

# Human-in-the-Loop SLAM

Samer Nashed and Joydeep Biswas

University of Massachusetts, Amherst  
140 Governors Drive  
Amherst, Massachusetts 01003

## Abstract

Building large-scale, globally consistent maps is a challenging problem, made more difficult in environments with limited access, sparse features, or when using data collected by novice users. For such scenarios, where state-of-the-art mapping algorithms produce globally inconsistent maps and require additional data collection, we introduce a systematic approach to incorporating sparse human corrections, which we term Human-in-the-Loop Simultaneous Localization and Mapping (HitL-SLAM). Given an initial factor graph for pose graph SLAM, HitL-SLAM accepts approximate, potentially erroneous, and rank-deficient human input; infers the intended correction via expectation maximization (EM); back-propagates the extracted corrections over the pose graph; and finally jointly optimizes the factor graph including the human inputs as human factor terms, to yield globally consistent large-scale maps. We thus contribute an EM formulation for inferring potentially rank-deficient human corrections to mapping, and human factor extensions to the factor graphs for pose graph SLAM that result in a principled approach to joint optimization of the pose graph while simultaneously accounting for multiple forms of human correction. We present empirical results showing the effectiveness of HitL-SLAM at generating globally accurate and consistent maps even when given poor initial estimates of the map.

## 1 Introduction

Accurate metric mapping of environments is essential for autonomous mobile robot function. For a variety of reasons, including operator inexperience, sensor limitations, large or time-sensitive environments, and algorithmic limitations, generating accurate metric maps from data collected over a single deployment may require more resources than are available. Furthermore, it may be impossible or prohibitively expensive to improve the map with data collected by redeploying the robot. In these instances, we propose a general human-in-the-loop framework, along with a specific algorithm, which dramatically improves map accuracy while requiring limited additional compute resources and imposing minimal constraints on the human.

The goal of Human-in-the-Loop SLAM (HitL-SLAM) is to leverage a human’s ability to outperform state-of-the-art algorithms in the data association problem, in order to construct more accurate maps. Moreover, humans often have meta-knowledge about an environment that presently far ex-

ceeds most robot learning capabilities. Allowing humans to endow robots with this knowledge can greatly increase robot capabilities and is necessary in certain cases where resources are constrained. Additionally, the information sharing between robot and human necessitates a method for gathering and evaluating human input which is robust to human error and requires minimal user effort. Fig. 1 presents an example of HitL-SLAM in practice.

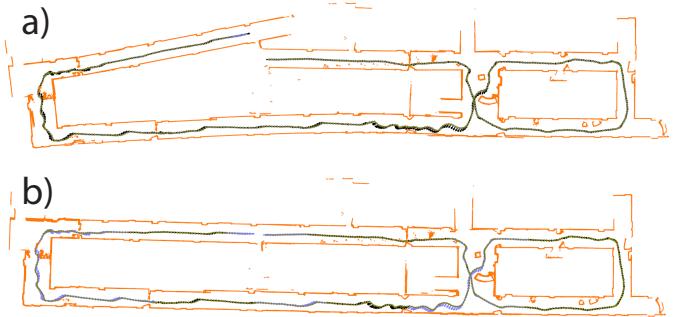


Figure 1: Before and after maps of a floor in our department. Observations are shown in orange, and poses are shown as black arrows. a) initial map with no loop closure and no human constraints. b) the final map produced by HitL-SLAM. Poses which are involved in one or more human constraints are shown in blue.

In this paper we present three contributions. First, a principled framework for dealing with human input motivated by both functional and theoretical considerations, which allows the human to efficiently convey their knowledge. Second, an algorithm for constructing human factors and performing joint factor graph optimization, which we use to solve HitL-SLAM. And third, a 2-stage SLAM backend, which uses a combination of analytical and numerical techniques to produce globally consistent maps with minimal distortion in the presence of rank deficient constraints.

HitL-SLAM is an iterative process, wherein the human and robot alternate proposing the most likely map. The robot operates on poses in a factor graph and displays the resulting map, while the human operates on the observations and those operations are translated to factors in the factor graph. The key idea is that humans impose constraints more natu-

rally by relating observations instead of the underlying poses since they are more likely to be knowledgeable about the structure of the observations rather than the poses. Our proposed approach thus consists of the following steps:

1. *Extract Likely Input*: Use EM (Dempster, Laird, and Rubin 1977) to decide the most likely parameters of human input (Section 4).
2. *Establish Initial Estimate*: Use COP-SLAM (Dubbelman and Browning 2015) to find a piece-wise optimal initial estimate. (Section 5a)
3. *Solve Optimization Problem*: Construct human factors and jointly solve a non-linear least-squares optimization problem. (Section 5b)

To the best of our knowledge, we present the first algorithm and framework for incorporating human input into the optimization of a pose-graph. Our approach is evaluated using a series of tests designed to measure the accuracy and scalability of HitL-SLAM. We find that HitL-SLAM is able to produce highly accurate maps in a variety of scenarios, and requires less time than it would take to re-deploy the robot over the same map.

