A Symmetric Objective Function for ICP

SZYMON RUSINKIEWICZ, Princeton University

The Iterative Closest Point (ICP) algorithm, commonly used for alignment of 3D models, has previously been defined using either a point-to-point or point-to-plane objective. Alternatively, researchers have proposed computationally-expensive methods that directly minimize the distance function between surfaces. We introduce a new symmetrized objective function that achieves the simplicity and computational efficiency of point-to-plane optimization, while yielding improved convergence speed and a wider convergence basin. In addition, we present a linearization of the objective that is exact in the case of exact correspondences. We experimentally demonstrate the improved speed and convergence basin of the symmetric objective, on both smooth models and challenging cases involving noise and partial overlap.

CCS Concepts: • **Computing methodologies → Shape analysis**.

Additional Key Words and Phrases: 3D Registration, 3D Alignment, Iterative Closest Point

ACM Reference Format:
Szymon Rusinkiewicz. 2019. A Symmetric Objective Function for ICP. *ACM Trans. Graph.* 38, 4, Article 85 (July 2019), 8 pages. https://doi.org/10.1145/3306346.3323037

1 INTRODUCTION

Registration of 3D shapes is a key step in both 3D model creation (from scanners or computer vision systems) and shape analysis. For rigid-body alignment based purely on geometry (as opposed to RGB-D), the most common methods are based on variants of the Iterative Closest Point (ICP) algorithm [Besl and McKay 1992]. In this method, points are repeatedly selected from one model, their nearest points on the other model (given the current best-estimate rigid-body alignment) are selected as correspondences, and an incremental transformation is found that minimizes distances between point pairs. The algorithm eventually converges to a local minimum of surface-to-surface distance.

Because ICP-like algorithms can be made efficient and reliable, they have become widely adopted. As a result, researchers have focused on both addressing the shortcomings of ICP and extending it to new settings such as color-based registration and non-rigid alignment. One particular class of improvements has focused on the loss function that is optimized to obtain an incremental transformation. For example, as compared to the original work of Besl and McKay, which minimized *point-to-point* distance, the method of Chen and Medioni [1992] minimized the distance between a point on one mesh and a plane containing the matching point and perpendicular to its normal. This *point-to-plane* objective generally results in faster convergence to the correct alignment and greater ultimate accuracy, though it does not necessarily increase the basin of convergence. Work by Fitzgibbon [2001], Mitra et al. [2004], and Pottmann et al. [2006] showed that both point-to-point and point-to-plane minimization may be thought of as approximations to minimizing the squared Euclidean distance function of the surface, and they presented algorithms that achieved greater convergence speed and stability, albeit at the cost of greater computational complexity and/or auxiliary data structures.

This paper proposes a symmetrized version of the point-to-plane objective for use in ICP, incorporating two key ideas. First, the plane in which the error is minimized is based on the surface normals of *both* points in the corresponding pair. Second, the optimization is performed in a "stationary" coordinate system, while both meshes are moved in opposite directions. These changes require a relatively small modification to the optimization problem being performed, and almost no increased computation per iteration, but result in improved convergence of ICP.

The reason for this improvement is that the symmetric objective is minimized whenever the pair of points lies on a second-order (constant-curvature) patch of surface, rather than being minimized only if the points are on a plane. Thus, we gain some of the same benefits as second-order distance function minimization methods, but without explicit computation of second-order surface properties, or the need for volumetric data structures to store an approximation to the squared Euclidean distance function.

In addition to the primary contribution of the new symmetric objective, we also introduce an alternative approach to linearization of rotations that allows us to reduce the optimization to a linear least-squares problem, while still solving for the exact transformation when correspondences are exact. We conduct experiments that demonstrate both greater per-iteration error reduction and an increase in the convergence basin for our proposed method.

