

HSM3D: feature-less global 6DOF scan-matching in the Hough/Radon domain

Andrea Censi, Stefano Carpin

Abstract—This paper presents HSM3D, an algorithm for global rigid 6DOF alignment of 3D point clouds. The algorithm works by projecting the two input sets into the Radon/Hough domain, whose properties allow to decompose the 6DOF search into a series of fast one-dimensional cross-correlations. No planes or other particular features must be present in the input data, and the algorithm is provably complete in the case of noise-free input. The algorithm has been experimentally validated on publicly available data sets.

I. INTRODUCTION

Finding the best alignment between two clouds of 3D points is a recurring problem in more than one discipline. Historically, it has been studied first in the vision community, and has become relevant in robotics only recently, with the availability of sensor configurations capable of generating dense 3D data. If a good first estimate of the unknown roto-translation is available, then the problem can be solved using “local” alignment methods, usually based on an iterative process. If such a first estimate is not available, then a different class of methods must be used. HSM3D belongs to this latter class of “global” methods.

A. Local alignment methods

Local iterative alignment algorithms usually work by establishing tentative correspondences between specific parts of the two inputs, followed by a minimization step that brings them together. The Iterated Closest/Corresponding Point (ICP) algorithm has been introduced over 15 years ago [1], [2] in the vision community and has seen numerous refinements [3]. In the robotics community, ICP and variants have been used since a few years for 2D mapping (e.g., [4]), and more recently for 3D (e.g., [5]). Some of the assumptions on the data in vision, e.g. very dense points clouds and small initial errors, do not always hold in robotics, therefore there is a wide margin for fruitful extensions. Another iterative alignment algorithm used in robotics is 3D-NDT [6], [7], [8], which is a generalization to 3D of the Normal Distributions Transform (NDT) [9].

These algorithms work very well for small initial errors, but can get trapped in local minima. The basic problem is that any procedure for establishing correspondences is inevitably sensitive to outliers. There are various heuristics which improve the robustness by correspondences rejection or reweighting, but still there are no guarantees on the basin of convergence. In practice, in structured environments, these methods tend to diverge when the initial angular error is sufficiently large.

A. Censi is with the Control & Dynamical Systems department, Division of Engineering and Applied Science, California Institute of Technology. Address: MC 107-81, 1200 E. California Blvd., 91125, Pasadena, CA (USA). E-mail: andrea@cds.caltech.edu

S. Carpin is with the School of Engineering, University of California. Address: 5200 North Lake Rd, 95348, Merced, CA (USA). E-mail: scarpin@ucmerced.edu

B. Global alignment methods

If the error in the initial roto-translation estimate cannot be handled by a local method, then a different class of algorithms must be used. We focus on methods that do not rely on the presence of specific features in the environment. In 3D, there are six parameters to estimate (three for representing the translation $t \in \mathbb{R}^3$, and three for the rotation $r \in SO(3)$) therefore any approach based on a brute-force search is infeasible. A popular line of attack is transforming the problem from the Euclidean space to another domain, where the six degrees of freedom can be partially decoupled. For example, Reyes *et al.* [10] show that the problem can be solved, by using Geometric Algebra, by a voting procedure in a certain extended space; unfortunately, for general motions, the cost is quadratic in the number of points in one scan, and therefore it is not practical.

Another strategy consists in decoupling rotation and translation by first finding a translation-invariant transformation of the input, and use this to recover the rotation. After the rotation has been recovered, finding the translation is relatively simple. This is the strategy employed by our algorithm, and also by some previous work. In Makadia *et al.* [11], the translation-invariant statistics is the Extended Gaussian Image (EGI) [12]. For each direction $s \in \mathbb{S}^2$, $EGI(s)$ is the fraction of surfaces having direction s . The rotation is recovered by computing the cross-correlation between the two EGIs, by using the Fourier Transform on $SO(3)$ [13]. The method proposed by [11] has two main limitations with respect to HSM3D: 1) It can only be used for smooth surfaces; and 2) It cannot deal with problems where the EGI is constant and therefore not informative, such as the case displayed in Fig. 1.

Global algorithms have a different set of challenges. If a voting/correlation procedure is used, then the computational complexity depends on the resolution of the solution, while methods based on correspondences such as ICP do not have this problem. Choosing the resolution is a trade-off also between precision and robustness, because large grids filter more noise in the data. Global methods are necessarily more complex because they must deal with the inherent ambiguity of the solution. For example, matching two range scan images of a corridor will most likely yield one hypothesis in which the corridor is matched upside-down. Therefore, there must a criterion for weighting the resulting hypotheses, but formulating such criterion is not trivial, because it must be robust to the possible partial overlapping of the two inputs.

C. Overview

HSM3D is designed to obtain a global matching with limited precision, with the idea that this estimate can be

refined later by a local algorithm such as ICP. HSM3D is the generalization of the algorithm presented in [14], which operated in 2D. The inputs to HSM3D are two sets of 3D points, possibly with associated orientation information. The output is a weighted list of roto-translation hypotheses. The overall steps are as follows:

- 1) The Hough/Radon transforms are computed for the two inputs.
- 2) The transforms are further reduced to obtain a translation invariant called “spectrum”, which is a function defined on the unit sphere.
- 3) Hypotheses for the rotation are obtained by matching the two spectra. An initial guess is obtained by matching local maxima in the two spectra, and the guess is refined by a uni-dimensional correlation.
- 4) For each rotation hypothesis, the corresponding translations are found by uni-dimensional correlations in the HT.
- 5) The hypotheses are weighted according to a similarity measure in the Euclidean space.

Section II introduces the notation, the Hough/Radon Transform and the relevant properties. HSM3D can use both *raw* 3D points, and *oriented* 3D points (i.e. points with a normal attached); these two cases correspond two slightly different definitions of the transform. The description of the algorithm is split in two sections. Section III focuses on the mathematics, and Section IV focuses on the implementations details, plus some particularizations for the case of range data. Finally, Section V reports experiments on publicly available data sets showing the different behaviors of HSM3D and ICP.

