2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Robust low-overlap 3-D point cloud registration for outlier rejection

John Stechschulte,[1] Nisar Ahmed,[2] and Christoffer Heckman[1]

***Abstract*— When registering 3-D point clouds it is expected that some points in one cloud do not have corresponding points in the other cloud. These non-correspondences are likely to occur near one another, as surface regions visible from one sensor pose are obscured or out of frame for another. In this work, a hidden Markov random field model is used to capture this prior within the framework of the iterative closest point algorithm. The EM algorithm is used to estimate the distribution parameters and learn the hidden component memberships. Experiments are presented demonstrating that this method outperforms several other outlier rejection methods when the point clouds have low or moderate overlap.**

I. Introduction

Depth sensing is an increasingly ubiquitous technique for robotics, 3-D modeling and mapping applications. Structured light sensors, lidar, and stereo matching produce point clouds, or 3-D points on surfaces. Algorithmic registration of point clouds is needed to use the resulting data, often without the aid of other measurements. Iterative closest point (ICP) [1] is commonly used for this purpose, although this time-tested approach has its drawbacks, including failure when registering clouds with low overlap [2], [3]. However, many applications of 3-D registration would be more efficient with low overlap clouds, such as combining partial scans into a complete 3-D model. To address this gap, we present a probabilistic model using a hidden Markov random field (HMRF), shown in Fig. 1, for inferring via the EM algorithm which points lie in the overlap. A successful alignment of clouds with low overlap is shown in Fig. 2.

推测哪个点在overlap中，这种方法可以用于判断alignment是否判断好，我感觉是个很好的方法；

ICP attempts to recover the optimal transformation to align two point clouds. The algorithm recovers the transformation that moves the "free" point cloud onto the "fixed" point cloud by iteratively:

1) finding the closest point in the fixed cloud to each point in the free cloud;
2) discarding some of these matches as outliers; and
3) computing and applying the rigid transform that optimally aligns the remaining points (the inliers), minimizing some measure of the nearest point distances found in step 1;

until convergence. The present work focuses on step 2 of the above sequence.

[1] Department of Computer Science, University of Colorado, Boulder, John.Stechschulte@colorado.edu, Christoffer.Heckman@colorado.edu

[2] Department of Aerospace Engineering, University of Colorado, Boulder, Nisar.Ahmed@colorado.edu

JS was supported by the U.S. Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) program while conducting this work. CH was supported by DARPA award no. N65236–16–1–1000.

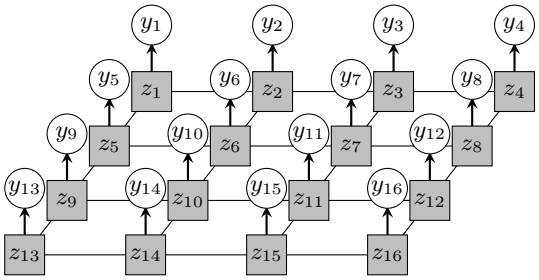

Fig. 1. Graphical model for nearest fixed point distance, shown for a 4×4 grid of pixels in the free depth map. At pixel i, the $Z_i \in \{\pm 1\}$ value is the unobserved inlier/outlier state, and $Y_i \in \mathbb{R}_{\geq 0}$ is the observed distance to the closest fixed point.

Which error measure is minimized in step 3 has been considered extensively [4], and is explored further in Section II. There are closed-form solutions [5] that minimize the point-to-point distance, including using the singular value decomposition [6] and the dual number quaternion method [7]. The present work minimizes the sum of squared point-to-point distances, for simplicity, although it is fundamentally independent of choice of error metric.

A crucial step in ICP is the rejection of outliers (step 2), generally resulting from non-overlapping volumes of space between two measurements. The original ICP formulation [1] does not discard any points and simply incurs error for each outlier. A proliferation of strategies have been proposed for discarding outliers [3], [8], [9], [10], [11], [12]. These methods threshold the residual distance from free points to the nearest point in the fixed cloud, but do not consider the spatial relations amongst points in the free cloud, and what information might be gained by considering whether a point's neighbors are inliers or outliers. In the present work we propose an alternative method where the distribution of residual distances is modeled as a mixture of two distributions: a Gaussian for inliers and a logistic distribution for outliers. Using a technique from image segmentation, a point's inlier or outlier state is modeled as being influenced by the state of its neighbors through a hidden Markov random field—an expectation we refer to as the "neighbor prior." The EM algorithm, with the use of a mean field approximation, allows for inference of the hidden state.

