An experimental assessment of the HSM3D algorithm for sparse and colored data

Stefano Carpin, Andrea Censi

***Abstract*— We recently introduced HSM3D, an algorithm to solve the six dimensional scan-matching problem without relying on features in the input, and whose solution does not depend on initial guesses. Building upon these new findings, in this manuscript we present a more detailed experimental study of the algorithm we proposed. In particular, we show how to improve the algorithm's performance also when matching point clouds produced by stereo cameras, given that this kind of input invalidates some of the assumptions we formerly identified in order to accelerate HSM3D's performance. We also show that by incorporating color information into the the algorithm it is possible to reduce the number of sporadic outliers in the solution set, thus providing a more reliable algorithm.**

I. Introduction

The increased availability of sensors capable of producing three dimensional point clouds has invigorated the interest in algorithms capable of estimating six dimensional rigid transformations providing a satisfactory match between two partially overlapping data sets. This problem is relevant in a variety of domains and not only in robotics. The robotics scenario is however challenging for a few reasons. Sensors mounted on mobile robots often deliver noisy data sets, thus offering little experimental evidence to validate or reject possible solutions. Moreover, sensors mounted on mobile robots change their position over time, and the inherent uncertainty of the robot's pose is transmitted to the sensor location and orientation. Finally, the algorithm merging two point clouds is commonly found inside more articulated control loops, thus asking for utmost speed while searching for a solution. Starting from these considerations, we recently developed HSM3D (Hough Scan Matched in 3D), a novel algorithm for registration of three dimensional point clouds [4]. HSM3D has some interesting features overcoming some of the outlined challenges:

- the algorithm is *global*, i.e. it does not require a starting point to compute its solution. From this point of view it is radically different from algorithms like Iterative Closest/Corresponding Point (ICP) for which the result of the iterative process depends on the starting point. ICP and similar algorithms can be considered *local*, i.e. they are well suited when there is a good initial estimate of the solution, so that the search for the optimal solution can be localized, and does not need to span the whole solution space.
- HSM3D does not rely on the presence of features in the input.
- HSM3D does not provide a single solution, but it rather computes a set of ranked hypotheses. This aspect is particularly useful to detect ambiguous situations, and can then be used to track multiple hypothesis, similarly to the principles used in particle filters.
- Under the hypothesis of noise free input, HSM3D can be proven to be complete, i.e. it computes the correct solution (up to symmetries in the data).

In our initial study, we focussed our attention on data sets produced by tilting range scanners. This particular configuration delivers very dense data sets allowing very efficient solutions. The hypothesis of dense data, however, cannot be applied for sensors like stereo camera where the denseness of the data is less evident and not uniform.

In this paper we illustrate how HSM3D can be modified and improved in order to efficiently process data coming from stereo cameras. In particular we show that for the case under consideration it is still possible to find opportunistic ways to reduce the computation time while retaining an excellent performance. We also extend the solution ranking method so that it takes into account the color information delivered with the point cloud. This additional information effectively reduces the number of occasional outliers found in the solution set. For example when the environment exhibits a high degree of geometrical symmetry, the original HSM3D algorithm sometimes wrongly estimates the best solution by taking a symmetric transformation. The paper is organized as follows. Related literature is shortly described in Section II, while the HSM3D algorithm is recalled in Section III. Section IV discusses some computational aspects of HSM3D and identifies possible improvements. These additions are experimentally validated in Section V, and conclusions are offered in Section VI.

II. Related literature

The use of scan matching algorithms for three dimensional point clouds is mainly motivated by the need to develop algorithms capable of producing three dimensional maps. Most robotics research in the area has dealt with point clouds produced by tilting range scanners. While there has been a flurry of papers on the topic, it is possible to roughly classify them into two categories. The majority of papers are based on the use of the well known Iterative Closest/Corresponding Point (ICP) algorithm [2]. ICP's main advantage is its simple implementation, but as evidenced in the introduction its main limitation comes from the necessity of starting its