II. THE HOUGH TRANSFORM AND HOUGH SPECTRUM

This section introduces the notation and theory upon which HSM3D is based. The theory extends the 2D case described in [14], which might be an easier preliminary reading.

A. Notation

Let \mathbb{S}^2 be the 2-sphere, i.e. the set of vectors in \mathbb{R}^3 with unitary modulus. Points in \mathbb{S}^2 represent surface orientations. For example, the set of all oriented planes in \mathbb{R}^3 is isomorphic to $\mathbb{S}^2 \times \mathbb{R}$: each plane is identified by a certain surface orientation $s \in \mathbb{S}^2$ and a scalar $\rho \in \mathbb{R}$ which accounts for the translation from the origin. Let $\text{SO}(3)$ be the group of rotations in the 3D space. In the axis-angle representation we write a rotation $r \in \text{SO}(3)$ as $r = (a, \theta)$, with $a \in \mathbb{S}^2$ and $\theta \in \mathbb{S}^1$, to indicate a rotation of θ along axis a . Rotations are operators that can be applied to points in \mathbb{R}^3 and \mathbb{S}^2 . For $p \in \mathbb{R}^3$ this operation is indicated as $(a, \theta) \cdot p$. Evidently, $(a, \theta) \cdot a = a$ for all θ . Elements of \mathbb{R}^3 can be seen as either points in space or translation operators. For $t, p \in \mathbb{R}^3$, we write $t \cdot p$ to indicate the action (translation) of t on p , that is $t \cdot p = t + p$. A notation like $t \cdot (r \cdot p)$, with $r \in \text{SO}(3)$, is equivalent to a rotation of p followed by a translation. We use \circ as the function composition operator. With this notation, $t \cdot (r \cdot p) = (t \circ r) \cdot p$. We use the term “image” in a loose sense. A 2D image is a function from the image space to the intensity

space (\mathbb{R}^+ in this paper). A 3D image is a function from \mathbb{R}^3 to \mathbb{R}^+ . If the image i_2 is a roto-translated copy of i_1 , functional notation makes it easy to express that by $i_2 = i_1 \circ t \circ r$, which is equivalent to $i_2(p) = i_1(r \cdot p + t)$. We also use “oriented” 3D images, i.e., functions from $\mathbb{R}^3 \times \mathbb{S}^2$ to \mathbb{R}^+ .

B. The Hough/Radon Transform

We assume the input data to be a 3D image $i : \mathbb{R}^3 \rightarrow \mathbb{R}^+$. If the input data is a list of 3D points $\{p_j\}$, then i is the sum of the corresponding Dirac impulses: $i(x) = \sum_j \delta(p_j - x)$.

Definition 1: The Hough Transform maps a 3D image into a function defined on $\mathbb{S}^2 \times \mathbb{R}$:

$$\text{HT} : (\mathbb{R}^3 \rightarrow \mathbb{R}^+) \rightarrow (\mathbb{S}^2 \times \mathbb{R} \rightarrow \mathbb{R}^+)$$

Given a kernel $k : \mathbb{R} \rightarrow \mathbb{R}^+$, the value of the HT of i at point $(s, \rho) \in \mathbb{S}^2 \times \mathbb{R}$ is defined as:

$$\text{HT}[i](s, \rho) = \int_{\mathbb{R}^3} i(v)k(\langle s, v \rangle - \rho) dv \quad (1)$$

If k is the Dirac’s delta, then $\text{HT}[i](s, \rho)$ is the integral of the function i along the plane defined by (s, ρ) . The presence of a smooth kernel is useful when dealing with sets of discrete points, otherwise the HT would be zero for almost all values of (s, ρ) .

A note about nomenclature: strictly speaking, the Radon transform corresponds to the definition when the kernel k is the Dirac impulse. The Radon transform has a rich theory behind, mainly developed in the context of inverse reconstruction problems arising in medical imaging. We refer the reader to [15] for a comprehensive explanation. In the engineering literature, the Hough transform for lines/planes is usually understood to correspond to a discretization of the continuous Radon transform. If ρ is discretized with cells of size δ , we recover a formal definition of the discretized Hough transform by setting k to be 1 in $[-\delta/2, \delta/2]$ and 0 elsewhere. Calling “Hough Transform” the transform with an arbitrary kernel k is an overloading of terms which we do for simplicity.

C. The Hough/Radon Transform for oriented points

This second definition is used when the points clouds are dense enough to possibly estimate the orientation of the surface they lie on. This definition is non-standard and slightly more complicated than the previous one, so it can be omitted at a first read without compromising the understanding of the paper. Suppose the input data is a set of oriented points $(p_j, \alpha_j) \in \mathbb{R}^3 \times \mathbb{S}^2$, where α_j is the orientation of the surface at point p_j (think of it as an arrow attached to it). Then the input image i can be thought as a function from $\mathbb{R}^3 \times \mathbb{S}^2$ to \mathbb{R}^+ and defined as $i(p, \alpha) = \sum_j \delta(\|p - p_j\| + \|\alpha - \alpha_j\|)$. The definition of the HT is modified as:

Definition 2: The oriented Hough Transform maps an oriented 3D image into a function defined on $\mathbb{S}^2 \times \mathbb{R}$:

$$\text{HT} : (\mathbb{R}^3 \times \mathbb{S}^2 \rightarrow \mathbb{R}^+) \rightarrow (\mathbb{S}^2 \times \mathbb{R} \rightarrow \mathbb{R}^+)$$

Given two kernels $k_1, k_2 : \mathbb{R} \rightarrow \mathbb{R}^+$, the value of the HT of i at point $(s, \rho) \in \mathbb{S}^2 \times \mathbb{R}$ is defined as:

$$\text{HT}[i](s, \rho) = \int_{\mathbb{R}^3} \int_{\mathbb{S}^2} i(v, \alpha) k_1(\langle s, v \rangle - \rho) k_2(\langle s, \alpha \rangle) dv d\alpha \quad (2)$$

The formula is way more complicated than the basic idea: for an oriented point to contribute to the value of the HT at (s, ρ) , it needs not only to lie close (as defined by the kernel k_1) to the plane (s, ρ) , but also to have an orientation compatible (as defined by the kernel k_2) with that plane.