2 RELATED WORK

Since the original ICP algorithms by Besl and McKay [1992] and Chen and Medioni [1992], there have been significant efforts to improve convergence and stability. For a comprehensive overview of many variants, see the surveys by Rusinkiewicz and Levoy [2001], Díez et al. [2015], and Pomerleau et al. [2015]. Much of this work focuses on finding better correspondences (e.g., by matching local surface properties or descriptors), performing outlier-tolerant optimization, or generalizing to non-rigid deformation. Here we focus specifically on methods that modify the objective function and/or the strategy for minimizing it.

Segal et al. [2009] generalize ICP to associate a probabilistic model (in practice, a covariance matrix) with each point. This allows for a "soft plane-to-plane" minimization that improves the

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *ACM Transactions on Graphics*, https://doi.org/10.1145/3306346.3323037.

matching of planar surfaces. Halber and Funkhouser [2017] have explored incorporating additional constraints between planes, such as parallelism or orthogonality, into registration.

Fitzgibbon [2001] proposes to directly minimize the distance between samples on one shape (referred to as "data") and the second shape itself (the "model"). This is done by computing the squared distance transform of the model, evaluating it at data locations, applying a robustifying kernel, and minimizing the result using Levenberg-Marquardt. Mitra et al. [2004] propose two methods for using the distance field of a shape for optimization: one based on local quadratic approximation at closest corresponding points, and the other based on a global hierarchical d^2*-Tree* data structure [Leopoldseder et al. 2003] that stores a bounded-error approximation to the global squared distance field. Pottmann et al. [2006] analyze the theoretical properties of distance-function minimization, and demonstrate its improved convergence.

The variants described above all perform *local* minimization, requiring an initial guess. This may be based on exhaustive search, matching of descriptors (such as spin images [Huber and Hebert 2003] or integral invariants [Gelfand et al. 2005]), or finding constrained point arrangements [Aiger et al. 2008]. In contrast, Yang et al. [2016] combine local registration with a branch-and-bound algorithm that yields a provably globally-optimal solution. The loss function, however, is still based on point-to-point, which is exploited for derivation of the error bounds for global search.

In this paper, we derive an objective that is closest in spirit to simple point-to-plane minimization, but locally converges to zero for quadratic, rather than just planar, patches. This is done by considering the normals of both points in a pair, though we do so in a way unrelated to Segal et al. [2009].

3 METHOD

3.1 Background and Motivation

Consider the problem of aligning surfaces $\mathcal{P}$ and $\mathcal{Q}$. This involves finding a rigid-body transformation $(\mathrm{R}|t)$ such that applying the transformation to $\mathcal{P}$ causes it to lie on top of $\mathcal{Q}$. The original ICP algorithm of Besl and McKay [1992] may be thought of as an instance of Expectation Maximization: the problem is solved by alternately computing pairs of corresponding points (p_i, q_i), where q_i is the *closest* point to p_i given the current transformation, and finding the transformation minimizing the *point-to-point* objective:

$$\mathcal{E}_{point} = \sum_i \left\| \mathrm{R}p_i + t - q_i \right\|^2. \tag{1}$$

Because this iteration converges slowly, authors including Fitzgibbon [2001], Mitra et al. [2004], and Pottmann et al. [2006] have re-cast alignment as iterative minimization of the squared Euclidean distance function of $\mathcal{Q}$, sampled at points p_i. The most accurate way to accomplish this is to pre-compute a data structure that stores at each point in space (an approximation to) the squared distance field, then use it at run-time in an optimization based on Levenberg-Marquardt [Fitzgibbon 2001] or Newton's method [Mitra et al. 2004]. This leads to fast convergence and a wide convergence basin, though at significant computational and storage cost. A simpler approach is to approximate the distance function based on the local surface at each corresponding point q_i. The "on-demand" method of Mitra et al. [2004] approximates the surface as locally quadratic, which requires evaluation of second-order surface properties (i.e., curvatures). Even more straightforward is to approximate the surface around q_i as planar, which only requires evaluation of surface normals $n_{q,i}$. Indeed, this approach dates back to the work of Chen and Medioni [1992], who minimized what has come to be called the *point-to-plane* objective:

$$\mathcal{E}_{plane} = \sum_i \Big[(\mathrm{R}p_i + t - q_i) \cdot n_{q,i} \Big]^2. \tag{2}$$

It can be shown that minimizing this objective is equivalent to Gauss-Newton minimization of squared Euclidean distance.

