# Point-to-hyperplane RGB-D Pose Estimation: Fusing Photometric and Geometric Measurements

Fernando Ireta, Andrew I. Comport

# Point-to-hyperplane RGB-D Pose Estimation: Fusing Photometric and Geometric Measurements

Fernando Ireta and Andrew Ian Comport

***Abstract*— The objective of this paper is to investigate the problem of how to best combine and fuse color and depth measurements for incremental pose estimation or 3D tracking. Subsequently a framework will be proposed that allows to formulate the problem with a unique measurement vector and not to combine them in an ad-hoc manner. In particular, the full color and depth measurement will be defined as a 4-vector (by combining 3D Euclidean points + image intensities) and an optimal error for pose estimation will be derived from this. As will be shown, this will lead to designing an iterative closest point approach in 4-dimensional space. A *k*d-tree is used to find the closest point in 4-space, therefore simultaneously accounting for color and depth. Based on this unified framework a novel point-to-hyperplane approach will be introduced which has the advantages of classic point-to-plane ICP but in 4-space. By doing this it will be shown that there is no longer any need to provide or estimate a scale-factor between different measurement types. Consequently this allows to increase the convergence domain and speed up the alignment, whilst maintaining the robust and accurate properties. Results on both simulated and real environments will be provided along with benchmark comparisons.**

## I. INTRODUCTION

Color and depth images acquired from RGB-D sensors are increasingly useful, especially in robotics for computing visual odometry, performing autonomous navigation and reconstructing 3D environments. One of the most fundamental problems is estimating the pose that relates measurements obtained from a moving sensor at different times. Some recent approaches have combined both measurements together in a limited hybrid manner.

The problem of pose estimation from color or depth images have each been individually studied in the computer vision and robotics literature. Classically color and depth measurements have been used separately in image-based and geometric-based pose estimation. In the case of depth images, the well known Iterative Closest Point (ICP) algorithm prevails [2] and in particular the point-to-plane ICP algorithm is especially efficient and robust [3]. On the other hand, color images have been used to estimate the pose of the camera using direct and dense error functions based on view synthesis [4]. The later will be referred to here as the image-based approach. It can be noted that feature-based image approaches first extract geometric information from the image before performing estimation on a geometric error. Feature-based approaches are a sub-part of the direct approach and wont be detailed here [9].

Whilst color and depth based pose estimation have been studied separately, similar solutions have been used for both using a non-linear iteratively re-weighted least squares

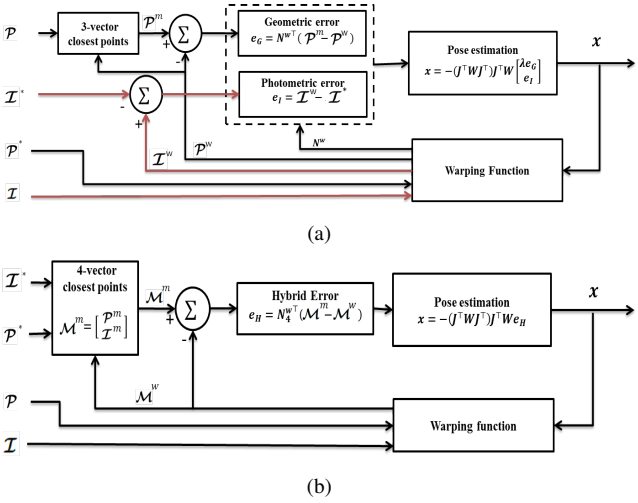



Fig. 1. Two hybrid-based approaches to estimate the unknown pose $\mathbf{x}$ between two sequent RGB-D frames. (a) A direct approach for the color images $\mathcal{I}$ and $\mathcal{I}^*$, and a point-to-plane algorithm for the geometric cloud of points $\mathcal{P}$ and $\mathcal{P}^*$ is used. (b) Proposed method: a matching stage considers the 4-vector for minimizing the integrated error with the point-to-hyperplane algorithm, which computes the normals $\mathbf{N}$ in 4D space. The scale factor $\lambda$ is no longer needed.

(IRLS) method. The general IRLS pipeline for pose estimation follows the common strategy across different measurement types:

1) Transform/warp the current measurement onto a reference frame using the last pose estimate.
2) Find the closest points between the two datasets.
3) Determine an error between the two datasets (and robust weights).
4) Estimate an incremental update on the pose.
5) Repeat to 1. until convergence.

Recently, several strategies have combined the color and depth measurements together in different ways and attempt to retain the respective benefits of each. The advantages of using both include increased efficiency, accuracy and robustness. Image-based approaches alone are dependant on texture in the images to constrain all degrees of freedom. For example, a wall with only horizontal lines would be degenerate. ICP approaches require sufficient geometry and are, for example, degenerate in the case of a movement parallel to a flat wall. In [21] a recent survey of the real-time performance of these approaches is provided. In this paper, those approaches which combine depth and color for robust and accurate pose estimation will be referred to as *hybrid* approaches (See Fig. 1).

Amongst the various hybrid methods, those of most interest are those that minimize a photometric and geometric error simultaneously in real-time [11], [12], [19], [20]. The two main differences in the proposed approaches are categorized as:

- How the closest points are determined between different RGB-D measurements.
- How the joint optimization is performed.