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

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

computation from an initial solution guess. In other words, ICP is an iterative minimization algorithm whose point of convergence is strongly influenced by the starting point for the minimization process. Numerous mapping algorithms based on ICP have been developed, and the reader is referred to the recent monograph by Nüchter for an up-to-date description of ICP-based methods [17], and to [12] for an example of application in the 2D case. The Normal Distribution Transformation (3DNDT) is a more recent approach also based on an iterative principle [13], [10]. 3DNDT is somehow similar to ICP inasmuch as it performs an iterative optimization process whose final result depends from the starting point. A recent paper attempts to make a comparison between ICP and 3DNDT [14], but no definitive conclusion can be derived, i.e. no algorithm prevails in an absolute sense. The use of local methods like ICP and 3DNDT for the three dimensional case has been mainly driven by the possibility to use odometry information to build an initial position/pose estimate. Few global algorithms have been proposed. The algorithm most similar to HSM3D is [15] and the reader is referred to [4] for a detailed comparison. Monocular, omnidirectional, and stereo cameras have been used more and more for localization and SLAM in recent years. Usually, camera images are preprocessed to extract features which are then tracked over multiple frames. For example, Davison *et al.* [6] used features extracted from monocular images to solve the SLAM problem. The more limited pose-tracking problem with visual input ("visual odometry") has been approached earlier, with both monocular [16] and stereo cameras [18]. This approach has been proven to be usable on mobile robots in outdoor unstructured environments [1]. The problems that these methods must solve are: 1) extracting stable features (popular options are Lucas-Kanade and SIFT features); 2) establishing feature correspondences across frames. Note that the methods that use stereo input for visual odometry only work on a subset of features, discarding most of the 3D information which is computed by the comparison of the two images. HSM3D, in comparison, use *all* the 3D points extracted by the stereo computation, regardless of the presence of particular features.

III. Algorithmic review

A short overview of the HSM3D algorithm is offered in this section. The reader is referred to [4] for an in depth discussion.

A. Problem definition and notation

A 3D image is a function $i : \mathbb{R}^3 \to \mathbb{R}^+$. In the following, images will be represented by finite sets of points in three dimensions, therefore we will write $p \in i$ to indicate that p is a point of image i. A rototranslation is a transformation mapping a point in $\mathbb{R}^3$ into a new point by first applying a 3D rotation r, and then a translation t. Depending on the context, a rotation r will be either represented by a 3×3 rotation matrix r, or using the axis angle-representation (a, θ), where $a \in \mathbb{R}^3$ is a vector of unitary modulus. Given an image i_1, a new image i_2 is obtained by applying to i_1 a rototranslation (r, t), i.e.

$$i_2 = \{p \mid p' \in i_1 \wedge p = rp' + t\}.$$

In the sequel we will use the notation $i \circ r$ and $i \circ t$ to indicate the image obtained after applying rotation r or translation t to image i. The problem we aim to solve is the following: given two images i_1 and i_2 such that i_2 is obtained from i_1 by applying an unknown rototranslation (r, t), recover r and t. In the following we will also use the symbol $\mathbb{S}^2$ to indicate the unit sphere, i.e. the set of three dimensional vectors of unary length. The reader should notice that elements of $\mathbb{S}^2$ represent 3D surface orientations. We will furthermore use the symbol $\langle \cdot \rangle$ to indicate the scalar product between vectors.

B. The 3D Hough Transform and Hough Spectrum

HSM3D works in two stages, i.e. it first recovers r and then t. In order to recover r, arguably the hardest step, the algorithm projects the input into the 3D Hough Spectrum, a concept that in its 2D version was introduced in [5]. The Hough Spectrum builds upon the Hough Transform of a 3D image. As the Hough Transform does not depend on the color of points in i, but only on their positions, it is helpful to think to i as the sum of Dirac's impulses centered on the points in i, i.e. $i(x) = \Sigma_j \delta(p_j - x)$. The Hough Transform of a 3D image maps a 3D image into a function defined on $\mathbb{S}^2 \times \mathbb{R}$

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

Given a kernel $k : \mathbb{R} \to \mathbb{R}^+$ the value of $\text{HT}[i]$ at point $(s, \rho) \in \mathbb{S}^2 \times \mathbb{R}$ is given by:

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

In order to appreciate this definition, it is necessary to consider that the set of oriented planes in $\mathbb{R}^3$ is isomorphic to $\mathbb{S}^2 \times \mathbb{R}^+$, i.e. each plane is univocally identified by an orientation and a distance from the origin. $\text{HT}[i](s, \rho)$ therefore adds the contribution of all points in i towards the plane identified by the couple s, ρ. The contribution is weighted by the kernel k. The reader should then now recognize that $k(\langle s, v \rangle - \rho)$ indeed measures how much v supports s, ρ.

