Point-to-hyperplane ICP: fusing different metric measurements for pose estimation

Fernando I Ireta Munoz, Andrew I. Comport

To appear in *Advanced Robotics*
Vol. 00, No. 00, August 2017, 1–17

FULL PAPER

Point-to-hyperplane ICP: Fusing Different Metric Measurements for Pose Estimation

Fernando I. Ireta Muñoz[1] and Andrew I. Comport[1]

[1]*Université Côte d'Azur, CNRS, I3S, 2000 Route des Lucioles, Bât. Euclide B, Sophia Antipolis, France*

(*v2.0 released January 2018*)

The objective of this article is to provide a generalized framework of a novel method that investigates the problem of combining and fusing different types of measurements for pose estimation. The proposed method allows to jointly minimize the different metric errors as a single measurement vector in n-dimensions without requiring a scaling factor to tune their importance. This paper is an extended version of previous works that introduced the Point-to-hyperplane ICP approach. In this approach an increased convergence domain and a faster alignment was demonstrated by considering a 4-dimensional measurement vector (3D Euclidean points + Intensity). The method has the advantages of the classic Point-to-plane ICP method, but extends this to higher dimensions. For demonstration purposes, this paper will focus on a RGB-D sensor that provides color and depth measurements simultaneously and an optimal error in higher dimensions will be minimized from this. Results on both, simulated and real environments will be provided and the performance of the proposed method will be carried on real-time visual SLAM.

Keywords: Point-to-hyperplane, Visual Odometry, RGB-D Pose estimation, Visual SLAM

1. Introduction

One of the most common problems in view registration is estimating the pose that relates sets of measurements obtained by a moving sensor (or sensors). This problem has been widely studied by the computer vision and robotics communities and it is specially applied for 3D reconstruction, visual odometry and autonomous navigation tasks.

Depending on the type of sensor, different types of measurements of the environment can be registered through pose estimation. Classically, when more than one type of sensor is employed for pose estimation, the alignment between the extended measurements has been achieved by minimizing each sensor's error separately or by jointly optimizing over each type of measurement in a so-called *hybrid*-manner.

Nowadays, the availability of RGB-D sensors such as the Microsoft Kinect V1, V2 or Asus Xtion have provided the possibility to acquire color and depth information simultaneously at a considerably high framerate, which has been useful for real-time pose estimation. The metric information obtained by RGB-D sensors has been individually studied in the literature. One case is by using depth images, where *geometric*-based methods, such as the well known Iterative Closest Point (ICP) [1] and its variants, have demonstrated the ability to obtain robust alignments when enough geometric information is available and they can obtain fast alignment if the datasets are closely overlapping. Particular variants such as the Point-to-plane ICP strategy [2] and the Generalized-ICP [3] have demonstrated to be the most effective and robust methods when combined with robust estimations approaches such as the M-estimators [4]. On the other hand, color images have

*Corresponding author. Email: ireta@i3s.unice.fr, Andrew.Comport@cnrs.fr

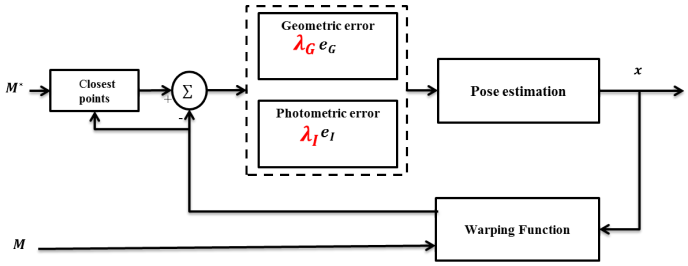


Figure 1. Classic hybrid approach diagram. The geometric and photometric error ($\mathbf{e}_G$ and $\mathbf{e}_I$, respectively) are jointly minimized. A tuning parameter $\boldsymbol{\lambda} = (\lambda_G, \lambda_I)$ weights the contribution of each measurement during the minimization process. The metric measurements are represented in a $i-th$ single vector $\mathbf{M}_i = \left[\mathbf{P}_i^\top \ \mathbf{I}_i\right]^\top \in \mathbb{R}^4$, which contains the 3D Euclidean points $\mathbf{P}$ and its associated intensities $\mathbf{I}$.

been used for pose estimation processes by performing *photometric*-based minimization. Strategies as direct approaches based on view synthesis [5] or feature-based strategies such as: SIFT (Scale-Invariant Feature Transform) [6], SURF (Speeded Up Robust Features) [7], BRIEF (Binary Robust Independent Elementary Features) [8] or ORB (Oriented FAST and Rotated BRIEF) [9] have been widely used. In [6], the feature descriptors are obtaining by computing a histogram of local oriented gradients around a keypoint, however, it can be computationally expensive due to the high dimensionality of its descriptors. An improved version has been proposed in [7], which relies on local gradient histograms but the matching is accelerated by using integral images. However, methods [6, 7] are highly discriminant and features of the same object under different illumination cannot be properly detected. A similar performance is shown in [8], but the feature matches are improved by training a set of classification trees and by using the Hamming distance as the metric for matching instead of Euclidean distance. Finally, an efficient alternative has been presented in [9], where an efficient computation of BRIEF features is performed.

It can be noted at this point that feature-based approaches first extract geometric information from the image before performing estimation on a geometric error. Therefore, for purposes of this article, feature-based approaches won't be detailed here since they can be considered as a sub-part of direct approaches.

The geometric-based and photometric-based approaches share much similarity and subsequently, the common pose estimation framework of both strategies involves the following non-linear Iteratively Re-weighted Least Squares (IRLS) process:

(1) Acquisition of the measurements at different times.
(2) Transform (warp) the measurements using the last pose estimate.
(3) Find the closest points between the sets of measurements.
(4) Minimize the robust weighted error functions.
(5) Estimate a new incremental update on the pose.
(6) Perform all the steps from 2 until convergence.

Recently, several strategies have combined *geometric* and *photometric*-based methods together to obtain the main benefits of each via a so-called *hybrid* method (The main recent surveys are cited in [10]). The advantages of hybrid approaches in combining different measurements include increased efficiency, accuracy and robustness for pose estimation processes. However, the contribution of each measurement during the minimization process should be weighted by a tuning parameter λ, which scales the relative importance of each measurement (Figure 1). Various prominent hybrid methods proposed in the literature are those that simultaneously minimize the geometric and photometric error functions in real-time such as [11–15]. The aforementioned methods differ in how the tuning parameter is estimated and how the closest points are found.

The cited hybrid strategies in this paper do not necessarily consider the color and depth simultaneously when computing the closest points. All the methods perform the closest point searching separately for both color and depth except for [14], which estimates the closest points using a kd-tree (k-dimensional) in a 4-dimensional space (3D Euclidean points + intensity). Finding the closest points by considering the fused information increases the accuracy of finding the true nearest neighbours, however, this approach requires an efficient search in a higher dimensional space. Depending on how the closest points are found, this step can potentially be the most computationally expensive part of the pose estimation pipeline.

The choice of λ has a huge influence while estimating the pose. If the parameter is well determined, then it can speed-up the alignment and increase the convergence rate. Depending on how λ is chosen, a variety of hybrid strategies have been categorized into *adaptive* or *non-adaptive* methods in [16]. Basically, adaptive methods are those that determine the tuning parameter at each iteration of the minimization process and the non-adaptive methods estimate λ only once and its value is used for all the following iterations. For purposes of this article, methods that perform real-time tasks for pose estimation are selected for comparison.

Adaptive methods such as [12] and [15] estimate the tuning parameter by obtaining the ratio between the Median Absolute Deviation (MAD) of the photometric and geometric error functions. In [11] the uncertainty between the metric measurements is compensated by computing the covariance matrix between both metric errors and in [14] the tuning parameter is obtained by a sigmoid function which increases the importance of the photometric error over the geometric error or viceversa. The strategy proposed in [13] is classified as the non-adaptive since λ is chosen experimentally.

The aim of this article is to provide an extended framework of a previous work on fusing different metric measurements via the Point-to-hyperplane ICP approach [17]. The invariance to any tuning parameter will be proven mathematically, which will demonstrate that the Point-to-hyperplane ICP method is invariant to λ in hybrid pose estimation processes if the normals are estimated in higher dimensions. Particularly, here the method is applied for RGB-D pose estimation in a 4D and 6D space by fusing both geometric and photometric techniques based on Point-to-plane ICP and direct methods, respectively. With respect to previous work, this paper addresses the issue of computing the 4D normal when geometric or photometric information together are not available. Various real RGB-D sequences that allow to better compare texture vs structure will be compared for the proposed method alongside hybrid strategies that estimate a scale factor.

This paper is organized as follows: In Section 2 an overview of the hybrid method briefly explains how the RGB-D pose estimation can be performed by jointly minimizing over the color and depth by using a direct method for the color and Point-to-plane ICP for the geometric 3D points. In this section it will be shown that the tuning parameter λ has a huge influence on these methods. In Section 3 the Point-to-hyperplane ICP method will be introduced and the invariance to the scale parameter will be demonstrated by minimizing the error as a single vector for n-dimensions. Finally, extended results for both, real and synthetic environments, will be shown.

