A Proof that Fusing Measurements Using Point-to-Hyperplane Registration is Invariant to Relative Scale

Fernando I Ireta Munoz, Andrew I Comport

▶ **To cite this version:**

Fernando I Ireta Munoz, Andrew I Comport. A Proof that Fusing Measurements Using Point-to-Hyperplane Registration is Invariant to Relative Scale. IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Sep 2016, Baden - Baden, Germany. hal-01358130

HAL Id: hal-01358130
https://hal.science/hal-01358130v1

Submitted on 31 Aug 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Proof that Fusing Measurements Using Point-to-Hyperplane Registration is Invariant to Relative Scale

Fernando I. Ireta Muñoz[1] and Andrew I. Comport[2]

***Abstract*— The objective of this paper is to demonstrate that the metric error between different types of measurements can be jointly minimized without a scaling factor for the estimation processes if a Point-to-hyperplane approach is employed. This article is an extension of previous work based on the Point-to-hyperplane approach in 4 dimensions applied to pose estimation, where the proposed method minimized a fused error (3D Euclidean points + Image intensities) and it was experimentally demonstrated that the method is invariant to the choice of scale factor. In this paper, the invariance to the scale factor will be mathematically demonstrated. By doing this, it will be shown how the proposed method can further improve the convergence domain in 4D (or higher dimensions) and speed up the alignment between augmented frames (color + depth) whilst maintaining the robust and accurate properties of hybrid approaches when different types of measurements are available.**

I. INTRODUCTION

View registration has been widely studied in the field of computer vision and it is especially applied in mobile robotics to perform autonomous navigation by computing visual odometry and reconstructing 3D maps of the environment. One of the most fundamental problems is estimating the pose that relates measurements obtained from a moving sensor at different times.

RGB-D sensors provide rich geometric and photometric information from the scene that can be registered. The alignment between frames is ideally computed by jointly optimizing over color and depth measurements in a so-called *hybrid*-based manner. Basically, *hybrid* approaches combine *geometric* techniques, such as the well know ICP algorithm and its variants, with *photometric* techniques (direct or feature-based methods) together in order to obtain the benefits of each.

Different approaches have been proposed in the literature to estimate the pose between two different RGB-D views. The main surveys were recently cited in [26]. For the purpose of this paper, we will focus on proving that the Point-to-hyperplane approach proposed in [12] is invariant to a scale factor λ. The Point-to-hyperplane minimization avoids the estimation of λ, which weights the contribution of each measurement type and is usually required for methods that minimize the geometric and photometric error simultaneously. The estimation of λ has been widely studied by the vision community and various strategies had been proposed. Depending on how it is estimated, the scale factor λ can be categorized as an "adaptive" or a "non-adaptive" coefficient.

*This work is supported by the European H2020 project: COMANOID, Université Côte dAzur, CNRS, I3S, France and CONACYT, México.

[1] ireta@i3s.unice.fr [2] Andrew.Comport@cnrs.fr

The *non-adaptive* category mostly involves strategies for dense 3D reconstruction from RGB-D images. The coefficient λ is computed only once and it is used to align all the following frames which contain similar information, such as [9], [16], [17], [5], [13], [27], [6]. A real-time RGB-D SLAM using a non-adaptive scale factor is found in [23], [25], [24] where λ was also set empirically to reflect the relative difference in metrics used for color and depth costs.

On the other hand, *adaptive* methods increase the importance of the geometric error or the photometric error to ensure that each measurement is in the same order of magnitude. They are however more complex methods that compute the adequate scalar factor for each RGB-D image. These methods are usually employed to perform real-time tasks as 3D visual tracking [18], [2], visual odometry [22], [20], [7] and SLAM [14], [15]. They improve the convergence rate, however, it can be computationally expensive to estimate a λ for each new RGB-D frame.

The aim of this paper is to give the mathematical proof of the invariance to λ if a Point-to-hyperplane technique is used for minimizing different types of metrics as a single combined error. In particular, in [12] we proposed a method to minimize a 4D joint error which is invariant to the scale factor λ where, as a side note, the alignment is accelerated by performing the searching of the nearest neighbours via 4D-vector. The method performs visual odometry on real and simulated environments by estimating the camera poses from RGB-D sequences. During the experiments, the invariance of the tuning parameter λ was empirically observed. The method is based on a Point-to-plane method for 3D Euclidean points [4], but the normals are estimated in 4D space using a Principal Component Analysis (PCA) algorithm, as is done in [19], where the eigenvector with the lowest eigenvalue is chosen as the normal. The normal is therefore closely related to the relative uncertainty in the measurements.

In order to provide the proof that the Point-to-hyperplane method is invariant to λ, this paper is structured as follows. Section II establishes a general pipeline that is common to different hybrid methods for RGB-D pose estimation. Regularly, these methods minimize the errors simultaneously and scale them to the same magnitude by λ, which weighs the contribution of each during the minimization process. In Section III, demonstrates that the Point-to-hyperplane method can minimize the error as a single vector without any influence by the choice of λ, and the approach is generalized for higher dimensions such as 6D color and depth. Finally, extended results with respect to [12] for both, real and simulated environments, will be shown.