The aforementioned hybrid approaches are somewhat ad-hoc because they do not necessarily consider the color and depth simultaneously when computing closest points. Furthermore, in the optimization stage they simply combine the classic ICP and image-based approaches by minimizing both error types simultaneously. This, however, requires the definition or estimation of a tuning parameter $\lambda$ which weights the respective contribution of each different measurement type.

First consider a fused version of the closest point search in Step 2 which is required for both ICP and image-based approaches. In the case of ICP alone, the closest points are often obtained by performing a *k*d-tree (k-dimensional) for nearest neighbours search. Alternatively, in the image-based approaches the image warping function finds the closest color values by view interpolation (nearest neighbour, bi-linear, bi-cubic,...) directly in image space. Of the recent hybrid approaches [11], [12], [19], [20], each performs finding the closest point search separately for both color and depth and no fused information is considered. Methods that consider both color and depth in the closest point matching stage include [10], [14], [13]. The former and later approaches use 3 channels of color and differ in the color spaces used while [14] considers only greyscale information. Finding the closest points using both color and depth increases the accuracy of finding the true nearest neighbour, however, this requires an efficient search in 4-space (3D points + intensity).

Now consider the joint optimization problem of the IRLS algorithm that minimizes both a fused ICP and image-based error. The large majority of classic approaches involve simply stacking the two error functions and minimizing the resulting joint error simultaneously [10], [14], [19], [12], [11], [20], [13]. All except [10] perform ICP point-to-plane combined with the image-based approach. The drawback of these approaches is that they require the definition of a tuning parameter $\lambda$ which weights the respective contribution of each measurement. These methods then vary in how this tuning parameter is determined. In [10], $\lambda$ is computed by estimating the ratio between the minimum and maximum values of both, color space and geometric errrors and the best value is chosen experimentally in this range. [14] proposes interestingly an adaptative $\lambda$ which is varied using a sigmoidal function which favors the ICP approach far from the solution and the image-based approach close to the minimum. This has the benefit of faster convergence and more accuracy at the solution. In [19], [12], the scale factor is automatically estimated as the ratio between the Median Absolute Deviations (MAD) of the color error and the median of the depth error (i.e. their relative robust variance). In [11], [17], $\lambda$ is estimated by computing the covariance of the residuals for each point individually assuming a t-distribution of the error. This improved the convergence rate, however, is computationally expensive to iteratively compute a $\lambda$ for each pixel.

The aim of this paper is to propose a unified framework for fusing both image-based and ICP strategies for pose estimation at each stage of the IRLS process. As will be shown, this leads to a novel point-to-hyperplane ICP approach in 4 dimensions (3D + Intensity) which could easily be extended to greater dimensions (for example color RGB). This formulation also naturally leads to a fused closest point search strategy that exploits both color and geometric information simultaneously. The approach used in the paper to find the closest points in 4D space uses a *k*d-tree, however, alternative search strategies could also be used. In practice, and for computational efficiency, the ANN (Approximate Nearest Neighbour) [15] algorithm is used. Furthermore, the *k*d-tree can be built only once from the reference image before the iterative loop, therefore maintaining efficiency.

The paper is organized as follows. Section II briefly explains the classic hybrid approach that jointly minimizes intensities and point-to-plane ICP. In Section III a novel point-to-hyperplane approach is introduced. Section IV provides the implementation details common to all the methods that were evaluated. Finally, simulated and real experimental results with benchmarks are presented in Section V.

## II. JOINT METHOD FOR COMBINING GEOMETRIC AND PHOTOMETRIC APPROACHES

Pose estimation from hybrid methods is achieved by fusing the geometric and photometric optimization functions and minimizing the errors simultaneously. The main feature of hybrid methods for estimating the camera poses, is that they constrain the pose estimation better and can converge faster than using the techniques alone.

The pose will be defined here as the homogeneous pose matrix $\mathbf{T}(\mathbf{x}) \in \mathbb{R}^{4\times4}$ which depends on a minimal parametrisation of 6 parameters which are defined here as the linear and angular velocity $\mathbf{x} = [\upsilon, \omega]^\top \in \mathbb{R}^6$. The homogeneous transformation matrix can be decomposed into rotational and translational components $\mathbf{T}(\mathbf{x}) = (\mathbf{R}(\mathbf{x}), \mathbf{t}(\mathbf{x})) \in \mathbb{SE}(3)$. The relationship between both is given by the exponential map as $\mathbf{T}(\mathbf{x}) = e^{[\mathbf{x}]_\wedge}$, with the operator $[\cdot]_\wedge$ as:

$$[\mathbf{x}]_\wedge = \begin{bmatrix} [\omega]_\times & \upsilon \\ 0 & 0 \end{bmatrix} \tag{1}$$

where $[\cdot]_\times$ is the skew symmetric matrix operator.

