hessian of the resulting linear system when minimising the residual becomes singular. This occurs because the translation on the image axes and rotation around the optical axis are unobservable, while the rotation around camera axes and translation on the optical axis are still observable. A similar reasoning can be applied with respect to the movements on the axes parallel and

Table 6: Absolute trajectory error (RMSE) in meters in the synthetic RGB-D dataset using different visibility ratio thresholds and comparison with state-of-the-art approaches

| Dataset | Visibility ratio threshold | | | | | Best |
|---|---|---|---|---|---|---|
| | No KF | 0.9 | 0.8 | 0.7 | 0.6 | in [13] |
| office_tr0 | 0.1081 | 0.0921 | 0.0253 | **0.0145** | 0.0193 | 0.0216 |
| office_tr1 | 0.0715 | 0.0384 | 0.0145 | **0.0114** | 0.0147 | 0.3778 |
| office_tr2 | 0.0956 | 0.0327 | 0.0080 | **0.0067** | 0.0086 | 0.0109 |
| office_tr3 | 0.0547 | 0.0550 | 0.0201 | 0.0108 | **0.0084** | 0.0838 |
| livingRoom_lt0 | 0.0982 | 0.0365 | 0.0102 | **0.0061** | 0.4846 | 0.0724 |
| livingRoom_lt1 | 0.0476 | 0.0229 | 0.0153 | 0.0162 | 0.0113 | **0.0054** |
| livingRoom_lt2 | 0.1798 | 0.1683 | 0.0780 | 0.0109 | **0.0108** | 0.0154 |
| livingRoom_lt3 | 0.1131 | 0.0950 | 0.0553 | 0.0365 | 0.0355 | **0.3554** |
| office_tr0_noNoise | 0.0040 | 0.0040 | 0.0040 | 0.0040 | **0.0027** | 0.0029 |
| office_tr1_noNoise | 0.0114 | 0.0114 | 0.0229 | 0.0096 | **0.0032** | 0.0385 |
| office_tr2_noNoise | 0.0293 | 0.0296 | 0.0295 | 0.0309 | 0.0085 | **0.0016** |
| office_tr3_noNoise | 0.0341 | 0.0341 | 0.0427 | 0.0222 | 0.0128 | **0.0021** |
| livingRoom_lt0_noNoise | 0.0431 | 0.0419 | 0.0403 | **0.0099** | 0.0183 | 0.1138 |
| livingRoom_lt1_noNoise | 0.0123 | 0.0123 | 0.0126 | 0.0071 | **0.0023** | **0.0023** |
| livingRoom_lt2_noNoise | 0.0058 | 0.0058 | 0.0058 | 0.0042 | 0.0028 | **0.0015** |
| livingRoom_lt3_noNoise | 0.0166 | **0.0136** | 0.0307 | 0.0319 | 0.0499 | 0.0200 |

perpendicular to a plane with arbitrary orientation viewed by the camera.

The condition number of a Hessian $\mathbf{H}$ can be defined as the quotient between its highest and lowest singular values:

$$\kappa(\mathbf{H}) = \frac{\sigma_{max}(\mathbf{H})}{\sigma_{min}(\mathbf{H})} \tag{34}$$

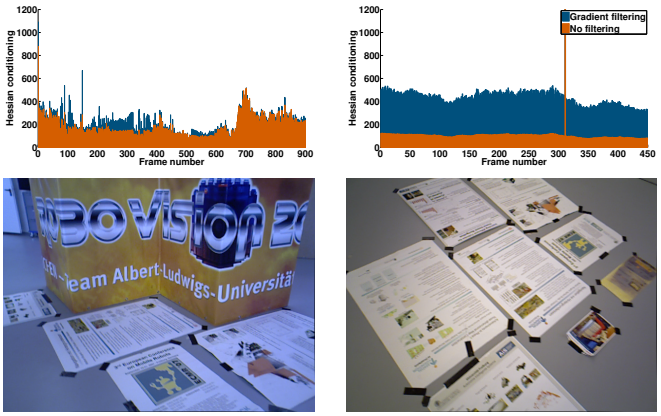
If $\mathbf{H}$ is singular then $\kappa(\mathbf{H}) \rightarrow \infty$. This means that a higher $\kappa(\mathbf{H})$ indicates a more ill-conditioned matrix and thus less accurate estimates in some of the motion degrees of freedom.