The Hough Transform enjoys two properties that are relevant for the derivation of the algorithm:

$$\text{HT}[i \circ r](s, \rho) = \text{HT}[i](r \cdot s, \rho) \quad (1)$$

$$\text{HT}[i \circ t](s, \rho) = \text{HT}[i](s, \rho + \langle t, s \rangle). \quad (2)$$

In plain words, the first property says that a rotation of the input translates to a rotation of its transform, while the second property states that a translation of the input maps into a translation of the transform. Having defined the Hough Transform, the definition of its spectrum is immediate. Let g be any translation-invariant functional mapping a function on the reals into the positive reals, like for example $g : f \to$

$||f||_2$. Then, the Hough Spectrum of a 3D image is a function mapping $\mathbb{S}^2$ into the positive reals

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

and is defined as follows:

$$\text{HS}[i](s) = g[\text{HT}[i](s, \cdot)]. \tag{3}$$

In colloquial terms the Hough Spectrum accumulates the Hough Transform along the directions identified by $\mathbb{S}^2$ according to the functional g. Given equations (1), (2) and (3), it is evident that the Hough Spectrum is invariant to translations, and that rotations of the input are mapped into rotations of the Hough spectrum defined over $\mathbb{S}^2$. In order to visualize the concept of Hough Spectrum, it may be helpful to think to it as function defined over the sphere $\mathbb{S}^2$ identifying which surface directions are more common on the input. The reader is however warned upfront that the whole framework does not rely on the assumption that the input features planar surfaces, and is instead well defined for any kind of data (see Figure 1).

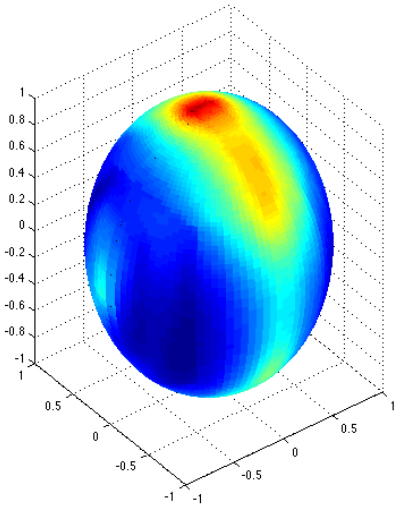

Fig. 1. The Hough Spectrum is a function defined over $\mathbb{S}^2$, and is built by accumulating the values of the three dimensional Hough Transform according to the functional g. The figure gives a graphical representation of how the Hough Spectrum looks like.

C. Algorithmic steps

The two steps to recover first the rotation and then the translation are here shortly summarized.

1) Recovering the rotation r: given i_1 and i_2 the algorithm first computes their spectra, i.e. $f_1 = \text{HS}[i_1]$ and $f_2 = \text{HS}[i_2]$. In order to gain some insights about the algorithm it is convenient to initially think that the two images are free of noise, so that f_2 and f_1 are related by a rotation in three dimensions. The overall rotation r is constructed as the combination of two successive rotations r_1 and r_2, i.e. $r = r_2 \circ r_1$. Let m_1 and m_2 be the global maximum[1] in f_1 and f_2 respectively. Rotation r_1 aims to move m_1 into m_2. In axis-angle representation the transformation can be written in closed form as follows:

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

After applying r_1 the two maxima coincide, but the two spectra may still not overlap because a further rotation r_2 around m_2 is still needed, i.e.

$$r_2 = (m_1, \theta).$$

The unknown angle θ can be derived by cross correlation between the spectra f_1 and f_2 into the smallest truncated cylinder containing $\mathbb{S}^2$ and parallel to the z axis. The precise description of this step is rather long, and the reader is referred to [4] for details.