## 2 Related Work

Solutions to robotic mapping and SLAM have improved dramatically in recent years, but a significant gap between state-of-the-art fully autonomous systems and the requirements for robust, large-scale deployment still exists, due in part to the difficulty of the data association problem (Dissanayake et al. 2011; Bailey and Durrant-Whyte 2006; Aulinis et al. 2008; Thrun and others 2002). The notion that humans and robots can or should collaborate in map building is not new, and has given birth to a field known as Human-Augmented Mapping (HAM).

Most work within HAM can be thought of as belonging to one of two groups, depending on whether the human and robot are conjunctive in time and space during exploration (C-HAM), or whether the human enters the loop remotely or after the fact (R-HAM). Many C-HAM techniques exist to address semantic (Nieto-Granda et al. 2010; Christensen and Topp 2010) and topological (Topp and Christensen 2006) mapping. A number of approaches have also been proposed for integrating semantic and topological information, along with human trackers (Milella et al. 2007) and interaction models (Topp et al. 2006), into *conceptual spatial maps* (Zender et al. 2007), which are typically organized in a hierarchical manner.

There are two major weaknesses implicit in these C-HAM approaches. First, a human must be present with the robot during exploration. This places physical constraints on the type of environments which can be mapped, as they must be accessible and traversable by a human. Second, these methods are inefficient with respect to the human’s attention, since most of the time the human’s presence is not critical to the robot’s function, for instance during navigation between waypoints. These approaches, which focus mostly on semantic and topological mapping, also typically assume that the robot is able to construct a nearly perfect metric map

entirely autonomously. While this is reasonable for small environments, metric mapping of large, dynamic spaces is still an open research question.

In contrast, most of the effort in R-HAM has been concentrated on either incorporating humans into the loop during deployment in order to teleoperate and gain situational awareness such as in the USAR problem (Murphy 2004; Nourbakhsh et al. 2005), or to do high level decision making such as goal assignment or coordination of multiple agents (Olson et al. 2013; Parasuraman et al. 2007; Doroodgar et al. 2010). Some R-HAM techniques for metric mapping and pose estimation have also been explored, but these involve either having the robot retrace its steps to fill in parts missed by the human (Kim et al. 2009) or by having additional agents and sensors in the environment (Kleiner, Dornhege, and Dali 2007), neither of which is efficient.

Ideally, a robot could explore an area only once with no need for human guidance or input during deployment, and later with minimal effort, a human could make any corrections necessary to achieve a near-perfect metric map. This is precisely what HitL-SLAM does, and it is worth noting that the HitL-SLAM framework and algorithm place no restrictions on the temporal or spatial conjunctivity of the human and robot.

## 3 Human-in-the-Loop SLAM

HitL-SLAM operates on a factor graph  $G$  defined as a set of poses  $X$  representing estimated poses along the robot’s trajectory, and a set of factors  $F = \{R, H\}$ , which encode information about both relative pose constraints arising from odometry and observations,  $R$ , and constraints supplied by the human, denoted  $H$ . In this report, the initial factor graph  $G_0$  provided by our SLAM front-end, EnML, does not contain any loop closure information beyond the length of the episode specified by EnML. However, HitL-SLAM is capable of handling constraints in  $G_0$  with or without loop closure.

HitL-SLAM runs iteratively, with the human specifying constraints on observations in the map, and the robot then enforcing those constraints along with all previous constraints to produce a revised estimate of the map. Since humans often enter only approximately correct input given a desired effect, some interpretation of the human input is necessary before human constraints can be computed and added to the factor graph. Formally, we say the robot first proposes an initial graph  $G_i = \{X_i, F_i\}$ , the human then supplies a set of correction factors  $H_i$ , and finally the robot re-optimizes the poses in the factor graph, producing  $G'_i = \{X'_i, F_i \cup H_i\}$ .

Each iteration of HitL-SLAM proceeds in two steps, shown in Fig. 2. First, the human input is gathered, interpreted, and a set of human correction factors are instantiated. Second, a combination of analytical and numerical techniques is used to jointly optimize the factor graph using the human factors and the relative pose factors, producing the final map.