The latter does indeed improve convergence rate relative to the point-to-point objective, and point-to-plane minimization has become the workhorse of most modern ICP-like implementations. However, point-to-plane ICP has been observed to have a narrower convergence basin than point-to-point [Mitra et al. 2004]. In addition, the residual at optimal alignment is zero only when the surface is locally flat, if the correspondences are not perfect (which is necessarily the case if point sets $\{p_i\}$ and $\{q_i\}$ differ in their sampling of the surface, as with 3D scans). This is important because the zero-set of the objective function defines what transformations are "free," in the sense that the surface is allowed to slide along itself to permit geometric features elsewhere to lock down the transformation. This is exploited by work such as the normal-space sampling of Rusinkiewicz and Levoy [2001], the covariance-eigenvector-directed sampling of Gelfand et al. [2003], and the stable sampling of Brown and Rusinkiewicz [2007], all of which bias the sampling to regions that are most necessary to constrain the aligning transformation.

We therefore seek to develop a new objective function whose zero-set allows a greater class of surfaces to "slide along" themselves at zero penalty. Thinking within the framework of Expectation Maximization, this makes the method as robust as possible to mis-estimation of point correspondences in the Expectation step.

3.2 A Symmetric Objective Function

Because the surfaces $\mathcal{P}$ and $\mathcal{Q}$ should be, up to noise, the same, we consider what residual the objective function will attain if we were to sample a pair of nearby points (p, q) on that surface. In the point-to-plane case, the error is

$$(p - q) \cdot n_q. \tag{3}$$

If we consider the possibility of sampling (p, q) anywhere within some small region of a surface, this will be zero only if the surface is perfectly flat. However, consider the more symmetric function[1]

$$(p - q) \cdot (n_p + n_q). \tag{4}$$

Examining the behavior of this function in 2D (see Figure 1), we see that it is zero whenever p and q are sampled from a circle, since n_p and n_q have opposite projections onto $p - q$. As rigid-body transformations are applied to $\mathcal{P}$, this expression will continue to evaluate

[1] If oriented normals are not available, then n_p and n_q should be adjusted to point in the same direction, by negating either one if their dot product is negative. This does not affect any of the discussion below, and only makes a difference to the outlier rejection strategy described in Section 4.3.

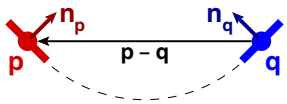

Fig. 1. For any points p and q sampled from a circular arc, the vector between them $p - q$ is perpendicular to the sum of normals $n_p + n_q$. This is the fundamental property exploited by the symmetric ICP formulation.

to zero as long as p and q end up in a relative position consistent with their placement on *some* circle (Figure 2, top). A similar property is true in 3D: Equation 4 evaluates to zero as long as p and q and their normals are consistent with *some cylinder*. Because it is difficult to describe, and especially to visualize, the set of (p, n_p) that lie on arbitrary cylinders containing a fixed (q, n_q) — it is a 4D space — Section 4.1 investigates a different property: Equation 4 also holds whenever p and q are consistent with a locally-second-order surface centered between them. While this constraint still provides a great deal of freedom for (p, n_p) to move relative to (q, n_q), it is a "more useful" form of freedom than provided by the point-to-plane metric. In particular, it constrains (p, n_p) to be consistent with a *plausible* extension of (q, n_q), unlike point-to-plane (Figure 2, bottom). Note that achieving this property does not require the evaluation of any higher-order information (i.e., curvature), which is a major benefit for computational efficiency and noise resistance.