The hybrid approach used to estimate the pose is depicted in Fig. 1(a). It defines an error function that minimizes the joint error between subsequent RGB-D image frames (see [12] for more detail) such as:

$$\mathbf{e}_{H_i} = \rho_i \begin{pmatrix} \lambda \left( \widehat{\mathbf{R}}\mathbf{R}(\mathbf{x})\mathbf{N}_i^* \right)^\top \left( \mathbf{P}_i^m - \Pi_3 \widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}) \overline{\mathbf{P}_i^*} \right) \\ \mathbf{I}_i \left( w(\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}), \mathbf{P}_i^*) \right) - \mathbf{I}_i^*(\mathbf{p}^*) \end{pmatrix} \in \mathbb{R}^4 \tag{2}$$

where the first row of equation (2) is the point-to-plane ICP error with projective data association and the second row is the photometric term. The superscript $*$ identifies the reference measurements, $\Pi_3 = [\mathbf{1}, \mathbf{0}] \in \mathbb{R}^{3\times4}$ is the projection matrix, $\mathbf{N}_i^* \in \mathbb{R}^3$ is the surface normal for each homogeneous 3D point $\overline{\mathbf{P}}_i \in \mathbb{R}^4$. The closest point $\mathbf{P}_i^m$ can be obtained by linearly interpolating the warped pixel coordinates into the current depth map as in [11], [12], [19]. The geometric warping function $w(\cdot)$ projects a reference 3D point $\mathbf{P}_i^* \in \mathbb{R}^3$ onto the current image plane. The closest image intensity is then found by interpolation of the current intensity function at the warped pixel coordinates to obtain the corresponding intensity as: $\mathbf{I}_i^w(\mathbf{p}_i^*) = \mathbf{I}_i(\mathbf{p}_i^w) \in \mathbb{Z}$. The 3D point is computed by back projection 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 $\mathbf{K} \in \mathbb{R}^{3\times3}$ is the calibration matrix which contains the intrinsic parameters of the camera, and $Z_i \in \mathbb{R}^+$ is the metric measurement for each pixel coordinate $\overline{\mathbf{p}_i} = [u_i\ v_i\ 1]^\top \in \mathbb{R}^3$ of the depth image.

The given non-linear error in Equation 2 is minimized iteratively using a Gauss-Newton approach to compute the unknown parameter $\mathbf{x}$ with increments given by:

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

where $\mathbf{J} = [\mathbf{J}_I\ \mathbf{J}_G]^\top$ represents the stacked Jacobian matrices obtained by derivating the stacked photometric and geometric error functions ($\mathbf{e}_I$ and $\mathbf{e}_G$ respectively), and the weight matrix $\mathbf{W}$ contains the weights $\rho_i$ associated to each set of coordinates obtained by M-estimation [8]. The photometric Jacobian $\mathbf{J}_I$ is computed using the efficient second order minimization method (ESM) [1]. The pose estimate $\mathbf{T}(\mathbf{x})$ is computed at each iteration and is updated incrementally as $\widehat{\mathbf{T}} \leftarrow \widehat{\mathbf{T}}\mathbf{T}(\mathbf{x})$ until convergence.

The parameter $\lambda$ is a constant that scales the relative error distributions. As mentioned in the introduction, many methods have been proposed to estimate this parameter ranging from manual tuning to estimation. Manually fixing $\lambda$ is not optimal nor efficient. Two efficient real-time possibilities include the ratio of the Median Absolute Deviations [19]: $\lambda = MAD(\mathbf{e}_I)/MAD(\mathbf{e}_G)$ and the adaptative lambda using the sigmoidal function [14], $\lambda(average(\mathbf{e}))$, which varies with the average distance between the two point clouds. As will be seen in the following section, $\lambda$ is not required if we consider a point-to-hyperplane approach.

## III. POINT-TO-HYPERPLANE METHOD

As mentioned in the introduction, the objective of this paper is to perform both closest point matching and minimization using a 4-vector containing color and depth. Since 4D space has an additional degree of freedom, the normal obtained for the 3D point-to-plane method will be orthogonal to a surface in 4D which spans both geometry and color. This surface will be referred in this paper as *hyperplane*. The 4-vector is defined as:

$$\mathbf{M}_i = \begin{bmatrix}\mathbf{P}_i & \mathbf{I}_i\end{bmatrix}^\top \in \mathbb{R}^4 \tag{4}$$

where the 3D Euclidean point $\mathbf{P}_i$ is fused with its associated greyscale intensity $\mathbf{I}_i$ in a single measurement vector.

Two measurement vectors, $\mathbf{M}_i^*$ and $\mathbf{M}_i$, obtained at different views of the same scene are generally not in correspondence. The fused error can then be defined as a 4D IRLS problem between two point clouds. If the hyper-normal is determined from the surface in 4D then it is possible to extend the optimisation which will be referred here as the *point-to-hyperplane* approach. The hybrid error function is then defined such as:

$$\begin{aligned}\mathbf{e}_{H_i} &= \rho_i\mathbf{N}_i^{*\top}(\mathbf{M}_i^* - \mathbf{M}_i^m) \in \mathbb{R}^4 \\ &= \rho_i\mathbf{N}_i^{*\top}\begin{pmatrix}\mathbf{P}_i^* - \Pi_3\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x})\overline{\mathbf{P}_i^m} \\ \mathbf{I}_i^* - \mathbf{I}_i\left(w(\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}), \mathbf{P}_i^*)\right)\end{pmatrix} \in \mathbb{R}^4\end{aligned} \tag{5}$$

where $\mathbf{M}_i^*$ is the reference 4D point, $\mathbf{M}_i^m$ corresponds to the warped closest points to image according to the unknown transformation $\hat{\mathbf{T}}\mathbf{T}(\mathbf{x})$. This is similar to equation 2 except that the normal is computed in 4D and the closest points can be determined in 4D. Also, note that the current measurements are all warped to the reference frame so that it is no longer necessary to rotate the normals.