2) Recovering the translation t: once r has been determined, t can be recovered as follows. Pick at least three linearly independent directions in $\mathbb{S}^2$, namely $s_1, s_2, s_3, \ldots$. Next, project the Hough Transforms along these directions, i.e. define the functions

$$h_1^{s_i}(\rho) = \text{HT}[i_1](s_i, \rho)$$
$$h_2^{s_i}(\rho) = \text{HT}[i_2](r \cdot s_i, \rho).$$

These functions are unidimensional projections of the Hough Transform along the considered directions. Because of Eq. 2 the following relationship holds:

$$h_1^{s_i}(\rho) = h_2^{s_i}(\rho + \langle r \cdot s_i, t \rangle).$$

Henceforth, by cross correlating $h_1^{s_i}$ and $h_2^{s_i}$ one can estimate $d_i = \langle r \cdot s_i, t \rangle$. Given three or more of these estimates it will be possible to recover t by solving a least square problem like the following:

$$\begin{bmatrix} (r \cdot s_1)^T \\ (r \cdot s_2)^T \\ (r \cdot s_3)^T \\ \vdots \end{bmatrix} t = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \end{bmatrix}.$$

D. Multihypothesis ranking

The steps just described will not lead to a unique solution for the problem. In fact, when recovering the rotation r_1, one should consider multiple local maxima in f_1 and f_2 and try all possible combinations. Moreover, when recovering both r_2 and t, the search for maxima in cross correlations will yield multiple peaks, and therefore in the end one is provided with a set of possible solutions to the original problem. This fact is not negative at all. On the contrary, one is provided with multiple hypotheses, so that possible ambiguous situations can be tracked. However, a method to rank these solutions is needed. This is achieved using an algorithm similar to RANSAC [8]. In order to rank a possible

[1]The global maximum may not exist, but at this level of the discussion it helps assuming it does. In practice this step is repeated for the largest n local maxima in the two spectra, so the step is well defined.

solution (r, t), a random subset of points are selected in i_2 and transformed according to (r, t). Next, transformed points are checked for consistency with the points in i_1, and each successful correspondence votes for the solution being evaluated. Rototranslations are eventually ranked by the number of received votes.

IV. Performance considerations and improvements

From a computational point of view, the algorithm relies on the preliminary construction of the Hough Transform, followed by the computation of its spectrum. Similarly to the much studied bidimensional Hough Transform, a discretization step is necessary for its approximated computation, and the choice of the discretization resolution represents a significant tradeoff between accuracy and speed. In practice, we are faced with the problem of computing a three dimensional version of the discrete Hough Transform mostly known in literature [**?**]. However, when stepping from two to three dimensions the situation becomes more complicated, as evidenced from the fact that in the 3D case the Hough Transform is a function defined over $\mathbb{S}^2 \times \mathbb{R}^+$, while in 2D it is defined over $[0, 2\pi] \times \mathbb{R}^+$. As it is cumbersome to directly define and compute an approximated function over $\mathbb{S}^2$, the diffeomorphism between $\mathbb{S}^2$ and the smallest cube surrounding $\mathbb{S}^2$ is exploited[2], i.e. each patch on the surface of the sphere is put into correspondence with a regular square patch on the faces of the cube. A key parameter is then the resolution used to divide each face of the cube into a number of equally-sized squares that will correspond to patches on the sphere. Moreover, for HT it is also necessary to consider its $\rho \in \mathbb{R}^+$ component, that will be discretized into a series of equally sized segments within two extremes, say ρ_{min} and ρ_{max}. Therefore, to compute the HT it is necessary to compute the contribution of every point in the given images towards every possible direction approximated on the cube, and every possible distance from the origin as defined in ρ's discretization. It is evident that if this step is executed in a brute-force manner it will be extremely time consuming. As pointed out in [4], if points in i are dense enough, it is possible to extend the above definition to consider the case of *oriented points*, i.e. points with an associated orientation. Indeed, when points are generated by a sensor that regularly samples a solid angle, for example tilting a range scanner, the orientation of the surface supporting a point can be reliably estimated by looking at the neighbors of the point itself. This observation leads to a notable improvement of the speed because it is not necessary to consider the contribution of a point towards all possible direction, and instead to restrict the computation only to the directions similar to the orientation of the surface supporting the point. In this paper, however, we focus our attention to three dimensional point clouds produced by stereo cameras. Preliminary experiments have outlined data delivered by this type of sensor is not dense enough to reliably extrapolate the orientation of the supporting surface, and therefore the accelerated version formerly described cannot be applied. However, there are other possible ways to gain some speed. As pointed out in [11] for the bidimensional case, the Hough Transform is robust enough that it can be often estimated by just considering a small subset of the input points. This observation was exploited in [3], where a 2D version of the algorithm here discussed was studied for the special problem of merging occupancy grid maps produced by multiple robots. The first improvement therefore consists of embedding the randomized Hough Transform into the the first step of the algorithm.