II. HYBRID-BASED RGB-D POSE ESTIMATION

The *hybrid*-based methods are useful when only geometric or color information alone are not significant enough to obtain a correct alignment. The main feature of these strategies, is that they can improve the robustness and accuracy of motion estimation than using only geometric or photometric minimization separately [26]. This section will give an overview of a general model that is common to all pose estimation approaches. In particular, the *hybrid*-based model presented in this paper will attempt to unify both, color and depth measurements, in a common framework.

The geometric and photometric techniques, share much similarity when estimating the pose. The strategy common to many classic techniques involves the following pipeline:

1) Acquire the set of measurements (color, depth, extracted features, etc) at different viewpoints.
2) Find the closest points between the datasets based on the current best pose estimation.
3) Minimize the weighted error function and estimate an incremental update for the pose.
4) Iteratively perform all the steps from 2 until convergence.

Therefore, if we develop the aforementioned stages and we consider that a RGB-D sensor is available, a 4D-vector measurement, defined here as $\mathbf{M}_i = [\mathbf{P}_i^\top \; I_i]^\top \in \mathbb{R}^4$, is obtained for the i-th point and its corresponding match is found in the other image. Each intensity value I_i is associated with an unique 3D Euclidean point $\mathbf{P}_i = [X_i \; Y_i \; Z_i]^\top \in \mathbb{R}^3$ which is computed by the back projection function such as: $\mathbf{P}_i = \mathbf{K}^{-1}\overline{\mathbf{p}_i}Z_i$, where $\mathbf{K} \in \mathbb{R}^{3\times3}$ is the intrinsic calibration matrix, $\overline{\mathbf{p}_i} = [p_{x_i} \; p_{y_i} \; 1]^\top \in \mathbb{R}^3$ are the homogeneous pixel coordinates and Z_i is the metric distance. Based on the corresponding point pairs between two datasets with an unknown pose $\mathbf{x}$, an $i-th$ error metric can be defined as:

$$\mathbf{e}_{H_i} = \boldsymbol{\lambda}\left(\mathbf{M}_i^* - f(\mathbf{M}_i, \mathbf{x})\right) \in \mathbb{R}^4 \tag{1}$$

where the superscript $*$ denotes reference measurements that correspond to a keyframe. This superscript will be used throughout this paper to denote the reference measurements.

As is shown in (1), the intensity is fused with the Euclidean distance with a weight matrix $\boldsymbol{\lambda}$ that scales the importance of the 3D geometric points with respect to the intensities such as:

$$\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda_G} & 0 \\ 0 & \lambda_I \end{bmatrix} \tag{2}$$

where $\boldsymbol{\lambda_G} = diag(\lambda_{G_1}, \lambda_{G_2}, \lambda_{G_3})$.

The given non-linear error in (1) is minimized iteratively using a Gauss-Newton approach to compute the unknown parameter $\mathbf{x}$ with increments given by:

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

where $\mathbf{J} = [\; \mathbf{J}_G^\top \; \mathbf{J}_I^\top \;]^\top$ represents the stacked Jacobian matrices obtained by derivating the stacked geometric and photometric error functions ($\mathbf{e}_G$ and $\mathbf{e}_I$, respectively), and

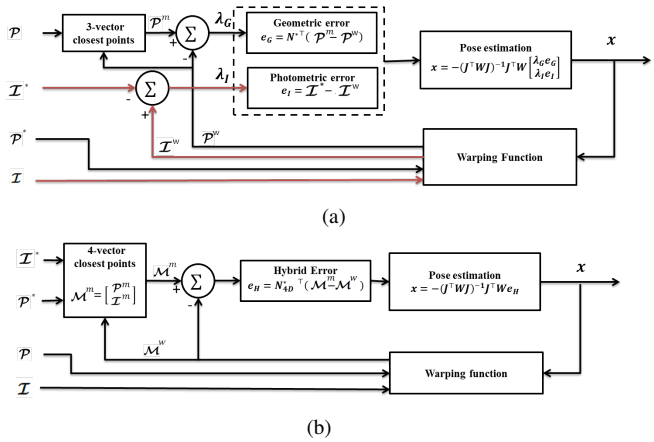


Fig. 1: *Hybrid-based approaches. (a) A Point-to-plane ICP algorithm is minimized simultaneously with a direct image-based method while searching the 3D closest points. The coefficient λ automatically scales both errors in the same magnitude for each new frame. (b) Point-to-hyperplane method [12], the closest points are estimated via 4D-vector and the error is minimized as a single vector, the coefficient λ is not longer any needed (see Section III).*

the weight matrix $\mathbf{W}$ contains the stacked weights associated with each set of coordinates obtained by M-estimation [10]. For the purposes of this paper, the Jacobian $\mathbf{J}_I$ is computed by using the Second Order Minimization (ESM) method [3]. Often, M-estimation is performed separately on each measurement vector since their scale is different.

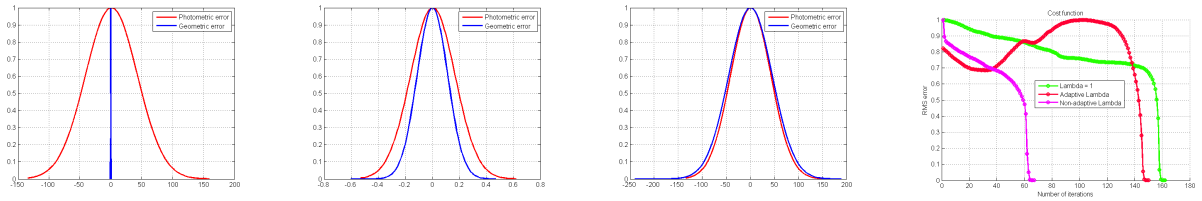
Finally, the pose estimation $\mathbf{T}(\mathbf{x})$ is computed at each iteration and is updated incrementally as $\widehat{\mathbf{T}} \leftarrow \widehat{\mathbf{T}}\mathbf{T}(\mathbf{x})$ until convergence.