When interpreting human input, we define human correction factors  $h \in H$  as tuples  $h = \{P_a, P_b, S_a, S_b, X_a, X_b, m\}$ , where  $P_a$  and  $P_b$  are sets

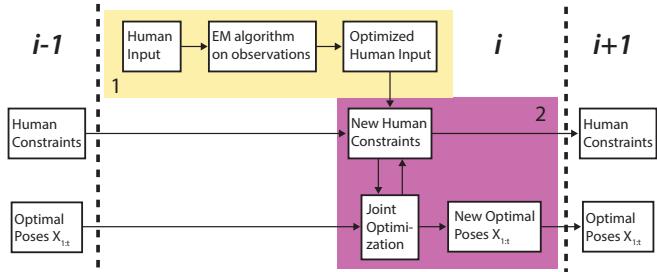


Figure 2: Flow of information during processing of the  $i^{\text{th}}$  human input. Block 1 (yellow) outlines the evaluation of human input, and block 2 (purple) outlines the factor graph construction and optimization processes. Note that the joint optimization process optimizes *both* pose parameters *and* human constraint parameters.

of points in  $\mathbb{R}^N$  (for N-dimensional HitL-SLAM) specified by the human. Sets  $S_a$  and  $S_b$  are subsets of all observations  $S$  and contain observations which are near the features defined by  $P_a$  and  $P_b$ , respectively.  $X_a$  and  $X_b$  are subsets of the poses  $X_{1:t}$ , where a given pose  $x \in X_a$  if there are observations in  $S_a$  which originate from  $x$ .  $X_b$  is defined analogously.  $m \in M$  defines the geometry of the constraints the human is enforcing over the observations, such as colocation, collinearity, perpendicularity, *etc.*

We frame the problem of interpreting human input as finding the observation subsets  $S_a, S_b$  and human input sets  $P_a, P_b$  which maximize the joint correction input likelihood,  $p(S_a, S_b, P_a, P_b | P_a^0, P_b^0, m)$ , which is the likelihood of selecting observation sets  $S_a, S_b$  and point sets  $P_a, P_b$ , given initial human input  $P_a^0, P_b^0$  and correction mode  $m$ . To find  $S_a, S_b$  and  $P_a, P_b$  we use the sets  $P_a^0, P_b^0$  and observations in a neighborhood around  $P_a^0, P_b^0$  as initial estimates in an Expectation Maximization approach, detailed in Section 4. As the pose parameters are adjusted during joint optimization in later iterations of HitL-SLAM, the locations of points in  $P_a, P_b$  may change, but once an observation is established as a member of  $S_a$  or  $S_b$  its status is not changed.

Once  $P_a, P_b$  and  $S_a, S_b$  are determined for a new constraint, then given  $m$  we can find the set of poses  $X_{1:t}^*$  which best satisfy all given constraints. We first compute an initial estimate  $X_{1:t}^0$  by analytic back-propagation of the most recent human correction factor, considering sequential constraints in the pose-graph. Next, we construct and solve a joint optimization problem over the relative pose factors  $f$  and the human factors  $h$ . This amounts to finding the set of poses  $X_{1:t}^*$  which minimize the sum of the cost of all factors,

$$X_{1:t}^* = \operatorname{argmin}_{X_{1:t}} \left[ \sum_{i=1}^{|R|} c_r(r_i) + \sum_{j=1}^{|H|} c_m(h_j) \right], \quad (1)$$

where  $c_r : R \rightarrow \mathbb{R}$  computes the cost from relative pose-graph factor  $r_i$ , and  $c_m : H_m \rightarrow \mathbb{R}$  computes the cost from human factor  $h_j$  with correction mode  $m$ . Section 5 covers the construction of the human factors and the formulation of the optimization problem.

## 4 Evaluating Human Input

Because humans do not have perfect control of a mouse or touch screen, what the human actually enters and what they intend to enter may differ slightly. We formulate the problem of evaluating human input as finding the sets  $S_a, S_b$  and  $P_a, P_b$  which the human most likely meant to identify ( $S_a, S_b$ ) and provide ( $P_a, P_b$ ), given the observations  $S$  and the initial human input  $P_a^0$  and  $P_b^0$ . To do this we use the EM algorithm, which maximizes the log likelihood

$$\sum_i \sum_{z_i} p(z_i | s_i, \theta^{\text{old}}) \log(p(z_i, s_i | \theta)), \quad (2)$$

where the parameters  $\theta = \{P_a, P_b\}$  are the human input, the  $s_i \in S$  are the observations, and the latent variables  $z_i$  are indicator variables denoting the inclusion or exclusion of  $s_i$  from  $S_a$  or  $S_b$ . The expressions for  $p(z_i | s_i, \theta^{\text{old}})$  and  $p(z_i, s_i | \theta)$  come from a generative model of human error based on the normal distribution,

$$p(z_i | s_i, \theta) = \frac{1}{\sqrt{2\sigma^2\pi}} \exp\left(-\frac{(s_i - f(s_i, \theta))^2}{2\sigma^2}\right). \quad (3)$$