Alternatively, it is also possible to find the closest points in the current 4-vector that solve the optimization problem as:

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

where $\mathbf{M}_w$ is the warped 4-vector. Solutions to both these cases will be considered but first lets consider the 4D normal.

In the point-to-hyperplane approach it is necessary to compute a 4D normal. Mathematically, at least four points in 4-space are needed to compute three vectors that will be used to perform a three-way cross product to obtains the orthogonal normal 4-vector. The cross product does not, however, account for the uncertainty of the points in its computation. Alternatively Principal Component Analysis (PCA) can be computed on covariance matrix of the nearest neighbours surrounding each point. In that case the smallest eigenvalue corresponds to the surface normal. Considering that $\mathbf{N}_i^m$ is directly obtained from $\mathbf{N}_i^*$, it is only necessary to compute the normals once for the reference image as in the case of the inverse compositional algorithm for image-based registration. The covariance matrix used in computing the normal 4-vector allows to project and weight the errors such that the point-to-hyperplane error is invariant to any tuning parameter $\lambda$.

As presented above in equation 5 and 6 it is possible to find the closest points either in the reference frame $\mathbf{M}_i^*$ (case 1) or the current frame $\mathbf{M}_i$ (case 2).

Consider first case 1. Computing in the reference frame has the advantage that several parameters can be precomputed only once and not at each iteration. In that case the reference normals can be precomputed along with a quick search strategy for finding closest points such as a kd-tree. In this paper the classic *k*d-tree with an ANN algorithm [15] is considered. The search function requires as inputs the current 4-vector $\mathbf{M}_i$ and a balanced *k*d-tree $\mathbf{k}^*$, which is created from the matrix $\boldsymbol{\mathcal{M}}^* = [\mathbf{M}_1^*\ \mathbf{M}_2^*\ \cdots\mathbf{M}_{mn}^*] \in \mathbb{R}^{4\times mn}$

that contains the reference 4D measurements. Each estimated matching point will be identified by an index, which is the result of a function that will be identified in this paper as: $\mathbf{M}_i^m = match(\mathbf{k}^*, \mathbf{M}_i)$, which gives the $i-th$ nearest neighbour for each current measurements vector $\mathbf{M}_i$ found in $\mathcal{M}^*$. Each node of the $k$d-tree contains a subset $B \subset \mathcal{M}^*$ of the reference dataset, which is divided at each level of the tree by the median of a different coordinate axis until reach an established number of elements for the end leafs. For this paper, the median of the so-called *buckets* $[B]_1, [B]_2, [B]_3, [B]_4$ are computed in that specific order to create the balanced tree. The operator $[\cdot]_j$ extract the $j-th$ row of the subset.

Consider now case 2. When finding the closest points in the current image it is not possible to recompute the kd-tree for each new image because it is computationally too expensive. In this case it is possible to consider approximating the closest point by simply search for the closest point in the image (as is done in equation 2). In this case nearest neighbour, bi-linear or bi-cubic interpolation can be performed [11], [12], [19], [20].

Both equations 5 and 6 can then be minimized iteratively as in 3.

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

Another strategy often used in the ICP literature (see for example [5], [7]), is to compute closest point matching only in the first iteration of the IRLS minimization loop. This allows to avoid to much computational complexity while obtaining the benefits of finding the closest points. In the first iteration, the transformation matrix $\mathbf{T}(\mathbf{x})$ is the identity matrix which simply means that the closest point is found by matching the current vector and the reference vector.

## IV. IMPLEMENTATION DETAILS

In this section, some parameters considered for the experiments will be established. The experiments were done for real and synthetic RGB-D greyscale images in MATLAB. The motivation for using synthetic data is that the generated images provide a groundtruth for evaluation, since the correspondences between the transformed views are known. A multi-resolution pyramid was used to improve the computational efficiency. The comparisons that will be presented in the following section were done at the second level of the pyramid (resolution $160 \times 120$).

There are two convergence criteria that stop the iterative loop for real and simulated experiments. The first one is a maximum number of iterations, which is established as 200, and the norm of the estimated rotation and translation. If the transformation matrix $\mathbf{T}(\mathbf{x})$ gets closer to the identity matrix, then the iterative loop stops. The parameters used to determine this second break is $norm(\mathbf{x}_R) < 1 \times 10^{-6}$ for rotation and $norm(\mathbf{x}_\mathbf{t}) < 1 \times 10^{-5}$ for translation. To estimate the closest points, the optimized search function of the FLANN library [15] is employed to find the true nearest neighbours.

To reject outliers, M-estimators were employed. They are more general because they permit the use of different minimization functions not necessarily corresponding to normally distributed data. In this paper, the Huber influence function was used for this purpose.