The bi-objective minimization has been introduced as an error function that minimizes the photometric and geometric error simultaneously for hybrid-based approaches. However, it depends on the computation of the tuning parameter λ, which has a huge influence on the minimization process. If it is well estimated, it can speed up the alignment between two different frames while maintaining robustness and accuracy. An example of the influence of the coefficient is shown in the Fig. 2, where each error is fitted into a normal Gaussian distribution.

The cited *hybrid*-based strategies that uses the adaptive coefficient λ [18], [2], [22], [20], [7], [14], [15] perform the ICP Point-to-Plane algorithm [4] and a direct image-based method [11] whilst minimizing the error simultaneously. Generally, these strategies minimize the following error function:

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

with $\boldsymbol{\lambda_G} = \mathbf{I}_3$, where $\mathbf{P}_i^w \in \mathbb{R}^3$ is the warped 3D point and I_i^w is the warped intensity. The 3D correspondences (matches) with their associated intensities are respectively defined by $\mathbf{P}_i^m$ and $\mathbf{I}_i^m$. In fact, the searching of the closest points is often the most computationally expensive performed stage in the pose estimation process. Several strategies can be used, such



(a) (b) (c) (d)

Fig. 2: *Influence on the error function residual (Equation (19)) of the scale coefficient when (a) it is not estimated ($\lambda = 1$) and greyscale units are compared to m, (b) when the intensities and 3D points are normalized (non-adaptive λ [16]) and (c) when an adaptive λ is estimated [22]. Finally in (d) their cost function at each iteration of minimization until reach convergence is shown. It is clearly seen that non-adaptive or adaptive coefficients attempt to preserve about the same contribution of each measurement type (The photometric error and geometric error distributions are shown in red and blue, respectively).*

as *k*d-trees, linear interpolation or performing feature-based methods using various correlation strategies.

From the cited methods above, only [18] matches the closest points using the 4D-vector in order to better constrain the search for the closest 3D points, but in that paper the minimization remains similar to the others (As is shown in Fig. 1(a)).

III. POINT-TO-HYPERPLANE

Based on the error function defined in (1) and its expanded form in (4), a Point-to-hypeplane minimization can be defined such that:

$$\mathbf{e}_{H_i} = \mathbf{N}_i^{*\top} \boldsymbol{\lambda} \left(\mathbf{M}_i^* - f(\mathbf{M}_i, \mathbf{x}) \right) \in \mathbb{R}^4 \tag{5}$$

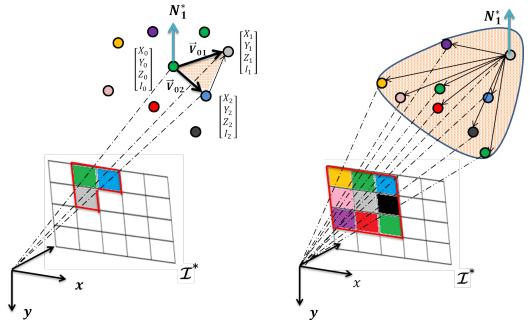
where $\mathbf{N}_i^* \in \mathbb{R}^4$ are the normals of the reference measurements and the scale parameter $\boldsymbol{\lambda} = diag\,(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ depends of the length of the measurement vector. If hybrid measurements are used, then the concept of Point-to-hyperplane is introduced and the integrated error is defined as follows:

$$\mathbf{e}_{H_i} = \mathbf{N}_i^{*\top} \left(\mathbf{M}_i^m - \mathbf{M}_i^w \right) \in \mathbb{R}^4 \tag{6}$$

where $\mathbf{M}_i^m$ denotes the match found between the reference and the transformed current measurements: $\mathbf{M_i^*} = [X_i^* \quad Y_i^* \quad Z_i^* \quad I_i^*]$ and $\mathbf{M_i^w} = [X_i^w \quad Y_i^w \quad Z_i^w \quad I_i^w]$, respectively. $\mathbf{M}_i^w$ is the measurement vector transformed by the geometric warping function $w(\cdot)$, which projects a 3D reference point $\mathbf{P}_i^* \in \mathbb{R}^3$ onto the current image plane.

It should be noted that the tuning parameters $\boldsymbol{\lambda}$ are not included in (6). This is due to the fact that the normals $\mathbf{N}^*$ are estimated by performing the cross product between the neighbouring reference points that forms an hyperplane, so that the distance of another point $\mathbf{M}_i^w$ to the formed hyperplane will be scaled by the geometric and photometric elements of $\boldsymbol{\lambda}$, which have not influence since all scale elements appears for each element of the error function. The coefficient $\boldsymbol{\lambda}$ is not longer needed since it has not effect in the error function (This demonstration will be shown below).

In order to extend (6) to higher dimensions and to demonstrate that the method is invariant to $\boldsymbol{\lambda}$, consider that the measurements vectors $\boldsymbol{\lambda}\mathbf{M}^*$ and $\boldsymbol{\lambda}\mathbf{M}$ contain j different types of measurements that are scaled by the same magnitude $\boldsymbol{\lambda}$.