Here,  $\sigma$  is the standard deviation of the human's accuracy when specifying points with the mouse, and  $f(s_i, \theta)$  is a distance function which evaluates the distance between a given observation  $s_i$  and the feature parameterized by  $\theta$ . Since all modes  $m$  in this study can be specified by one or more line segments and  $p(z_i | s_i, \theta)$  is convex, then if we define  $f(\theta)$  as the distance between the specified line segment and the given observation  $s_i$ , the EM problem reduces to iterative least-squares fitting over a changing subset of  $S$  which occur near the latest estimates for  $P_a, P_b$ .

Once the initial  $P_a, P_b$  have been determined, along with observations  $S_a, S_b$ , we can find the poses responsible for those observations  $X_a, X_b$ , thus fully defining the human correction factor  $h$ . To make this process more robust, a given pose is only allowed in  $X_a$  or  $X_b$  if there exist a minimum of  $T_p$  elements in  $S_a$  or  $S_b$  corresponding to that pose, where  $T_p$  is a threshold related to observation density.

## 5 Solving HitL-SLAM

### Initializing Human Corrections

HitL-SLAM allows three types of constraints. More complicated data associations may be constructed from a combination of the following:

1. *Colocation*: A full rank constraint specifying that two sets of observations are colocated.
2. *Collinearity*: A rank deficient constraint specifying alignment of two sets of observations along a single translation direction, as well as orientation.
3. *Co-orientation*: A rank deficient constraint specifying only orientation relationships.

Although the user may select sets of observations in any order, we define  $P_a$  to be the input which selects observations  $S_a$  arising from poses  $X_a$  such that  $\forall x_i \in X_a$  and

$x_j \in X_b$ ,  $i < j$ , where  $X_b$  is defined analogously by observations  $S_b$  specified by input  $P_b$ . That is, all poses  $x_i \in X_a$  occur before all poses  $x_j \in X_b$ .

Given  $P_a$  and  $P_b$ , we find the affine transformation  $A$  which transforms the constellation  $P_b$  to the correct location relative to constellation  $P_a$ , as specified by mode  $m$ . If the correction mode is rank deficient, we force the motion of the observations as a whole to be zero along the null space dimensions. For co-orientation, this means that the translation components of  $A$  are zero, and for colinearity the translation along the axis of colinearity is zero. To be clear, this does not mean that any given point  $p_i \in P_b$  will not experience motion along one or more null dimensions. Only the motion of the center of mass of the constellation is constrained in this way. Fig. 3 shows the effect of applying different types of constraints to a set of point clouds. Note that the rank deficient constraints do not produce motion of the point cloud as a whole along the null space dimensions.

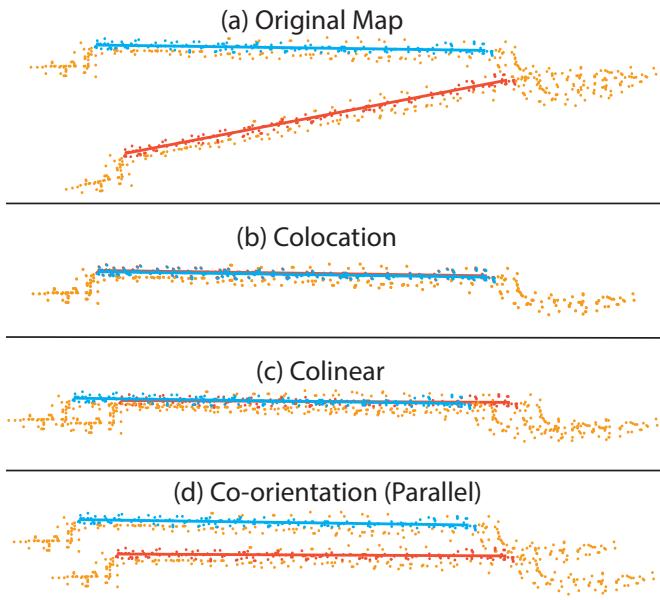


Figure 3: Result of transforming observation point clouds based on different human constraints, showing (a) Original map, (b) Colocation constraint, (c) Colinear constraint, (d) Co-orientation constraint. In all subfigures  $P_a$  and  $P_b$  are given by the red and blue lines, respectively.  $S_a$  and  $S_b$  are shown as the red and blue point clouds.  $S \setminus (S_a \cup S_b)$  appear in orange.