To formulate an objective function, we consider minimizing Equation 4 with respect to transformations applied to the surfaces $\mathcal{P}$ and/or $\mathcal{Q}$. Although most previous work applies a rigid-body transformation to only one of the surfaces (e.g., the transformation in $\mathcal{E}_{plane}$ is applied only to $\mathcal{P}$), we consider a symmetric split of the transformation: we imagine evaluating the metric in a fixed, "neutral" coordinate system and applying *opposite* transformations to $\mathcal{P}$ and $\mathcal{Q}$. Thus, we can formulate a symmetric objective as:

$$\mathcal{E}_{symm\text{-}RN} = \sum_i \left[(\mathrm{R}p_i - \mathrm{R}^{-1}q_i + t) \cdot (\mathrm{R}n_{p,i} + \mathrm{R}^{-1}n_{q,i}) \right]^2, \quad (5)$$

where the final transformation from $\mathcal{P}$ to $\mathcal{Q}$ is now $\mathrm{R} \circ \mathrm{T} \circ \mathrm{R}$ (with T being the translation matrix). We will refer to this as the *rotated-normals* ("-RN") version of the symmetric objective.

This splitting of rotations has a number of advantages when we consider linearizing the objective (Section 3.3):

- It reduces linearization error by optimizing for *half* of the rotation angle.
- It further reduces error, because the error in linearizing $Rp - q$ is proportional to $p \cdot a$, where a is the rotation axis, while the error in linearizing $\mathrm{R}p - \mathrm{R}^{-1}q$ is proportional to $(p - q) \cdot a$. Except for extreme misalignment, $p - q$ is usually smaller than p (i.e., the distance from p to the origin).
- It enables reduction of linearization error to zero when correspondences are exact (Section 4.2).

We have also explored a simpler version of this objective in which the normals are not rotated. That is, the direction of minimization per point pair remains fixed, while the points themselves are rotated in opposite directions:

$$\mathcal{E}_{symm} = \sum_i \left[(\mathrm{R}p_i - \mathrm{R}^{-1}q_i + t) \cdot (n_{p,i} + n_{q,i}) \right]^2. \quad (6)$$

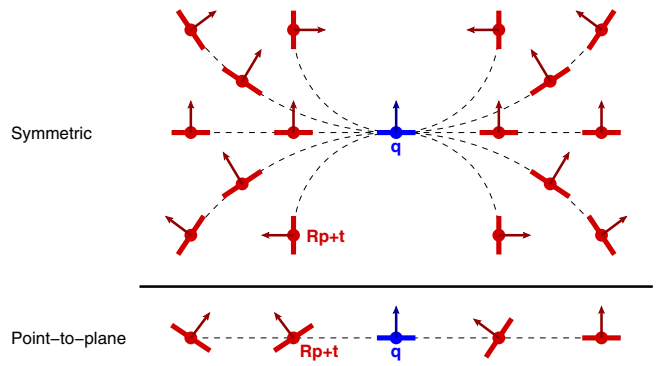

Fig. 2. **Top:** As p moves relative to q, the property $(p - q) \cdot (n_p + n_q) = 0$ holds as long as there is *some* circular arc with which p, q, n_p, and n_q are consistent. **Bottom:** This is in contrast to the point-to-plane metric, which is zero when p is in the plane defined by q and n_q, regardless of n_p.

Why might this be a reasonable simplification to make? Consider the sum of two unit-length vectors in 2D. Applying opposite rotations to the vectors preserves the direction of their sum, so that the contribution of each point pair to the two variants of the objective would be the same up to a scale. In 3D, this is not true for all rotation axes, but approaches true as n_p approaches n_q. The experiments in Section 4.3 show that the two objectives lead to similar convergence, but $\mathcal{E}_{symm}$ leads to simpler derivations and implementation. Therefore, the remainder of this paper adopts $\mathcal{E}_{symm}$ as the *symmetric objective*.

3.3 Linear Approximation

The traditional method for converting an objective function involving rotations into an easily-optimized linear least-squares system is to make the approximations $\cos\theta \sim 1$, $\sin\theta \sim \theta$, for small incremental rotations θ. This converts the rotation matrix R into a linear form, which then yields a linear least-squares system.