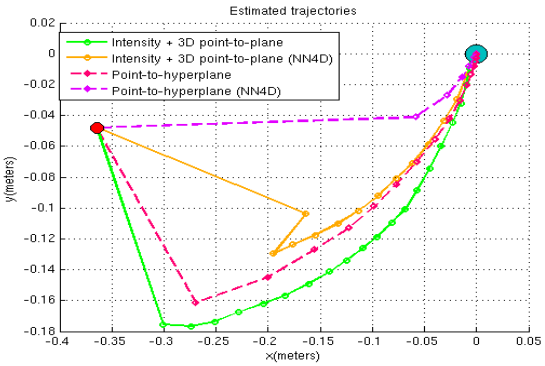



Fig. 3. Example of 4 estimated camera trajectories between a pair of images with a random pose in a simulated environment. The green and red dot indicate the initial and the final pose. Where the strategies 2, 3 & 4 improve the strategy 1, obtaining more direct trajectories. 1000 synthesized images with a random pose were equally tested, obtaining a similar performance.

The normal in 4 dimensions are adjusted to a $3 \times 3$ window that perform the PCA algorithm to find the covariance matrix of the fused error, leading to find the smallest eigenvalue which corresponds to the normal parameter.

All the experiments were validated on a workstation with Ubuntu 14.04, Intel Core i7-4770K and 16 GB RAM.

## V. RESULTS

An introduction to the improvement can be seen in Fig. 3, where the tracking trajectories estimated by 4 strategies are shown. These strategies are identified in this paper as follows:

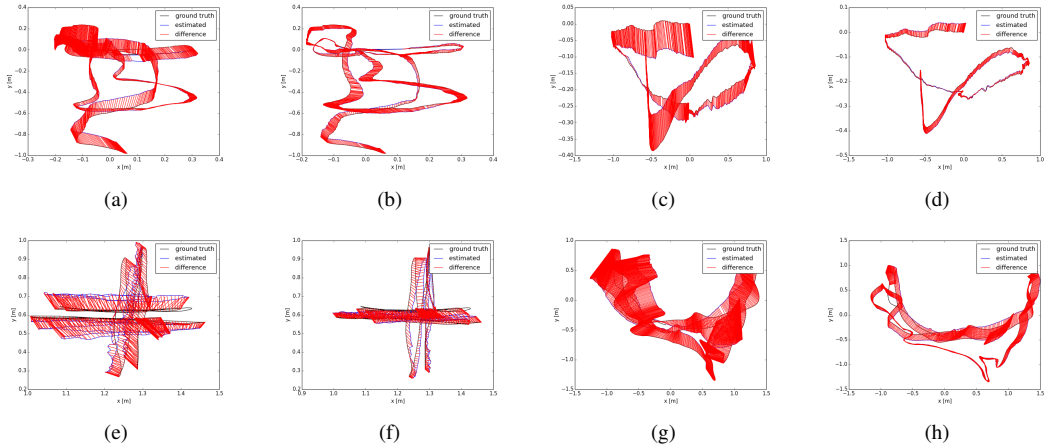
1) Direct approach + geometric point-to-plane
2) Direct approach + geometric point-to-plane (NN4D)
3) Point-to-hyperplane
4) Point-to-hyperplane (NN4D)

The legend NN4D means that the closest points were estimated in the first iteration only by finding the nearest neighbours with a $k$d-tree in the 4-vector. Bi-objective interpolation at each iteration was used for the method 1 & 3.

During the experimental part, it was seen that the parameter $\lambda$ , which is used in previous approaches, does not change the performance or accuracy of the pose estimation when the point-to-hyperplane approach is used. The normal vector is directly related to the covariance at the 4D points [16]. In Fig. 4, an example of a cost function is compared for the hybrid methods with and without the scale parameter (assigned as the normalization of the intensities), where real and simulated images with a random position between them can be equally employed to demonstrate it.

### A. Simulated environment

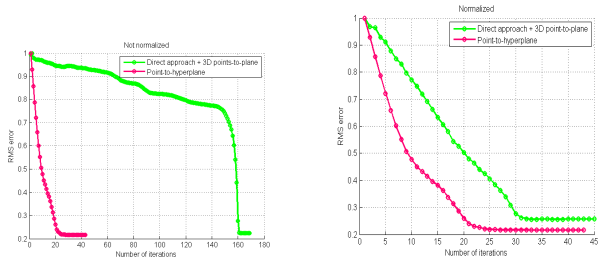
*1) Experiment 1 - Tracking:* The first experiment demonstrates the improvement in the convergence for synthesized images, where a random transformation is applied to the reference image. The new synthesized image is considered




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

(e) (f) (g) (h)

Fig. 2. Examples of the Absolute Trajectory Error evaluation, the first and third column show the results obtained by methods that combine the direct approach with the geometric point-to-plane method. The second and fourth column for the point-to-hyperplane method. The trajectories presented here are obtained for the simulated (a)(b) lvr/traj0, (c)(d) lvr/traj2, [6] and the real (e)(f) fr1/xyz, (g)(h) fr1/room [18] sequences.




(a) (b)

Fig. 4. Example of the cost function obtained after the alignment of two images of the same scene at different poses. When a $\lambda$ is provided (b) the performance of hybrid approaches is improved due that it scales the error functions. On the other hand, when the scale factor is not provided (a) the point-to-hyperplane keeps about the same performance. Similar results are obtained for synthetic or real images.

as the current image and the methods finds the alignment between each new image and the reference. The proposed methods are compared in computational time taken by the iterative loop and the number of iterations until convergence, where is already known that the transformation matrix is about the identity in the solution.