After finding  $A$  we then consider the poses in  $X_b$  to constitute points on a rigid body, and transform that body by  $A$ . The poses  $x_k$  such that  $\forall x_j \in X_b$ ,  $k > j$ , are treated similarly, such that the relative transformations between all poses occurring during or after  $X_b$  remain unchanged.

If  $X_a \cup X_b$  does not form a contiguous sequence of poses, then this explicit change creates at least one discontinuity between the earliest pose in  $X_b$ ,  $x_0^b$  and its predecessor,  $x_c$ . We define affine transformation  $C$  such that  $x_0^b = A_{cb}C x_c$ , where  $A_{cb}$  was the original relative transformation between

$x_c$  and  $x_0^b$ . Given  $C$ , and the pose and covariance estimates for poses between  $X_a$  and  $X_b$ , we use COP-SLAM over these intermediate poses to transform  $x_c$  without inducing further discontinuities.

The idea behind COP-SLAM is essentially a covariance-aware distribution of translation and rotation across many poses, such that the final pose in the pose-chain ends up at the correct location and orientation. The goal is to find a set of updates  $U$  to the relative transformations between poses in the pose-chain such that  $C = \prod_{i=1}^n U_i$ . An approximation could of course be to take the  $n^{\text{th}}$  root of  $C$ , but COP-SLAM achieves piece-wise optimality with respect to transformation magnitude normalized by covariance, by weighting the magnitude of the update rotations and translations according to the inverse covariance.

COP-SLAM has two primary weaknesses. First, it requires uncertainty estimates to be isotropic, which is not true in general. Second, COP-SLAM deals poorly with nested loops, where it initially produces good pose estimates but during later adjustments may produce inconsistencies between observations, since COP-SLAM is not able to simultaneously satisfy both current and previous constraints. Because of these issues, we use COP-SLAM as an initial estimate to a non-linear optimization problem, which produces a more robust, globally consistent map.

### Hit-SLAM Optimization

Without loop closure, a pose-chain of  $N$  poses has  $O(N)$  factors. With most loop closure schemes, each loop can be closed by adding one additional factor per loop. In Hit-SLAM, the data provided by the human is richer than most front-end systems, and reflecting this in the factor graph could potentially lead to a prohibitively large number of factors. If  $|X_a| = n$  and  $|X_b| = m$ , then a naive algorithm which adds a factor between every pair  $(x_i, x_j)$ , where  $x_i \in X_a$  and  $x_j \in X_b$ , would add  $mn$  factors *every loop*. This is a poor approach for two reasons. One, the large number of factors can slow down the optimizer and potentially prevent it from reaching the global optimum. And two, this formulation implies that every factor is independent of every other factor, which is incorrect.

Thus, we propose a method for reasoning about human factors jointly, in a manner which creates a constant number of factors per loop while also preserving the structure and information of the input. Given a human factor  $h = \{P_a, P_b, S_a, S_b, X_a, X_b, m\}$ , we define  $c_m$  as the sum of three residuals,  $R_a$ ,  $R_b$ , and  $R_p$ . The definitions of  $R_a$  and  $R_b$  are the same regardless of  $m$ :

$$R_a = \left( \frac{\sum_{i=1}^{|S_a|} \delta(s_i^a, P_a)}{|S_a|} \right)^{\frac{1}{2}}, \quad R_b = \left( \frac{\sum_{i=1}^{|S_b|} \delta(s_i^b, P_b)}{|S_b|} \right)^{\frac{1}{2}}. \quad (4)$$

Here,  $\delta(s, P)$  denotes the squared Euclidean distance from observation  $s$  to the closest point on the feature defined by the set of points  $P$ . All features used in this study are line segments, but depending on  $m$ , more complicated features with different definitions for  $\delta(s, P)$  may be used.  $R_a$  implicitly enforces the interdependence of different  $x_a \in X_a$ ,

since moving a pose away from its desired relative location to other poses in  $X_a$  will incur cost due to misaligned observations. The effect on  $X_b$  by  $R_b$  is analogous.