D. Properties

The following proposition are two properties of the HT which are relevant for HSM3D algorithm.

Proposition 1: Let $r \in \text{SO}(3)$ and $t \in \mathbb{R}^3$. Then:

$$\text{HT}[i \circ r](s, \rho) = \text{HT}[i](r \cdot s, \rho) \quad (3)$$

$$\text{HT}[i \circ t](s, \rho) = \text{HT}[i](s, \rho + \langle t, s \rangle) \quad (4)$$

The proofs are omitted; these are standard in the case of Definition 1 and easily derived for Definition 2.

Any other transform which satisfies (3) and (4) can be used in the algorithm in place of the HT.

E. The Hough spectrum

Next we define the Hough Spectrum, as a successive reduction of the HT to a function on the sphere, which is invariant to translations.

Definition 3: Let g be any translation-invariant functional that maps a function on the reals to \mathbb{R}^+ :

$$g : (\mathbb{R} \rightarrow \mathbb{R}^+) \rightarrow \mathbb{R}^+$$

An example of admissible g is $f \mapsto \|f\|_2$. Then the Hough Spectrum (HS) maps a 3D image to a function defined on the sphere:

$$\text{HS} : (\mathbb{R}^3 \rightarrow \mathbb{R}^+) \rightarrow (\mathbb{S}^2 \rightarrow \mathbb{R}^+)$$

and is defined as:

$$\text{HS}[i](s) = g[\text{HT}[i](s, \cdot)]$$

From the properties of the HT, the following two properties of the HS follow:

Proposition 2: Let $r \in \text{SO}(3)$ and $t \in \mathbb{R}^3$. Then the HS is invariant to translations of the input:

$$\text{HS}[i \circ t] = \text{HS}[i]$$

and is rotated by a rotation of the input:

$$\text{HS}[i \circ r] = \text{HS}[i] \circ r$$

III. HSM3D IN THEORY

Let us consider the following problem: given two images $i_1, i_2 : \mathbb{R}^3 \rightarrow \mathbb{R}^+$ such that, for some $r \in \text{SO}(3)$ and $t \in \mathbb{R}^3$,

$$i_1 = i_2 \circ t \circ r + \epsilon$$

where ϵ is a noise term, determine r and t .

HSM3D first recovers r by comparing the HS of the two images; then, after r is known, t can be recovered by uni-dimensional correlation of the HT.

EXAMPLES OF HOUGH SPECTRUM FOR NON-PLANE-STRUCTURED INPUTS

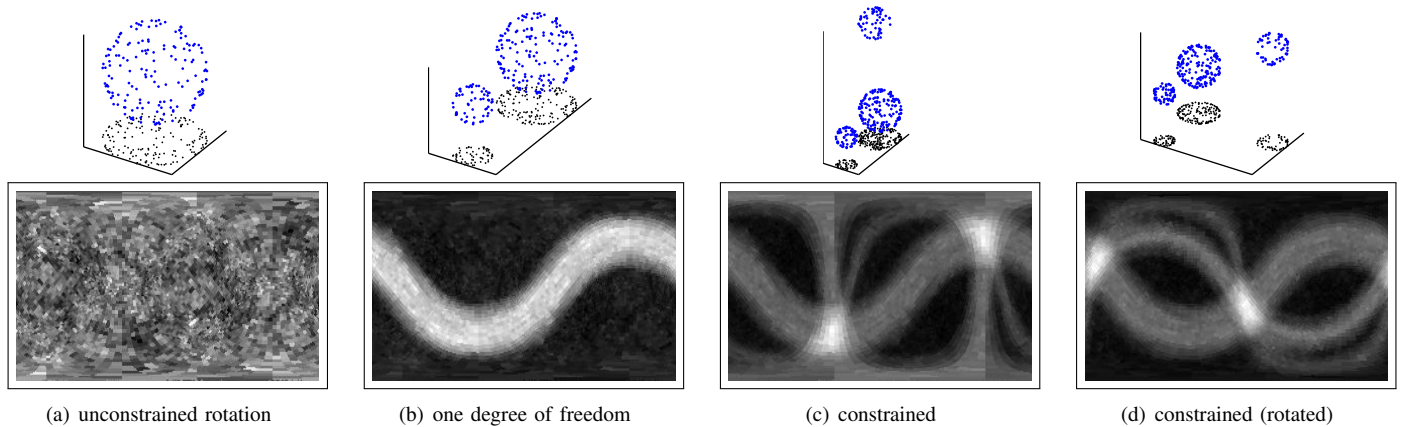


Figure 1. This figure shows that it is not necessary for the input to contain planes or flat features for HSM3D to work, and that it can work in situations where the Extended Gaussian Image (EGI) [12], used in [11], is constant and therefore not informative. On the top row some sample inputs are displayed, while the bottom row presents the corresponding HS, as projected to the cylinder oriented along the z axis. Note that while the spectrum is a function on the sphere the figure presents its projection on a cylinder for display purposes only (see section III for a more detailed description of such projection). (a): If the input is a perfect sphere, its HS is constant. In the case of random samples of a sphere it is just noise. No alignment is possible: the rotation is totally unconstrained. (b): If the input is a couple of spheres, then it is possible to constrain two dimensions of the rotation, as there is one axis of symmetry. The HS can shift in the direction of the belt. (c) If there are three non-aligned spheres, then the rotation is fully constrained. Note the peaks of the HS even though there are no planes in the input. Finally (d) shows the spectrum of a roto-translated input. Note that the EGI of (b), (c) and (d) would have been constant.