The evaluation of how bilateral filtering helps into a better estimation of the conditioning of the Hessian was performed on two sequences of the TUM RGB-D dataset: one showing a rich structure, composed by several planes and one showing a poor structure, with only one plane (Fig. 7). It can be observed that the effect on the Hessian conditioning of filtering the inverse depth map is almost unnoticeable for the scenario with rich structure, but in the case of a planar environment filtering makes more evident the ill-posedness of the Hessian, which otherwise would likely remain unnoticed.

## 5.5 3D reconstruction

Though for volumetric 3D mapping we use the original functions in KinectFusion, a good quality of the reconstructed dense 3D volume depends critically on the drift introduced by the visual odometry algorithm. Thus, we also present qualitative results of our approach showing the reconstruction of some of the tested RGB-D datasets from the TUM, and also two different datasets acquired by us in one laboratory of approximately 90 $m^2$ and the corridor of our department with a length of more than 80 m.

Our acquisitions were carried out with an Asus Xtion Pro RGB-D camera attached to a laptop by an arm-clamp system. The camera was only calibrated with a linear




(a) (b)

Figure 7: Hessian conditioning with and without filtering of the inverse depth gradient map in visual odometry with geometric error minimisation only in (left) structurally rich sequence and (right) structurally poor sequence.

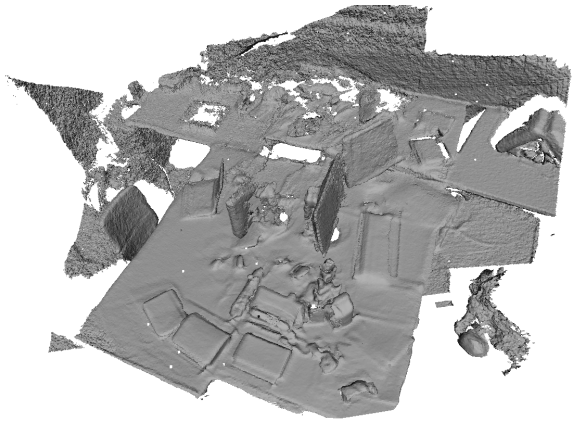


Figure 8: Dense 3D reconstruction of the $fr1/desk$ dataset.