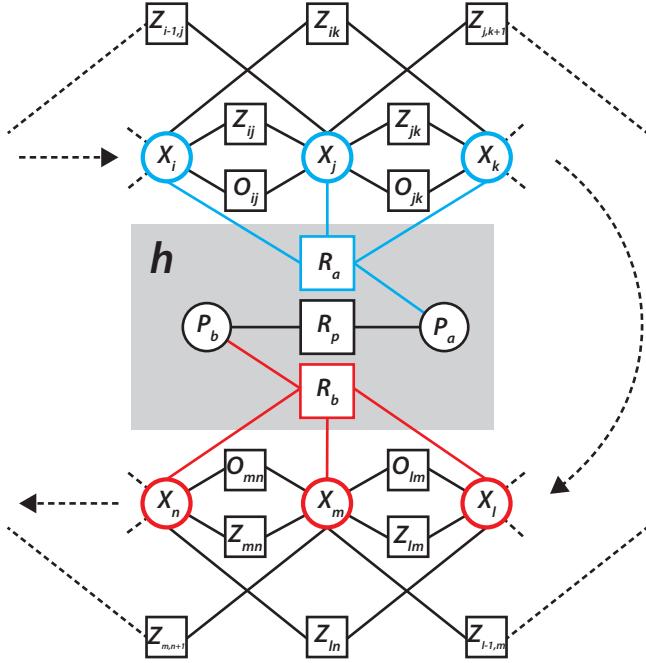


Figure 4: Subset of a factor graph containing a human factor  $h$ . Factors  $R_a$  and  $R_b$  drive observations in  $S_a$  and  $S_b$  toward features  $P_a$  and  $P_b$ , respectively. Factor  $R_p$  enforces the geometric relationship between  $P_a$  and  $P_b$ . Note that parameters in  $X_a$  (blue poses) and  $X_b$  (red poses) as well as  $P_a$  and  $P_b$  are jointly optimized.

The relative constraints between poses in  $X_a$  and poses in  $X_b$  are enforced indirectly by the third residual,  $R_p$ . Depending on the mode, colocation (+), colinearity (-), co-orientation parallel (||), co-orientation perpendicular ( $\perp$ ), the definition changes:

$$\begin{aligned} R_3^+ &= K_1 \|cm_b - cm_a\| + K_2(1 - (\hat{n}_a \cdot \hat{n}_b)) \\ R_3^- &= K_1 \|(cm_b - cm_a) \cdot \hat{n}_a\| + K_2(1 - (\hat{n}_a \cdot \hat{n}_b)) \\ R_3^{\parallel} &= K_2(1 - (\hat{n}_a \cdot \hat{n}_b)) \\ R_3^{\perp} &= K_2(\hat{n}_a \cdot \hat{n}_b) \end{aligned} \quad (5)$$

Here,  $cm_a$  and  $cm_b$  are the centers of mass of  $P_a$  and  $P_b$ , respectively, and  $\hat{n}_a$  and  $\hat{n}_b$  are the unit normal vectors for the feature (line) defined by  $P_a$  and  $P_b$ , respectively.  $K_1$  and  $K_2$  are constants that determine the relative costs of translational error ( $K_1$ ) and rotational error ( $K_2$ ). The various forms of  $R_p$  all drive the constellation  $P_b$  to the correct location and orientation relative to  $P_a$ . During optimization the solver is allowed to vary pose locations and orientations, and by doing so the associated observation locations, as well as points in  $P_a$  and  $P_b$ . Fig. 4 illustrates the topology of the human constraints in our factor graph.

## 6 Results

Evaluation of our method is carried out through two experiments. First, we construct a data set by limiting the range of our robot's laser so that it sees a wall only when very close. We then drive it around a room for which we have ground truth. This creates sequences of "lost" poses throughout the pose-chain which have to rely purely on odometry to localize, thus accruing error over time. We then impose human constraints on the resultant map and compare to ground truth, shown in Fig. 5. HitL-SLAM finds a room width of 6.31m, while our hand measurement gives a width of 6.33m. Additionally, HitL-SLAM produces opposite walls which are within  $1^\circ$  of parallel. Note that due to the limited sensor range, at no point are both walls simultaneously visible to the robot. Thus, correctness must come from proper application of human constraints to the "lost" poses between wall observations.

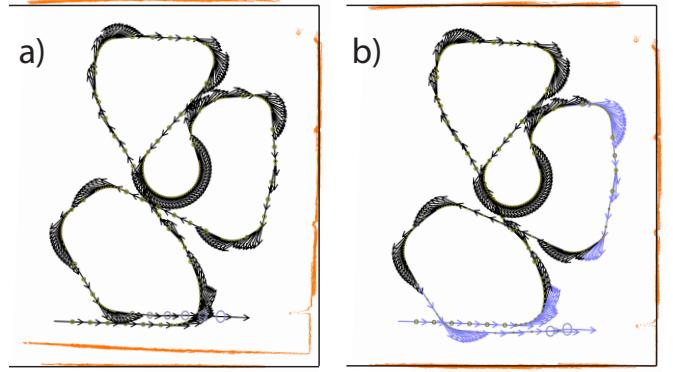


Figure 5: Lost poses experiment. Observations are shown in orange, poses are black arrows, and ground truth (walls) is represented by the black lines. Poses involved in human constraints are colored blue. The initial map is shown in a), and the final map is shown in b).