2. Hybrid RGB-D pose estimation

Hybrid approaches have been useful when color or depth alone are not significant enough for obtaining a correct alignment between RGB-D frames, that have been acquired at different times and that are not in correspondence. A IRLS pose estimation process can be employed to minimize the geometric or photometric error functions separately, but hybrid methods estimate the unknown pose by iteratively minimizing the non-linear error functions simultaneously. Hybrid methods can converge faster than using geometric or photometric approaches individually, and attempt to retain the main benefits of each by weighting their respective contribution.

2.1. *Joint error for pose estimation*

The generated errors between two sets of extended measurements (color + depth) can be jointly minimized to estimate the pose since the color and depth pose estimation pipeline shares too much similarity. So-called hybrid methods have been introduced to minimize both error functions simultaneously, where a 3D Euclidean point $\mathbf{P}_i \in \mathbb{R}^3$ is associated with an unique intensity $\mathbf{I}_i$ by weighting each contribution with an uncertainty factor $\boldsymbol{\lambda}$. Consider here two augmented point clouds obtained at different times. Let $\mathbf{M}^*$ be the reference point cloud and $\mathbf{M}$ be the current point cloud measurements. The hybrid error function for the $i-th$ joint measurement vector, $\mathbf{e}_{G_i}$ and $\mathbf{e}_{I_i}$ (geometric and photometric error, respectively) can be represented as:

$$\mathbf{e}_{H_i} = \boldsymbol{\lambda} \begin{bmatrix} \mathbf{e}_{G_i} \\ \mathbf{e}_{I_i} \end{bmatrix} = \boldsymbol{\lambda} \begin{bmatrix} \mathbf{P}_i^* - w(\mathbf{P}_i, \mathbf{x}) \\ \boldsymbol{I}_i^* - w(\boldsymbol{I}_i, \mathbf{x}) \end{bmatrix} \in \mathbb{R}^6 \tag{1}$$

where $\boldsymbol{\lambda} = diag(\lambda_X, \lambda_Y, \lambda_Z, \lambda_I) = \begin{bmatrix} \boldsymbol{\lambda}_G & 0 \\ 0 & \lambda_I \end{bmatrix}$ are the tuning parameters which weigh the contribution of each error function. For 3D Euclidean points, $\boldsymbol{\lambda}_G = diag(\lambda_X, \lambda_Y, \lambda_Z)$. The alignment between the measurement vector $\mathbf{M}^* = [\mathbf{P}^* \ \mathbf{I}^*]^\top$ and $\mathbf{M} = [\mathbf{P} \ \mathbf{I}]^\top$ is found by iteratively minimizing the error function (1). This involves transforming the current dataset $\mathbf{M}$ with the estimated pose $\mathbf{x}$ with the transformation function represented here as: $w(\cdot)$. A 3D Euclidean point can be determined by using the depth-back-projection function as: $\mathbf{P}_i = \mathbf{K}^{-1}\overline{\mathbf{p}_i} \ Z_i = \begin{bmatrix} X_i \ Y_i \ Z_i \end{bmatrix}^\top \in \mathbb{R}^3$, where $Z_i \in \mathbb{R}^+$ is the depth measurement for each pixel coordinate $\overline{\mathbf{p}_i} = [u_i \ v_i \ 1]^\top \in \mathbb{R}^3$ of the depth image and $\mathbf{K} \in \mathbb{R}^{3\times3}$ is the intrinsic camera calibration matrix as:

$$\mathbf{K} = \begin{bmatrix} fw & fs_\theta & c_x \\ 0 & fh & c_y \\ 0 & 0 & 1 \end{bmatrix} \mathbb{R}^{3\times3} \tag{2}$$

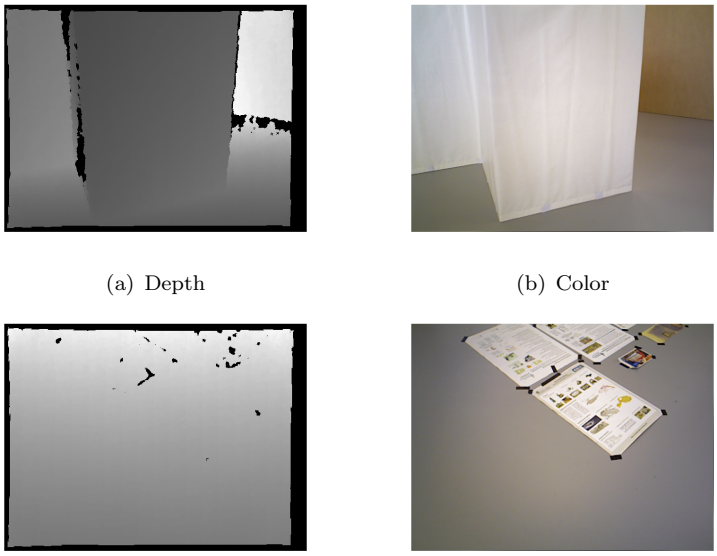
where f is the focal distance, s_θ is the skew angle of a pixel (which is usually set to 0), w and h is the width and height of the image, respectively and c_x, c_y are the coordinates of the center of the image.

The 6DOF (Degrees of freedom) pose parameter $\mathbf{x}$ can be decomposed into rotational and translational components and it will be defined here via the homogeneous transformation matrix $\mathbf{T}(\mathbf{x}) = \begin{bmatrix} \mathbf{R}(\mathbf{x}) & \mathbf{t}(\mathbf{x}) \\ \mathbf{0}_3 & 1 \end{bmatrix} \in \mathbb{SE}(3)$ which is parametrized by the linear $\boldsymbol{v} \in \mathbb{R}^3$ and angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$, respectively. The relationship between the velocity twist and the homogeneous pose matrix is given by the exponential map as $\mathbf{T}(\mathbf{x}) = e^{[\mathbf{x}]\wedge}$, with the operator $[\cdot]_\wedge$ defined as: $[\mathbf{x}]_\wedge = \begin{bmatrix} [\boldsymbol{\omega}]_\times & \boldsymbol{v} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in SE(3)$ where $[\cdot]_\times$ is the skew symmetric operator.

Here the non-linear error defined in (1) is minimized iteratively using a Gauss-Newton approach to compute the unknown 6DOF pose parameters with increments given by:

$$\mathbf{x} = -(\mathbf{J}^\top \mathbf{W} \mathbf{J})^{-1} \mathbf{J}^\top \mathbf{W} \begin{bmatrix} \boldsymbol{\lambda}_G \mathbf{e}_G \\ \lambda_I \mathbf{e}_I \end{bmatrix} \in \mathbb{R}^6 \tag{3}$$

where $\mathbf{J} = \begin{bmatrix} \mathbf{J}_G^\top \ \mathbf{J}_I^\top \end{bmatrix}^\top$ is the stacked Jacobian matrix obtained by deriving the stacked error functions, and the weight matrix $\mathbf{W} = diag(\rho_1, \rho_2, \cdots, \rho_n)$ contains the stacked weights associated with each set of coordinates obtained by M-estimation [4]. Often, robust M-estimation is performed separately for each different measurement type.



(a) Depth (b) Color

(c) Depth (d) Color

Figure 2. Examples of RGB-D images: Texture vs Color. (a) and (b) are images taken from an environment where the geometric measurements are more significant than photometric measurements while (c) and (d) were taken from a scenario where the texture is more significant than geometry. The depth image is visualized as an intensity image, where black indicates a non valid depth measurement.

The cited hybrid strategies [11–15] perform the Point-to-Plane ICP algorithm [2] and a direct image-based method [5] whilst minimizing the error simultaneously. Generally, these approaches minimize an error function [1] similar to:

$$\mathbf{e}_{H_i} = \begin{pmatrix} \lambda_G \left(\mathbf{N}_i^{*\top} \left(\mathbf{P}_i^m - \mathbf{P}_i^w \right) \right) \\ \lambda_I \left(\boldsymbol{I}_i^m - \boldsymbol{I}_i^w \right) \end{pmatrix} \in \mathbb{R}^6 \tag{4}$$