A. Recovering the rotation r

For recovering the rotation, start by computing the HS for both inputs, obtaining two functions $f_1 = \text{HS}[i_1]$ and $f_2 = \text{HS}[i_2]$. An example of how these functions might look like is shown in Fig. 1 (c) and (d). Proposition 2 implies that

$$f_1 = f_2 \circ r + \epsilon'$$

Now assume that by observing f_1 and f_2 , one can determine that a point m_1 on f_1 corresponds to a point m_2 on f_2 . One way to do this is by looking at the local maxima of the functions. If m_1 corresponds to m_2 , it then follows that

$$m_2 = r \cdot m_1$$

This relation provides only two constraints for r . In fact the solution r can be factorized as $r_2 \circ r_1$, where r_1 is a rotation that moves m_1 onto m_2 , and r_2 is a rotation that leaves m_2 fixed. Rotation r_1 can be written in the axis-angle form as

$$r_1 = (m_1 \wedge m_2, \arccos(\langle m_1, m_2 \rangle))$$

where $\langle m_1, m_2 \rangle$ is the usual scalar product. In the axis-angle representation the rotation r_2 that leaves m_2 fixed is simply

$$r_2 = (m_2, \theta)$$

for some θ yet to be determined.

The angle θ can be inferred by cross-correlation of the projections of f_1, f_2 onto certain cylinders. Let $\mathbb{K} = \mathbb{S}^1 \times (-1, 1)$ be the (truncated) cylinder. Our goal is now to define a correspondence between \mathbb{S}^2 and \mathbb{K} . By using cylindrical coordinates for \mathbb{K} and the natural embedding of \mathbb{S}^2 in \mathbb{R}^3 , the map Q can be written as:

$$\begin{aligned} Q : \mathbb{K} &\rightarrow \mathbb{S}^2 - \{(0, 0, \pm 1)\} \\ (\varphi, a) &\mapsto (\sqrt{1-a^2} \cos(\varphi), \sqrt{1-a^2} \sin(\varphi), a) \end{aligned}$$

Let e_3 be the unit vector identifying the z axis. Then a rotation around e_3 commutes with Q : $(e_3, \theta) \circ Q = Q \circ (e_3, \theta)$.

Next define the transformation $\Lambda : \mathbb{S}^2 \times \mathbb{S}^2 \rightarrow \text{SO}(3)$ such that, for $\langle u, v \rangle = 0$, $\Lambda(u, v)$ is the rotation that maps e_3 to u and e_1 to v (where e_1 is the unit vector that identifies the x axis). Notice this definition is well-posed because a rotation is uniquely identified by its action on two linearly independent vectors. Note also that for all $r \in \text{SO}(3)$,

$$r \circ \Lambda(u, v) = \Lambda(r \cdot u, r \cdot v)$$

$$\Lambda(u, v) \circ (e_3, \theta) = \Lambda(u, (u, \theta) \cdot v)$$

We then define c_1 and c_2 , two functions from the truncated cylinder to the positive reals, that are the projection of f_1 and f_2 to the cylinders aligned to m_1 and m_2 . Choose any q such that $\langle q, m_1 \rangle = 0$ and define $c_1, c_2 : \mathbb{K} \rightarrow \mathbb{R}^+$ as

$$c_1 \triangleq f_1 \circ \Lambda(m_1, q) \circ Q \quad (5)$$

$$c_2 \triangleq f_2 \circ \Lambda(m_2, r_1 \cdot q) \circ Q \quad (6)$$

This is the relation between c_1 and c_2 :

$$\begin{aligned} c_1 &= f_1 \circ \Lambda(m_1, q) \circ Q \\ &= f_2 \circ r_2 \circ r_1 \circ \Lambda(m_1, q) \circ Q \\ &= f_2 \circ r_2 \circ \Lambda(r_1 \cdot m_1, r_1 \cdot q) \circ Q \\ &= f_2 \circ (m_2, \theta) \circ \Lambda(m_2, r_1 \cdot q) \circ Q \\ &= f_2 \circ \Lambda(m_2, (m_2, \theta) \cdot r_1 \cdot q) \circ Q \\ &= f_2 \circ \Lambda(m_2, r_1 \cdot q) \circ (e_3, \theta) \circ Q \\ &= f_2 \circ \Lambda(m_2, r_1 \cdot q) \circ Q \circ (e_3, \theta) \\ &= c_2 \circ (e_3, \theta) \end{aligned}$$

That is, in coordinates, $c_1(\alpha, z) = c_2(\alpha + \theta, z)$. Hence θ can be recovered by cross-correlation of c_1 and c_2 . Notice that even though c_1 and c_2 are bi-dimensional, only a one-dimensional correlation is needed.

B. Further improvement for the rotation estimate

We have seen that the first rotation r_1 is obtained by matching m_1 and m_2 , and then we can find r_2 for compensating the rest of the rotation. In practice, however, some error will inevitably affect the first correspondence. This error stems from noise in the images, and because of the finite resolution of the buffer where the local maxima search is performed. The idea, then, is to do further adjustments of the alignment along other axes.

Given that we have just compensated along direction m_1 , a direction perpendicular to m_1 is chosen, then the other perpendicular direction, and so on, cycling through these three axis and repeating the previous procedure. Viewing the sphere from the ‘‘top’’ of m_1 , this amounts of locally roto-translating the images to obtain a better overlap¹.