(a) (b)

Fig. 3: *At least two vectors are needed to compute (a) the plane for 3 dimensions and (b) three vectors for 4 dimensions (In this paper 8 vectors are computed). In general, for N dimensions a $N-1$ vectors are needed. The points that forms the plane (or hyperplane) are selected depending on its associated pixel coordinates. Any pixel on the image can be selected as a central pixel (except for the corners) to compute the distance with its surrounding neighbours.*

The general form of the equation of a plane for 3D geometric points is $ax + by + cz + d = 0$, where $\langle x, y, z \rangle$ are the coordinates of the 3D point and $\langle a, b, c \rangle$ defines the normal vector. Therefore, an hyperplane of dimension j can be defined as follows:

$$N_1^* \lambda_1 M_1^* + N_2^* \lambda_2 M_2^* + \cdots + N_j^* \lambda_j M_k^* + d = 0 \tag{7}$$

where $d = -N_1^* \lambda_1 M_1^* - N_2^* \lambda_2 M_2^* - \cdots - N_j^* \lambda_j M_l^*$. The normals are calculated by performing the cross product in j dimensions such that:

$$\mathbf{N}_i^* = \mathbf{V}^{*1} \times \mathbf{V}^{*2} \times \cdots \times \mathbf{V}^{*j-1} \in \mathbb{R}^j \tag{8}$$

where each $m = 1, 2, \cdots, j-1$ vector $\mathbf{V}^{*m} = \boldsymbol{\lambda} \left(\mathbf{M}_k^* - \mathbf{M}_l^* \right)$ is similarly computed with the Point-to-point distance equation. The integer index $k \neq l$ selects the N closest points lying in the hyperplane. N is the number of vectors employed to compute the normal (For this paper $N = 8$). The minimum number of vectors that are required to perform the multidimensional cross product, depends on the number of dimensions used in the measurements vector (See Fig. 3).

The elements of the normal in (8) can be expressed as:

$$\mathbf{N}_i^* = \begin{bmatrix} c(\lambda_1)[\mathbf{M}_{k-l}^*]_1 & c(\lambda_2)[\mathbf{M}_{k-l}^*]_2 & \cdots & c(\lambda_j)[\mathbf{M}_{k-l}^*]_j \end{bmatrix}^\top \quad (9)$$

where the operator $[\cdot]_i$ extracts the $i-th$ row of $\left(\mathbf{M}_k^* - \mathbf{M}_l^*\right) \in \mathbb{R}^j$, and the operator $c(\lambda_i)$ corresponds to the product of the elements of the diagonal of $\boldsymbol{\lambda}$ except for λ_i such as:

$$c(\lambda_i) \underset{i \neq j}{=} \prod_i^j diag(\boldsymbol{\lambda}), \quad i = 1, 2, \cdots, j \quad (10)$$

The equation to compute the distance $\mathbf{e}_H$ of a point $\boldsymbol{\lambda}\mathbf{M}_i$ to the hyperplane, which is formed by the reference points $\boldsymbol{\lambda}\mathbf{M}_i^*$, can be represented in general form such as:

$$\mathbf{e}_{H_i} = \frac{[\mathbf{N}_i^*]_1 \lambda_1 M_1 + \cdots + [\mathbf{N}_i^*]_j \lambda_j M_j + d}{\sqrt{([\mathbf{N}_i^*]_1 \lambda_1)^2 + \cdots + ([\mathbf{N}_i^*]_j \lambda_j)^2}} \quad (11)$$

where $d = -[\mathbf{N}_i^*]_1^* \lambda_1 M_1^* - \cdots - [\mathbf{N}_i^*]_j^* \lambda_j M_j^*$ and the operator $[\mathbf{N}_i^*]_j = c(\lambda_j)[\mathbf{M}_{k-l}^*]_j$ extracts the $j-th$ element of (9).

Equation (11) can be easily represented in Hessian normal form as:

$$\mathbf{e}_{H_i} = -\overrightarrow{\mathbf{N}_\mathbf{i}^*}^\top \left(\mathbf{M}_i^* - \mathbf{M}_i\right) \quad (12)$$

where $\overrightarrow{\mathbf{N}_\mathbf{i}^*}$ is the normalization of the normal j-vector such as:

$$\overrightarrow{\mathbf{N}_\mathbf{i}^*} = \begin{bmatrix} \frac{[\mathbf{M}_{k-l}^*]_1}{\sqrt{([\mathbf{M}_{k-l}^*]_1)^2 + \cdots + ([\mathbf{M}_{k-l}^*]_j)^2}} \\ \vdots \\ \frac{[\mathbf{M}_{k-l}^*]_j}{\sqrt{([\mathbf{M}_{k-l}^*]_1)^2 + \cdots + ([\mathbf{M}_{k-l}^*]_j)^2}} \end{bmatrix} \in \mathbb{R}^j \quad (13)$$

that demonstrates the invariance to $\boldsymbol{\lambda}$ in (12), due to the fact that all its diagonal factors appears for both, numerator and denominator, and for each $j-th$ element of (13) as: $c(\lambda_j)\lambda_j$. Applied to the error function in (6), the following lemma is established.

Lemma 3.1: The integrated error $\mathbf{e}_H$ in $j-th$ dimension is invariant to the relative scale λ if it is minimized by a Point-to-hyperplane method.

