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

Fernando I. Ireta Muñoz\textsuperscript{1} and Andrew I. Comport\textsuperscript{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 $\lambda$. The Point-to-hyperplane minimization avoids the estimation of $\lambda$, 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 $\lambda$ has been widely studied by the vision community and various strategies had been proposed. Depending on how it is estimated, the scale factor $\lambda$ can be categorized as an "adaptive" or a "non-adaptive" coefficient.

The non-adaptive category mostly involves strategies for dense 3D reconstruction from RGB-D images. The coefficient $\lambda$ 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 $\lambda$ 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 $\lambda$ for each new RGB-D frame.

The aim of this paper is to give the mathematical proof of the invariance to $\lambda$, 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 $\lambda$ 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 $\lambda$ 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 $\lambda$, 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 $\lambda$, 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 $\lambda$, 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.

\textsuperscript{*}This work is supported by the European H2020 project: COMANOID, Université Côte d’Azur, CNRS, I3S, France and CONACYT, México.

\textsuperscript{1}ireta@i3s.unice.fr \textsuperscript{2}Andrew.Comport@cnrs.fr
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 $M_i = [P_i^c, I_i]^T \in \mathbb{R}^4$, 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 $P_i = [X_i, Y_i, Z_i]^T \in \mathbb{R}^3$ which is computed by the back projection function such as: $P_i = K^{-1}p_iZ_i$, where $K \in \mathbb{R}^{3x3}$ is the intrinsic calibration matrix, $p_i = [p_{x_i}, p_{y_i}, 1]^T \in \mathbb{R}^3$ 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 $x$, an $i-th$ error metric can be defined as:

$$e_{H_i} = \lambda \left( M_i^e - f(M_i, x) \right) \in \mathbb{R}^4$$  \hspace{1cm} (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 $\lambda$ that scales the importance of the 3D geometric points with respect to the intensities such as:

$$\lambda = \begin{bmatrix} \lambda_G & 0 \\ 0 & \lambda_I \end{bmatrix} \hspace{1cm} (2)$$

where $\lambda_G = \text{diag}(\lambda_{G1}, \lambda_{G2}, \lambda_{G3})$.

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

$$x = - (J^T W J)^{-1} J^T W \left[ \lambda_G e_G \lambda_I e_I \right]$$  \hspace{1cm} (3)

where $J = [ J^T_G, J^T_I ]^T$ represents the stacked Jacobian matrices obtained by derivating the stacked geometric and photometric error functions ($e_G$ and $e_I$, respectively), and the weight matrix $W$ contains the stacked weights associated with each set of coordinates obtained by M-estimation [10]. For the purposes of this paper, the Jacobian $J_i$ 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 $T(x)$ is computed at each iteration and is updated incrementally as $T \leftarrow \hat{T} T(x)$ 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 $\lambda$, which has a huge influence on the minimization process. If it is well estimated, it can speed up the alignment between two 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 $\lambda$ [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:

$$e_{H_i} = \left( \lambda_G \left( N_i^e (P_i^m - P_i^w) \right) \right) \in \mathbb{R}^4 \hspace{1cm} (4)$$

with $\lambda_G = I_3$, where $P_i^m \in \mathbb{R}^3$ is the warped 3D point and $P_i^w$ is the warped intensity. The 3D correspondences (matches) with their associated intensities are respectively defined by $P_i^m$ and $I_i$. 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
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 $\lambda$ [16]) and (c) when an adaptive $\lambda$ 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 $kd$-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-hyperplane minimization can be defined such that:

$$e_{H_i} = N_i^\top \lambda (M_i^* - f(M_i, x)) \in \mathbb{R}^4$$  \hspace{1cm} (5)

where $N_i \in \mathbb{R}^4$ are the normals of the reference measurements and the scale parameter $\lambda = \text{diag} (\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ depends on 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:

$$e_{H_i} = N_i^\top (M_i^m - M_i^p) \in \mathbb{R}^4$$  \hspace{1cm} (6)