One of the strengths of HSM3D resides in its ability to solve the *global* alignment problem, i.e. not requiring a preliminary estimation of the solution. However, as evidenced in [4], the best ranking solution is sometimes far from the real solution. When this happens, it is very often the case that the best solution exhibits an error of 180 degrees. This is easy to explain in indoor environments, where large planar surfaces like floors and ceilings contribute to the generation of highly symmetric Hough Spectra leading to a rotation estimate in which the best ranking solution is tilted upside down. While this problem can be mitigated by tracking multiple hypothesis, i.e. by considering not just the best ranking solution, but the best n ranking solutions, it would still be beneficial to minimize the occurrence of these sporadic outliers. This goal can be achieved by integrating the provided color information into the multihypothesis ranking step. To be precise, when a solution candidate needs to be ranked, it is rewarded not only for its ability to map corresponding points to nearby positions, but also for finding agreements with the pixel colors.

V. Experimental setup and results

In [4] HSM3D was compared to ICP while processing 3D images produced by a tilting laser range finder. In this paper instead we process data points produced by a stereocamera. A P3AT robot is remotely controlled and moves around in an indoor office environment. A Bumblebee2 camera is used to collect data. The camera captures two images with resolution 320×240 and returns a 3D image. The 3D image is obtained through a triangulation process performed by a proprietary software library shipped with the camera. For every point the sensor returns its spatial coordinates, its gray scale color, and the point in the image plane that was used to triangulate its position (this last piece of information is not used in our algorithm). Source code and data sets used in the experiments described in this section are available for download on http://robotics.ucmerced.edu. The code is written in C++, does not rely on external libraries, and at the moment has not been engineered for high performance, therefore timing information provided later on should only be considered to extrapolate performance trends, and not as indicators of absolute performance.

[2]Thinking to $\mathbb{S}^2$ as the unitary sphere centered in the origin, the diffeomorphism maps $\mathbb{S}^2$ into the cube centered in the origin and with edge length 2.

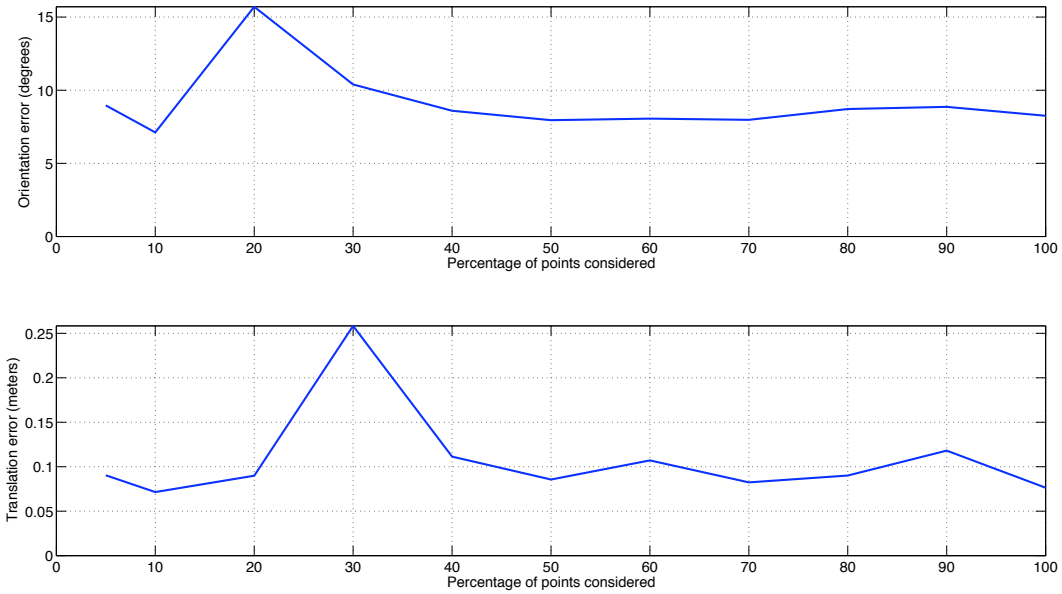

Fig. 2. This test measures the change in performance of the HSM3D algorithm as the percentage of points considered to compute the Hough Transform decreases. The top panel shows the average orientation error as a function of the percentage of randomly chosen points, while the bottom panel displays the average translation error.