For larger maps where exact ground truth was not available, we introduce a different metric for evaluation. We define the inconsistency  $I_{i,j}$  to be the area (or volume in 3D) which observations from pose  $x_i$  show as free space and observations from pose  $x_j$  show as occupied space. Although there are cases where inconsistency is not a useful metric, for instance a map of a long hallway without loop closure may be consistent even if it is severely bent and does not correspond well to ground truth, it is often what humans look at when evaluating a map and allows us to monitor the progress of HitL-SLAM over multiple iterations. On average, the inconsistency of the final map was reduced to 9% of its initial value. This makes a compelling argument for the use of inconsistency as a metric for evaluating maps, as well as a tool to guide robots and humans when deciding what constraints to impose on a given map to increase accuracy. Fig. 6 offers some qualitative examples of our algorithm's performance.

All maps shown in Fig. 6 were completed by the human in under 15 mins, well within the time it would take to redeploy the robot and then run our localization algorithm on both datasets to generate a new map from both deployments.

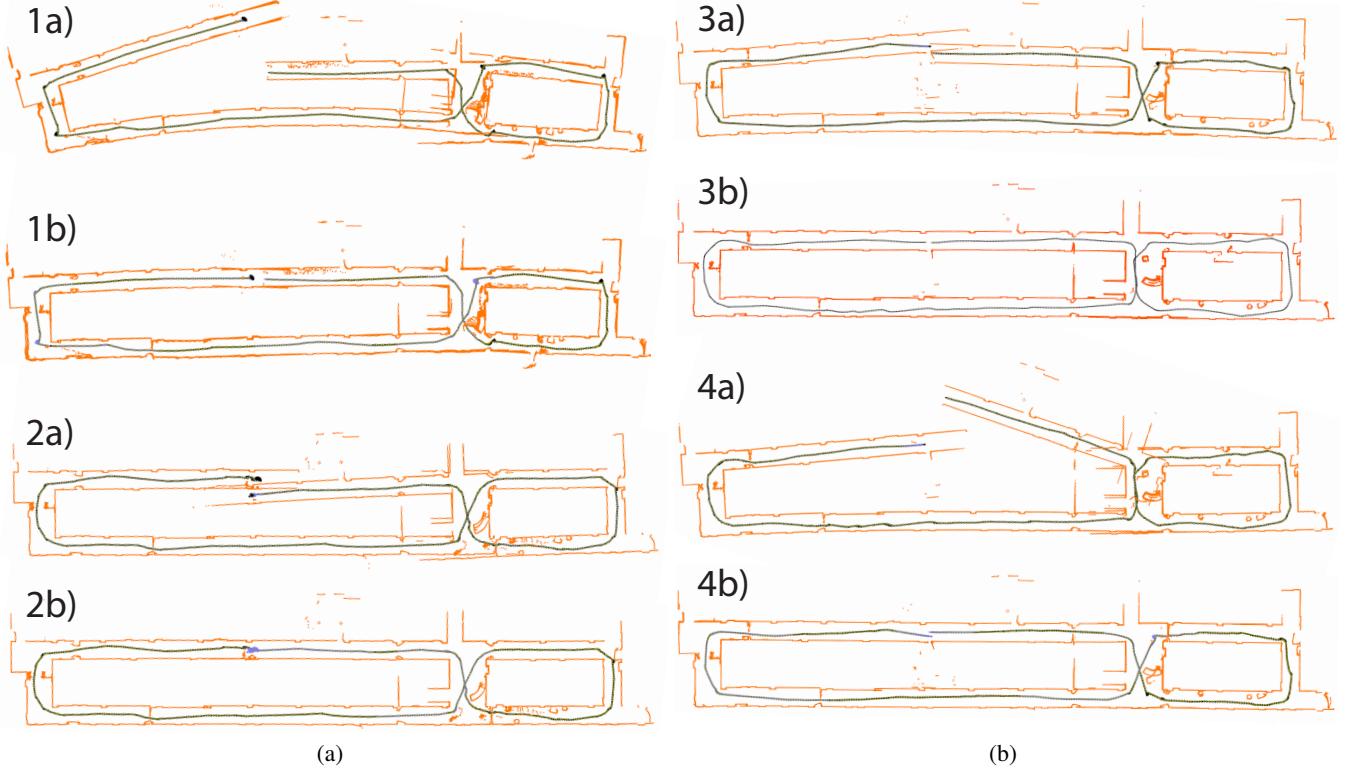


Figure 6: Initial and final maps from HitL-SLAM. Each map is of the same floor, and consist of between 600 and 700 poses. All maps labeled a) are the initial maps, and all those labeled b) are the final maps. Observations are shown in orange and poses are shown as arrows. Poses which are part of a human constraint are blue, while those which are not are in black. Note the varying degree of degradation in each map; some took only a couple human inputs, while others required closer to 10. Our factor graph formulation allows the optimizer to reach minima in an acceptable amount of time, even when there are hundreds of poses involved in human constraints.