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

It should be noted that the tuning parameters $\lambda$ are not included in (6). This is due to the fact that the normals $N_i$ are estimated by performing the cross product between the neighbouring reference points that forms an hyperplane, so that the distance of another point $M_i^w$ to the formed hyperplane will be scaled by the geometric and photometric elements of $\lambda$, which have not influence since all scale elements appears for each element of the error function. The coefficient $\lambda$ 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 $\lambda$, consider that the measurements vectors $\lambda M^p$ and $\lambda M$ contain $j$ different types of measurements that are scaled by the same magnitude $\lambda$. The general form of the equation of a plane for 3D geometric points is $ax + by + cz + d = 0$, where $(x, y, z)$ are the coordinates of the 3D point and $(a, b, c)$ defines the normal vector. Therefore, an hyperplane of dimension $j$ can be defined as follows:

$$N_i^j \lambda_1 M_i^1 + N_i^2 \lambda_2 M_i^2 + \cdots + N_i^j \lambda_j M_i^j + d = 0$$  \hspace{1cm} (7)

where $d = -N_i^j \lambda_1 M_i^1 - N_i^2 \lambda_2 M_i^2 - \cdots - N_i^j \lambda_j M_i^j$. The normals are calculated by performing the cross product in $j$ dimensions such that:

$$N_i^j = V^{*1} \times V^{*2} \times \cdots \times V^{*j-1} \in \mathbb{R}^j$$  \hspace{1cm} (8)

where each $m = 1, 2, \cdots, j-1$ vector $V^{*m} = \lambda (M_i^k - M_i^l)$ 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:
\[ N_j^\ast = [c(\lambda_1)(M_{j-1})_1, c(\lambda_2)(M_{j-2})_2, \cdots, c(\lambda_j)(M_{k-j})_j]^T \]  
where the operator $[\cdot]_i$ extracts the $i$-th row of $(M_k^i - M_j^i) \in \mathbb{R}^j$, and the operator $c(\lambda_j)$ corresponds to the product of the elements of the diagonal of $\hat{\lambda}$ except for $\lambda_j$ such as:
\[ c(\lambda_j) = \prod_{i \neq j} diag(\lambda_i), \quad i = 1, 2, \cdots, j \]  
(10)
The equation to compute the distance $e_{Hj}$ of a point $\lambda M_i$ to the hyperplane, which is formed by the reference points $\lambda M_i^j$, can be represented in general form such as:
\[ e_{Hj} = \frac{[N_j^0]^T \lambda M_i + \cdots + [N_j^j]^T \lambda M_j + d}{\sqrt{([N_j^0]^T \lambda)^2 + \cdots + ([N_j^j]^T \lambda)^2}} \]  
(11)
where $d = -[N_j^0]^T \lambda M_i^0 - \cdots - [N_j^j]^T \lambda M_j^j$ and the operator $[N_j^j]^T \lambda M_{k-j}^j$ extracts the $j$-th element of (9). Equation (11) can be easily represented in Hessian normal form as:
\[ e_{Hj} = -\mathbf{N}^T j (\mathbf{M}_j - \mathbf{M}_i) \]  
(12)
where $\mathbf{N}_i^j$ is the normalization of the normal $j$-vector such as:
\[ \mathbf{N}_i^j = \left[ \frac{[M_{j-1}]}{\sqrt{([M_{j-1}]^T)^2 + \cdots + ([M_{k-j}]^T)^2}}, \cdots, \frac{[M_{k-j}]}{\sqrt{([M_{j-1}]^T)^2 + \cdots + ([M_{k-j}]^T)^2}} \right]^T \in \mathbb{R}^j \]  
(13)
that demonstrates the invariance to $\lambda$ 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 $e_{Hj}$ in $j$-th dimension is invariant to the relative scale $\lambda$ if it is minimized by a Point-to-hyperplane method.