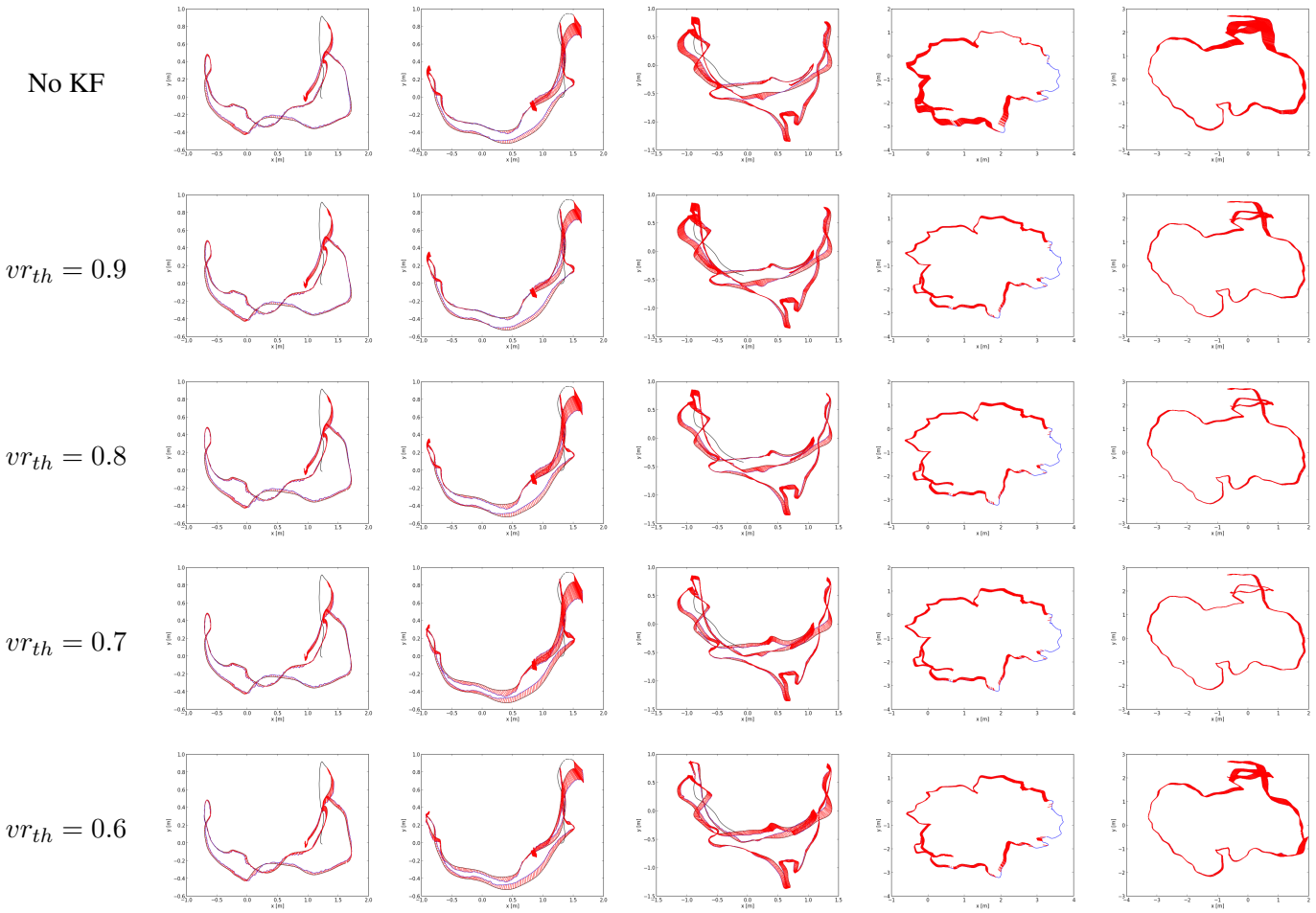
pinhole model without distortion parameters for the RGB sensor. The depth sensor is not calibrated, taking the depth values directly as provided by the sensor; and we use the hard-coded stereo pair calibration for depth and RGB registration. All the reconstructions are obtained just using our modified version of KinectFusion without performing loop closure.

Qualitative results for the datasets $fr1/desk$ , $fr1/room$ and $fr2/desk$ are shown in Figs. 8, 9 and 10. It can be observed the great level of detail in $fr1/desk$, which indicates a low drift in the reconstruction. In the $fr1/room$ there are some zones, like the table at right, where the quality of the 3D reconstruction is poor. This occurs generally when mapping the same area under large camera motion and when revisiting a previously mapped place. In these cases new depth maps integrated into the mapped volume conflict with the stored map generating artefacts. However for zones which are swept during less time, as occurs in the rest of the sequence, the drift during mapping is low and the map reconstruction is more accurate. In

Table 5: Absolute trajectory error (RMSE, median and max) in meters using different visibility ratio thresholds for keyframe switching and comparison with state-of-the-art approaches