By employing the neighbor prior, the HMRF model can infer which points are outliers in high- and low-overlap cloud pairs. Although exact inference for an MRF model is intractable in applications of reasonable size, the mean field approximation provides sufficient accuracy at a reasonable computational expense. Further justification and development of the neighbor prior is discussed at length in Sections II

978-1-5386-6027-0/19/$31.00 ©2019 IEEE

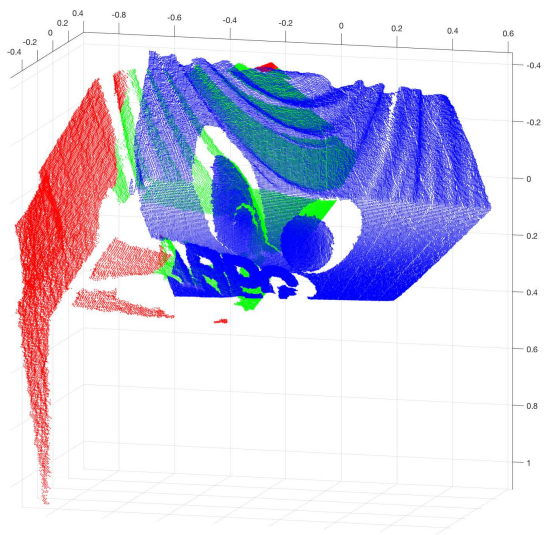

Fig. 2. A successful registration with only 36% overlap. The blue points are the fixed cloud, and the green and red points together are the free cloud, which has been registered. The green points are inliers and the red are outliers, as inferred by the HMRF method.

and III. Sections IV and V present experiments and results on two challenging benchmark dense 3D datasets, showing this method outperforms two state-of-the-art methods, particularly on point clouds with low overlap. To the authors' knowledge, this is the first example of an HMRF model being employed for 3-D registration, and these promising results show that the model is a viable and attractive alternative to other more complex probabilistic scan matching techniques. In particular, the HMRF's relative simplicity and ease of deployment make it scalable and adaptable to different environments where little or no prior knowledge or training data are available.

II. Related Work

Several attempts have been made to improve registration performance in the low-overlap regime. The hybrid genetic/hill-climbing algorithm of Silva, et al. [13] shows success with overlaps down to 55% but at the great computational expense of a stochastic global search. Good low-overlap performance is claimed in [14], which defines a "direction angle" on points and then aligns clouds in rotation using a histogram of these, and in translation using correlations of 2-D projections; although not described as such, a Manhattan world assumption is made. Rotational alignment is recovered using extended Gaussian images in [2], and refined with ICP, showing success with overlap as low as 45%. HMRF ICP is simpler than these methods, and generally performs well at overlaps that are still lower.

Robust statistics or improved error metrics can make ICP more robust to outliers. The point-to-plane [15] metric, which measures the distance projected onto the surface normal of the fixed point, is frequently used and more robust in the case of limited overlap [16]. Sparse norms are used within ICP in [17]. These additions are mostly orthogonal to the work presented, and so could be implemented within HMRF ICP, possibly resulting in further performance improvements.

Improvements to ICP have been achieved through better rejection of outlier correspondences—a thread of research that the current work fits into. Several methods work by thresholding the residual distances, and differ simply in how the threshold is determined: fixed distance [8], residual percentile [3], standard deviations from the mean residual [9], or median absolute deviations from the median residual [10]. These methods inherently assume a large overlap fraction, so are brittle to overlap variations. Zhang [11] presents an adaptive threshold tuned with a distance parameter that does not have direct physical significance. Fractional ICP dynamically adapts the fraction of point correspondences to be used [18], although this method explicitly assumes a large fraction of overlap, with a penalty for smaller overlap fractions. Finally, there are a few outlier-rejection methods that cannot be summarized as simple thresholds. Enqvist, et al. [12] leverage the fact that distances between corresponding points within a cloud will be invariant under rigid motion and find the largest set of consistent correspondences to identify inliers. In [19], points are classified as inliers or one of three classes of outliers: occluded, unpaired (outside the frame), or outliers (sensor noise). The current work is more closely related to the simple threshold methods, but uses a probabilistic model that better captures the expected residual distribution.

In contrast to the current method, EM-ICP [20], [21] employs EM to recover point correspondences. The correspondence is the hidden variable and the E-step computes assignment probabilities, and the implied rigid transformation is recovered in the M-step. This approach assumes that *every* point in the free cloud corresponds to some point in the fixed cloud, although it may be straightforward to include an "unassigned" value to the hidden variable states.

Since ICP requires an informed initialization, significant work has been devoted to achieving global registration or finding an approximate alignment from any initial state, which is then refined with ICP. In Super 4PCS [22], [23], an initial alignment is found by matching sets of 4 coplanar points, using ratios invariant to rigid transformation; good performance with low overlap is claimed, and is used as a basis for comparison in this work. Fast global registration between two or more point clouds is achieved in [24] by finding correspondences only once based on point features, then using robust estimators to mute the impact of spurious correspondences. Branch-and-bound algorithms can guarantee global optimality, and several variations have been applied to the registration problem to search over transformations [25] or over point or feature correspondences [26]. With the exception of Super 4PCS, low-overlap performance is not addressed in these works, and many require final refinement, thereby offering an opportunity to apply HMRF ICP.

Finally, Ramos, et al. [27] and Sun, et al. [28] use conditional random fields to discriminatively match 2-D lidar scans in an ICP setting with impressive and reliable results. Although extending the method to 3-D registration is theoret-

ically straightforward, the computational cost would increase significantly. These methods use an AdaBoost classifier to combine several derived features, which requires training the classifier with ground truth alignments. The classifier may not generalize well to new observations that are not drawn from the same distribution as the training data. In contrast, the generative neighbor prior leveraged in this work is a direct consequence of sensor geometry, and thus does not require training data.