**Proof:** Consider for simplicity the 3D case instead of 4D. Three hybrid 3D points that belong to the same cloud of points $\mathcal{M} \in \mathbb{R}^{3 \times N}$ (2D points + intensity) such as: $\mathbf{M}_0 = [X_0, Y_0, I_0]^T$, $\mathbf{M}_1 = [X_1, Y_1, I_1]^T$, and $\mathbf{M}_2 = [X_2, Y_2, I_2]^T$, and consider one warped point $\mathbf{M}_w = [X_w, Y_w, I_w]^T$ which represent any element of the warped point cloud $\mathcal{M}^w \equiv \Pi_3 \mathcal{M}$. 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: $\lambda \mathcal{M}$ and $\lambda \mathcal{M}^w$, where $\lambda$ is defined as $\lambda = diag(\lambda_\chi, \lambda_\alpha, \lambda_\beta)$. The coefficients of $\hat{\lambda}_j\mathcal{M}_i$ are introduced in a 3D point-to-hyperplane error function such that:
\[ e_{Hj} = \mathbf{N}_i^j \top \lambda (\mathbf{M}_j - f(\mathbf{M}_i, x)) \in \mathbb{R}^3 \]  
(14)
where the normals $\mathbf{N}_i^j$ are computed by performing the cross product of the vectors formed from $\lambda \mathcal{M}$. Considering the 3 hybrid points $\mathbf{M}_0$, $\mathbf{M}_1$ and $\mathbf{M}_2$ as reference points, the normal $\mathbf{N}_i^j = [N_{X_i} N_{Y_i} N_{I_i}]^T$ is defined here as the cross product between the vectors $\mathbf{V}^{01}$ and $\mathbf{V}^{02}$, which are defined as $\mathbf{V}^{01} = \lambda (\mathbf{M}_1 - \mathbf{M}_0)$ and $\mathbf{V}^{02} = \lambda (\mathbf{M}_2 - \mathbf{M}_0)$, that provides all the elements of the normal $\mathbf{N}_j^i$ at $\mathbf{M}_0$ such that:
\[ \mathbf{N}_j^i = \begin{bmatrix} \lambda_{\alpha} \lambda_{\chi} (\mathbf{V}^{01} \mathbf{V}^{02}) - \mathbf{V}^{01} \mathbf{V}^{02} \\ \lambda_{\alpha} \lambda_{\alpha} (\mathbf{V}^{01} \mathbf{V}^{02}) - \mathbf{V}^{01} \mathbf{V}^{02} \\ \lambda_{\alpha} \lambda_{\alpha} \mathbf{V}^{01} \mathbf{V}^{02} \end{bmatrix} = \begin{bmatrix} \lambda_{\alpha} \lambda_{\chi} N_{X_i} \\ \lambda_{\alpha} \lambda_{\alpha} N_{Y_i} \\ \lambda_{\alpha} \lambda_{\alpha} N_{I_i} \end{bmatrix} \]  
(15)
Equation (14) can be rewritten as follows:
\[ e_{Hj} = -\mathbf{N}_j^i \top (\mathbf{M}_j - \mathbf{M}_i) \]  
(17)
The unknown parameter $x$ is estimated by following the same pipeline of the hybrid-based methods, where (3) is rewritten as follows:
\[ x = -(J^T w J)^{-1} J^T W e_{Hj} \]  
IV. RESULTS
All the experiments presented in this paper were performed on both, real and synthetic, RGB-D grayscale images.