| Visibility ratio | fr1/desk | | | fr1/desk2 | | | fr1/room | | | fr2/desk | | | fr3/office | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RMSE | median | max | RMSE | median | max | RMSE | median | max | RMSE | median | max | RMSE | median | max |
| No KF | **0.032** | 0.027 | **0.078** | 0.070 | 0.047 | 0.189 | **0.087** | **0.077** | 0.218 | 0.170 | 0.144 | 0.299 | 0.186 | 0.135 | 0.515 |
| $vr_{th} = 0.9$ | 0.033 | **0.026** | 0.086 | **0.066** | **0.044** | **0.180** | 0.097 | 0.086 | 0.195 | **0.075** | 0.077 | **0.104** | 0.082 | 0.036 | 0.143 |
| $vr_{th} = 0.8$ | 0.033 | 0.027 | 0.087 | 0.081 | 0.052 | 0.230 | 0.088 | 0.085 | **0.162** | 0.077 | 0.076 | 0.111 | **0.064** | **0.021** | **0.112** |
| $vr_{th} = 0.7$ | 0.033 | 0.028 | 0.083 | 0.092 | 0.069 | 0.257 | 0.096 | 0.077 | 0.195 | 0.078 | **0.075** | 0.109 | 0.091 | 0.040 | 0.173 |
| $vr_{th} = 0.6$ | 0.043 | 0.038 | 0.097 | 0.073 | 0.053 | 0.199 | 0.158 | 0.123 | 0.305 | 0.102 | 0.098 | 0.232 | 0.279 | 0.193 | 0.950 |
| VP [33] | - | **0.018** | **0.066** | - | - | - | - | 0.144 | 0.339 | - | **0.093** | **0.116** | - | - | - |
| ICP+RGB-D [47] | - | 0.069 | 0.234 | - | - | - | - | 0.158 | 0.421 | - | 0.119 | 0.362 | - | - | - |
| 6D RGB-D odometry [8] | - | - | - | - | - | - | 0.095 | **0.067** | 0.254 | 0.197 | 0.174 | 0.416 | - | - | - |
| SDF tracking [4] | 0.035 | - | - | 0.062 | - | - | 0.078 | - | - | - | - | - | 0.040 | - | - |
| RGB-D SLAM [10]∗ | 0.023 | - | - | 0.043 | - | - | 0.084 | - | - | 0.057 | - | - | 0.032 | - | |
| MRSMap [42]∗ | 0.043 | - | - | 0.049 | - | - | 0.069 | - | - | 0.052 | - | - | 0.042 | - | |
| RGB+D+KF+Opt [21]∗ | **0.021** | - | - | **0.046** | - | - | **0.053** | - | - | **0.017** | - | - | 0.035 | - | - |
| RGb-D SLAM Vol. Fusion[48]∗ | 0.037 | 0.031 | 0.078 | 0.071 | - | - | 0.075 | 0.068 | 0.231 | 0.034 | 0.028 | 0.079 | **0.030** | - | - |

∗ with loop closure and pose-graph optimisation




(a) fr1/desk (b) fr1/desk2 (c) fr1/room (d) fr2/desk (e) fr3/office

Figure 4: Trajectories on real TUM datasets. Estimated trajectory is shown in blue, ground truth is in black. Error between visual estimate and ground truth is shown in red.