A. *Use of the randomized Hough Transform*

The first experiment aims to verify how the use of the randomized Hough Transform affects the performance of the algorithm and its time requirements. The experiment is organized as follows. Given a 3D image i_1 we apply a random transformation (r, t) to obtain i_2, and we then pass these two images to HSM3D. The performance of the algorithm is measured in terms of the difference between (r, t) and the best ranking transformation returned by the algorithm. This procedure is applied to a dataset consisting of 20 3D images, and repeated for decreasing percentages of considered points, from 100% down to 5%. Accuracy results are presented in figure 2. It can be observed that even while dramatically decreasing the percentage of considered points, the performance of the algorithm does not significantly deteriorate. Table I instead illustrates the average time spent as a function of the percentage of considered points. As a frame of reference, reported times are referred to the execution of the algorithm on an Intel dual core at 3.06 GHz. It can be observed that by reducing the number of considered points the time significantly drops.

10	20	30	40	50	60	70	80	90	100
5.7	6.8	6.9	7.6	8.8	10.1	10.7	11.2	11.5	14.6

TABLE I

COMPUTATION TIME AS A FUNCTION OF THE PERCENTAGE OF POINTS. TOP ROW: PERCENTAGE OF POINTS. BOTTOM ROW: COMPUTATION TIME IN SECONDS.

B. *Integration of color*

The setup of this experiment is similar to the previous one, i.e. a random transformation is applied to a 3D image, and we measure the ability to recover it. Results are illustrated in figure 3, where the hypotheses produced while solving 1200 random tests have been ranked taking color into consideration (data series *combined*) and then without considering the color (data series *weight*). Given that random transformations consider rotations varying from 0 to 180 degrees, the experiment confirms that performance of HSM3D is independent from the entity of the random transformation to be recovered. In addition, and more importantly, it also shows that by integrating color information the number of outlier solutions displaying a 180 degrees error is exactly halved.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a thorough experimental validation of the HSM3D algorithm we recently developed. In particular, we have focussed our analysis on point clouds produced by stereo cameras, thus complementing our former findings obtained with tilting laser scanners. The use of this type of input is particularly interesting because it invalidates some of the conditions we formerly outlined to expedite the algorithm. In order to overcome the limitations deriving from this different input, we introduced a randomized version of the Hough Transform, and we verified that the performance of the algorithm remains good even when a large part of the input is disregarded. This result was known for the 2D case, but the the best of our knowledge it was never observed nor exploited for the 3D case. The use of the randomized Hough Transform also significantly reduced the time needed to solve the registration problem. Finally, we have integrated the use of color into the hypothesis ranking step, thus halving the number of sporadic solutions affected by large errors in the orientation component. These