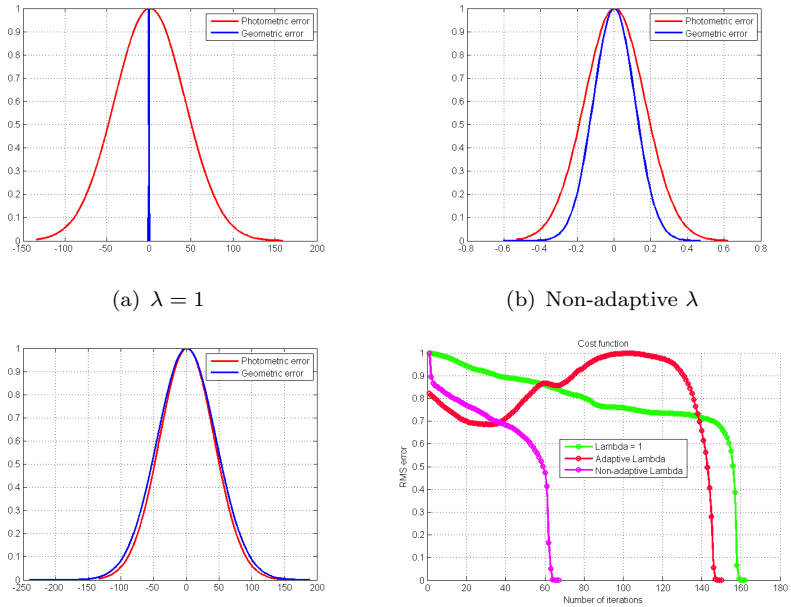
where $\mathbf{P}_i^w = \Pi_3 \widehat{\mathbf{T}}\mathbf{T}\left(\mathbf{x}\right) \overline{\mathbf{P}_i^*} \in \mathbb{R}^3$ is the warped 3D point and $\mathbf{N}_i^* \in \mathbb{R}^3$ is the surface normal for each 3D point $\overline{\mathbf{P}}_i^* \in \mathbb{R}^4$. In the photometric term, $\boldsymbol{I}_i^w = \boldsymbol{I}\left(\omega(\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}), \overline{\mathbf{P}}_i^*)\right)$ is the warped image through the geometric warping function $\omega(\cdot)$ as:

$$\overline{\mathbf{p}_i}^w = \frac{\mathbf{K}\Pi_3 \widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}) \overline{\mathbf{M}_i}^*}{\mathbf{e}_3^\top \widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}) \overline{\mathbf{M}_i}^*} = \begin{bmatrix} u_i^w \\ v_i^w \\ 1 \end{bmatrix} = \begin{bmatrix} c_x + f_x X_i^w / Z_i^w \\ c_y + f_y X_i^w / Z_i^w \\ 1 \end{bmatrix} \in \mathbb{R}^3 \tag{5}$$

where $\Pi_3 = [\mathbf{1}, \mathbf{0}] \in \mathbb{R}^{3\times4}$ projects the 4×4 pose matrix onto the 3×4 space and $\mathbf{e}_3^\top = [0\ 0\ 1]$ extracts the depth component of a transformed 3D point.

The closest image intensity is found by interpolating the current intensity function at the warped pixel coordinates. Therefore, the corresponding intensities can be estimated as: $\mathbf{I}_i^w(\mathbf{p}_i^*) = \mathbf{I}_i(\mathbf{p}_i^w) \in \mathbb{Z}^+$. The 3D point correspondences and the matched intensities are defined as $\mathbf{P}_i^m$ and $\mathbf{I}_i^m$, respectively. Finally, the pose estimation $\mathbf{T}(\mathbf{x})$ is computed at each iteration and is updated incrementally as: $\widehat{\mathbf{T}} \leftarrow \widehat{\mathbf{T}}\mathbf{T}(\mathbf{x})$ until convergence.

[1] Note that in (4) the parameter $\lambda_G = det(\boldsymbol{\lambda_G})$ is a scalar for the geometric point-to-plane error



(a) $\lambda = 1$ (b) Non-adaptive λ

(c) Adaptive λ (d) Cost functions

Figure 3. Influence of the scale coefficient on the error function residuals (Equation (13)) when (a) λ is not estimated, (b) when the intensities and 3D points are normalized between $[0, 1]$ [18] and (c) when λ is estimated at each iteration [15]. An example of a cost function is shown in (d), which indicates that the number of iterations can be improved when a good choice of λ is made.

2.2. *Uncertainty between depth and color*

Since the geometric and photometric measurements and subsequently their uncertainties are not in the same units or order of magnitude, the contribution in the minimization process of each measurement should be weighted to compensate for the relative uncertainty between the different error functions. Consider as an example the case when the intensity is almost uniform in the scene but the geometric features are not (Figure 2(b) and 2(a)). The pose estimation function should give more importance to the geometric features since the errors generated in the photometric term are not significant enough to constrain all degrees of freedom. On the other hand, the opposite case can be found when rich texture can be registered from flat surfaces (Figure 2(d) and 2(c), the geometric information does not constrain all degrees of freedom for obtaining the alignment. An example of the influence of $\boldsymbol{\lambda} = diag(\lambda_X, \lambda_Y, \lambda_Z, \lambda_I)$ in the minimization process is shown in Figure 3 where a Gaussian distribution has been fitted into the residuals.

As is shown in (1), each intensity is associated with its corresponding 3D Euclidean point through a matrix $\boldsymbol{\lambda}$ that scales the importance of the geometric points w.r.t. the intensities. As mentioned in the introduction, many methods have been proposed to choose this parameter ranging from manual tuning to more complex estimation approaches. Manually fixing $\boldsymbol{\lambda}$ is not optimal nor efficient for real-time applications, and estimating its adequate value can require extra computational cost. Various strategies, which obtain $\boldsymbol{\lambda}$ in different ways, have been cited in [17]. Three efficient real-time possibilities will be considered here and they will be compared in the results section. They include adaptive methods: such as the ratio of the Median Absolute Deviations (MAD) [15] or computing the covariance matrix for each measurement vector as in [11], and non-adaptive methods: using the normalization of the metric measurements to scale the relative error distributions.

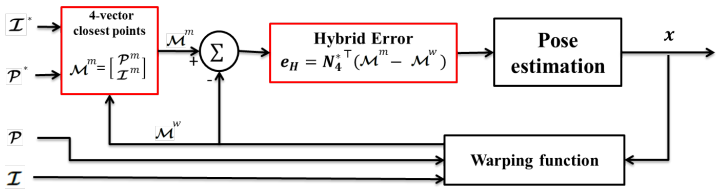


Figure 4. Point-to-hyperplane approach diagram. In the first iteration, the closest points can be estimated by finding in a kd-tree with the 4D vector and linear interpolation in the following iterations. The method is invariant to a tuning parameter λ.

3. Point-to-hyperplane ICP

The aim of this paper is to present a method that performs RGB-D registration by minimizing over color and depth components simultaneously in such a way that it is invariant to the scale between both measurements. The n-D space generated by considering measurement vector of n-Dimensions, has additional degrees of freedom. The proposed method therefore consist in extending the classic Point-to-plane method [2] for 2D and 3D points, to higher dimensions. Therefore, the estimated n-dimensional normal (as well as the 3D normal) will be orthogonal to a surface in n-dimensions which spans both geometry and color. This n-dimensional surface will be referred in this paper as the *hyperplane*.

Based on the Point-to-plane method for 3D points, an error function in higher dimensions can be defined as follows:

$$\mathbf{e}_{H_i} = \lambda \mathbf{N}_i^{*\top} \left(\mathbf{M}_i^* - w(\mathbf{M}_i, \mathbf{x}) \right) \tag{6}$$

where a tuning parameter $\lambda = det(\boldsymbol{\lambda})$ is added to deal with the uncertainty between different measurements and the normal $\mathbf{N}_i^{*\top}$ is perpendicular to the formed hyperplane.

For the purpose of this article, 4 dimensions will be considered (3D Euclidean points + intensity) for the experimentation (See Figure 4). The 4-vector is defined as $\mathbf{M}_i = [\mathbf{P}_i^\top \ \mathbf{I}_i]^\top \in \mathbb{R}^4$. The normals $\mathbf{N}^*$ are computed on the reference 4D measurements vector $\mathbf{M}^*$, which will be referred throughout this paper as the reference dataset. Therefore, the pose vector $\mathbf{x}$ can be estimated by iteratively minimizing the error function that projects the Point-to-point distance onto the normal direction as:

$$\mathbf{x} = -(\mathbf{J}^\top \mathbf{W} \mathbf{J})^{-1} \mathbf{J}^\top \mathbf{W} \mathbf{e}_H \tag{7}$$

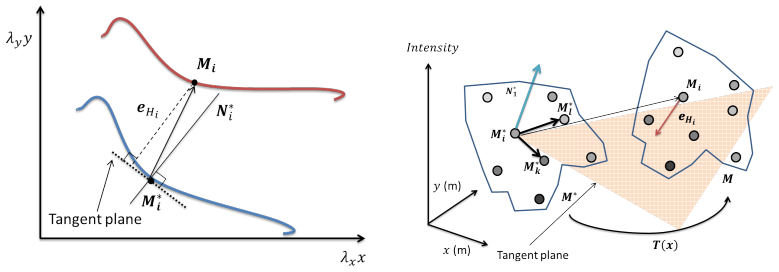
3.1. *Invariance to a tuning parameter $\boldsymbol{\lambda}$*

The invariance to any scale factor $\boldsymbol{\lambda} = diag(\lambda_1, \lambda_2, \cdots, \lambda_n)$ in the Point-to-hyperplane ICP method was experimentally observed in [17] and a demonstration to this invariance is given in [16], where the mathematical proof of the following lemma is given:

Lemma. *The integrated error $\mathbf{e}_H$ in n-dimensions is invariant to the relative scale λ if it is minimized by a Point-to-hyperplane ICP method.*

$$\mathbf{e}_{H_i} = \mathbf{N}_i^{*\top} \left(\mathbf{M}_i^* - w(\mathbf{M}_i, \mathbf{x}) \right) = \lambda \mathbf{N}_i^{*\top} \left(\mathbf{M}_i^* - w(\mathbf{M}_i, \mathbf{x}) \right) \tag{8}$$

The projection of the error onto the normal direction has the effect of canceling out the effect of $\lambda = det(\boldsymbol{\lambda})$ between the geometric and photometric error since the direction of the normal is



(a) 2D hyperplane (b) 3D hyperplane

Figure 5. Principle of the Point-to-hyperplane applied in 2 dimensions Point-to-line and 3 dimensions Point-to-plane. It should be noted that the axis x and y are not in the same scale in (a). A tuning parameter λ is added in order to demonstrate the invariance of the Point-to-hyperplane method in the 2D case. The distance $\mathbf{e}_{H_i}$ is the result of projecting the vector $\overrightarrow{\mathbf{M}_i^*\mathbf{M}_i}$ onto the normal direction $\mathbf{N}_i^*$.

invariant to any tuning parameter. The mathematical proof given in [16], demonstrated that the error function is not influenced by any scale parameter if the n-dimensional normals are normalized. In fact, for this paper it was observed that the invariance to the scale parameter is due to the fact that the elements of $\boldsymbol{\lambda}$ have no influence on the direction of the **estimated normals**. For purposes of simplicity, this invariance will be shown for the 2D case here.