We instead pursue a linearization that starts with the Rodrigues rotation formula for the effect of a rotation R on a vector v:

$$\mathrm{R}v = v\cos\theta + (a \times v)\sin\theta + a\,(a \cdot v)\,(1 - \cos\theta), \quad (7)$$

where a and θ are the axis and angle of rotation. We observe that the last term in (7) is quadratic in the incremental rotation angle θ, so we drop it to linearize:

$$\begin{aligned} \mathrm{R}v &\approx v\cos\theta + (a \times v)\sin\theta \\ &= \cos\theta\,\big(v + (\tilde{a} \times v)\big), \end{aligned} \quad (8)$$

where $\tilde{a} = a\tan\theta$. Substituting into (6),

$$\begin{aligned} \mathcal{E}_{symm} &\approx \sum_i \Big[\cos\theta\,(p_i - q_i) \cdot n_i + \\ &\qquad \cos\theta\,\big(\tilde{a} \times (p_i + q_i)\big) \cdot n_i \;+\; t \cdot n_i \Big]^2 \\ &= \sum_i \cos^2\theta \Big[(p_i - q_i) \cdot n_i + \\ &\qquad \big((p_i + q_i) \times n_i\big) \cdot \tilde{a} \;+\; n_i \cdot \tilde{t} \Big]^2, \end{aligned} \quad (9)$$

where $n_i = n_{p,i} + n_{q,i}$ and $\tilde{t} = t/\cos\theta$. We now make the additional approximation of weighting the objective by $1/\cos^2\theta$, which approaches 1 for small θ. Finally, for better numerical stability, we

normalize the (p_i, q_i) by translating each point set to the origin and adjusting the solved-for translation appropriately. This yields:

$$\sum_i \left[(\tilde{p}_i - \tilde{q}_i) \cdot n_i \; + \; \big((\tilde{p}_i + \tilde{q}_i) \times n_i\big) \cdot \tilde{a} \; + \; n_i \cdot \tilde{t} \right]^2, \qquad (10)$$

where $\tilde{p}_i = p_i - \overline{p}$ and $\tilde{q}_i = q_i - \overline{q}$. This is a least-squares problem in $\tilde{a}$ and $\tilde{t}$, and the final transformation from $\mathcal{P}$ to $\mathcal{Q}$ is:

$$\mathrm{trans}(\overline{q}) \circ \mathrm{rot}\Big(\theta, \tfrac{\tilde{a}}{\|\tilde{a}\|}\Big) \circ \mathrm{trans}(\tilde{t}\cos\theta) \circ \mathrm{rot}\Big(\theta, \tfrac{\tilde{a}}{\|\tilde{a}\|}\Big) \circ \mathrm{trans}(-\overline{p}), \qquad (11)$$

where $\theta = \tan^{-1} \|\tilde{a}\|$.

Note that the new linearization results in the same system of equations as would the traditional approach. What changes is how the solved-for variables $\tilde{a}$ and $\tilde{t}$ are *interpreted.* This produces a modest increase in accuracy but, more importantly, is necessary to obtain the property that the linearization is exact for exact correspondences (see Section 4.2). We may interpret (10) as a Gauss-Newton step applied to (6), using the Gibbs representation of rotations.

4 THEORETICAL AND EXPERIMENTAL RESULTS

4.1 The Symmetric Error Is Zero When Corresponding Points Are Consistent With a Quadratic Surface

Assume that points p and q are located on a height field $z = h(x, y)$ that may be approximated locally as second-order. We construct a coordinate system centered at their geodesic midpoint m, with the surface tangent to xy (see Figure 3). The height of that surface relative to the tangent plane may be expressed as a quadratic function of xy displacement away from m:

$$\Delta z = \frac{1}{2} \begin{pmatrix} \Delta x & \Delta y \end{pmatrix} \begin{pmatrix} e & f \\ f & g \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}. \qquad (12)$$

This is an *even* function, so if p and q are displaced by equal amounts in opposite directions from m, then their z coordinates are equal, and $p - q$ is parallel to the xy plane. Conversely, the perturbation of the surface normal away from $\hat{z}$ is an *odd* function:[2]

$$\begin{pmatrix} \Delta n_x \\ \Delta n_y \end{pmatrix} = - \begin{pmatrix} e & f \\ f & g \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}. \qquad (13)$$

Therefore, $n_p + n_q$ is parallel to the z axis, and so it must be perpendicular to $p - q$.