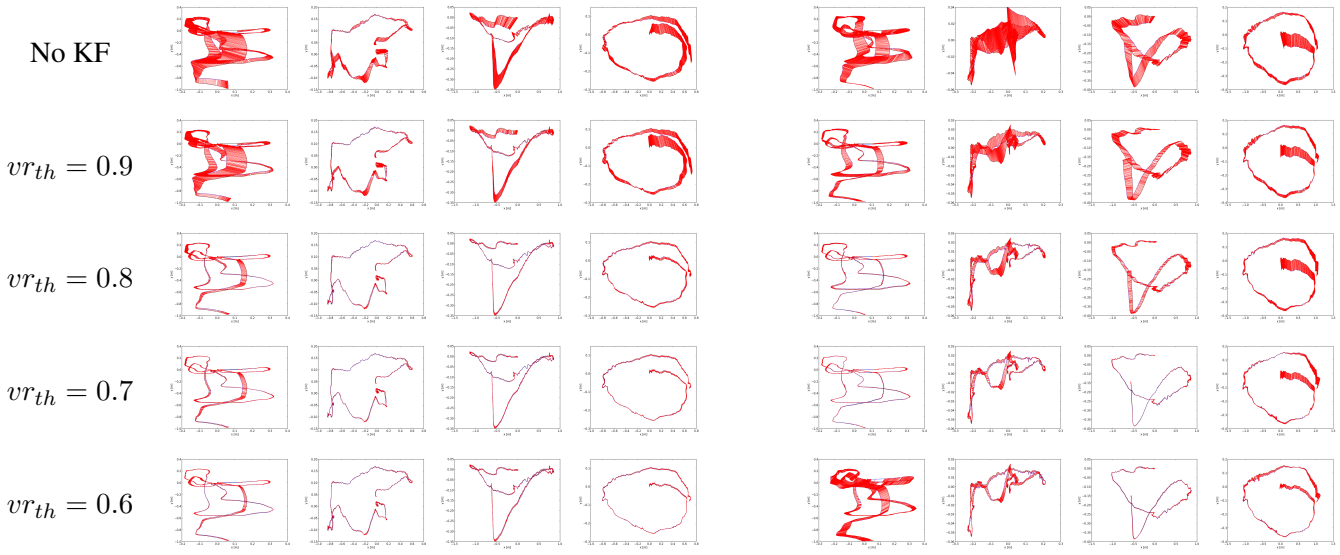



Figure 5: Trajectories on office (a)-(d) and living room (e)-(h) synthetic datasets with simulated noise. Estimated trajectory is shown in blue, ground truth is in black. Error between visual estimate and ground truth is shown in red.

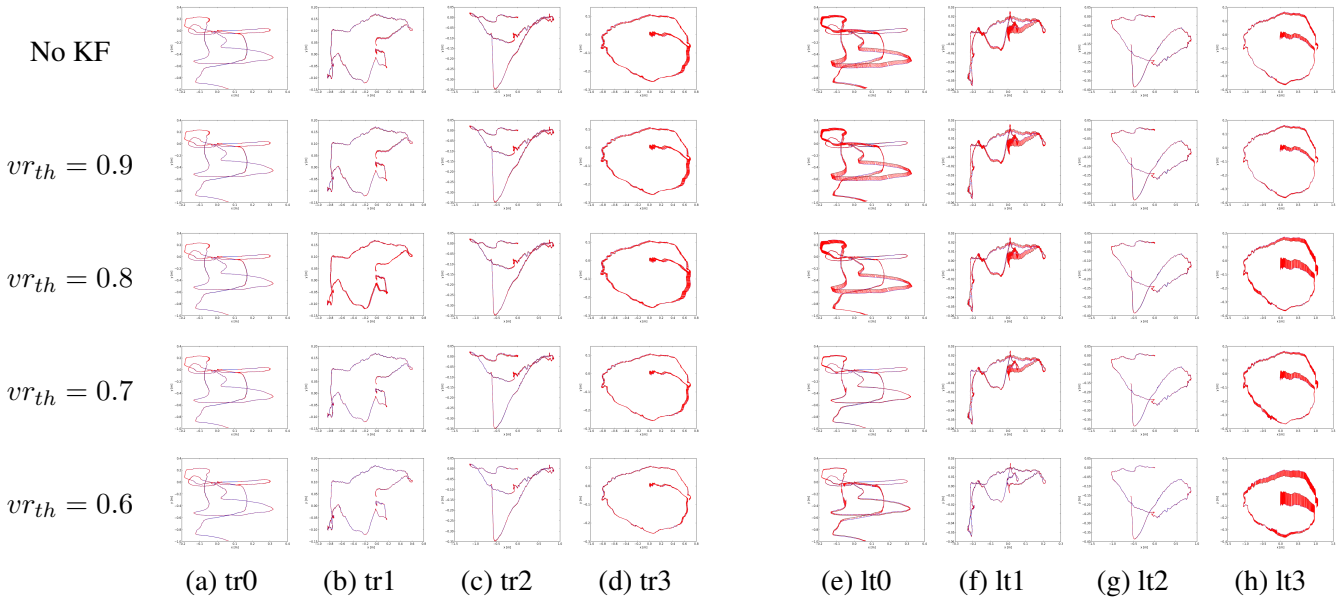



Figure 6: Trajectories on office (a)-(d) and living room (e)-(h) synthetic datasets without noise. Estimated trajectory is shown in blue, ground truth is in black. Error between visual estimate and ground truth is shown in red.