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

Proof: Consider for simplicity the 3D case instead of 4D. Three hybrid 3D points that belong to the same cloud of points $\boldsymbol{\mathcal{M}} \in \mathbb{R}^{3\times n}$ (2D points + intensity) such as: $\mathbf{M}_0 = \begin{bmatrix} X_0 & Y_0 & I_0 \end{bmatrix}^\top$, $\mathbf{M}_1 = \begin{bmatrix} X_1 & Y_1 & I_1 \end{bmatrix}^\top$ and $\mathbf{M}_2 = \begin{bmatrix} X_2 & Y_2 & I_2 \end{bmatrix}^\top$, and consider one warped point $\mathbf{M}_\mathbf{w} = \begin{bmatrix} X_w & Y_w & I_w \end{bmatrix}^\top$ which represent any element of the warped point cloud $\boldsymbol{\mathcal{M}}^w \overset{\mathbf{f(x)}}{=} \Pi_3 \mathbf{T}(\mathbf{x}) \overline{\mathcal{M}_2}$. Note that the j-D case is an extension of this basic case.

In order to balance the magnitude of the metric measurements, a scalar factor is applied to all the points as: $\boldsymbol{\lambda}\boldsymbol{\mathcal{M}}$ and $\boldsymbol{\lambda}\boldsymbol{\mathcal{M}}^w$, where $\boldsymbol{\lambda}$ is defined as $\boldsymbol{\lambda} = diag(\lambda_X, \lambda_Y, \lambda_I)$. The coefficients of $\boldsymbol{\lambda}$ are introduced in a 3D point-to-hyperplane error function such that:

$$\mathbf{e}_{H_i} = \mathbf{N}_i^{*\top} \boldsymbol{\lambda} \left(\mathbf{M}_i^* - f(\mathbf{M}_i, \mathbf{x})\right) \in \mathbb{R}^3 \quad (14)$$

where the normals $\mathbf{N}^*$ are computed by performing the cross product of the vectors formed from $\boldsymbol{\lambda}\boldsymbol{\mathcal{M}}$. Considering the 3 hybrid points $\mathbf{M}_0$, $\mathbf{M}_1$ and $\mathbf{M}_2$ as reference points, the normal $\mathbf{N}_i^* = \begin{bmatrix} N_X & N_Y & N_I \end{bmatrix}^\top$ is defined here as the cross product between the vectors $\mathbf{V}^{01}$ and $\mathbf{V}^{02}$, which are defined as $\mathbf{V}^{01} = \lambda\left(\mathbf{M}_1 - \mathbf{M}_0\right)$ and $\mathbf{V}^{02} = \lambda\left(\mathbf{M}_2 - \mathbf{M}_0\right)$. that provides all the elements of the normal $\mathbf{N}^*$ at $\mathbf{M}_0$ such that:

$$\mathbf{N}_i^* = \begin{bmatrix} \lambda_Y \lambda_I \left(V_Y^{01} V_I^{02} - V_I^{01} V_Y^{02}\right) \\ \lambda_X \lambda_I \left(V_I^{01} V_X^{02} - V_X^{01} V_I^{02}\right) \\ \lambda_X \lambda_Y \left(V_X^{01} V_Y^{02} - V_Y^{01} V_X^{02}\right) \end{bmatrix} = \begin{bmatrix} \lambda_Y \lambda_I N_X \\ \lambda_X \lambda_I N_Y \\ \lambda_X \lambda_Y N_I \end{bmatrix} \quad (15)$$

Equation (14) can be rewritten as follows:

$$\mathbf{e}_{H_i} = \lambda_X \lambda_Y \lambda_I \left(N_X (X_i - X_w) + N_Y (Y_i - Y_w) + N_I (I_i - I_w)\right)$$

The normalization of the normal is calculated such as:

$$\overrightarrow{\mathbf{N}_i} = \begin{bmatrix} \frac{a}{\sqrt{a^2+b^2+c^2}} & \frac{b}{\sqrt{a^2+b^2+c^2}} & \frac{c}{\sqrt{a^2+b^2+c^2}} \end{bmatrix}^\top \quad (16)$$

where $a = \lambda_X \lambda_Y \lambda_I N_X$, $b = \lambda_X \lambda_Y \lambda_I N_y$ and $c = \lambda_X \lambda_Y \lambda_I N_I$. The error function of (14) can be minimized just as:

$$\mathbf{e}_{H_i} = \overrightarrow{\mathbf{N}_i}^\top \left(\mathbf{M}_i - \mathbf{M}_w\right) \quad (17)$$

The unknown parameter $\mathbf{x}$ is estimated by following the same pipeline of the hybrid-based methods, where (3) is rewritten as follows:

$$\mathbf{x} = -(\mathbf{J}^\top \mathbf{W} \mathbf{J})^{-1} \mathbf{J}^\top \mathbf{W} \mathbf{e}_H \quad (18)$$