In formulas, we seek (small) rotations r_3, r_4 , etc. This is the procedure to obtain the j -th rotation r_j . Let $r_{1:j-1} = r_{j-1} \circ \dots \circ r_1$. Let a_j be the axis we choose at this step, and let s_j be any support vector such that $\langle a_j, s_j \rangle = 0$. Then compute:

$$c_1^j \leftarrow f_1 \circ \Lambda(a_j, s_j) \circ Q \quad (7)$$

$$c_2^j \leftarrow f_2 \circ \Lambda(r_{1:j-1} \cdot a_j, r_{1:j-1} \cdot s_j) \circ Q \quad (8)$$

$$\hat{\theta} \leftarrow \arg \max_{\theta \in \mathbb{S}^1} \langle c_1^j, c_2^j \circ (e_3, \theta) \rangle \quad (9)$$

$$r_j \leftarrow (r_{1:j-1} \cdot a_j, \hat{\theta})$$

C. Alternative methods for recovering the rotation

Once one has obtained the two spectra, an alternative method to recover the rotation is the one detailed in [11]. The advantage is that it is more robust, because it considers all possible rotations, instead of relying of the heuristics of matching local maxima. The drawback is that it is slower: the computational complexity is $O(L^3 \log^2 L)$, where L is the number of harmonics considered. L is roughly proportional to the resolution of the solution. For HSM3D, the complexity can be computed as follows. The resolution of the solution is

¹Actually, that is exactly what is happening, given that the action of $\text{SO}(3)$ on \mathbb{S}^2 is locally isomorphic to the action of $\text{SE}(2)$ on \mathbb{R}^2 .

roughly equivalent to the resolution R of the grid representing the function on the sphere. Looking for the minima takes $O(R^2)$ and $O(R^2)$ is also the cost for computing of the cylindrical images c_1 and c_2 according to (5)-(6). The circular cross-correlation of $O(R) \times O(R)$ cylindrical images along *one* direction can be done in $O(R) \times O(R \log R)$ using the Fast Fourier Transform. The cost of successive refinements is negligible with respect to the first correlation. Overall, the cost is $O(kR^2 \log R)$ where k is the number of matchings. The precise value of k depends on the complexity of the input.

D. Recovering the translation

Let us suppose now that the rotation r is known. Choose a direction $\bar{s} \in \mathbb{S}^2$. Define the two functions $h_1^{\bar{s}}, h_2^{\bar{s}} : \mathbb{R} \rightarrow \mathbb{R}^+$ as

$$\begin{aligned} h_1^{\bar{s}}(\rho) &= \text{HT}[i_1](\bar{s}, \rho) \\ h_2^{\bar{s}}(\rho) &= \text{HT}[i_2](r \cdot \bar{s}, \rho) \end{aligned}$$

These are uni-dimensional slices of the HT. From Proposition 1 it follows that

$$\begin{aligned} h_1^{\bar{s}}(\rho) &= \text{HT}[i_1](\bar{s}, \rho) \\ &= \text{HT}[i_2 \circ t \circ r](\bar{s}, \rho) \\ &= \text{HT}[i_2 \circ t](r \cdot \bar{s}, \rho) \\ &= \text{HT}[i_2](r \cdot \bar{s}, \rho + \langle r \cdot \bar{s}, t \rangle) \\ &= h_2^{\bar{s}}(\rho + \langle r \cdot \bar{s}, t \rangle) \end{aligned}$$

Therefore, by cross-correlation of $h_1^{\bar{s}}$ and $h_2^{\bar{s}}$ one can get an estimate $\delta^{\bar{s}}$ of $\langle r \cdot \bar{s}, t \rangle$. By repeating this over at least three linear independent directions $\bar{s} = \{\bar{s}_1, \bar{s}_2, \bar{s}_3, \dots\}$ it is possible to obtain an estimate of t by linear least squares:

$$\begin{bmatrix} (r \cdot \bar{s}_1)^T \\ (r \cdot \bar{s}_2)^T \\ \vdots \end{bmatrix} t = \begin{bmatrix} \delta^{\bar{s}_1} \\ \delta^{\bar{s}_2} \\ \vdots \end{bmatrix}$$

Evidently, when doing the correlation of $h_1^{\bar{s}}$ and $h_2^{\bar{s}}$ there will be more than one local maxima which should be considered if one expects ambiguous data. In general, there will be more than one translation hypothesis for each rotation.

E. Weighting the hypotheses

If one assumes the data generation model is $i_1 = i_2 \circ t \circ r + \epsilon$, that is, there is additive noise, then to evaluate the quality of an hypothesis $\langle \hat{r}, \hat{t} \rangle$, it suffices to evaluate the dot product $\langle i_1, i_2 \circ \hat{t} \circ \hat{r} \rangle$ in the image space, or the dot product in the Hough domain. This works as a generic mechanism, but there are cases, such as the scan-matching scenario, in which the “noise” is mainly the fact that there is only partial overlap between scans. Moreover, the fact that input data are coherent scan views gives much more structure to the problem that can be exploited to define a better measure than the dot product. In general, the assessment of the solution quality must be related to the data generation model.

F. Completeness

This algorithm is complete if there is no noise on the inputs, in the sense that it reports the correct solution, along with all symmetries, if any. To see why, note that, if there is no noise, the spectra are exactly two rotated copies of each other. Hence, by considering enough local maxima, eventually one correct pairing will be determined. If the first match between m_1 and m_2 is correct, the rest of HSM3D finds the correct solution because of the properties of the cross-correlation operator: if two signals are identical up to a delay, that delay is the absolute maximum of the cross-correlation (up to symmetries and periodicity).

IV. HSM3D IN PRACTICE

After having presented the general ideas underlying the HSM3D algorithm, we now discuss its actual implementation, along with some tweaks that are advantageous in the particular case of range-finder data. Obviously, not all details can be covered in this section. The source code for HSM3D is available in Matlab at <http://purl.org/censi/2008/hsm3d> and in C++ at <http://robotics.ucmerced.edu/Software>.

A. Representing functions on the sphere

Since the HS maps 3D images into functions defined on the unit sphere, a suitable representation for these functions is needed. This can be achieved by putting into correspondence cells in a buffer to points on the sphere. More precisely, each cell in the buffer shall correspond to a small patch on the surface of the unit sphere. The requirement is that the size of the patches is somehow regular. The value stored in a buffer cell corresponds to an approximated value of the function in the associated patch. A tradeoff between efficiency and ease of implementation guided our design choice. Specifically, we project the unit sphere into the smallest cube that contains the sphere, using the diffeomorphism

$$\begin{aligned} \varphi : \mathbb{S}^2 &\rightarrow \text{cube} \\ s &\mapsto s / \|s\|_\infty \end{aligned}$$

and we then use a regular lattice on each of the six faces of the cube. Fig. 2 shows the resulting map.