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 $1 \times 10^{-6}$ in rotation and $1 \times 10^{-5}$ 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].
\[ e_{Hj} = \lambda (\mathbf{R R}(x) \mathbf{N}_j^0)^\top (\mathbf{P}_i^T - \Pi_3 \mathbf{T}^T(x) \mathbf{P}_i^T) - \mathbf{I} \begin{bmatrix} \Pi_3 (\mathbf{T}(x), \mathbf{P}_i^T) \\ \mathbf{I} \end{bmatrix} \in \mathbb{R}^4 \]  
(19)
matrix of the 8 closest points for each point, where the
eigenvector with the lowest eigenvalue corresponds to the
normal vector $N_{8D}$. 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 $\lambda$, which are computed based on the strategies [16]
(The intensities are normalized $I_i = I_i/255$) and an adaptive
$\lambda$ [22] (where the scale parameter is the ratio between
the Median Absolute Deviation (MAD) of the errors $\lambda_G =
\text{MAD}(e_i)/\text{MAD}(e_G)$). The minimization of the error pre-
sented in (19) will also be compared with a $\lambda = 1$ ($\lambda$ is
not estimated).

1) Simulated environment: In order to verify the perform-
ance of the method, 1000 synthesized images were gen-
erated 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 I 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 $\lambda$ 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 $\lambda$ for hybrid-based approaches.
Here, the proposed method is compared with an adaptive $\lambda$
strategy that normalizes the intensities, 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.

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 $\lambda$ for hybrid-based approaches.
Here, the proposed method is compared with an adaptive $\lambda$
strategy that normalizes the intensities, 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>
<thead>
<tr>
<th>Method</th>
<th># Iterations</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Hybrid-based + non-adaptive $\lambda$ [16]</td>
<td>124,2280</td>
<td>1.5403</td>
</tr>
<tr>
<td>2) Hybrid-based + adaptive $\lambda$ [22]</td>
<td>152,3790</td>
<td>1.9019</td>
</tr>
<tr>
<td>3) Hybrid-based ($\lambda$ is not estimated)</td>
<td>155,0030</td>
<td>1.9406</td>
</tr>
<tr>
<td>4) Point-to-hyperplane</td>
<td>44,6280</td>
<td>0.4679</td>
</tr>
</tbody>
</table>

V. CONCLUSION

In this paper, it is proven mathematically that the Point-
to-hyperplane approach [12] is invariant to $\lambda$. The normals
have been obtained by performing the multidimensional cross
product of vectors in $j$ dimension and the $\lambda$ 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 com-
pute normals, such as color or IR measurements.