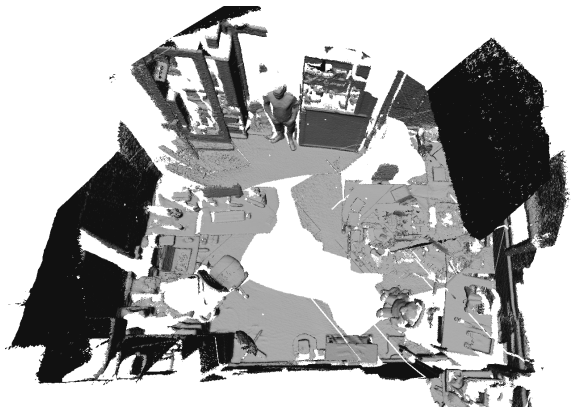


Figure 9: Dense 3D reconstruction of the $fr1/room$ dataset. Note how the shape of the room is accurately captured. Black part on the right top corner of the $fr1/room$ map corresponds to the ceiling reconstruction viewed from outside the volume.

the $fr2/desk$ the same zone is being constantly mapped moving slowly the camera in a loop around the desk. The final reconstruction shows a high precission without having applied loop closures nor any type of map correction.

Results for our laboratory and the corridor sequences are shown in the figure 11 and 12 respectively. The accuracy of the reconstruction can be assessed from the comparison to RGB images from similar points of view. Note also that given that our method is frame-to-frame, the drift both in the laboratory, reflected in the mismatch in the right wall, and in the corridor, reflected in the slight curvature of its side view, are relatively low.

## 6 Conclusions

In this paper we have presented a new visual odometry system on GPU based on the alignment between consecutive frames by minimisation both on the photometric and geometric error. Our system is implemented as an extension of the KinectFusion implementation Kinfu Large Scale in PCL, where the original ICP algorithm for frame alignment and visual odometry computation has been completely substituted by our method. The main contribution of our proposal is using the inverse depth instead of the depth to parametrise the geometric error, as well as allowing to switch between different robust estimators, residuals' scale estimators or geometric error parametrisation for comparative purposes. Our method shows its competitiveness with other state-of-the-art methods outperforming them in the majority of the tested datasets in terms of Relative Pose Error (RPE) and showing low Absolute Trajectory Error (ATE) in spite of not performing loop closure. With the introduction of some changes to increase the computational performance our system is able to reach a performance above 30 Hz

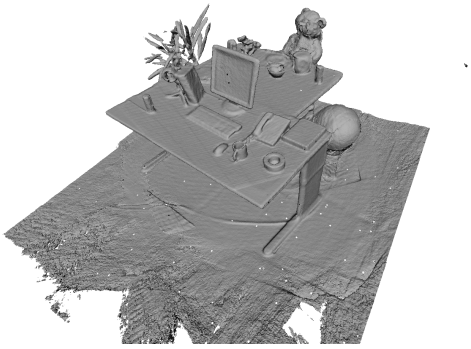


Figure 10: Dense 3D reconstruction of the $fr2/desk$ dataset. Note the high accuracy of the final reconstruction without having performed loop closure.

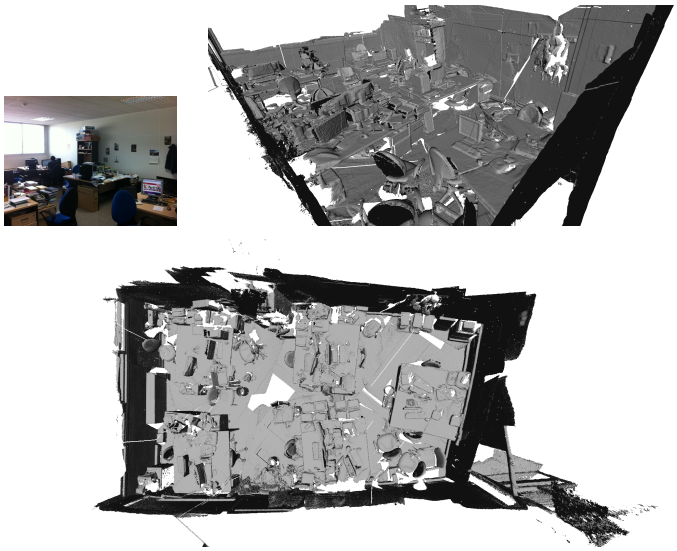


Figure 11: (Top-left) RGB image of the laboratory, (Top-right) KinectFusion 3D reconstruction using our method for visual odometry and (bottom) plant view of the complete 3D mesh.

with the GPU device nVidia GeForce 660GTX used in the experiments, without hindering the accuracy of the method. Also we show that extensions on the method such us taking reference frames for odometry estimation and performing bilateral filtering on the gradients of the images can first improve the accuracy when the camera moves slowly, and secondly allow for a better detection of bad odometry estimates.