Note that the property that the error vanishes near a second-order patch of surface does not hold for point-to-point, point-to-plane, or even the method of Mitra et al. [2004]. The latter, for example, considers a quadratic approximant to the square of the Euclidean distance function, which can be minimized only at a plane, line, or point. An objective function that vanishes near a curved surface, however, would require a higher-order approximation. This partially explains the faster convergence of $\mathcal{E}_{symm}$, as observed in the experiments of Section 4.3.

Note also that, as p and q move away from being consistent with a second-order surface, the error in Equation 4 remains well-behaved: it is just linear in positions and normals. This is in contrast to the (squared) Euclidean distance function, whose Hessian diverges at the medial surface.

[2]After perturbation, n must be re-normalized, but the the dependency on $(\Delta x, \Delta y)$ remains odd.

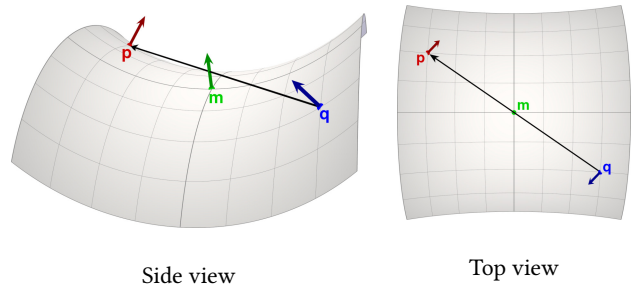

Fig. 3. A second-order patch of surface around m, the geodesic midpoint between p and q. Because the variation of height and normal relative to m are even and odd, respectively, $p - q$ and $n_p + n_q$ are parallel and perpendicular to the tangent plane at m, and so Equation 4 holds.

4.2 The Linearization is Exact for Exact Correspondences

Unlike the traditional linearization of rotations, we observe that the linear least-squares problem in (10) produces an *exact* result when correspondences (p_i, q_i) are correct. This is because the approximation of $1/(\cos\theta)^2$ as 1 involves a *multiplicative* factor that may be interpreted as a weight, and so the only *additive* term that is actually dropped contains a factor of $(\tilde{p}_i - \tilde{q}_i) \cdot a$. However, if correspondences are correct and the points are center-of-mass normalized, then $\tilde{p}_i - \tilde{q}_i$ is guaranteed to be perpendicular to the rotation axis, and hence zero error is introduced by the linearization. We have verified experimentally that this is the case, up to roundoff error.

This an unexpected result, because previous techniques that solve for exact rotations have tended to involve nonlinear optimization, and hence require multiple iterations. There do exist several closed-form solutions for the minimization of $\mathcal{E}_{point}$, but all are based on SVD or eigenvector problems [Eggert et al. 1997]. To the best of our knowledge, there are no published techniques that exactly optimize $\mathcal{E}_{plane}$ or any related metric with a single linear solve, even for exact correspondences.

4.3 $\mathcal{E}_{symm}$ Accelerates Per-Iteration Convergence

In an ICP implementation, of course, correspondences will not be exact. Nevertheless, we observe that $\mathcal{E}_{symm}$ produces faster convergence than $\mathcal{E}_{point}$ and $\mathcal{E}_{plane}$. We conduct an experiment in which we start with two copies of a mesh, then move one copy away from its ground-truth position and orientation. We execute a *single* iteration of ICP to align the shifted copy back towards the original. We measure error, both before and after that ICP iteration, as the root-mean-square distance between actual vertex positions and their ground-truth locations, where the mesh is scaled such that the root-mean-squared vertex distance from the center of mass is 1. No outlier rejection is performed.