B. Computing the HT/HS

To compute the HT, set up a three-dimensional buffer (a lattice on $\mathbb{S}^2 \times [\rho_{\min}, \rho_{\max}]$) that represents a sampled version of (1). The HT can be computed by cycling through each discretized orientation s and 3D point in the input, with a complexity which is linear in the number of points, quadratic in the angular resolution, and independent of the ρ resolution. However, if the input image has been produced by dense laser scans, this computation can be greatly sped up by estimating the surface normals and using the oriented HT. Then, the kernel k_2 can be chosen such that each oriented point (p, α) influences only one discretized orientation s (the closest to α). Apart from employing a different definition for the HT, the rest of the algorithm remains the same as described in the previous section. We treat each sensor point as a small surface

patch. The orientation s of the patch is given by the extracted normal. The area A of the patch is estimated by the distance to the neighboring points in the 3D scan. The computation of the HT is greatly simplified because only the entry for (s, ρ) is increased by A , therefore there is no costly loop for iterating over all possible iterations. In practice, for 3D scans containing about 20 000 points, using the alternative definition cuts the time for creating the HT of various orders of magnitude and yields a computational time of fractions of a second.

C. Finding the maxima of the spectra

After the HS has been computed, its local maxima can be easily found by inspection of the local maxima on the buffer cells associated with the cube. Correspondences could be sought by a brute force pairing of all maxima, but this approach would be computationally too demanding. For this reason only a small subset of the local maxima are considered for mutual correspondences. In particular, we limit the search to the first n_1 maxima of f_1 and the first n_2 maxima of f_2 . The choice for the values of these parameters is once again driven by a tradeoff between accuracy and speed (in the experiments, $n_1 = 2$ and $n_2 = 6$).

D. The least-square step

The directions for the least-squares estimations of t are chosen to be the local maxima of the HS as they are those for which t is well constrained. At least 3 are needed and we chose up to 8, as estimating the translation turns out to be computationally cheap with respect to estimating the rotation. The LS can arrange a weight for each observation; we choose this to be the value of the correlation between h_1^s and h_2^s .

E. Evaluating the quality of one hypothesis

As outlined in section III, HSM3D does not produce a single solution, but rather a number of possible solutions. This proliferation derives from the various ways that maxima in f_1 can be matched to maxima in f_2 , and from the multiple local maxima emerging when computing the translations associated with a given rotation. For this reason, a method to evaluate the quality of this set of potential solutions is needed. The idea shortly mentioned in section III-E is not robust enough in the

general case where point clouds not completely overlapping have to be matched. We here describe an approach appropriate for the case of a robot equipped with a three dimensional range scanner. To evaluate the quality of a roto-translation guess, points in one scan are put in the same reference frame as the other scan according to the guess. Each point is then compared with the closest sensor reading. If it is coherent (according to a threshold), it scores as one point in the metric. This is a very crude approximation to the computation of a particularly relaxed log-likelihood; it has to be relaxed because we expect the roto-translations to be affected by moderate error. In order to expedite the ranking, the consistency check is not performed for all points, but rather for a randomly chosen subset. The size of the random set is subject to an obvious tradeoff between accuracy and speed. Eventually, solutions are ranked accordingly to the number of consistencies found in the random set. Before concluding this section it is worth outlining that the presence of multiple hypotheses to be ranked is not necessarily a nuisance. On the contrary, in highly symmetric environments where multiple transformations may consistently relate two successive scans, the ability to identify a set of roughly equivalent solutions is a plus, because the algorithm is not required to commit too early to a single model explaining the data gathered by the sensors.

V. EXPERIMENTS

In order to evaluate the performance of HSM3D, we have performed a number of experiments with 3D scans from three dimensional laser range finders. In the experiments described in the following we used the public dataset *dat_avz* available from <http://kos.informatik.uni-osnabrueck.de/3Dscans/>². Each scan features about 21000 points, and was taken while exploring the interior of a building. Results presented in this section were obtained with the C++ implementation of the HSM3D algorithm. Due to the still exploratory stage of this research, parameters were not fine tuned, and the current implementation of the algorithm favors clarity rather than performance. Just to give the reader a rough idea about the current computational

²The data set was recorded by Andreas Nüchter from the University of Osnabrück.

REPRESENTING FUNCTIONS ON THE SPHERE

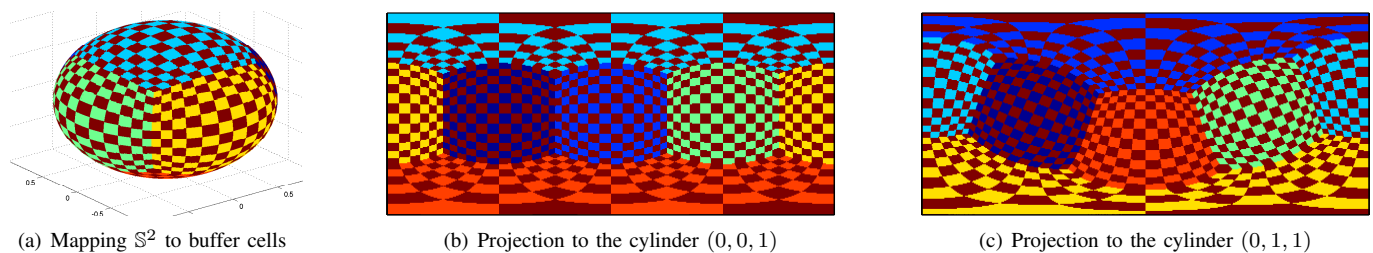


Figure 2. To implement HSM3D, it is necessary to establish correspondences between patches on the sphere with cells in a buffer. We chose to map the sphere to the cube that contains it, and map a simple buffer for each cube face. Fig. (a) shows the resulting patches on the sphere. The other needed operation is mapping the sphere to a cylinder wrapped around it. In (b), the sphere is mapped to the cylinder with orientation along the z axis. In (c), it is mapped to a rotated cylinder.

cost of HSM3D, on a 2.13GHz Pentium the average execution time is slightly above 3 seconds.