Consider two sets of measurements taken at different times as $\mathbf{M}^* = [\mathbf{X}^* \ \mathbf{Y}^*]^\top$ and $\mathbf{M} = [\mathbf{X} \ \mathbf{Y}]^\top$ where each measurement $(\mathbf{X}, \mathbf{Y})$ has a different scale (e.g. centimeters and millimeters). The measurements are scaled to the same order of magnitude by the tuning parameter $\boldsymbol{\lambda} = diag(\lambda_X, \lambda_Y)$. The equation of the tangent line $ax + by + c = 0$ (where $\langle a, b \rangle$ are the coordinates of the normal) at an $i-th$ reference point can be obtained from the definition of the slope of a line:

$$\frac{y - \lambda_Y Y_i^*}{x - \lambda_X X_i^*} = \frac{\lambda_Y (Y_{i+1}^* - Y_i^*)}{\lambda_X (X_{i+1}^* - X_i^*)} \tag{9}$$

that can be written as: $\lambda_Y(Y_{i+1}^* - Y_i^*)x + \lambda_X(X_i^* - X_{i+1}^*)y + c = 0$, where $c = \lambda_X\lambda_Y(X_{i+1}^* Y_i^* - Y_{i+1}^* X_i^*)$. The normal $\langle a, b \rangle = \left\langle \lambda_Y(Y_{i+1}^* - Y_i^*), \lambda_X(X_i^* - X_{i+1}^*) \right\rangle$ can be represented for simplicity as: $\overrightarrow{\mathbf{N}_i^*} = det\,(\boldsymbol{\lambda}) \cdot \boldsymbol{\lambda}^{-1}\mathbf{V}_i^*$ where $\mathbf{V}_i^* = [Y_{i+1}^* - Y_i^* \quad X_i^* - X_{i+1}^*]^\top$. For simplicity this last will be written as: $\mathbf{V}_i^* = [N_{1_i}^* \ N_{2_i}^*]^\top$

The projection of the point-to-point error $\overrightarrow{\mathbf{M}_i^*\mathbf{M}_i} = \boldsymbol{\lambda}(\mathbf{M}_i - \mathbf{M}_i^*)$ onto the normal direction $\overrightarrow{\mathbf{N}_i^*}$ defines the distance of a point to a line (Figure 5(a)). It is clearly seen that the error function can be computed as: $\mathbf{e}_{H_i} = \overrightarrow{\mathbf{N}_i^*}^\top \boldsymbol{\lambda}\,(\mathbf{M}_i^* - \mathbf{M}_i)$. Replacing $\overrightarrow{\mathbf{N}_i^*}$ into this error function as: $\mathbf{e}_{H_i} = det\,(\boldsymbol{\lambda}) \cdot \boldsymbol{\lambda}^{-1}\mathbf{V}_i^{*\top}\boldsymbol{\lambda}\,(\mathbf{M}_i^* - \mathbf{M}_i)$ allows to rewrite it as $\mathbf{e}_{H_i} = \lambda_X\lambda_Y\left(N_{1_i}^*(X_i - X_i^*) + N_{2_i}^*(Y_i - Y_i^*)\right)$, where is demonstrated that the tuning parameter $\boldsymbol{\lambda}$ has no effect on the minimization process since it has no influence on the direction of the normal and it scales its magnitude only.

The invariance for the 3D case has been demonstrated in [16] and it has been extended to n-dimensions. In this paper, a better presentation of the proof will be given. The normal in three dimensions is obtained by performing the cross product between two hybrid vectors, scaled by $\boldsymbol{\lambda} = diag(\lambda_X, \lambda_Y, \lambda_I)$ (2D geometric points + intensity) as: $\mathbf{N}_i^* = det(\boldsymbol{\lambda}) \cdot \boldsymbol{\lambda}^{-1}\left(\mathbf{V}^{ik} \times \mathbf{V}^{il}\right)$, where $\mathbf{V}^{ik}$ and $\mathbf{V}^{il}$ are defined as the $k-th$ and the $l-th$ closest point to $\mathbf{M}_i^*$ that lies on the reference dataset as $\mathbf{V}^{ik} = \boldsymbol{\lambda}\,(\mathbf{M}_k^* - \mathbf{M}_i^*)$ and $\mathbf{V}^{il} = \boldsymbol{\lambda}\,(\mathbf{M}_l^* - \mathbf{M}_i^*)$, respectively (See Figure 5(b)).

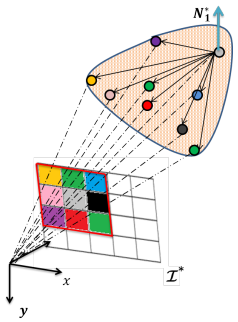


Figure 6. The neighbours of a central pixel (in grey) in a 3×3 window on a image are employed to associate its closest 3D points and estimate the normal to the connecting plane (or hyperplane). Any pixel on the image can be selected as a central pixel to compute the distance with its surrounding neighbours (except for the corners).

Therefore, a 3D normal at the $i-th$ reference point is obtained as follows:

$$\mathbf{N}_i^* = \begin{bmatrix} \lambda_Y\lambda_I\left(V_Y^{ik}V_I^{il} - V_I^{ik}V_Y^{il}\right) \\ \lambda_X\lambda_I\left(V_I^{ik}V_X^{il} - V_X^{ik}V_I^{il}\right) \\ \lambda_X\lambda_Y\left(V_X^{ik}V_Y^{il} - V_Y^{ik}V_X^{il}\right) \end{bmatrix} = \begin{bmatrix} \lambda_Y\lambda_I N_1 \\ \lambda_X\lambda_I N_2 \\ \lambda_X\lambda_Y N_3 \end{bmatrix} \tag{10}$$

The error function generated between two sets of 3D hybrid measurements can be defined then as: $\mathbf{e}_{H_i} = det(\boldsymbol{\lambda})\left(\mathbf{V}^{ik}\times\mathbf{V}^{il}\right)^\top(\mathbf{M}_i^* - \mathbf{M}_i)$.

The estimation of the normal presented in (10) can be easily extended to higher dimensions. The n-dimensional normal is estimated by performing the n-dimensional cross product between the $n-1$ vectors such as:

$$\mathbf{N}_i^* = det(\boldsymbol{\lambda})\cdot\boldsymbol{\lambda}^{-1}\left(\mathbf{V}^1\times\mathbf{V}^2\times\cdots\times\mathbf{V}^{n-1}\right)\in\mathbb{R}^{n\times1} \tag{11}$$

The n-dimensional error function [16] between two sets of measurements can be defined as: $\mathbf{e}_{H_i} = det(\boldsymbol{\lambda})\boldsymbol{\lambda}^{-1}\left(\mathbf{V}^1\times\mathbf{V}^2\times\cdots\times\mathbf{V}^{n-1}\right)^\top\boldsymbol{\lambda}(\mathbf{M}_i^* - \mathbf{M}_i)$, which can be re-written as:

$$\mathbf{e}_{H_i} = det(\boldsymbol{\lambda})\left(\mathbf{V}^1\times\mathbf{V}^2\times\cdots\times\mathbf{V}^{n-1}\right)^\top(\mathbf{M}_i^* - \mathbf{M}_i) \tag{12}$$

where $det(\boldsymbol{\lambda}) = \lambda_1\lambda_2\cdots\lambda_n$.

The aforementioned normals can be computed by performing an n-dimensional cross product but other strategies can be equally used. In the Generalized-ICP strategy, the PCA (Principal Component Analysis) is computed. The eigenvector associated with its lowest eigenvalue is considered as the normal. Recently, an alternative solution has been provided in [3], where the normals are efficiently and accurately computed by performing the Prewitt operator on projected spherical coordinates onto a spherical range image. This approach, however, only applies to the 3D case. For the 4-dimensional case presented in this article, a PCA analysis was performed to estimate the 4D normal, which can be written as: $\mathbf{N}_i^* = \begin{bmatrix}\lambda_Y\lambda_Z\lambda_I N_{1_i}^* & \lambda_X\lambda_Z\lambda_I N_{2_i}^* & \lambda_X\lambda_Y\lambda_I N_{3_i}^* & \lambda_X\lambda_Y\lambda_Z N_{4_i}^*\end{bmatrix}^\top$, where N_1, N_2, N_3 and N_4 are the components of the normal. These components can be estimated by considering the lowest eigenvalues of the nearest 4D points to a central $i-th$ 4D point (Figure 6). Therefore, equation (6) can be rewritten for the 4-dimensional space as follows:

$$\mathbf{e}_{H_i} = det(\boldsymbol{\lambda})\left(N_{X_i}(X_i^* - X_i^w) + N_{Y_i}(Y_i^* - Y_i^w) + N_{Z_i}(Z_i^* - Z_i^w) + N_{I_i}(I_i^* - I_i^w)\right)$$

where $det(\boldsymbol{\lambda}) = \lambda_X\lambda_Y\lambda_Z\lambda_I$.