Furthermore, this data set contains 2 features which may not be solved by re-deployment. One is a severely bent hallway, in subfigure 1a), and the other is a sensor failure, in subfigure 4a) which caused the robot to incorrectly estimate its heading by roughly 30 degrees between two poses about a fifth of the way into its deployment. Combined, these results show that incorporating human input into metric mapping can be done in a principled, computationally tractable manner, which allows us to solve some types of metric mapping problems in less time and with higher accuracy than previously possible, at the expense of a small amount of human input.

## 7 Conclusion

We present Human-in-the-Loop SLAM (HitL-SLAM), an algorithm designed to leverage human ability and meta-knowledge as they relate to the data association problem for robotic mapping. HitL-SLAM contributes a generalized framework for interpreting human input using the EM algorithm, as well as a factor graph based algorithm for incorporating human input into pose-graph SLAM. Future work in this area should proceed towards further minimizing the human requirements, and extending this method for higher dimensional SLAM and for different sensor types.

## References

- Aulinas, J.; Petillot, Y. R.; Salvi, J.; and Lladó, X. 2008. The slam problem: a survey. In *CCIA*, 363–371. Citeseer.
- Bailey, T., and Durrant-Whyte, H. 2006. Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics & Automation Magazine* 13(3):108–117.
- Christensen, H. I., and Topp, E. A. 2010. Detecting region transitions for human-augmented mapping.
- Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* 39(1):1–38.
- Dissanayake, G.; Huang, S.; Wang, Z.; and Ranasinghe, R. 2011. A review of recent developments in simultaneous localization and mapping. In *2011 6th International Conference on Industrial and Information Systems*, 477–482. IEEE.
- Doroodgar, B.; Ficocelli, M.; Mobedi, B.; and Nejat, G. 2010. The search for survivors: Cooperative human-robot interaction in search and rescue environments using semi-autonomous robots. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2858–2863. IEEE.
- Dubbelman, G., and Browning, B. 2015. Cop-slam: Closed-

- form online pose-chain optimization for visual slam. *IEEE Transactions on Robotics* 31(5):1194–1213.
- Kim, S.; Cheong, H.; Park, J.-H.; and Park, S.-K. 2009. Human augmented mapping for indoor environments using a stereo camera. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5609–5614. IEEE.
- Kleiner, A.; Dornhege, C.; and Dali, S. 2007. Mapping disaster areas jointly: Rfid-coordinated slam by humans and robots. In *2007 IEEE International Workshop on Safety, Security and Rescue Robotics*, 1–6. IEEE.
- Milella, A.; Dimiccoli, C.; Cicirelli, G.; and Distante, A. 2007. Laser-based people-following for human-augmented mapping of indoor environments. In *Artificial Intelligence and Applications*, 169–175.
- Murphy, R. R. 2004. Human-robot interaction in rescue robotics. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 34(2):138–153.
- Nieto-Granda, C.; Rogers, J. G.; Trevor, A. J.; and Christensen, H. I. 2010. Semantic map partitioning in indoor environments using regional analysis. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 1451–1456. IEEE.
- Nourbakhsh, I. R.; Sycara, K.; Koes, M.; Yong, M.; Lewis, M.; and Burion, S. 2005. Human-robot teaming for search and rescue. *IEEE Pervasive Computing* 4(1):72–79.
- Olson, E.; Strom, J.; Goeddel, R.; Morton, R.; Ranganathan, P.; and Richardson, A. 2013. Exploration and mapping with autonomous robot teams. *Communications of the ACM* 56(3):62–70.
- Parasuraman, R.; Barnes, M.; Cosenzo, K.; and Mulgund, S. 2007. Adaptive automation for human-robot teaming in future command and control systems. Technical report, DTIC Document.
- Thrun, S., et al. 2002. Robotic mapping: A survey. *Exploring artificial intelligence in the new millennium* 1:1–35.
- Topp, E. A., and Christensen, H. I. 2006. Topological modelling for human augmented mapping. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2257–2263. IEEE.
- Topp, E. A.; Huettenrauch, H.; Christensen, H. I.; and Ek-lundh, K. S. 2006. Bringing together human and robotic environment representations-a pilot study. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4946–4952. IEEE.
- Zender, H.; Jensfelt, P.; Mozos, O. M.; Kruijff, G.-J. M.; and Burgard, W. 2007. An integrated robotic system for spatial understanding and situated interaction in indoor environments. In *AAAI*, volume 7, 1584–1589.