IV. RESULTS

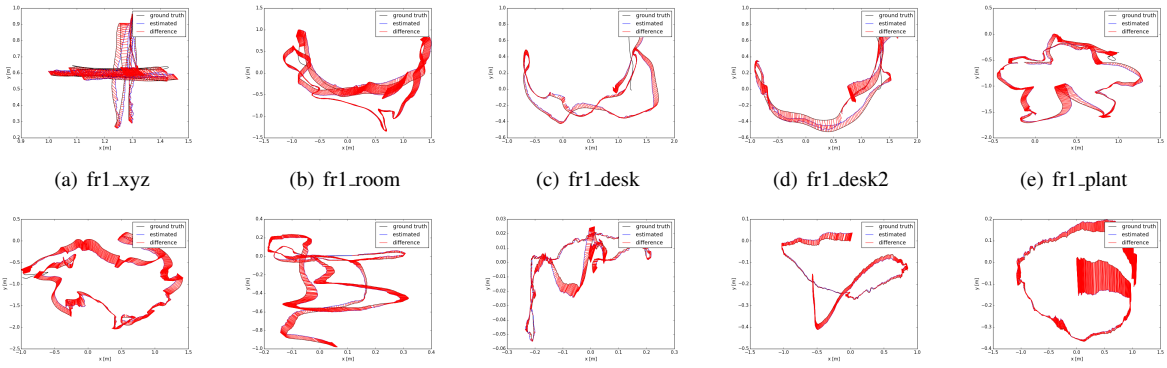
All the experiments presented in this paper were performed on both, real and synthetic, RGB-D grayscale images. In order to improve computational efficiency, a multiresolution pyramid was used. The iterative closest points minimization can be stopped by two criteria: when the maximum number of iterations (200) is reached or if the norm of the transformation matrix is less than 1×10^{-6} in rotation and 1×10^{-5} in translation. The Huber influence function was employed in the M-estimator to reject outliers and obtain more robust estimations. Only one M-estimator was used for the unified measurement vector as opposed to two in [15].

The Point-to-hyperplane method is compared here with the following hybrid error function, which weights the minimization between the geometric and photometric error [15].

$$\mathbf{e}_{H_i} = \begin{pmatrix} \boldsymbol{\lambda} \left(\widehat{\mathbf{R}}\mathbf{R}(\mathbf{x})\mathbf{N}_i^*\right)^\top \left(\mathbf{P}_i^m - \Pi_3 \widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}) \overline{\mathbf{P}_i^*}\right) \\ \mathbf{I}_i \left(w(\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}), \mathbf{P}_i^*)\right) - \mathbf{I}_i^*(\mathbf{p}_i^*) \end{pmatrix} \in \mathbb{R}^4 \quad (19)$$

where $\mathbf{P}_i^m \in \mathbb{R}^3$ is the closest point in the current cloud, $\hat{\mathbf{R}} \leftarrow \hat{\mathbf{R}}\mathbf{R}(\mathbf{x})$ is the incremental update of the rotations, $\mathbf{N}_i^* = [N_{x_i}\ N_{y_i}\ N_{z_i}]^\top$ are the normals of the reference points and $\Pi_3 = [\mathbf{1}, \mathbf{0}] \in \mathbb{R}^{3\times 4}$ is the projection matrix. For this strategy, the normals are only estimated for the geometric points (e.g. a classic Point-to-plane [4] approach). The selection is done by choosing the neighbouring pixels in the color image as is shown in Fig. 3(a).

In the case of the 4-dimensional space, a 3×3 window is selected to compute the normals as is shown in Fig. 3(b). Based on the Generalized-ICP algorithm [19], the Principal Component Analysis (PCA) is employed on the covariance



(a) fr1_xyz (b) fr1_room (c) fr1_desk (d) fr1_desk2 (e) fr1_plant

(f) fr1_teddy (g) lvr_traj0 (h) lvr_traj1 (i) lvr_traj2 (j) lvr_traj3

Fig. 4: *Examples of the Absolute Trajectory Error evaluation obtained by the Point-to-Hyperplane method. The benchmarck datasets [8] and [21] were used.*

matrix of the 8 closest points for each point, where the eigenvector with the lowest eigenvalue corresponds to the normal vector $\mathbf{N}^*_{i4D}$. An alternative algorithm that can be used to compute the normals is given in [1].

For the comparisons, the Point-to-hyperplane method is compared with different strategies that compute a non-adaptive λ, which are computed based on the strategies [16] (The intensities are normalized $\lambda_I = I_i/255$) and an adaptive λ [22] (where the scale parameter is the ratio between the Median Absolute Deviation (MAD) of the errors $\lambda_G = MAD(\mathbf{e}_I)/MAD(\mathbf{e}_G)$). The minimization of the error presented in (19) will also be compared with a $\lambda = 1$ (λ is not estimated).

1) Simulated environment: In order to verify the performance of the method, 1000 synthesized images were generated with random poses. The alignment process between the generated images and its reference image ensures exact correspondences at the solution. The mean of number of iterations and computational time until reaching convergence is shown in Table I. The normals were computed only once for the reference image, obtaining 9.29 seconds.

2) Real environments: For the test with real RGB-D images the Freiburg 1 sequences from [21] were employed to perform Visual Odometry with frame-to-frame tracking in the same way as the simulated environment [8]. The performance of the hybrid-based techniques that estimate an adaptive λ is compared to the Point-to-hyperplane technique. The Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) between the estimated trajectories and their respective groundtruth trajectories (Table II) are compared. It should be noted that the averages in time and number of iterations is less than the averages obtained by the Point-to-hyperplane method for sequences *desk* and *floor*, but obtaining also less error. With respect to the previous results in [12], the experiments were carried with different strategies that estimate the scale parameter λ for hybrid-based approaches. Here, the proposed method is compared with an adaptive λ strategy that normalizes the intesities, in order to demonstrate the invariance to the scale factor. A comparison of the ATE for some of the sequences are shown in Fig. 4.