4. Results

In order to evaluate the Point-to-hyperplane ICP method, some parameters considered for the experiments are established. All the experiments were performed on both, real and synthetic RGB-D grayscale images in MATLAB. Furthermore, visual SLAM in real-time was performed in C++. All the experiments were validated on a PC with Ubuntu 14.04, Intel core i7-4770K and 16GB ram.

A multi-resolution pyramid was used to improve computational efficiency (resolution: 160×120 at the top), where a pose is estimated at the top of the pyramid and the estimated transformation is employed to initializate the transformation in the next level until reaching the base of the pyramid.

The minimization process can be stopped by two criteria: an established maximum number of iterations (200 iterations for the experiments performed in this paper) or if the norm of the pose parameter is less than 1×10^{-6} in rotation and 1×10^{-5} in translation. To reject outliers, the Huber influence function was employed in only one M-estimator (as opossed to [11, 12] where the M-estimation is performed separately for color and depth). The M-estimation allows the use of different minimization functions not necessarily corresponding to normally distributed data.

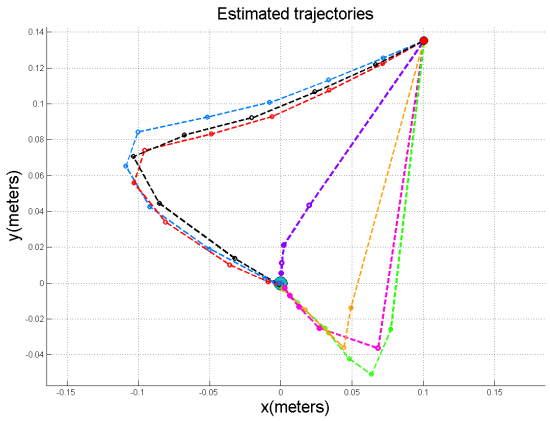
The Point-to-hyperplane ICP method is compared in this paper with variants (which differ in how the uncertainty factors $\boldsymbol{\lambda} = diag(\lambda_X, \lambda_Y, \lambda_Z, \lambda_I)$ are estimated) of the error function proposed in [12]. For this strategy, the classic Point-to-plane [2] approach is employed to minimize the geometric term and a direct method for the photometric term as:

$$\mathbf{e}_{H_i} = \boldsymbol{\lambda} \begin{pmatrix} \left(\widehat{\mathbf{R}}\mathbf{R}(\mathbf{x})\mathbf{N}_i^*\right)^\top \left(\mathbf{P}_i^m - \Pi_3 \widehat{\mathbf{T}}\mathbf{T}\left(\mathbf{x}\right) \overline{\mathbf{P}_i^*}\right) \\ \mathbf{I}_i\left(w(\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}), \mathbf{P}_i^*)\right) - \mathbf{I}_i^*\left(\mathbf{p}_i^*\right) \end{pmatrix} \in \mathbb{R}^4 \tag{13}$$

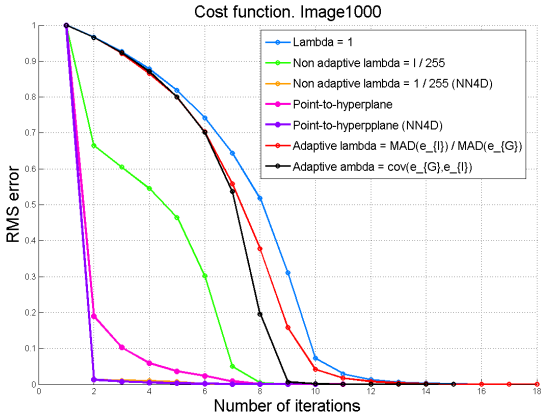
where the first and second row correspond to the geometric and photometric error function, respectively. $\mathbf{P}_i^m \in \mathbb{R}^3$ is the closest 3D Euclidean point in the current cloud, $\hat{\mathbf{R}} \leftarrow \hat{\mathbf{R}}\mathbf{R}(\mathbf{x})$ is the incremental update of rotations, $\mathbf{N}_i^* = [N_{x_i}\ N_{y_i}\ N_{z_i}]^\top$ are the normals of the reference points and $\Pi_3 = [\mathbf{1}, \mathbf{0}] \in \mathbb{R}^{3\times4}$ is the projection matrix. For the purposes of this paper, the photometric term is minimized by using the Second Order Minimization (ESM) method [19].

A strategy to locally find the closest points is needed for computing the normals. The nearest neighbours to a central pixel in the image are considered to find its associated 3D point. At least three 4D points should be considered to estimate the 4D normal. A 3×3 window was considered in this paper for the experiments, which estimate the resultant normal of the 8 nearest neigbours (Figure 6). For the real-time application, however, a 2×2 window is employed to speed up the performance. It should be mentioned here that the computational cost is linear to the size of this window, but the accuracy is increased.

For the comparisons, the performance of the Point-to-hyperplane ICP method is compared with three different strategies that compute a non-adaptive or adaptive $\boldsymbol{\lambda}$: 1) The intensities are normalized $\lambda_I = I_i/255$) (non adaptive), 2) an adaptive $\boldsymbol{\lambda}$ as in [15], where the scale parameter is the ratio between the Median Absolute Deviation (MAD) of the errors $\lambda_G = MAD(\mathbf{e}_I)/MAD(\mathbf{e}_G)$ and 3) the covariance matrix of the metric errors as $\boldsymbol{\lambda} = cov(e_G, e_I)$. For this last strategy, the T-distribution was employed to reject outliers as in [11]. The minimization of the error presented in (13) will also be compared with a $\boldsymbol{\lambda} = eye(1)$ ($\boldsymbol{\lambda}$ is not estimated) in order to demonstrate that the parameter $\boldsymbol{\lambda}$ can improve the hybrid methods if it is well estimated. Alternatively, the estimation of the closest points were also done by searching a kd-tree (Labeled in Figure 7(b) as NN4D). This strategy demonstrated a better performance while aligning the frames when they are not close enough, but increasing the computational cost.



(a) Estimated trajectories

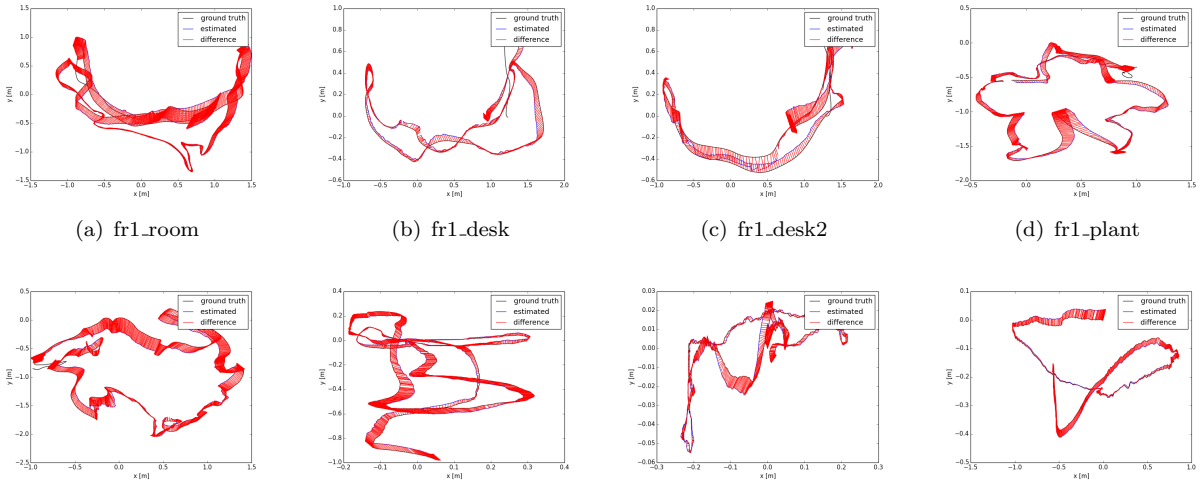


(b) Cost functions

Figure 7. Example of the estimated trajectories between a current and a reference frame in the transformation space. (a) The green and red dot indicates the initial and final pose respectively. The Point-to-hyperplane method improves the other hybrid methods by obtaining more direct trajectories and a less number of iterations (b). A similar performance was observed in the 1000 synthetic frames that were equally tested. The label NN4D indicates that the nearest neighbours were obtained by using a kd-tree in the first iteration.

Table 1. Averages in time and in the number of iterations until convergence for 1000 synthesized Images at Random Poses. The legend NN4D or NN6D indicates that the closest points were estimated in the first iteration only by searching the nearest neighbours in the 4D or 6D kd-tree.