Few existing methods specifically address the problem of aligning clouds with low overlap—more often, high overlap is assumed. By using an appropriate probabilistic model that captures the neighbor prior, this assumption need not be made, and the resulting method works equally well with high and low overlap.

III. Probabilistic Neighbor Prior Model

Let $B_i \in \mathbb{R}^4$ be a point in the free cloud in homogeneous coordinates. In the first step of ICP, we find the closest point to B_i in the fixed cloud, which we will denote $C_j \in \mathbb{R}^4$. The distance between them, $y_i = \|B_i - C_i\|$, is the observed residual. $\boldsymbol{Y}$ will denote all Y_i random variables, with specific instantiations represented as $\boldsymbol{y}$ and y_i, respectively.

Many sources of depth data generate data with a 2-D lattice structure—the pixel grid—in which each pixel has four nearest neighbors. We exploit these neighbor relations to model the distribution of the closest point distances $\boldsymbol{Y}$. Neighbor relations can also be defined on unstructured point clouds, as discussed below, although a grid topology is more intuitive.

Given observed distances $\boldsymbol{y}$, we wish to decide which points are inliers. Our prior beliefs are: (i) inliers will generally lie closer to their respective closest point than outliers; and (ii) neighbors of inliers are likely inliers, and neighbors of outliers are likely outliers—the neighbor prior. To capture these priors, we model the distribution of $\boldsymbol{Y}$ as a mixture of two distributions, one for inliers and one for outliers, where a point's mixture membership is dependent on its nearest neighbors. That is, we capture the second prior using a hidden Markov random field on the inlier/outlier state of a point.

A. Probabilistic model

The graphical model for data with a grid topology is shown in Fig. 1. The distribution of $\boldsymbol{Y}$ conditionally depends on the hidden field $\boldsymbol{Z}$, which is Gibbs distributed with a parameter β that controls the strength of the neighbor influence. The Gibbs distribution is calculated based on the energy of a given configuration, $P_G(\boldsymbol{Z}) = W^{-1}\exp(-H(\boldsymbol{Z}))$, where W is a normalization term called the "partition function", $W = \sum_{\boldsymbol{z}} \exp(-H(\boldsymbol{z}))$.

We use the energy function $H(\boldsymbol{z}) = -\beta \sum_{i' \sim i} w_{i,i'} z_i z_{i'}$ where $w_{i,i'}$ is the edge weight, and $\beta \geq 0$ is a parameter controlling the interaction strength. If all neighbor relations are considered equally, such as in a grid topology with no privileged edges, the weights are all 1. More generally, they can capture the strength of a particular neighbor interaction.

Function `ICP` (*Initial transform* T_{init}, *clouds* $\boldsymbol{B}$ *and* $\boldsymbol{C}$, *field parameter* β, *thresholds*) **:**
- KDtree = BuildKdTree($\boldsymbol{C}$)
- $T = T_{\text{init}}$
- $\boldsymbol{B} = T \times \boldsymbol{B}$
- $\boldsymbol{I}, \boldsymbol{y}$ = KDtree.NearestNeighbors($\boldsymbol{B}$)
- Initialize $\tilde{\boldsymbol{z}}$ to 1 except for highest 10% of $\boldsymbol{y}$, which are initialized to -1.
- **do**
  - **do**
    - θ = M-step($\boldsymbol{y}, \tilde{\boldsymbol{z}}$)
    - $\tilde{\boldsymbol{z}}$ = E-step($\boldsymbol{y}, \tilde{\boldsymbol{z}}, \theta, \beta$)
  - **while** *some* $\tilde{\boldsymbol{z}}$ *value changed sign*
  - T_{step} = localize($\boldsymbol{B}, \boldsymbol{C}, \tilde{\boldsymbol{z}}$)
  - $T = T_{\text{step}} \times T$
  - $\boldsymbol{B} = T_{\text{step}} \times \boldsymbol{B}$
  - $\boldsymbol{I}, \boldsymbol{y}$ = KDtree.NearestNeighbors($\boldsymbol{B}$)
- **while** T_{step} *sufficiently large*
- **return** T

Function `E-step` $(\boldsymbol{y}, \tilde{\boldsymbol{z}}, \theta, \beta)$ **:**
- Calculate update, $\tilde{\boldsymbol{z}} \leftarrow \mathbb{E}[\boldsymbol{Z}|\boldsymbol{y}, \tilde{\boldsymbol{z}}, \theta]$
- **return** $\tilde{\boldsymbol{z}}$

Function `M-step` $(\boldsymbol{y}, \tilde{\boldsymbol{z}})$ **:**
- Calculate MLE of θ from complete data log likelihood, assuming $\mathbb{E}[\boldsymbol{Z}] = \tilde{\boldsymbol{z}}$
- **return** θ

Algorithm 1: The full HMRF ICP algorithm. Pyramiding has been omitted for clarity, as well as iteration limits on loops.