The averages shown in Table I demonstrates that the point-to-hyperplane method improve the convergence. However, the time shown does not consider the computation of normals or construction of the *k*d-tree. On the other hand, the matching in 4-vector space improves the hybrid methods by estimating more direct trajectories between the images.

TABLE I

AVERAGES IN TIME AND IN NUMBER OF ITERATIONS UNTIL CONVERGENCE FOR 1000 SYNTHESIZED IMAGES AT RANDOM POSES.

| Method | # Iterations | Time (sec) |
|---|---|---|
| 1) Intensity + point-to-plane | 65.6470 | 0.6076 |
| 2) Intensity + point-to-plane (NN4D) | 34.2320 | 0.3550 |
| 3) Point-to-hyperplane | 53.2410 | 0.5604 |
| 4) **Point-to-hyperplane (NN4D)** | **12.8330** | **0.1567** |

TABLE II

AVERAGES IN TIME AND IN NUMBER OF ITERATIONS FOR REAL IMAGES UNTIL CONVERGENCE.

| Method | # Iterations | Time (sec) |
|---|---|---|
| 1) Intensity + point-to-plane | 53.1489 | 0.6991 |
| 2) Intensity + point-to-plane (NN4D) | 49.8511 | 0.6662 |
| 3) Point-to-hyperplane | 37.1277 | 0.4338 |
| 4) **Point-to-hyperplane (NN4D)** | **34.5319** | **0.4136** |

*2) Experiment 2 - Visual odometry:* The methods were evaluated on the ICL-NUIM RGB-D benchmark dataset [6]. The benchmarck contains multiple datasets, where a virtual RGB-D sensor captures images in a synthesized living room. All data is compatible with the evaluation tools available for the TUM RGB-D dataset [18]. Therefore, in order to evaluate the estimated trajectories, the ATE (Absolute Trajectory Error) and RPE (Relative Pose Error) are compared alongside an accurate groundtruth trajectory in Table III.

## B. Real environment

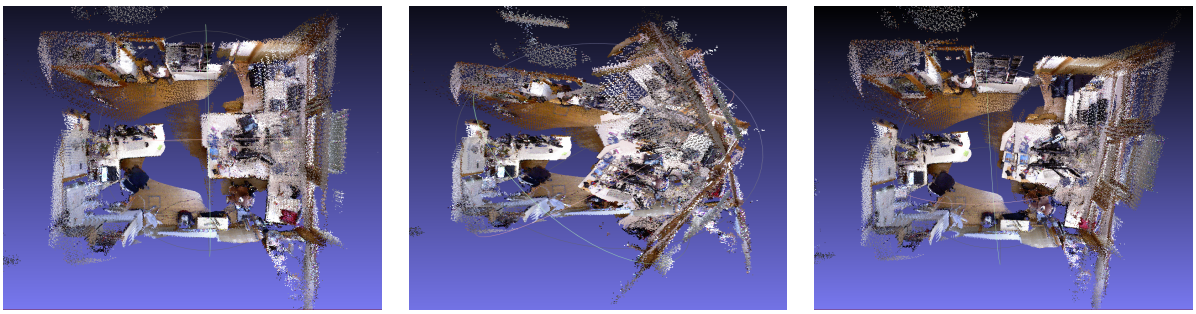
*1) Experiment 1: Improving the convergence domain:* The second experiment is carried on real images, it consists in moving the camera along one axis where each new image obtained generates a bigger error for each different position along the axis. RGB-D frames from the fr1/xyz TUM sequence [18] (images 740 - 790) were selected to verify the performance of the methods.

The first image was established as the reference image, and the tracking is performed by aligning the following images into the reference. Is demonstrated in Table II that the new approaches can get faster convergence even when the images are taken from different far positions, which aims to perform 3D reconstruction when keyframes are employed. However, since the proposed methods here does not include a keyframe detector, the methods could fails due to blurred images in real sequences, obtaining wrong pose estimations (As is shown in Fig. 5). In order to show the improvements in the convergence domain, the simulated trajectories are used

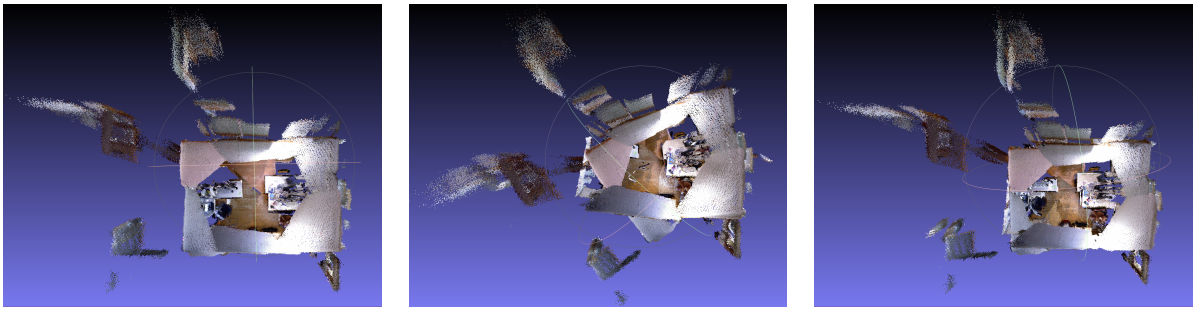
TABLE III

RELATIVE POSE ERROR (RPE) AND ABSOLUTE TRAJECTORY ERROR (ATE) FOR THE SIMULATED AND REAL DATASET [6], [18]. IT CAN BE SEEN THAT THE POINT-TO-HYPERPLANE METHODS (3 & 4) IMPROVE THE HYBRID METHODS THAT COMBINE THE DIRECT APPROACH AND THE GEOMETRIC POINT-TO-PLANE (1 & 2) IN THE MAJORITY OF DATASET FOR THE RPE TRANSLATIONAL EVALUATION AND IN ALL DATASET FOR RPE ROTATIONAL EVALUATION.