Method	**# Iterations**	**Time (sec)**
Hybrid ($\boldsymbol{\lambda} = ones$)	157.668	2.046
Hybrid + non-adaptive $\lambda_I = I_i/255$	124.419	1.598
Hybrid + non-adaptive $\lambda_I = I_i/255$ (NN4D)	116.609	1.563
Hybrid + adaptive $\lambda_G = MAD(e_I)/MAD(e_G)$ [15]	154.966	2.010
Hybrid + adaptive $\boldsymbol{\lambda} = cov(\mathbf{e}_G, \mathbf{e}_I)$ [11]	155.455	6.079
Point-to-hyperplane (3D points + Intensity)	**48.038**	**0.531**
Point-to-hyperplane (NN4D)	**13.224**	**0.191**
Point-to-hyperplane (3D points + RGB)	96.79	2.1572
Point-to-hyperplane (NN6D)	79.439	1.7978



(a) fr1_room (b) fr1_desk (c) fr1_desk2 (d) fr1_plant

(e) fr1_teddy (f) lvr_traj0 (g) lvr_traj1 (h) lvr_traj2

Figure 8. Examples of the Absolute Trajectory Error evaluation obtained by the Point-to-Hyperplane method. The benchmarck datasets [20] and [21] were used.

4.1. *Simulated environments*

The performance of the aforementioned strategies were compared in a synthetic environment. The motivation for using synthetic data is that the generated images provide a groundtruth for the evaluation since the transformation between the frames is known. For the comparisons, 1000 synthesized RGB-D frames were generated with random poses and Gaussian noise was added.

Averages in time and number of iterations are shown in Table 1 [2]. The time shown, however, does not consider the computation of the normals or construction of the kd-tree in the reference image. Therefore, for the employed RGB-D image in this experiment, the normals obtained 9.33 seconds for a 3×3 window and the construction of the kd-tree took 0.0056 seconds in MATLAB. For comparisons in this experiment, the matching points obtained by the kd-tree were used by considering 4D and 6D points. It was observed that the searching of the closest points by using this strategy reduces the number of iterations and convergence time if it is performed in the first iteration only. The chances to find the true nearest neighbours increases when more dimensions are considered. This is useful when the overlapping area between RGB-D frames is not large enough. However, the searching of the closest points in the kd-tree require extra computational time. Therefore, for purposes of this paper, only 4 dimensions were considered (This balances the computational cost and accuracy while estimating the pose). Figure 7(a) shows an example of the estimated trajectories in the convergence domain by different strategies.

4.2. *Real environments*

The well known *living room* ICL-NUIM RGB-D [20], *freiburg1* and *freiburg3* TUM [21] benchmark datasets were employed to perform visual odometry to compare different hybrid strategies. For this experiment, a frame-to-frame alignment was employed. The estimated poses were used to evaluate the ATE (Absolute Trajectory Error) and RPE (Relative Pose Error). Various examples of the ATE evaluation for the Point-to-hyperplane ICP method are shown in Figure 8, where it can be seen that the Point-to-hyperplane ICP method can obtain close solutions w.r.t. the groundtruth without employing extra strategies for pose refinement or loop closure methods.

[2]In order to better present the performance of the strategies, the best obtained results are displayed in bold in all tables.

Table 2. Averages in Time (miliseconds), number of iterations, Relative Pose Error (RPE) and Absolute Trajectory Error (ATE) for the synthetic dataset [20]. It can be seen that the Point-to-hyperplane method [17] improves hybrid methods that combine the direct approach and the geometric Point-to-plane approach.

Sequence	Method	ATE (m)		RPE translational (m)		RPE rotational (deg)		AVERAGE	
		RMSE	MEAN	RMSE	MEAN	RMSE	MEAN	Time(sec)	#Iterations
lvr/traj0	1	**0.107**	**0.096**	0.003	0.002	0.083	0.067	0.203	15.68
	2	**0.107**	**0.096**	0.003	0.002	0.083	0.067	0.230	16.51
	3	0.128	0.114	**0.002**	**0.001**	**0.042**	**0.026**	**0.179**	**16.07**
	4	0.128	0.114	**0.002**	**0.001**	**0.042**	**0.026**	0.214	17.23
	5	0.230	0.210	0.006	0.004	0.132	0.105	0.541	41.47
	6	0.320	0.300	0.007	0.005	0.179	0.141	1.308	33.25
lvr/traj1	1	0.211	0.190	0.003	0.003	0.082	0.072	0.227	17.73
	2	0.211	0.190	0.003	0.003	0.082	0.072	0.240	17.95
	3	**0.041**	**0.032**	**0.001**	**0.001**	**0.021**	**0.017**	**0.148**	**14.87**
	4	**0.041**	**0.032**	**0.001**	**0.001**	**0.021**	**0.017**	0.181	15.99
	5	0.397	0.330	0.006	0.005	0.128	0.112	0.451	38.89
	6	0.341	0.291	0.007	0.006	0.155	0.136	1.228	34.30
lvr/traj2	1	0.152	0.146	0.003	0.003	0.085	0.074	0.189	16.13
	2	0.152	0.146	0.003	0.003	0.085	0.074	0.220	16.61
	3	**0.039**	**0.036**	**0.001**	**0.001**	**0.024**	**0.019**	**0.172**	**16.06**
	4	**0.039**	**0.036**	**0.001**	**0.001**	**0.024**	**0.019**	0.203	17.07
	5	0.323	0.297	0.007	0.005	0.139	0.118	0.519	41.93
	6	0.398	0.363	0.008	0.007	0.176	0.149	1.205	34.78
lvr/traj3	1	0.445	0.403	0.003	0.003	0.120	0.097	0.300	22.65
	2	0.445	0.403	0.003	0.003	0.119	0.097	0.317	22.95
	3	0.080	0.066	0.001	0.001	0.044	0.027	**0.185**	**15.97**
	4	**0.072**	**0.056**	**0.001**	**0.001**	**0.044**	**0.027**	0.212	16.62
	5	0.526	0.459	0.005	0.004	0.145	0.119	0.584	46.51
	6	0.484	0.436	0.007	0.006	0.179	0.150	1.689	42.07

The numerical results are shown in Table 2, where the methods are listed as follows:

(1) Hybrid + non-adaptive $\lambda_I = I_i/255$
(2) Hybrid + non-adaptive $\lambda_I = I_i/255$ (NN4D*[3])
(3) Point-to-hyperplane
(4) Point-to-hyperplane (NN4D)
(5) Hybrid + adaptive $\lambda_G = MAD(\mathbf{e}_I)/MAD(\mathbf{e}_G)$ [15]
(6) Hybrid + adaptive $\boldsymbol{\lambda} = cov(\mathbf{e}_G, \mathbf{e}_I)$ [11]

From Table 2, it can be seen that the Point-to-hyperplane methods improve other methods while obtaining less computational cost and less number of iterations. It was observed that when the frame-to-frame alignment is employed, the Strategies 2 and 4 obtain about the same results as Strategies 1 and 3, respectively. Therefore, the results for these strategies are shown together in Table 3, where it can be noted that the adaptive λ methods obtained less ATE and RPE error for the *freiburg3* sequences (benchmark structure vs texture), however the computational cost is high w.r.t. Point-to-hyperplane strategies. It can be noted that the Point-to-hyperplane ICP method obtained more robust results in challenging 360 degree scenarios.

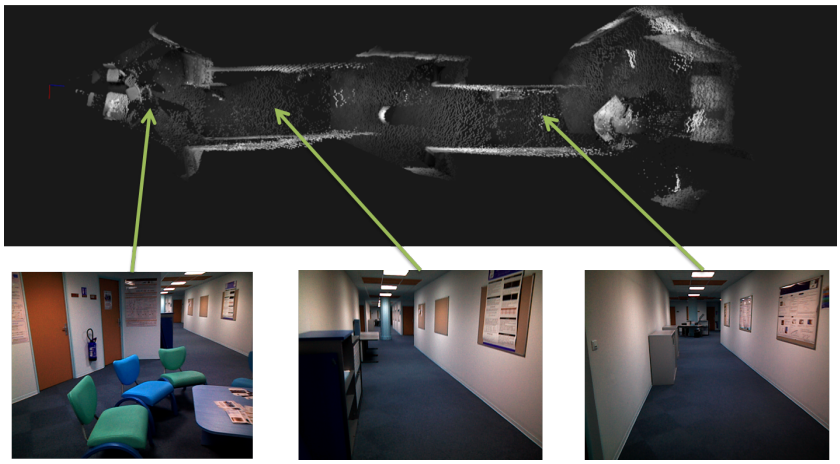
The results obtained by performing visual odometry in the synthetic environment demonstrated the robustness of the Point-to-hyperplane ICP method when rich color and depth features can be associated. During the experiments in real scenarios, the Point-to-hyperplane ICP method obtained better estimations in challenging sequences with blurred images such as fr1/room and fr1/360 (closed loops), demonstrating the robustness of the method. It was observed in the experiments that adaptive methods can improve the accuracy of the pose estimation methods when rich geometric and photometric information is available (as the case of sequences fr/3), however they were not robust enough for closed loop sequences.