TABLE I: *Averages in Time and in Number of Iterations until Convergence for 1000 Synthesized Images at Random Poses.*

Method	# Iterations	Time (sec)
1) Hybrid-based + non-adaptive λ [16]	124.2280	1.5403
2) Hybrid-based + adaptive λ [22]	152.3790	1.9019
3) Hybrid-based (λ is not estimated)	155.0030	1.9406
4) Point-to-hyperplane	**44.6280**	**0.4679**

V. CONCLUSION

In this paper, it is proven mathematically that the Point-to-hyperplane approach [12] is invariant to $\boldsymbol{\lambda}$. The normals have been obtained by performing the multidimensional cross product of vectors in j dimension and the $\boldsymbol{\lambda}$ coefficients are shown to not influence the minimization process. Evaluations in the experiments, show that more accurate results are obtained for the Point-to-hyperplane method when the normals are estimated with the PCA algorithm instead of the cross product. However, the former algorithm requires extra computational time which is linear with the number of nearest neighbours selected in the image [1].

We aim to generalize this approach to any measurement fusion approach for which enough data is available to compute normals, such as color or IR measurements.

REFERENCES

[1] H. Badino, D. Huber, Y. Park, and T. Kanade. Fast and accurate computation of surface normals from range images. In *IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011.

[2] A. Ercil Batu Akan, M. Cetin. 3D Head tracking using normal flow constraints in a vehicle environment. In *Biennial on DSP for in-Vehicle and Mobile Systems*, Istanbul, Turkey, June 2007.

[3] Selim Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *IEEE International Conference on Intelligent Robots and Systems*, Sendai, Japan, Sept 2004.

[4] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *IEEE International Conference on Robotics and Automation*, Sacramento, CA, USA, Apr 1991.

[5] L. Douadi, M.-J. Aldon, and A. Crosnier. Pair-wise registration of 3D/Color data sets with ICP. In *IEEE International Conference on Intelligent Robots and Systems*, Beijing, China, Oct 2006.

TABLE II: *Averages in Time (miliseconds), number of iterations, Relative Pose Error (RPE) and Absolute Trajectory Error (ATE) for the simulated and real dataset [8], [21]. It can be seen that the Point-to-hyperplane method [12] improves hybrid methods that combine the direct approach and the geometric Point-to-plane approach.*

Sequence	Method	RPE translational (m)			RPE rotational (deg)			ATE (m)			AVERAGE	
		RMSE	MEAN	STD	RMSE	MEAN	STD	RMSE	MEAN	STD	Time(ms)	#Iterations
fr1/xyz	Adaptive λ [22]	0.033	0.030	0.014	2.025	1.741	1.034	0.095	0.087	0.039	386.1	29.73
	Point-to-hyperplane	**0.021**	**0.019**	**0.008**	**1.106**	**0.998**	**0.477**	**0.045**	**0.038**	**0.024**	**300.8**	**26.94**
fr1/rpy	Adaptive λ [22]	0.062	0.050	0.037	3.161	2.887	1.288	0.136	0.115	0.072	515.3	**39.52**
	Point-to-hyperplane	**0.038**	**0.032**	**0.020**	**2.820**	**2.652**	**0.959**	**0.035**	**0.032**	**0.015**	**444.9**	39.74
fr1/360	Adaptive λ [22]	**0.146**	0.118	**0.086**	4.171	3.844	1.621	0.520	0.484	0.188	654.3	49.23
	Point-to-hyperplane	0.152	**0.114**	0.100	**3.159**	**2.859**	**1.343**	**0.322**	**0.296**	**0.125**	**460.1**	**40.41**
fr1/room	Adaptive λ [22]	0.076	0.060	0.048	3.285	2.912	1.520	0.434	0.404	0.158	436.3	33.48
	Point-to-hyperplane	**0.056**	**0.047**	**0.030**	**2.673**	**2.329**	**1.313**	**0.174**	**0.152**	**0.086**	**374.6**	**33.36**
fr1/desk	Adaptive λ [22]	0.047	0.039	0.027	2.826	2.503	1.312	0.108	0.104	0.029	**383.4**	**30.20**
	Point-to-hyperplane	**0.044**	**0.036**	**0.025**	**2.309**	**2.027**	**1.106**	**0.071**	**0.067**	**0.023**	408.1	36.04
fr1/desk2	Adaptive λ [22]	**0.058**	**0.051**	**0.027**	3.483	3.026	1.725	0.189	0.174	0.075	574.8	43.79
	Point-to-hyperplane	0.060	**0.051**	0.031	**3.026**	**2.641**	**1.478**	**0.133**	**0.116**	**0.065**	**496.0**	**43.43**
fr1/floor	Adaptive λ [22]	0.094	**0.038**	0.086	4.660	1.953	4.231	0.772	0.666	0.391	**318.1**	**24.80**
	Point-to-hyperplane	**0.080**	0.051	**0.062**	**3.909**	**1.915**	**3.408**	**0.473**	**0.405**	**0.244**	355.2	31.67
fr1/plant	Adaptive λ [22]	0.106	0.067	0.082	3.941	3.223	2.268	0.324	0.296	0.132	560.5	42.35
	Point-to-hyperplane	**0.055**	**0.043**	**0.034**	**2.130**	**1.947**	**0.864**	**0.101**	**0.093**	**0.037**	**394.5**	**34.88**
fr1/teddy	Adaptive λ [22]	0.096	0.081	0.051	3.410	3.021	1.583	0.615	0.553	0.271	553.0	42.16
	Point-to-hyperplane	**0.070**	**0.056**	**0.043**	**2.287**	**1.954**	**1.187**	**0.169**	**0.158**	**0.059**	**423.5**	**36.86**
lvr/traj0	Adaptive λ [22]	**0.001**	**0.001**	**0.001**	0.044	0.035	**0.027**	0.128	0.114	0.057	275.6	22.37
	Point-to-hyperplane	0.002	**0.001**	0.002	**0.042**	**0.026**	0.033	**0.050**	**0.046**	**0.019**	**172.4**	**16.07**
lvr/traj1	Adaptive λ [22]	0.002	**0.001**	**0.001**	0.048	0.041	0.024	0.114	0.104	0.046	298.9	23.74
	Point-to-hyperplane	**0.001**	**0.001**	**0.001**	**0.021**	**0.017**	**0.013**	**0.041**	**0.032**	**0.026**	**140.0**	**13.51**
lvr/traj2	Adaptive λ [22]	0.002	**0.001**	**0.001**	0.044	0.039	0.021	0.074	0.067	0.030	323.7	25.52
	Point-to-hyperplane	**0.001**	**0.001**	**0.001**	**0.024**	**0.019**	**0.014**	**0.039**	**0.036**	**0.016**	**165.7**	**15.53**
lvr/traj3	Adaptive λ [22]	0.002	**0.001**	**0.001**	0.070	0.053	0.045	0.218	0.202	0.082	284.3	22.77
	Point-to-hyperplane	**0.001**	**0.001**	**0.001**	**0.044**	**0.027**	**0.035**	**0.080**	**0.066**	**0.045**	**178.9**	**16.82**