Note that calculation of W, and therefore exact calculation of $P_G(\boldsymbol{z})$, requires a sum over all possible configurations, so is exponential in the number of nodes. For a nontrivial field this is intractable. We avoid this issue with the mean field approximation [29], which assumes a fixed configuration $\tilde{\boldsymbol{z}} = \mathbb{E}[P_G(\boldsymbol{Z}|\beta)]$ and approximates the Gibbs distribution with independent components conditioned on $\tilde{\boldsymbol{z}}$,

$$P_G(\boldsymbol{Z}|\beta) \approx \prod_i P_{\text{mf}}(Z_i|\beta, \tilde{\boldsymbol{z}}). \tag{1}$$

As the components are independent, it is no longer necessary to exhaust over $\boldsymbol{z}$ configurations. The components also depend only on local information:

$$P_{\text{mf}}(z_i|\beta, \tilde{\boldsymbol{z}}) = \frac{\exp\left(\beta \sum_{i' \sim i} w_{i,i'} z_i \tilde{z}_{i'}\right)}{\exp\left(\beta \sum_{i' \sim i} w_{i,i'}(+1)\tilde{z}_{i'}\right) + \exp\left(\beta \sum_{i' \sim i} w_{i,i'}(-1)\tilde{z}_{i'}\right)}. \tag{2}$$

We assume that inliers are normally distributed and outliers are logistically distributed (as discussed further below). The approximate complete data likelihood can be written,

$$f_{\text{mf}}(\boldsymbol{Y}, \boldsymbol{Z}|\beta, \theta, \tilde{\boldsymbol{z}}) = \prod_i P_{\text{mf}}(z_i|\beta, \tilde{\boldsymbol{z}}) \left(N(y_i|\mu_{+1}, \sigma_{+1})\right)^{(1+z_i)/2} \left(L(y_i|\mu_{-1}, s_{-1})\right)^{(1-z_i)/2}, \tag{3}$$

with parameters β, μ_{-1}, s_{-1}, μ_{+1}, σ_{+1}. We will use θ to represent all Gaussian and logistic parameters, that is, $\theta = \{\mu_{-1}, s_{-1}, \mu_{+1}, \sigma_{+1}\}$.

B. Applying the EM algorithm

The maximum likelihood model parameters and hidden state are estimated using the EM algorithm, similarly to the image segmentation model presented in [30], [31], [32]. In the E-step, the current estimate of the normal and logistic parameters, along with the current mean field and the observed residuals, are used to find the expected value of the hidden field. Then, in the M-step, the estimates for the normal and logistic parameters are updated using the maximum likelihood estimates from the observed data and the expected value of the hidden field. The full HMRF ICP algorithm is shown in Algorithm 1.

The complete data log likelihood can be bounded below by taking the expectation with respect to the hidden $\boldsymbol{Z}$ values, due to Jensen's inequality. Because of linearity of expectation, and since z_i appears linearly in the complete data log likelihood, this amounts to replacing z_i with $\mathbb{E}[z_i|y_i, \tilde{\boldsymbol{z}}, \theta, \beta] = 2P(z_i = 1|y_i, \tilde{\boldsymbol{z}}, \theta, \beta) - 1$ throughout Equation 3. The EM algorithm iteratively calculates these probabilities in the E-step, and then chooses parameters to maximize the resulting expected likelihood in the M-step.

1) The E-step: The mean field approximation allows for a closed-form E-step, where we calculate,

$$P(z_i = +1|y_i, \tilde{\boldsymbol{z}}, \theta, \beta) \propto \exp\left(\beta \sum_{i' \sim i} w_{i,i'} \tilde{z}_{i'}\right) \frac{1}{\sqrt{2\pi\sigma_{+1}^2}} \exp\left(\frac{(y_i - \mu_{+1})^2}{2\sigma_{+1}^2}\right) \tag{4}$$

$$P(z_i = -1|y_i, \tilde{\boldsymbol{z}}, \theta, \beta) \propto \exp\left(-\beta \sum_{i' \sim i} w_{i,i'} \tilde{z}_{i'}\right) \frac{\exp\left(-\frac{y_i - \mu_{-1}}{s}\right)}{s\left(1 + \exp\left(-\frac{y_i - \mu_{-1}}{s}\right)\right)^2} \tag{5}$$

The necessary normalization factor is the same for the two calculations, so is simply their sum.

2) The M-step: Now, the expected hidden values can be used to update the mean field and problem parameters in the M-step using maximum likelihood estimates. Updating the mean field is simply adopting the expected z_i values calculated in the E-step. The MLEs of the normal and logistic distribution parameters are calculated in the normal way, but the data are weighted by the probabilities that they came from the given distribution. For the inliers,

$$n_{+1} = \sum_i \left(\frac{1 + z_i}{2}\right) \tag{6}$$

$$\mu_{+1} = \frac{1}{n_{+1}} \sum_i \frac{1 + z_i}{2} y_i \tag{7}$$

$$\sigma_{+1} = \sqrt{\frac{1}{n_{+1}} \sum_i \frac{1 + z_i}{2} y_i^2 - \mu_{+1}^2}. \tag{8}$$