[3]The legend NN4D means that the closest points were estimated by a kd-tree in the first iteration only

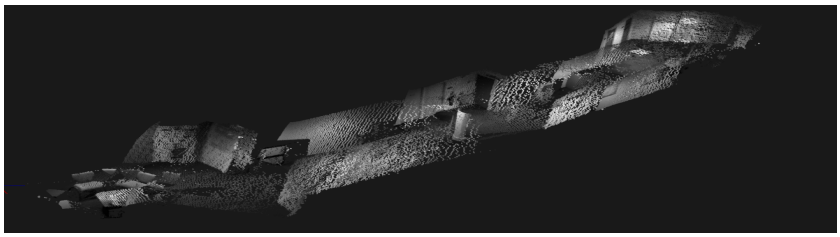
Table 3. Average in Time (miliseconds), number of iterations, Relative Pose Error (RPE) and Absolute Trajectory Error (ATE) for the dataset freiburg1 and freiburg3 [21].

Sequence	Method	ATE (m)		RPE translational (m)		RPE rotational (deg)		AVERAGE	
		RMSE	MEAN	RMSE	MEAN	RMSE	MEAN	Time(sec)	#Iterations
fr1/xyz	1 & 2	0.068	0.064	0.041	0.035	2.485	2.068	0.227	18.41
	3 & 4	**0.045**	**0.038**	**0.021**	**0.019**	**1.106**	**0.998**	**0.301**	26.94
	5	0.092	0.087	0.040	0.035	2.407	2.034	0.364	28.97
	6	0.086	0.080	0.041	0.036	2.421	2.080	0.946	**24.51**
fr1/rpy	1 & 2	0.102	0.087	0.040	0.034	2.918	**2.566**	**0.273**	**21.16**
	3 & 4	**0.035**	**0.032**	**0.038**	**0.032**	**2.820**	2.652	0.445	36.79
	5	0.129	0.111	0.045	0.037	3.034	2.643	0.288	23.15
	6	0.131	0.114	0.046	0.038	2.947	2.594	0.997	25.09
fr1/360	1 & 2	0.353	0.332	0.111	0.090	3.917	3.550	0.295	22.95
	3 & 4	0.322	0.296	0.152	0.114	**3.159**	**2.859**	0.460	38.68
	5	**0.190**	**0.179**	**0.094**	**0.085**	4.113	3.583	**0.285**	**22.18**
	6	0.268	0.245	0.191	0.149	5.210	4.539	0.962	24.55
fr1/room	1 & 2	0.375	0.353	0.075	0.055	3.373	2.836	**0.255**	**19.75**
	3 & 4	**0.152**	**0.131**	**0.056**	**0.047**	**2.673**	**2.329**	0.375	33.36
	5	0.323	0.286	0.063	0.051	3.250	2.743	0.308	23.34
	6	0.363	0.305	0.068	0.055	3.434	2.902	0.896	23.41
fr1/desk	1 & 2	**0.064**	**0.060**	**0.043**	**0.036**	2.738	2.403	**0.236**	**19.78**
	3 & 4	0.071	0.067	0.044	**0.036**	**2.310**	**2.028**	0.408	34.98
	5	0.069	0.065	0.044	**0.036**	2.580	2.233	0.300	23.79
	6	0.065	0.062	0.045	0.037	2.667	2.285	0.877	24.19
fr1/desk2	1 & 2	**0.083**	**0.079**	**0.058**	**0.049**	3.816	3.133	**0.243**	**20.00**
	3 & 4	0.133	0.116	0.060	0.051	**3.026**	**2.641**	0.496	38.33
	5	0.560	0.233	0.624	0.154	24.448	7.055	0.328	25.02
	6	0.409	0.188	0.463	0.129	16.883	5.887	0.940	25.20
fr1/floor	1 & 2	**0.124**	**0.105**	**0.035**	**0.023**	**1.946**	**1.364**	**0.198**	**15.61**
	3 & 4	0.473	0.405	0.080	0.051	3.909	1.915	0.355	31.67
	5	0.273	0.224	0.085	0.037	2.846	1.754	0.281	21.64
	6	2.050	1.765	0.384	0.089	21.096	4.932	0.873	22.75
fr1/plant	1 & 2	0.066	0.056	0.035	0.030	1.740	1.568	**0.262**	**20.63**
	3 & 4	0.101	0.093	0.055	0.043	2.130	1.947	0.395	34.88
	5	0.067	0.055	**0.031**	**0.027**	**1.608**	**1.405**	0.329	25.48
	6	**0.065**	**0.054**	0.033	0.028	1.623	1.427	0.914	23.52
fr1/teddy	1 & 2	0.260	0.219	**0.061**	**0.046**	**1.996**	**1.656**	**0.282**	**20.97**
	3 & 4	**0.169**	**0.158**	0.070	0.056	2.287	1.954	0.424	36.86
	5	0.303	0.267	0.086	0.049	2.432	1.734	0.322	23.74
	6	0.339	0.299	0.100	0.055	2.681	1.821	0.926	24.12
fr3/s_t_far	1 & 2	0.136	0.133	0.028	0.025	0.936	0.856	0.201	16.71
	3 & 4	0.401	0.361	0.067	0.061	1.561	1.428	**0.172**	**15.52**
	5	**0.044**	**0.040**	0.024	0.021	0.707	0.639	0.274	22.32
	6	**0.044**	0.042	**0.021**	**0.018**	**0.649**	**0.590**	0.617	16.32
fr3/s_t_near	1 & 2	0.152	0.143	0.027	0.021	1.568	1.176	**0.161**	**13.59**
	3 & 4	0.185	0.157	0.045	0.036	1.862	1.525	0.240	21.32
	5	**0.056**	**0.052**	**0.018**	**0.016**	**0.965**	**0.848**	0.345	27.43
	6	**0.066**	0.063	0.019	0.016	0.997	0.856	0.813	21.25

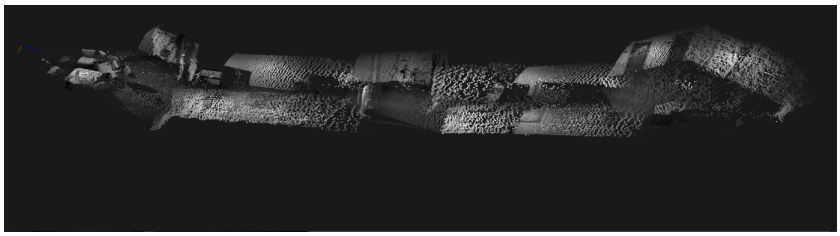
On the other hand, non adaptive methods achieved faster convergence and they obtained better estimations for sequences as fr1/desk, fr1/desk2 (where the sequences contain several sweeps) and fr1/floor, where the geometric features are not significant enough but texture. An example of the performance of the Point-to-hyperplane method can be shown in Figure 10, where the 3D reconstruction of loop closed sequences is obtained by transforming the cloud of points by the estimated 6DOF pose parameter.



(a) Top view



(b) Perspective view



(c) Side view

Figure 9. 3D reconstruction of a long corridor by the Point-to-hyperplane method

4.3. *Visual SLAM*

The Point-to-hyperplane ICP method has been implemented successfully for real time applications by employing the ASUS Xtion sensor. The proposed method here has demonstrated its robustness in a long corridor which contain similar geometric and photometric features (See Figure 9). The depth map obtained by the sensor contain holes in the depth map which are not valid values. This can generate a problem in the estimation of the normal for the Point-to-hyperplane ICP. In previous works, the corresponding intensities with non-valid depth values were considered as outliers. Another solution is to interpolate the surrounding valid depth values or assign the value of the closest point. By doing this, the method can achieve the main advantages of the Point-to-plane ICP or the direct method if either the color or the 3D point are not available.

The implementation of the Point-to-hyperplane ICP method in real-time can be an alternative for performing visual navigation, 3D reconstruction and localization of robotics platforms. The method can be also used for combining other types of measurements obtained by different sensors.

5. Conclusion

In this article, extended results of previous works on the Point-to-hyperplane strategy were shown. Particularly, the method has been extended to higher dimensions (3D Euclidean points + intensity and 3D Euclidean points + 3 channels of color) and it was mathematically demonstrated that the joint error function projected onto the normal direction has the effect of canceling out the λ tuning parameter since it does not change the direction of the normal. The future aim is to exploit other types of measurements for estimating the pose.

Two strategies for obtaining the closest points were compared, kd-trees and bilinear interpolation. For the benchmark sequences presented, both strategies obtained about the same performance. The bilinear interpolation was employed for the real-time application where a 2×2 window, which has been used to improve the computational cost. The real-time visual SLAM is running under CPU, obtaining a mean computational time of 1236 ms for the estimation of the normals in 4 dimensions. As a future work, the proposed method here will be implemented using GPU and a refinement method will be employed by estimating the global pose for a keyframe-to-frame and keyframe-to-keyframe tracking.

The reconstructions can be refined by any post-processing algorithms. The post-processing refinement strategies were not introduced here, but strategies that perform global convergence can be considered [22–24]. The refinement of the Point-to-hyperplane method has been recently performed by estimating the global pose of a RGB-D frame w.r.t. the generated 3D model in [25].