REFERENCES

putation of surface normals from range images. In IEEE International
Conference on Robotics and Automation, Shanghai, China, May 2011.
constraints in a vehicle environment. In Biennial on DSP for in-Vehicle
[3] Selim Benhimane and E. Malis. Real-time image-based tracking of
planes using efficient second-order minimization. In IEEE Interna-
tional Conference on Intelligent Robots and Systems, Sendai, Japan,
range images. In IEEE International Conference on Robotics and
Automation, Sacramento, CA, USA, Apr 1991.
3D/Color data sets with ICP. In IEEE International Conference on
<table>
<thead>
<tr>
<th>Sequence</th>
<th>Method</th>
<th>RPE translational (m)</th>
<th>RPE rotational (deg)</th>
<th>ATE (m)</th>
<th>AVERAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RMSE</td>
<td>MEAN</td>
<td>STD</td>
<td>RMSE</td>
</tr>
<tr>
<td>fr1/xyz</td>
<td>Adaptive λ [22]</td>
<td>0.033</td>
<td>0.030</td>
<td>0.014</td>
<td>2.025</td>
</tr>
<tr>
<td></td>
<td>Point-to-hyperplane</td>
<td>0.021</td>
<td>0.019</td>
<td>0.008</td>
<td>1.106</td>
</tr>
<tr>
<td>fr1/py</td>
<td>Adaptive λ [22]</td>
<td>0.062</td>
<td>0.050</td>
<td>0.037</td>
<td>3.161</td>
</tr>
<tr>
<td></td>
<td>Point-to-hyperplane</td>
<td>0.038</td>
<td>0.032</td>
<td>0.020</td>
<td>2.820</td>
</tr>
<tr>
<td>fr1/360</td>
<td>Adaptive λ [22]</td>
<td>0.146</td>
<td>0.118</td>
<td>0.086</td>
<td>4.171</td>
</tr>
<tr>
<td></td>
<td>Point-to-hyperplane</td>
<td>0.152</td>
<td>0.114</td>
<td>0.100</td>
<td>3.159</td>
</tr>
<tr>
<td>fr1/room</td>
<td>Adaptive λ [22]</td>
<td>0.076</td>
<td>0.060</td>
<td>0.048</td>
<td>3.285</td>
</tr>
<tr>
<td></td>
<td>Point-to-hyperplane</td>
<td>0.056</td>
<td>0.047</td>
<td>0.030</td>
<td>2.673</td>
</tr>
<tr>
<td>fr1/desk</td>
<td>Adaptive λ [22]</td>
<td>0.047</td>
<td>0.039</td>
<td>0.027</td>
<td>2.826</td>
</tr>
<tr>
<td></td>
<td>Point-to-hyperplane</td>
<td>0.054</td>
<td>0.046</td>
<td>0.035</td>
<td>2.507</td>
</tr>
<tr>
<td>fr1/desk2</td>
<td>Adaptive λ [22]</td>
<td>0.058</td>
<td>0.051</td>
<td>0.027</td>
<td>3.483</td>
</tr>
<tr>
<td></td>
<td>Point-to-hyperplane</td>
<td>0.060</td>
<td>0.051</td>
<td>0.031</td>
<td>3.026</td>
</tr>
<tr>
<td>fr1/door</td>
<td>Adaptive λ [22]</td>
<td>0.094</td>
<td>0.083</td>
<td>0.066</td>
<td>4.660</td>
</tr>
<tr>
<td></td>
<td>Point-to-hyperplane</td>
<td>0.080</td>
<td>0.051</td>
<td>0.062</td>
<td>3.909</td>
</tr>
<tr>
<td>fr1/plant</td>
<td>Adaptive λ [22]</td>
<td>0.106</td>
<td>0.067</td>
<td>0.082</td>
<td>3.941</td>
</tr>
<tr>
<td></td>
<td>Point-to-hyperplane</td>
<td>0.055</td>
<td>0.043</td>
<td>0.034</td>
<td>2.130</td>
</tr>
<tr>
<td>fr1/teddy</td>
<td>Adaptive λ [22]</td>
<td>0.096</td>
<td>0.081</td>
<td>0.051</td>
<td>3.410</td>
</tr>
<tr>
<td></td>
<td>Point-to-hyperplane</td>
<td>0.070</td>
<td>0.056</td>
<td>0.026</td>
<td>2.287</td>
</tr>
<tr>
<td>lvr/traj0</td>
<td>Adaptive λ [22]</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.044</td>
</tr>
<tr>
<td></td>
<td>Point-to-hyperplane</td>
<td>0.002</td>
<td>0.001</td>
<td>0.002</td>
<td>0.042</td>
</tr>
<tr>
<td>lvr/traj1</td>
<td>Adaptive λ [22]</td>
<td>0.002</td>
<td>0.001</td>
<td>0.001</td>
<td>0.048</td>
</tr>
<tr>
<td></td>
<td>Point-to-hyperplane</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.041</td>
</tr>
<tr>
<td>lvr/traj2</td>
<td>Adaptive λ [22]</td>
<td>0.002</td>
<td>0.001</td>
<td>0.001</td>
<td>0.044</td>
</tr>
<tr>
<td></td>
<td>Point-to-hyperplane</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.024</td>
</tr>
<tr>
<td>lvr/traj3</td>
<td>Adaptive λ [22]</td>
<td>0.002</td>
<td>0.001</td>
<td>0.001</td>
<td>0.070</td>
</tr>
</tbody>
</table>