## 7 Acknowledgements

This work was supported by Ministerio de Economía y Competitividad and European Union (project DPI2012-31781) and Ministerio de Educación, Cultura y Deporte (scholarship FPU AP2012-5507)

Figure 12: (Top-left) RGB image of the corridor, (Top-right) KinectFusion 3D reconstruction with our visual odometry method, (middle) side view and (bottom) plant view of the complete 3D mesh. Note the challenging of the sequence due to the poor texture of the corridor and light reflexes.

## References


[1] C. Audras, A. Comport, M. Meilland, and P. Rives. Real-time dense rgb-d localisation and mapping. *Aust. Conf. on Robotics and Automation (ACRA)*, 2011.

[2] A. E. Beaton and J. W. Tukey. The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics*, 16(2):147–185, 1974.

[3] E. Bondarev, F. Heredia, R. Favier, L. Ma, and P. H. N. de With. On photo-realistic 3d reconstruction of large-scale and arbitrary-shaped environments. In *IEEE Consumer Communications and Networking Conf. (CCNC)*, pages 621–624, 2013.

[4] E. Bylow, J. Sturm, C. Kerl, F. Kahl, and D. Cremers. Real-time camera tracking and 3d reconstruction using signed distance functions. In *Robotics: Science and Systems (RSS)*, 2013.

[5] J. Civera, A. J. Davison, and J. M. M. Montiel. Inverse depth parametrization for monocular slam. *IEEE Trans. on Robotics*, 24(5):932–945, 2008.

[6] D. Damen, A. P. Gee, W. W. Mayol-Cuevas, and A. Calway. Egocentric real-time workspace monitoring using an rgb-d camera. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1029–1036, 2012.

[7] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *International Conference on Computer Vision*, volume 2, pages 1403–1410, 2003.

[8] H. Dong, N. Figueroa, and A. El Saddik. Towards consistent reconstructions of indoor spaces based on 6d rgb-d odometry and kinectfusion. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1796–1803, 2014.

[9] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. An evaluation of the RGB-D SLAM system. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[10] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard. 3D mapping with an RGB-D camera. *IEEE Transactions on Robotics*, 30(1):177–187, 2014.

[11] J. Engel, J. Sturm, and D. Cremers. Semi-dense visual odometry for a monocular camera. In *IEEE Int. Conf. on Computer Vision (ICCV)*, 2013.

[12] D. Gutiérrez-Gómez, W. Mayol-Cuevas, and J. J. Guerrero. Inverse depth for accurate photometric and geometric error minimisation in rgb-d dense visual odometry. In *Proc. IEEE/RSJ Int. Conf. on Robotics and Automation (ICRA)*, 2015.

[13] A. Handa, T. Whelan, J. McDonald, and A. J. Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *International Confernce on Robotics and Automation*, 2014.

[14] 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 *Int. Symp. on Experimental Robotics (ISER)*, 2010.

[15] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using kinect-style depth cameras for dense 3D modeling of indoor environments. *Int. J. of Robotics Research (IJRR)*, 31(5):647–663, 2012.

[16] E. Herbst, X. Ren, and D. Fox. Rgb-d flow: Dense 3-d motion estimation using color and depth. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2276–2282, 2013.

[17] P. W. Holland and R. E. Welsch. Robust Regression Using Iteratively Reweighted Least-Squares. *Communications in Statistics: Theory and Methods*, A6:813–827, 1977.

[18] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1–3):185 – 203, 1981.

[19] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Fox, and N. Roy. Visual odometry and mapping for autonomous flight using an rgb-d camera. In *In Proc. of the Int. Symp. of Robotics Research (ISRR)*, 2011.

[20] P. J. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101, 1964.

[21] C. Kerl, J. Sturm, and D. Cremers. Dense visual slam for rgb-d cameras. In *IEEE/RSJ Conf. on Intelligent Robots and Systems (IROS)*, pages 2100–2106, 2013.

[22] C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for rgb-d cameras. In *Int. Conf. on Robotics and Automation (ICRA)*, pages 3748–3754, 2013.