For the outliers,

$$n_{-1} = \sum_i \left(\frac{1 - z_i}{2}\right) \tag{9}$$

$$\mu_{-1} = \frac{1}{n_{-1}} \sum_i \frac{1 - z_i}{2} y_i \tag{10}$$

$$s_{-1} = \frac{\sqrt{3}}{\pi} \sqrt{\frac{1}{n_{-1}} \sum_i \frac{1 - z_i}{2} y_i^2 - \mu_{-1}^2}. \tag{11}$$

C. Fitting the outlier distribution

It was observed that some outlier distributions had multiple modes or heavy tails, as distant regions were observed. To better model this distribution, outlier residuals for 100 pairs of frames from each of the three datasets described in Section IV were analyzed. A Kolmogorov-Smirnov statistic was used to compare the empirical CDF with the normal distribution and six heavy-tail distributions: the gamma, logistic, log-logistic, log-normal, Rayleigh, and Student's t distributions. No distribution outperforms all others in all cases, but the logistic distribution shows good average performance and is simple to estimate within the EM framework. Nevertheless, the worst cloud pairs still fail to align—in particular, those with multimodal outlier residuals. A mixture model would likely be more effective in these cases, but would also be significantly more computationally expensive.

D. Accelerating EM convergence

The EM algorithm is accelerated via a pyramid method: we downsample the $\boldsymbol{Z}$ and $\boldsymbol{Y}$ fields, allow EM to converge for this smaller model, then upsample the converged $\boldsymbol{Z}$ to initialize the larger model. For grid topologies, each pyramid level has a quarter the number of pixel sites as the level below it (half in each dimension). For unstructured clouds, discussed below, each pyramid level has half the number of points as the level below it. Although pyramiding is a common technique in computer vision, such as in the pyramidal Lucas Kanade optical flow algorithm [33], this is the first application to initializing a Markov random field.

E. Unstructured clouds

The pixel grid provides an intuitive structure on which to define the Markov random field, but the method can be extended to any point cloud by defining neighbor relations independent of underlying topology. We measure the "neighborliness" of two points by their distance, and apply a Gaussian kernel, $\exp\left(-\|B_i - B_j\|/2\sigma^2\right)$, to generate an appropriate edge weight. The σ parameter is set to half the average nearest-neighbor distance.

Calculating weights for every pair of points would require N^2 work, but only the nearest neighbors to a given point will have non-negligible values, by design. Thus, neighbor weights are only calculated for the k-nearest neighbors of each point, with $6 \leq k \leq 10$ giving satisfactory experimental results. Applying the pyramid acceleration to this neighborhood structure, however, is not straightforward. The obvious downsampling and upsampling methods available with a grid cannot be used for an arbitrary graph. Instead, we sample half

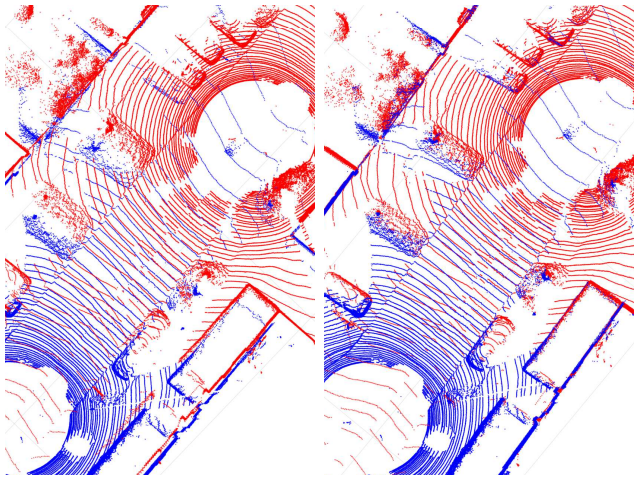

Fig. 3. Successful registration of two lidar spins, taken from the KITTI odometry dataset [34]. On the left, the initial registration from the RTK GPS poses. On the right, the registration refined by HMRF ICP. No information about the data topology was used.

the points at random to build a new layer on the pyramid. The new layer needs neighbor weights, as well, which we calculate by squaring the neighbor matrix for the lower level, with diagonal elements set to 0. This captures the two-hop weight between points in the lower neighbor matrix. Finally, once EM converges in a pyramid layer, the resulting $\boldsymbol{Z}$ field is upsampled using the lower-level neighborhood matrix to initialize the points that were not sampled in the higher-level neighborhood.

Figure 3 shows the successful alignment of two lidar spins, taken from the KITTI odometry dataset [34]. The ground truth poses in this dataset are from RTK GPS and IMU measurements, which have high accuracy over long sequences, but are not sufficiently accurate for the purpose of our experiments, described below.

IV. Experiments

The HMRF method is compared with Go-ICP [35] and Super 4PCS [23], as well as ICP with five other outlier rejection methods: (1) no outlier rejection, (2) keeping 90% of the points with smallest residuals, (3) keeping points whose residuals were less than 2.5 standard deviations above the mean, (4) keeping points whose residuals were less than 5.2 median absolute deviations above the median (the so-called "X84" criterion), and (5) the dynamic threshold from Zhang [11]. All of the ICP implementations (including HMRF ICP) use the point-to-point sum of squares distance metric.