[6] Pierre Georgel, Selim Benhimane, and Nassir Navab. A unified approach combining photometric and geometric information for pose estimation. In *Proceedings of the British Machine Vision Conference*, Leeds, United Kingdom, September 2008.

[7] T. Han, C. Xu, R. Loxton, and L. Xie. Bi-objective optimization for robust RGB-D visual odometry. In *The 27th Chinese Control and Decision Conference*, Qingdao, China, May 2015.

[8] A. Handa, T. Whelan, J. McDonald, and A.J. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *IEEE International Conference on Robotics and Automation*, Hong Kong, China, May 2014.

[9] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. *The 12th International Symposium on Experimental Robotics*, chapter RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[10] P.J. Huber, J. Wiley, and W. InterScience. *Robust statistics*. Wiley New York, 1981.

[11] Michal Irani and P. Anandan. About direct methods. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, pages 267–277, London, UK, 1999. Springer-Verlag.

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

[13] A. E. Johnson and Sing Bing Kang. Registration and integration of textured 3-D data. In *International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, Washington, DC, USA, 1997. IEEE Computer Society.

[14] C. Kerl, J. Sturm, and D. Cremers. Dense visual SLAM for RGB-D cameras. In *IEEE International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 2013.

[15] M. Meilland and A.I. Comport. On unifying key-frame and voxel-based dense visual SLAM at large scales. In *International Conference on Intelligent Robots and Systems*, Tokyo, Japan, November 2013. IEEE/RSJ.

[16] H. Men, B. Gebre, and K. Pochiraju. Color point cloud registration with 4D ICP algorithm. In *IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011.

[17] J. Pauli Michael Korn, M. Holzkothen. Color supported generalized-ICP. In *International Conference on Computer Vision Theory and Applications*, Lisbon, Portugal, 2014.

[18] L. Morency and T. Darrell. Stereo tracking using ICP and normal flow constraint. In *16th International Conference on Pattern Recognition*, Quebec, Canada, 2002.

[19] A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009.

[20] F. Steinbruecker, C. Kerl, J. Sturm, and D. Cremers. Large-scale multi-resolution surface reconstruction from RGB-D sequences. In *International Conference on Computer Vision*, Sydney, Australia, 2013.

[21] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IEEE International Conference on Intelligent Robot and Systems*, Vilamoura, Algarve, Portugal, Oct. 2012.

[22] T.M. Tykkälä, C. Audras, and A.I Comport. 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*, Barcelona, Spain, November 2011.

[23] T. Whelan, H. Johannsson, M. Kaess, J.J. Leonard, and J. McDonald. Robust real-time visual odometry for dense RGB-D mapping. In *IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, May 2013.

[24] Thomas Whelan, Michael Kaess, Hordur Johannsson, Maurice Fallon, John J. Leonard, and John Mcdonald. Real-time large-scale dense RGB-D SLAM with volumetric fusion. *International Journal of Robotics Research*, 34(4-5):598–626, April 2015.

[25] Thomas Whelan, Stefan Leutenegger, Renato Salas Moreno, Ben Glocker, and Andrew Davison. Elasticfusion: Dense slam without a pose graph. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

[26] Yu Zhang Zheng Fang. Experimental evaluation of RGB-D visual odometry methods. *International Journal of Advanced Robotic Systems*, 2015.

[27] Qian-Yi Zhou and Vladlen Koltun. Color map optimization for 3D reconstruction with consumer depth cameras. *ACM Trans. Graph.*, July 2014.