[23] K. Khoshelham and S. O. Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012.

[24] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. IEEE and ACM Int. Symp. on Mixed and Augmented Reality (ISMAR)*, 2007.

[25] S. Klose, P. Heise, and A. Knoll. Efficient Compositional Approaches for Real-Time Robust Direct Visual Odometry from RGB-D Data. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[26] K. Konolige and P. Mihelich. Technical description of kinect calibration. `http://wiki.ros.org/kinect_calibration/technical`, 2010, [Online; accessed 25-September-2014].

[27] K. L. Lange, R. J. A. Little, and J. M. G. Taylor. Robust Statistical Modeling Using the t Distribution. *J. of the American Statistical Association*, 84(408):881–896, 1989.

[28] W. L. D. Lui, T. J. J. Tang, T. Drummond, and W. H. Li. Robust egomotion estimation using ICP in inverse depth coordinates. In *IEEE Int. Conf. on Robotics and Automation, ICRA*, pages 1671–1678, 2012.

[29] M. Martinez and R. Stiefelhagen. Kinect unleashed: Getting control over high resolution depth maps. In *IAPR International Conference on Machine Vision Applications, (MVA)*, pages 247–250, 2013.

[30] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid. Rslam: A system for large-scale mapping in constant-time using stereo. *International Journal of Computer Vision*, 94(2):198–214, 2010.

[31] M. Meilland and A. Comport. On unifying key-frame and voxel-based dense visual SLAM at large scales. In *IEEE/RSJ Int. Conf.on Intelligent Robots and Systems (IROS)*, 2013.

[32] M. Meilland, A. Comport, and P. Rives. Dense visual mapping of large scale environments for real-time localisation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[33] M. Meilland and A. I. Comport. Super-resolution 3d tracking and mapping. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 5717–5723, 2013.

[34] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real-time localization and 3d reconstruction. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[35] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Int. Symp. on Mixed and Augmented Reality (ISMAR)*, pages 127–136, 2011.

[36] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *Proc. of Int. Conf. on Computer Vision (ICCV)*, pages 2320–2327, 2011.

[37] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23:3–20, 2006.

[38] L. M. Paz, P. Piniés, J. D. Tardós, and J. Neira. Large scale 6 dof slam with stereo-in-hand. *IEEE Transactions on Robotics*, 24(5):946–957, October 2008.

[39] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[40] J. Solà, T. Vidal-Calleja, J. Civera, and J. M. Montiel. Impact of landmark parametrization on monocular ekf-slam with points and lines. *Int. J. on Computer Vision*, 97(3):339–368, 2012.

[41] F. Steinbrücker, J. Sturm, and D. Cremers. Real-time visual odometry from dense rgb-d images. In *ICCV Workshops*, pages 719–722, 2011.

[42] J. Stuckler and S. Behnke. Integrating depth and color cues for dense multi-resolution scene mapping using rgb-d cameras. In *IEEE Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 162–167, 2012.

[43] J. Stückler and S. Behnke. Multi-resolution surfel maps for efficient dense 3d modeling and tracking. *J. Visual Communication and Image Representation*, 25(1):137–147, 2014.

[44] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the Int. Conf. on Intelligent Robot Systems (IROS)*, 2012.

[45] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Int. Conf. on Computer Vision (ICCV)*, pages 839–846, 1998.

[46] T. Tykkala, C. Audras, and A. I. Comport. Direct iterative closest point for real-time visual odometry. In *IEEE Int. Conf. on Computer Vision Workshops (ICCV)*, 2011.

[47] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. McDonald. Robust real-time visual odometry for dense rgb-d mapping. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013.

[48] T. Whelan, M. Kaess, R. Finman, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald. Real-time large scale dense rgb-d slam with volumetric fusion. *The International Journal of Robotics Research (IJRR)*, 2014.

[49] T. Whelan, M. Kaess, J. J. Leonard, and J. B. McDonald. Deformation-based loop closure for large scale dense RGB-D SLAM. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[50] Z. Zhang. Parameter estimation techniques: a tutorial with application to conic fitting. *Image and Vision Computing*, 15(1):59 –76, 1997.