| Sequence | Method | RPE translational (m) | | | RPE rotational (deg) | | | ATE (m) | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | RMSE | MEAN | STD | RMSE | MEAN | STD | RMSE | MEAN | STD |
| fr1/xyz | 1 & 2 | 0.033 | 0.030 | 0.014 | 2.025 | 1.741 | 1.034 | 0.095 | 0.087 | 0.039 |
| | **3 & 4** | **0.021** | **0.019** | **0.008** | **1.106** | **0.998** | **0.477** | **0.045** | **0.038** | **0.024** |
| fr1/rpy | 1 & 2 | 0.062 | 0.050 | 0.037 | 3.161 | 2.887 | 1.288 | 0.136 | 0.115 | 0.072 |
| | **3 & 4** | **0.038** | **0.032** | **0.020** | **2.820** | **2.652** | **0.959** | **0.035** | **0.032** | **0.015** |
| fr1/360 | 1 & 2 | **0.146** | 0.118 | **0.086** | 4.171 | 3.844 | 1.621 | 0.520 | 0.484 | 0.188 |
| | **3 & 4** | 0.152 | **0.114** | 0.100 | **3.159** | **2.859** | **1.343** | **0.322** | **0.296** | **0.125** |
| fr1/room | 1 & 2 | 0.076 | 0.060 | 0.048 | 3.285 | 2.912 | 1.520 | 0.434 | 0.404 | 0.158 |
| | **3 & 4** | **0.056** | **0.047** | **0.030** | **2.673** | **2.329** | **1.313** | **0.174** | **0.152** | **0.086** |
| fr1/desk | 1 & 2 | 0.047 | 0.039 | 0.027 | 2.826 | 2.503 | 1.312 | 0.108 | 0.104 | 0.029 |
| | **3 & 4** | **0.044** | **0.036** | **0.025** | **2.309** | **2.027** | **1.106** | **0.071** | **0.067** | **0.023** |
| fr1/desk2 | 1 & 2 | **0.058** | **0.051** | **0.027** | 3.483 | 3.026 | 1.725 | 0.189 | 0.174 | 0.075 |
| | **3 & 4** | 0.060 | **0.051** | 0.031 | **3.026** | **2.641** | **1.478** | **0.133** | **0.116** | **0.065** |
| fr1/floor | 1 & 2 | 0.094 | **0.038** | 0.086 | 4.660 | 1.953 | 4.231 | 0.772 | 0.666 | 0.391 |
| | **3 & 4** | **0.080** | 0.051 | **0.062** | **3.909** | **1.915** | **3.408** | **0.473** | **0.405** | **0.244** |
| fr1/plant | 1 & 2 | 0.106 | 0.067 | 0.082 | 3.941 | 3.223 | 2.268 | 0.324 | 0.296 | 0.132 |
| | **3 & 4** | **0.055** | **0.043** | **0.034** | **2.130** | **1.947** | **0.864** | **0.101** | **0.093** | **0.037** |
| fr1/teddy | 1 & 2 | 0.096 | 0.081 | 0.051 | 3.410 | 3.021 | 1.583 | 0.615 | 0.553 | 0.271 |
| | **3 & 4** | **0.070** | **0.056** | **0.043** | **2.287** | **1.954** | **1.187** | **0.169** | **0.158** | **0.059** |
| lvr/traj0 | 1 & 2 | **0.001** | **0.001** | **0.001** | 0.044 | 0.035 | **0.027** | 0.128 | 0.114 | 0.057 |
| | **3 & 4** | 0.002 | **0.001** | 0.002 | **0.042** | **0.026** | 0.033 | **0.050** | **0.046** | **0.019** |
| lvr/traj1 | 1 & 2 | 0.002 | **0.001** | **0.001** | 0.048 | 0.041 | 0.024 | 0.114 | 0.104 | 0.046 |
| | **3 & 4** | **0.001** | **0.001** | **0.001** | **0.021** | **0.017** | **0.013** | **0.041** | **0.032** | **0.026** |
| lvr/traj2 | 1 & 2 | 0.002 | **0.001** | **0.001** | 0.044 | 0.039 | 0.021 | 0.074 | 0.067 | 0.030 |
| | **3 & 4** | **0.001** | **0.001** | **0.001** | **0.024** | **0.019** | **0.014** | **0.039** | **0.036** | **0.016** |
| lvr/traj3 | 1 & 2 | 0.002 | **0.001** | **0.001** | 0.070 | 0.053 | 0.045 | 0.218 | 0.202 | 0.082 |
| | **3 & 4** | **0.001** | **0.001** | **0.001** | **0.044** | **0.027** | **0.035** | **0.080** | **0.066** | **0.045** |