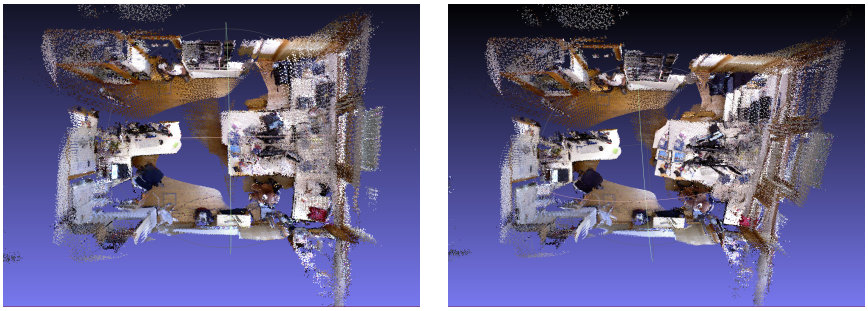
Funding

This work is supported by the European H2020 project: COMANOID, Université Côte d'Azur, CNRS, I3S, France and CONACYT (Consejo Nacional de Ciencia y Tecnología) under grant [265807], Mexico.

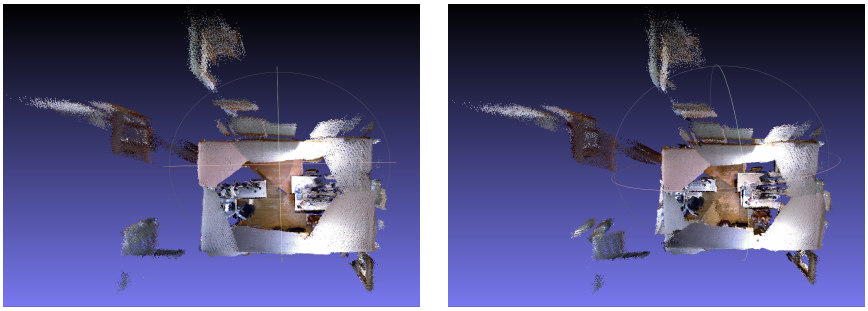
References

[1] Besl P, McKay N. A method for registration of 3-d shapes. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 1992 Feb;14:239–256.
[2] Chen Y, Medioni G. Object modeling by registration of multiple range images. In: IEEE International Conference on Robotics and Automation; Apr. Sacramento, CA, USA; 1991.
[3] Segal A, Haehnel D, Thrun S. Generalized-ICP. In: Proceedings of Robotics: Science and Systems; June. Seattle, USA; 2009.
[4] Huber P, Wiley J, InterScience W. Robust statistics. Wiley New York; 1981.
[5] Irani M, Anandan P. About direct methods. In: Proceedings of the International Workshop on Vision Algorithms: Theory and Practice. London, UK: Springer-Verlag; 1999. p. 267–277.
[6] Lowe D. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision. 2004 Nov;60:91–110.
[7] Bay H, Ess A, Tuytelaars T, et al. Speeded-up robust features (surf). Computer Vision and Image Understanding. 2008 Jun;110:346–359.
[8] Calonder M, Lepetit V, Strecha C, et al. Brief: Binary robust independent elementary features. Berlin, Heidelberg: Springer Berlin Heidelberg; 2010. p. 778–792.
[9] Rublee E, Rabaud V, Konolige K, et al. Orb: An efficient alternative to sift or surf. In: Proceedings of the 2011 International Conference on Computer Vision; ICCV '11. Washington, DC, USA: IEEE Computer Society; 2011. p. 2564–2571.
[10] Zheng Fang YZ. Experimental evaluation of RGB-D visual odometry methods. International Journal of Advanced Robotic Systems. 2015;12:26.
[11] Kerl C, Sturm J, Cremers D. Dense visual SLAM for RGB-D cameras. In: IEEE International Conference on Intelligent Robots and Systems. Tokyo, Japan; 2013.

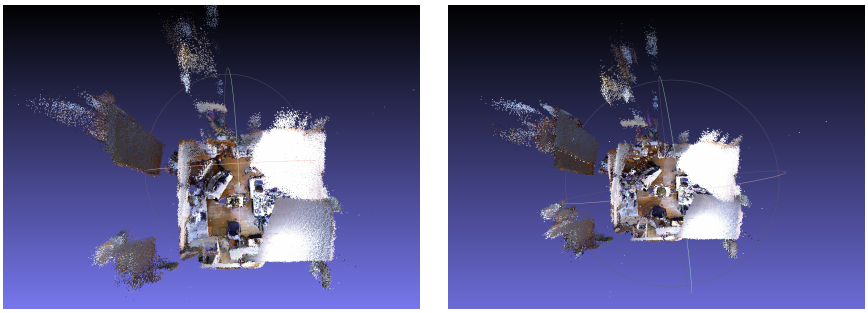
[12] Meilland M, Comport A. On unifying key-frame and voxel-based dense visual SLAM at large scales. In: International Conference on Intelligent Robots and Systems; November. Tokyo, Japan: IEEE/RSJ; 2013.
[13] Whelan T, Leutenegger S, Moreno R, et al. Elasticfusion: Dense slam without a pose graph. In: Proceedings of Robotics: Science and Systems; July. Rome, Italy; 2015.
[14] Morency L, Darrell T. Stereo tracking using ICP and normal flow constraint. In: 16th International Conference on Pattern Recognition. Quebec, Canada; 2002.
[15] Tykkälä T, Audras C, Comport A. Direct Iterative Closest Point for Real-time Visual Odometry. In: The Second international Workshop on Computer Vision in Vehicle Technology: From Earth to Mars in conjunction with the International Conference on Computer Vision; November. Barcelona, Spain; 2011.
[16] Ireta Muñoz FI, Comport AI. A proof that fusing measurements using point-to-hyperplane registration is invariant to relative scale. In: IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems. Baden-Baden, Germany; 2016.
[17] Ireta Muñoz FI, Comport AI. Point-to-hyperplane RGB-D pose estimation: Fusing photometric and geometric measurements. In: IEEE International Conference on Intelligent Robots and Systems. Deajeon, South Korea; 2016.
[18] Men H, Gebre B, Pochiraju K. Color point cloud registration with 4D ICP algorithm. In: IEEE International Conference on Robotics and Automation; May. Shanghai, China; 2011.
[19] Benhimane S, Malis E. Real-time image-based tracking of planes using efficient second-order minimization. In: IEEE International Conference on Intelligent Robots and Systems; Sept. Sendai, Japan; 2004.
[20] Handa A, Whelan T, McDonald J, et al. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In: IEEE International Conference on Robotics and Automation; May. Hong Kong, China; 2014.
[21] Sturm J, Engelhard N, Endres F, et al. A benchmark for the evaluation of RGB-D SLAM systems. In: IEEE International Conference on Intelligent Robot and Systems; Oct. Vilamoura, Algarve, Portugal; 2012.
[22] Yang J, Li H, Jia Y. Go-icp: Solving 3d registration efficiently and globally optimally. In: Proceedings of the 2013 IEEE International Conference on Computer Vision; ICCV '13. Washington, DC, USA: IEEE Computer Society; 2013. p. 1457–1464.
[23] Campbell D, Petersson L. GOGMA: globally-optimal gaussian mixture alignment. CoRR. 2016; abs/1603.00150.
[24] Straub J, Campbell T, How JP, et al. Efficient globally optimal point cloud alignment using bayesian nonparametric mixtures. CoRR. 2016;abs/1603.04868.
[25] Ireta Muñoz F, Comport A. Generalized global rgb-d registration: Collaborative local and global pose estimation by fusing color and depth. In: IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems. Daegu, South Korea; 2017.



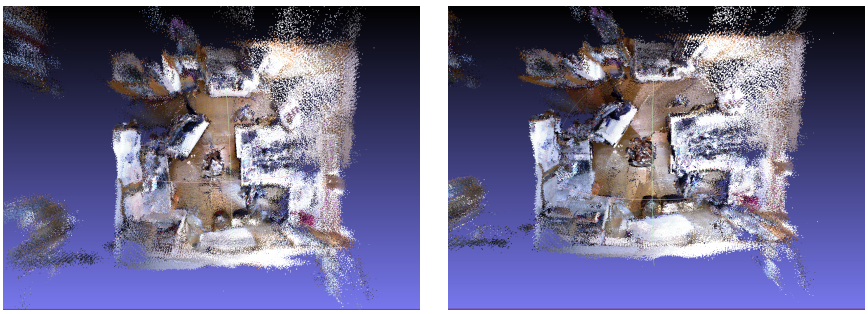
(a) fr1/room groundtruth (b) **Point-to-hyperplane**



(c) fr1/360 groundtruth (d) **Point-to-hyperplane**



(e) fr1/plant groundtruth (f) **Point-to-hyperplane**



(g) fr1/teddy groundtruth (h) **Point-to-hyperplane**

Figure 10. 3D reconstruction of sequences freiburg1: room, 360, plant and teddy (shown at each row, respectively). In the first column the groundtruth obtained by an external motion capture system is shown, the second column shown the result of the Point-to-hyperplane method. This difficult 360 degree sequence with motion blur shows that the proposed method can achieve more robust estimations.