The method is applied to several datasets: the shark sequence is a tabletop scene, taken by an Asus Xtion Pro; the remaining experiments were run on ten publicly available RGB-D SLAM datasets [36] from the TUM Computer Vision Group (specifically, those datasets in the "Handheld SLAM" category). Ground truth poses are known for both sequences. The sensor in the desk and room sequences is tracked using an external motion tracking system. The poses in the shark sequence are those estimated in the tracking stage of InfiniTAM[37] during the reconstruction of the scene. The camera calibrations are known and used to appropriately unproject the depth maps and generate point clouds. In the shark sequence, 100 pairs of frames were selected at random, but stratified to include a variety of overlap ratios. Similarly, 90 frames were selected from the TUM datasets: one each from each dataset and overlap decile, except 0–10%. The overlap is estimated using the ground truth poses and the relative distance to each free point's nearest neighbor in its own cloud versus in the fixed cloud. The threshold for this comparison was chosen to err on the side of overestimating the overlap ratio.

The frame pairs were aligned using ground truth poses, and translated so that the coordinate origin was at the centroid of the fixed frame. Then, the free cloud alignment was perturbed by rotating 18 degrees about a random axis through the origin. The same initialization was used for all methods for a given frame pair (including the global methods, Go-ICP and Super 4PCS). HMRF ICP was configured with 4 pyramid levels. Before the first ICP transformation was calculated, EM was limited to 150 iterations at each pyramid level, although this limit was rarely met; after applying the first transformation, EM was limited to 5 steps at each pyramid level. All experiments were executed in MATLAB on a workstation with 8 Intel® Xeon® E5620 CPUs at 2.40GHz, and with 48 GB RAM. The Go-ICP and Super 4PCS implementations are available from the authors [35], [23]. Both are written in C++ and are single-threaded. Parameters for these methods were chosen based on the provided demos, and their execution time was limited to 120 seconds. The Super 4PCS algorithm requires a prior estimate of the cloud overlap, which was provided from the overlap estimate described above. All code can be found at `https://github.com/JStech/ICP`.

V. Results

The results were noisy and no method was consistently best. Complete results are available in the supplementary materials. HMRF ICP performed best among the ICP variants, so we compare it to Go-ICP and Super 4PCS here.

Fig. 4 shows summary plots of the rotation error, translation error, and elapsed time for each method. At high overlaps, Go-ICP often recovered the transformation most accurately. However, its performance deteriorated quickly as the overlap decreased: only 7 Go-ICP tests with overlaps below 70% ran to completion within the time limit and returned transformations. Super 4PCS performed well at moderate overlaps, and often recovered the most accurate transformation in rotation and translation for overlaps around 50%. At very low overlaps (below 20%), no method performed reliably. Super 4PCS was very fast at moderate-to-high overlaps, even running single-threaded. However, it had a higher variance than HMRF ICP, in both accuracy and elapsed time. Note that the elapsed time of HMRF ICP cannot be compared directly with that of Go-ICP and Super 4PCS, as HMRF ICP was implemented entirely in MATLAB (and was thus able to take advantage of built-in