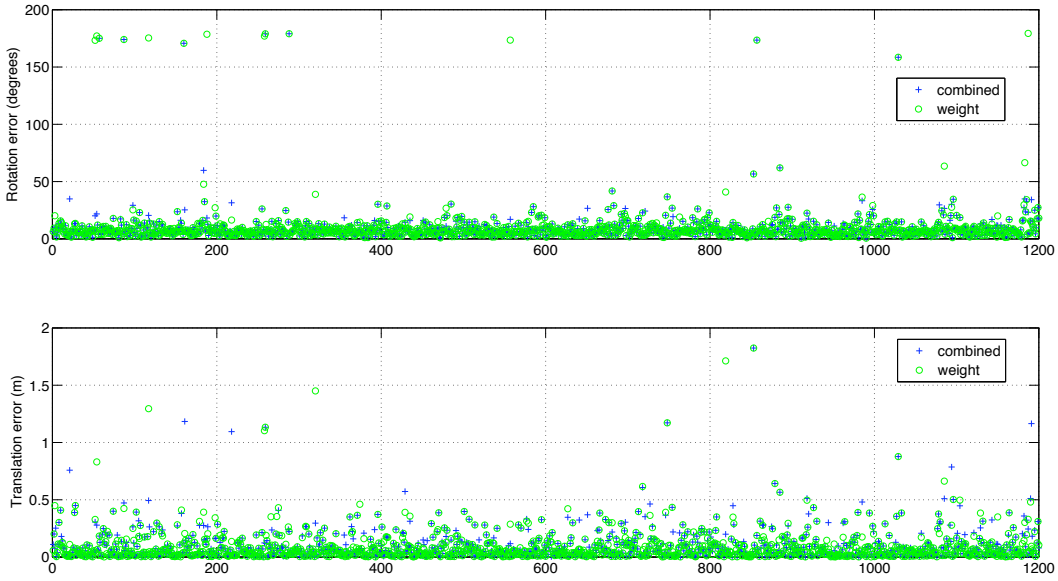

Fig. 3. This test outlines the global performance of the algorithm over 1200 randomly chosen transformations. The top chart outlines that by incorporating information color into the hypothesis ranking step the number of solutions exhibiting large rotation errors halves (green dots represent the error of the best solution when ranking does not consider color, while blue crosses represent the error of the best solution when color is also taken into consideration). A similar trend is observed for the translation error in the bottom panel.

erroneous hypotheses are typically caused by geometrical symmetries in the environment under consideration, and can be effectively rejected by taking color into consideration. After carefully analyzing the performance of the algorithm, it appears there are two directions to follow in order to obtain further performance increases and shoot for an algorithm that can be integrated into a robot controller with strict time requirements. The first one consists in switching to a different representation for the Hough Spectum domain, i.e. for $\mathbb{S}^2$. The diffeomorphism with the cube should be replaced with a different representation, like [9], so that one is not bound to use a time consuming iterative approach when building the Hough Transform by considering plenty of uninteresting directions. Secondly, the algorithm could be further accelerated by dismissing unpromising solutions early during the computation. At the moment the algorithm builds a numerous set of candidate solutions by pairwise matching rotations and translations, even when it could be possible to already rule out certain combinations that appear odd since the very beginning. These research direction will be pursued in the near future.

REFERENCES

[1] M. Agrawal and K. Konolige. Real-time localization in outdoor environments using stereo vision and inexpensive GPS. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2006.

[2] P. Besl and N. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 239–256, 1992.

[3] S. Carpin. Merging maps via Hough transform. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1878–1883, 2008.

[4] A. Censi and S. Carpin. HSM3D: feature-less global 6DOF scan-matching in the Hough/Radon domain. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3899–3906, 2009.

[5] A. Censi, L. Iocchi, and G. Grisetti. Scan matching in the Hough domain. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2739–2744, 2005.

[6] A. Davison. Realtime SLAM with a single camera. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2003.

[7] R. Duda and P. Hart. Use of the hough transform to detect lines and curves in the pictures. *Communications of the ACM*, 15(1):11–15, 1972.

[8] M.A. Fischler and R.C. Bolles. Randomized sample consensus: a paradigm for model fitting with applications to image analysis and automated carthography. *Communications of the ACM*, 24:381–385, 1981.

[9] K. M. Gorski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, , and M. Bartelmann. Healpix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere. *The Astrophysical Journal*, 622(2):759–771, 2005.

[10] B. Huhle, M. Magnusson, W. Strasser, and A. Lilienthal. Registration of colored 3D point clouds with a kernel-based extension to the Normal Distribution Transform. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 4025–4030, 2008.

[11] N. Kiryati, Y. Eldar, and A.M. Bruckstein. A probabilistic Hough transform. *Pattern recognition*, 24(4):303–316, 1991.

[12] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

[13] M. Magnusson, A. Lilienthal, and T. Duckett. Scan registration for autonomous mining vehicles using 3D-NDT. *Journal of Field Robotics*, 24(10):803–827, 2007.

[14] M. Magnusson, A. Nüchter, S. Lörken, A.J. Lilienthal, and J. Hertzberg. 3D mapping the Kvarntorp mine – a field experiment for evaulation of 3D scan matching algorithms. In *Proceedings of the IEEE Conference on Robotics and Automation*, 2009 (to appear).

[15] A. Makadia, A. Patterson, and K. Daniilidis. Fully authomatic registration of 3D point clous. In *Proceedeings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2006.

[16] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *Prooceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[17] A. Nüchter. *3D robotic mapping*. STAR. Springer, 2009.

[18] C. F. Olson, L. H. Matthies, and M. Schoppers. Robust stereo ego-motion for long-distance navigation. In *Prooceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000.