We test a total of six objective functions, of which four are $\mathcal{E}_{point}$, $\mathcal{E}_{plane}$, $\mathcal{E}_{symm\text{-}RN}$, and $\mathcal{E}_{symm}$. The two additional objectives are:

- **Quadratic:** the method of Mitra et al. [2004] that minimizes a locally-quadratic approximant to the squared Euclidean distance function. The implementation uses the "on demand" method described in that paper, in which the approximation uses curvature information at the closest point.

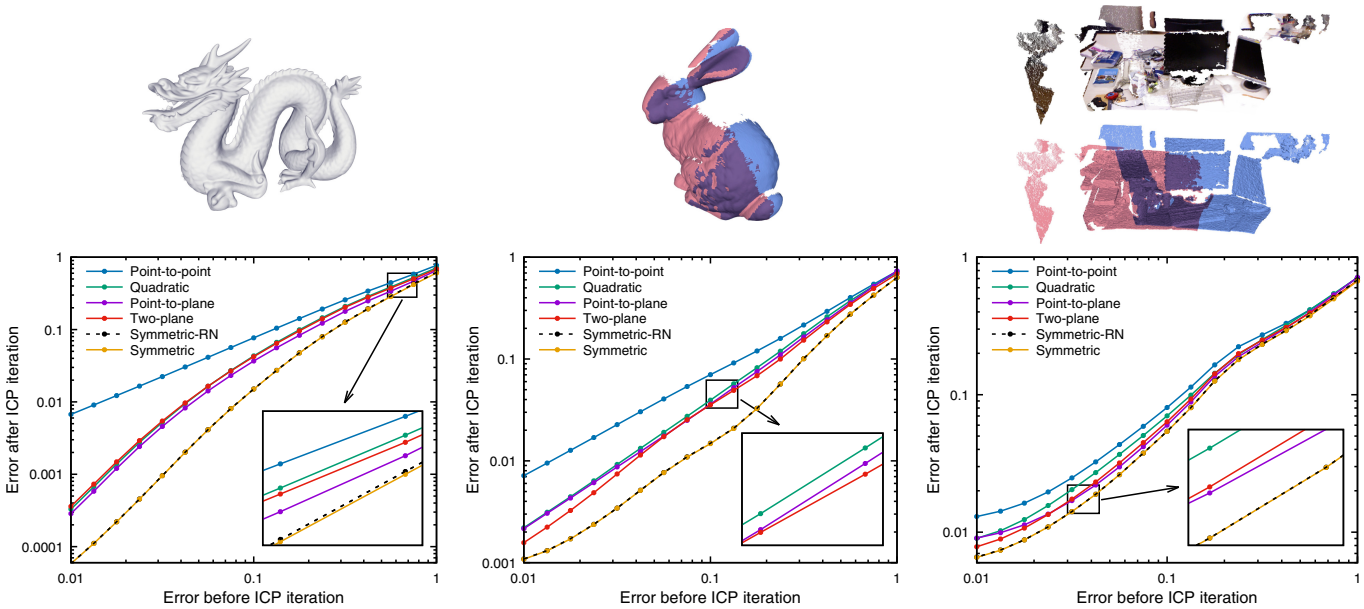

Fig. 4. **Left:** Error decrease due to one ICP iteration on the dragon model, aligned to itself. Ground-truth errors before and after the ICP iteration are shown on the x and y axes, respectively, of this log-log plot. The proposed symmetric objective results in significantly faster decrease of error at each iteration. **Center:** Error decrease due to one iteration of ICP, aligning bun090 to bun000 (illustrated in blue and red, respectively, with the areas of overlap in purple). **Right:** Error decrease in one ICP iteration, aligning two scans from the TUM RGB-D dataset. Note the slower convergence because of the high level of noise.