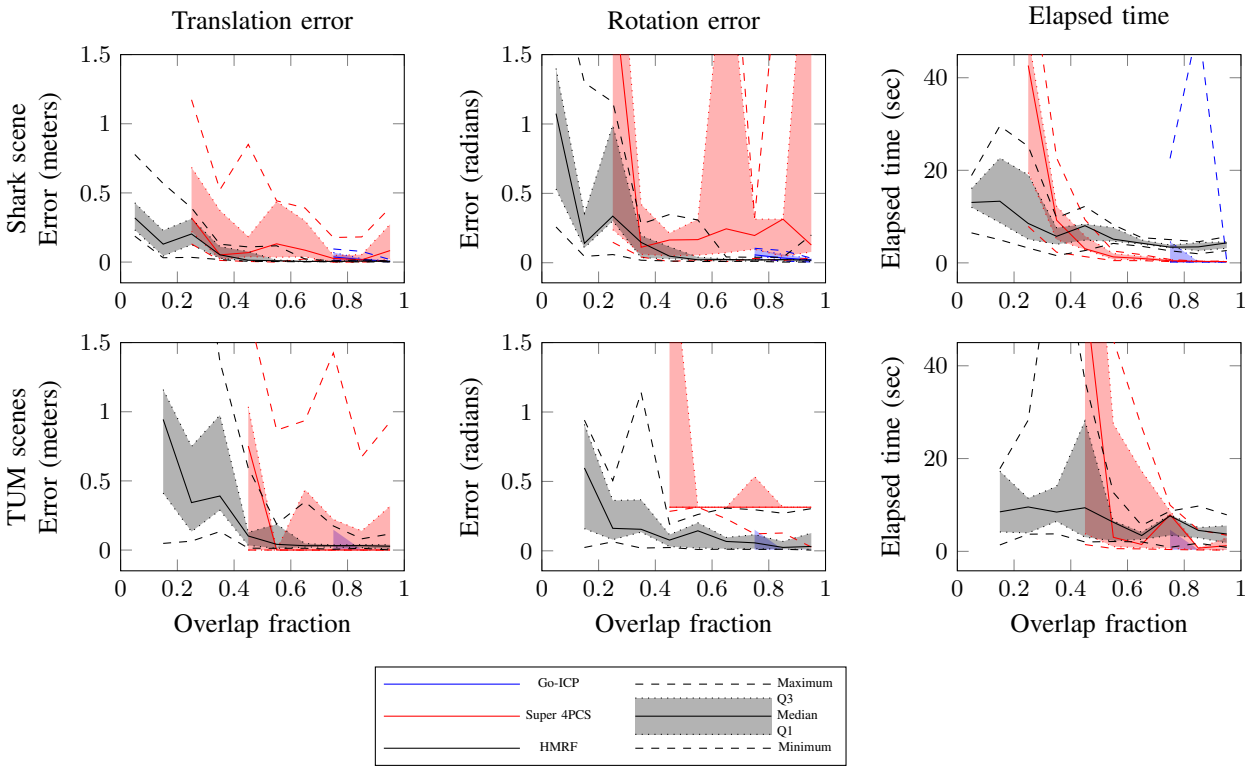

Fig. 4. Error and elapsed time plots for shark scene (top) and all TUM scenes (bottom). For each method, the data are aggregated by overlap decile, and the minimum, first quartile, median, third quartile, and maximum are shown. A method is only plotted if it ran successfully for at least half of the cases in the decile: hence the truncated Go-ICP and Super 4PCS plots.

multithreading optimizations), whereas Go-ICP and Super 4PCS were single-threaded in C++ (although then had the advantage of being compiled). However, the performance trends can still be reliably understood: all methods perform more slowly as overlap fraction drops, although HMRF ICP's performance does not deteriorate as quickly as the other methods.

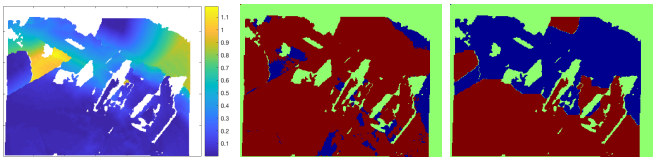

Fig. 5. Example $\boldsymbol{Y}$, $\tilde{\boldsymbol{z}}^{(1)}$, and $\tilde{\boldsymbol{z}}^{(375)}$ (at convergence). This frame has 36% overlap with the fixed frame. The left image shows the observed distance to the nearest fixed point in the initial configuration; yellow is farther, blue is nearer, and the white areas are unobserved. In the right two images, blue pixels are outliers and red pixels are inliers, green pixels are unobserved.

VI. Discussion and Conclusions

The good performance of HMRF ICP at low overlaps can be understood by considering the example frame with 36% overlap shown in Fig. 5. The three images all represent values before the first transformation is applied to the free cloud: the first is the residual distance to the nearest fixed point, the second is the initial setting of the $\tilde{\boldsymbol{z}}$ field, and the third is the converged $\tilde{\boldsymbol{z}}$ field before the first iteration of ICP. The HMRF model flexibly adapts to the small proportion of inlier points, in particular it eliminates outliers across the top of the image. The pixels that are unobserved (that is, the sensor returns no depth measurement) occupy 31% of the image. In the initial $\tilde{\boldsymbol{z}}$ field, 90% of observed pixels are considered inliers, and the remaining 10% are considered outliers. After 375 initial EM iterations, the $\tilde{\boldsymbol{z}}$ field has converged, and now has only 54% inliers and 46% outliers. By eliminating these outliers before the first transformation is calculated, divergence from the nearby optimum is avoided. An example alignment is shown at `https://youtu.be/w4eVOgd7Zes`.

Despite the use of the logistic distribution to model outliers, there were still cases where alignment failed because of multimodal outlier residuals. To address this failure mode, a still better representation of the residual distributions for outliers is necessary. For instance, modeling the outliers as a Gaussian mixture of several components could allow the field to fit the distant outliers with one Gaussian, and the nearer outliers as another. Using robust norms [17], [38] and the point-to-plane metric [15] would also make HMRF ICP more robust to imperfect estimations of which points lie in the overlap of the two clouds.

The HMRF model for the overlap of point clouds being aligned via ICP has demonstrated advantages at low overlap without sacrificing performance at high overlap. The HMRF model captures the neighbor prior and describes observed inlier/outlier behavior well, and so can adapt to the particular clouds being aligned. This would prove useful in the construction of models from 3-D scanner data, as fewer scans would be required, or aligning depth readings at a low frame rate, allowing greater differences between frames.

REFERENCES

[1] P. J. Besl, N. D. McKay, *et al.*, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[2] A. Makadia, A. Patterson, and K. Daniilidis, "Fully automatic registration of 3D point clouds," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, pp. 1297–1304, IEEE, 2006.

[3] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, "The trimmed iterative closest point algorithm," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 3, pp. 545–548, IEEE, 2002.

[4] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pp. 145–152, IEEE, 2001.

[5] D. W. Eggert, A. Lorusso, and R. B. Fisher, "Estimating 3-D rigid body transformations: a comparison of four major algorithms," *Machine Vision and Applications*, vol. 9, no. 5-6, pp. 272–290, 1997.

[6] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 5, pp. 698–700, 1987.