A. Global versus local performance: a comparison with ICP

The goal of this experiment is to show the ability of HSM3D to provide an appropriate estimate of the roto-translation independently from the magnitude of the involved translations and rotations. The setup is as follows: image i_1 is generated from a random 3D scan extracted from the dataset, and image i_2 is obtained by applying a random rotation r and a random translation t to i_1 , i.e. $i_2 = i_1 \circ t \circ r$. To be precise, 12 different angles are considered, ranging from 15 to 180 degrees with increases of 15 degrees. For every angle, 40 random axes are generated. Translations are instead generated using a uniform distribution over the interval $[-0.5, 0.5]$ along the x , y and z axis. The data are then processed by HSM3D and ICP in order to retrieve an estimate for r and t . For comparison with ICP, we used the ICP implementation provided by the Insight Toolkit [16]. In all experiments, ICP’s starting point for the alignment process is set to the identity transformation. Figure 3(a) illustrates the results we obtained with ICP. As expected, the error affecting its solutions gets progressively worse as the rotation angle increases, i.e. as the identity transformation becomes a much less accurate starting point for the minimization process. This trend is consistent with the expectations, since ICP implements a gradient descent approach whose final outcome depends on the goodness of the starting point. For the specific data set used in the experiment, ICP’s performance breaks down for random rotations of 75 degrees. While this specific value for the cutting point is ultimately dataset and problem dependent, its existence is instead a general fact. On the other hand, as outlined in Fig. 3, HSM3D’s performance exhibits instead a consistent trend, thus substantiating our initial claim concerning the algorithm’s ability to perform global scan matching. In order to better understand the charts describing its performance, and the few outliers occasionally appearing therein, the reader should keep in mind that we have considered only the first solution according to the ranking formerly described. Due to the random sampling involved in the ranking process, the existence of symmetries and under-constrained situations in the data, and the relative precision of HSM3D, the best ranked solution is not necessarily the “correct” one. For the cases with large error, a solution closer to the ground truth is usually present with a lower ranking.

B. Sensitivity to noise

The goal of this experiment is to evaluate how HSM3D’s performance degrades with increasing noise affecting the input data. The setup is similar to the previous one, i.e. a 3D image i_2 is obtained from i_1 through the application of random rotations and translations. However, before applying these transformations, i_1 is in this case corrupted by Gaussian noise with zero mean and increasing variance. More precisely, random noise is added to the range-finder readings, while the directions of the beams are not affected. This artificial

noise sums with the natural noise contained in the data. In particular the reader should observe that increased noise in the data will eventually invalidate the assumptions we made in Section IV-B, i.e. that normals can be directly extracted from the range scanner data. Table ?? shows the results, for a specific rotation angle, namely 15 degrees (we can consider only one angle as the previous experiment showed that HSM3D is insensitive to the magnitude of the roto-translation), for increasing noise variance. 5000 runs were considered for each combination of angle and variance. The table shows the percentage of “success” cases, defined as those for which the recovered rotation had a rotation error smaller or equal than 5 degrees. The experiments confirm that HSM3D with the oriented Hough transform is rather robust to a high level of noise in the range readings.

σ	0	0.01	0.02	0.04	0.06	0.08	0.1	0.12
%	96.2	95.7	94.5	95	90	91.5	91.5	90

Table I
PERCENTAGE OF SOLUTIONS WITH A ROTATION ERROR SMALLER THAN 5 DEGREES AS A FUNCTION OF THE STANDARD DEVIATION (σ) OF THE ARTIFICIAL NOISE.

VI. CONCLUSIONS AND FUTURE WORK

We have presented HSM3D, an algorithm that decomposes a global 6DOF search into a series of uni-dimensional correlations. After having projected the input data sets in to the Hough domain, the algorithm sequentially recovers the rotation and then the translation. The most appealing characteristic of HSM3D resides in its ability to efficiently perform a global search, a quality not found in formerly presented solutions like ICP and 3D-NDT. When the two inputs correspond perfectly it is possible to prove that the algorithm returns the true solution along with symmetries, if any. The completeness of the actual algorithm depends on the amount of noise in the data and on the resolutions chosen for computing the transform. This analysis is part of future work. Moreover it is necessary to better understand how to choose two important parameters: the angular (s) and linear (ρ) resolution of the Hough transform. Coarser grids give more robustness/speed, while finer grids offer more precision. The gain/loss ratio is still to be investigated and is most likely problem dependent to some extent. For the experiments in this paper, little care was devoted to fine tuning these parameters. Future work includes coming up with a rational way of choosing these two numbers and other parameters like for example the number of hypothesis to track. As of now, the algorithm is under-engineered, in the sense that it works in cases like 3D range-finder data and cases such as the inputs pictured in Fig. 1. That is because we have been trying to understand the wide applicability before specializing it to particular tasks. A different domain in which we will try it is the case of stereo-vision, where one can hopefully exploit HSM3D’s tolerance to noise. HSM3D is currently implemented in a very naive way, and it produces hundreds of possible hypotheses (say a dozen rotations, and