(a) fr1/room groundtruth (b) Intensity + 3D Point to plane (c) **Point-to-hyperplane**



(d) fr1/360 groundtruth (e) Intensity + 3D Point to plane (f) **Point-to-hyperplane**

Fig. 5. 3D reconstruction of sequences fr1/room and fr1/360 (first and second row, respectively). In first column the groundtruth obtained with an external motion capture system is shown, the column in the middle shown the results of the direct approach + 3D point to plane algorithm and the last column shown the result of the point-to-hyperplane method. It can be seen clearly that the proposed method can achieve more robust estimations. Therefore, the method could be improved with strategies as loop closure detection algorithms and keyframes detectors strategies.

instead for the video attachment in this paper.

*2) Experiment 2 - Visual odometry:* The tracking on full benchmark sequences from TUM [18] will compare the performance between the proposed methods. In this paper, only the dataset fr1 was considered to compare the performance frame-to-frame of the proposed methods. For both, simulated and real sequences, the online tool was used with the default settings to evaluate the ATE and RPE (Table III).

## VI. CONCLUSION

A novel point-to-hyperplane strategy was proposed based on a 4D vector which contains geometry and color. Two main advantages of this unified framework are underlined. First it is shown that the sensor pose can be estimated by minimizing the combined error without computing a scale parameter between them. Second, it is shown that nearest neighbour techniques can be used in 4-space to improve the convergence rate and domain for IRLS pose estimation. Experimental results and analysis are provided which compare two variants of the new point-to-hyperplane approach with the classic hybrid approach. The results show improved computation time and faster convergence on well known benchmarks. A demonstration of the performance of the alignment and the estimation of the camera poses (Fig. 6) can be seen in the video attachment of this paper.

Future work will be dedicated to testing the new approach in a simultaneously localization and mapping context which will provide for interesting comparisons with the map reconstruction. In this paper only the ANN algorithm was employed. In future works alternative and more efficient search strategies will be investigated.

## ACKNOWLEDGMENTS

Research presented in this paper is funded by CONACYT, Mexico, and CNRS-I3S, France. We would like to thank Dr. Maxime Meilland for his valuable comments and discussions.

## REFERENCES

[1] Selim Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 1, pages 943–948 vol.1, Sept 2004.

[2] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256, Feb 1992.

[3] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image Vision Comput.*, 10(3):145–155, April 1992.

[4] Andrew I Comport, Ezio Malis, and Patrick Rives. Accurate quadrifocal tracking for robust 3d visual odometry. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 40–45. IEEE, 2007.

[5] S. Druon, M.J. Aldon, and A. Crosnier. Color constrained icp for registration of large unstructured 3d color data sets. In *Information Acquisition, 2006 IEEE International Conference on*, pages 249–255, Aug 2006.

[6] A. Handa, T. Whelan, J. McDonald, and A.J. Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 1524–1531, May 2014.

[7] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. *Experimental Robotics: The 12th International Symposium on Experimental Robotics*, chapter RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments, pages 477–491. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

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

[9] Fernando I. Ireta and Andrew I. Comport. Direct matching for improving image-based registration. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2015.

[10] Andrew Edie Johnson and Sing Bing Kang. "registration and integration of textured 3d data ". *Image and Vision Computing*, 17(2):135 – 147, 1999.

[11] C. Kerl, J. Sturm, and D. Cremers. Dense visual slam for rgb-d cameras. In *Proc. of the Int. Conf. on Intelligent Robot Systems (IROS)*, 2013.

[12] 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, 3-8 November 2013. IEEE/RSJ.

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

[14] L. Morency and T. Darrell. Stereo tracking using icp and normal flow constraint. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 4, pages 367–372 vol.4, 2002.

[15] Marius Muja and David G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36, 2014.

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

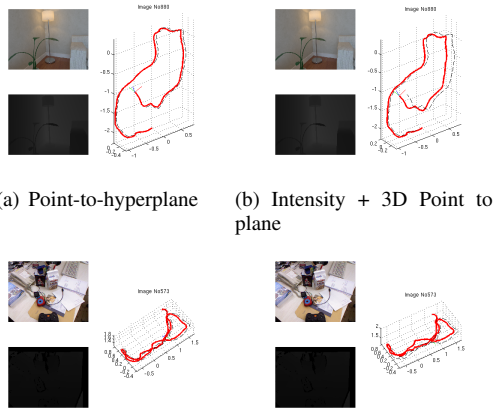
[17] F. Steinbruecker, C. Kerl, J. Sturm, and D. Cremers. Large-scale multi-resolution surface reconstruction from rgb-d sequences. Sydney, Australia, 2013.

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

[19] 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 6-13 2011.

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

[21] Qian-Yi Zhou and Vladlen Koltun. Color map optimization for 3d reconstruction with consumer depth cameras. *ACM Trans. Graph.*, 33(4):155:1–155:10, July 2014.



(a) Point-to-hyperplane (b) Intensity + 3D Point to plane

(c) Point-to-hyperplane (d) Intensity + 3D Point to plane

Fig. 6. Tracking simulator. (Attached video). The groundtruth is shown in black and the estimated trajectory in red for the Benchmark sequences.