[7] M. W. Walker, L. Shao, and R. A. Volz, "Estimating 3-D location parameters using dual number quaternions," *CVGIP: Image Understanding*, vol. 54, no. 3, pp. 358–367, 1991.

[8] G. Turk and M. Levoy, "Zippered polygon meshes from range images," in *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, pp. 311–318, ACM, 1994.

[9] T. Masuda, K. Sakaue, and N. Yokoya, "Registration and integration of multiple range images for 3-D model construction," in *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, vol. 1, pp. 879–883, IEEE, 1996.

[10] A. Fusiello, U. Castellani, L. Ronchetti, and V. Murino, "Model acquisition by registration of multiple acoustic range views," *Computer VisionECCV 2002*, pp. 558–559, 2002.

[11] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International Journal of Computer Vision*, vol. 13, no. 2, pp. 119–152, 1994.

[12] O. Enqvist, K. Josephson, and F. Kahl, "Optimal correspondences from pairwise constraints," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1295–1302, IEEE, 2009.

[13] L. Silva, O. R. P. Bellon, and K. L. Boyer, "Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 762–776, 2005.

[14] Y. Ma, Y. Guo, J. Zhao, M. Lu, J. Zhang, and J. Wan, "Fast and accurate registration of structured point clouds with small overlaps," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2016.

[15] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pp. 2724–2729, IEEE, 1991.

[16] J. Salvi, C. Matabosch, D. Fofi, and J. Forest, "A review of recent range image registration methods with accuracy evaluation," *Image and Vision Computing*, vol. 25, no. 5, pp. 578–596, 2007.

[17] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," in *Computer Graphics Forum*, vol. 32, pp. 113–123, Wiley Online Library, 2013.

[18] J. M. Phillips, R. Liu, and C. Tomasi, "Outlier robust ICP for minimizing fractional RMSD," in *3-D Digital Imaging and Modeling, 2007. 3DIM'07. Sixth International Conference on*, pp. 427–434, IEEE, 2007.

[19] T. Masuda and N. Yokoya, "A robust method for registration and segmentation of multiple range images," *Computer Vision and Image Understanding*, vol. 61, no. 3, pp. 295–307, 1995.

[20] S. Granger and X. Pennec, "Multi-scale EM-ICP: A fast and robust approach for surface registration," in *European Conference on Computer Vision*, pp. 418–432, Springer, 2002.

[21] J. Hermans, D. Smeets, D. Vandermeulen, and P. Suetens, "Robust point set registration using EM-ICP with information-theoretically optimal outlier handling," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 2465–2472, IEEE, 2011.

[22] D. Aiger, N. J. Mitra, and D. Cohen-Or, "4-points congruent sets for robust pairwise surface registration," *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3, p. 85, 2008.

[23] N. Mellado, D. Aiger, and N. J. Mitra, "Super 4PCS fast global pointcloud registration via smart indexing," in *Computer Graphics Forum*, vol. 33, pp. 205–215, Wiley Online Library, 2014.

[24] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *European Conference on Computer Vision*, pp. 766–782, Springer, 2016.

[25] C. Papazov and D. Burschka, "Stochastic global optimization for robust point set registration," *Computer Vision and Image Understanding*, vol. 115, no. 12, pp. 1598–1609, 2011.

[26] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann, "Robust global registration," in *Symposium on Geometry Processing*, vol. 2, p. 5, 2005.

[27] F. T. Ramos, D. Fox, and H. F. Durrant-Whyte, "CRF-matching: Conditional random fields for feature-based scan matching," in *Robotics: Science and Systems*, 2007.

[28] Z. Sun, J. Van de Ven, F. Ramos, X. Mao, and H. Durrant-Whyte, "Inferring laser-scan matching uncertainty with conditional random fields," *Robotics and Autonomous Systems*, vol. 60, no. 1, pp. 83–94, 2012.

[29] D. Chandler, *Introduction to modern statistical mechanics*. Oxford University Press, 1987.

[30] Z. Kato and T.-C. Pong, "A Markov random field image segmentation model for color textured images," *Image and Vision Computing*, vol. 24, no. 10, pp. 1103–1114, 2006.

[31] J. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 259–302, 1986.

[32] G. Celeux, F. Forbes, and N. Peyrard, "EM procedures using mean field-like approximations for Markov model-based image segmentation," *Pattern Recognition*, vol. 36, no. 1, pp. 131–144, 2003.

[33] J.-Y. Bouguet, "Pyramidal implementation of the affine Lucas Kanade feature tracker," *Intel Corporation*, vol. 5, no. 1-10, p. 4, 2001.

[34] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[35] J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-ICP: a globally optimal solution to 3D ICP point-set registration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2241–2254, 2016.

[36] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[37] O. Kähler, V. A. Prisacariu, C. Y. Ren, X. Sun, P. Torr, and D. Murray, "Very high frame rate volumetric integration of depth images on mobile devices," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 11, pp. 1241–1250, 2015.

[38] P. Mavridis, A. Andreadis, and G. Papaioannou, "Efficient sparse ICP," *Computer Aided Geometric Design*, vol. 35, pp. 16–26, 2015.