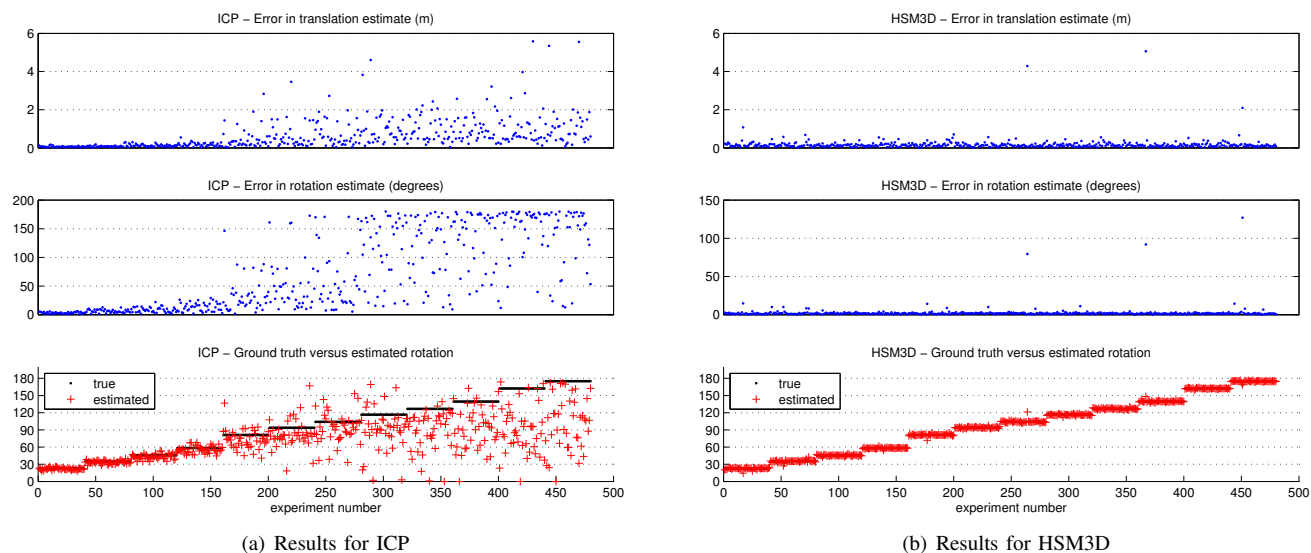


Figure 3. (a) The chart illustrates the accuracy of ICP for random transformations (a, θ) where the axis a is randomly sampled and angle θ is progressively increased from 15 to 180 degrees with steps of 15 degrees. The top panel displays the error of the recovered translation, while the middle panel displays error in the recovered rotation. The bottom panel contrasts the absolute rotation estimations provided by ICP with the ground truth, and clearly shows the increasing trend in the angle θ . (b) This chart outlines the global aspect of HSM3D. The errors in the recovered translation and rotation do not increase when the random rotation gets larger. The reader should notice that the plot displays only the result provided by the transformation with the highest ranking.

a dozen translations for each rotation) even though many of them could be discarded in an early phase. The main operation, cross-correlation, can be also implemented much more efficiently by using the Fast Fourier Transform. There are many possible improvements in the particular case of 3D robotic mapping with range-finders and many improvements may be obtained including domain specific knowledge in the search. For example, when the robot is in a corridor, HSM3D routinely produces not only the right solution, but also three solutions that correspond to the symmetries of the environment: one with a flip around the z axis, one with a flip around the x axis, and one with the double flip. However, in most practical applications, the robot is very unlikely to flip upside-down and this prior information can be incorporated in the algorithm to narrow the search domain.

Acknowledgements. Thanks to Li Na for tips on integration.

REFERENCES

- [1] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *Proceedings of the IEEE International conference on Robotics and Automation (ICRA)*, (Sacramento, CA, USA), pp. 2724–2729, Apr. 1991.
- [2] P. Besl and N. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.
- [3] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, June 2001.
- [4] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, 1997.
- [5] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann, "6D SLAM — 3D mapping outdoor environments," *Journal of Field Robotics*, vol. 24, no. 8-9, pp. 699–722, 2007.
- [6] E. Takeuchi and T. Tsubouchi, "A 3-D scan matching using improved 3-D Normal Distributions Transform for mobile robotic mapping," *Proceedings of the IEEE/RSJ International conference on Intelligent Robots and Systems (IROS)*, pp. 3068–3073, Oct. 2006.
- [7] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," *Journal of Field Robotics*, vol. 24, no. 10, pp. 803–827, 2007.
- [8] B. Huhle, M. Magnusson, W. Strasser, and A. Lilienthal, "Registration of colored 3D point clouds with a kernel-based extension to the Normal Distributions Transform," *Proceedings of the IEEE International conference on Robotics and Automation (ICRA)*, pp. 4025–4030, May 2008.
- [9] P. Biber and W. Strasser, "The Normal Distributions Transform: a new approach to laser scan matching," *Proceedings of the IEEE/RSJ International conference on Intelligent Robots and Systems (IROS)*, vol. 3, pp. 2743–2748 vol.3, Oct. 2003.
- [10] L. Reyes, G. Medioni, and E. Bayro, "Registration of 3D points using geometric algebra and tensor voting," *Int. J. Comput. Vision*, vol. 75, no. 3, pp. 351–369, 2007.
- [11] A. Makadia, A. Patterson, and K. Daniilidis, "Fully automatic registration of 3D point clouds," in *Computer Vision and Pattern Recognition*, 2006.
- [12] B. K. P. Horn, "Extended Gaussian Images," *Proceedings of the IEEE*, vol. 72, no. 2, pp. 1671–1686, 1984.
- [13] G. S. Chirikjian and A. B. Kyatkin, *Engineering Applications of Non-commutative Harmonic Analysis*. Boca Raton, FL: CRC Press, 2001.
- [14] A. Censi, L. Iocchi, and G. Grisetti, "Scan matching in the Hough domain," in *Proceedings of the IEEE International conference on Robotics and Automation (ICRA)*, (Barcelona, Spain), pp. 2739–2744, 2005.
- [15] L. Debnath and D. Bhatta, *Integral Transforms and Their Applications*. Chapman & All/CRC, 2007.
- [16] L. Ibanez, W. Schroeder, L. Ng, and J. Cates, *The ITK software guide*. Kitware Inc., 2005.