- **Two-plane:** minimizing the sum of squared distances to planes defined by *both* n_p and n_q:

$$\mathcal{E}_{two\text{-}plane} = \sum_i \left[(\mathrm{R}p_i - \mathrm{R}^{-1}q_i + t)\cdot(\mathrm{R}n_{p,i})\right]^2 + \left[(\mathrm{R}p_i - \mathrm{R}^{-1}q_i + t)\cdot(\mathrm{R}^{-1}n_{q,i})\right]^2. \tag{14}$$

  This approach is a symmetrized version (with split rotations) of methods previously used by Tagliasacchi et al. [2015] and Luong et al. [2016], which in turn were related to the Hausdorff metric by Tkach et al. [2016].

Figure 4, left, shows the result of this experiment on the dragon model [Curless and Levoy 1996]. The graph compares the error after an iteration of alignment (plotted along the y axis) to error before alignment (x axis), for each method. Each datapoint represents the average over 1000 trials, each having a different starting transformation (that produces the same initial error).

We see that the behaviors of the different methods fall into three general categories. The point-to-point objective is the slowest: it reliably exhibits *linear* convergence. The point-to-plane, two-plane, and quadratic methods exhibit *superlinear* convergence, while the two symmetric objectives (whose curves lie essentially on top of each other) are even faster: they have a higher convergence order (greater slope of the curve) for large misalignment, and a similar convergence order but more favorable constant once alignment improves. Note that the graph is on a log-log scale, so the vertical distance between the curves can represent improvement by as much as an order of magnitude in per-iteration convergence.

Figure 4, center, shows a similar experiment, but for incomplete and partially-overlapping range scans (the bun000 and bun090 scans from the bunny model [Turk and Levoy 1994]). In this case, some form of robust estimation is necessary to account for partial overlap, and this may be done using techniques including explicit outlier rejection [Turk and Levoy 1994; Pulli 1999; Chetverikov et al. 2005], consistency conditions [Pajdla and Van Gool 1995; Dorai et al. 1998], M-estimation [Fitzgibbon 2001], or sparsity-inducing norms [Bouaziz et al. 2013]. Because it is not our intention to exhaustively explore options for robust estimation, we adopt a relatively simple outlier-rejection strategy that excludes matches having normals with negative dot product, and pairs with (point-to-point) distance greater than 2.5 times the standard deviation of distances at each iteration [Masuda et al. 1996]. (The standard deviation is estimated robustly as 1.4826 times the median distance.) The relative ordering of the ICP variants is similar to the whole-model case, but the variants are closer to each other, due to noise and imperfect outlier rejection — these meshes only have an overlap (measured as intersection over union, or IOU) of approximately 23%.

Finally, Figure 4, right, shows the experimental convergence for a pair of scans of an indoor office environment, specifically timestamps 1305031104.030279 and 1305031108.503548 of the freiburg1_xyz sequence from the TUM RGB-D dataset [Sturm et al. 2012]. This dataset is qualitatively different from the bunny, since it contains many planar surfaces (which might be expected to boost the performance of $\mathcal{E}_{plane}$ and related variants), but also has some warp and scanning noise from the Kinect sensor (which leads to a nontrivial residual, decreasing the difference between variants). Even in this scenario, which might be expected to provide the least benefit to $\mathcal{E}_{symm}$, the symmetric objective outperforms the others.

4.4 ICP with $\mathcal{E}_{symm}$ Has a Wide Convergence Basin

To measure how reliably the different ICP objectives reach the correct alignment, we repeatedly misalign two meshes by a given angle (about a random axis) and a given translation magnitude (in a random direction), then run ICP for a fixed number of iterations. For consistency, all variants sample points from both meshes, find closest points (according to Euclidean distance) on the other mesh, and use the outlier rejection strategies described above. The success of ICP is determined by whether points are within a threshold (1% of mesh size) of their ground-truth locations after alignment.

Figure 5 shows the convergence basin for the six ICP variants from the previous experiment, plus two variants that apply the Levenberg-Marquardt algorithm, either to $\mathcal{E}_{plane}$ as in the work of Fitzgibbon [2001], or to the new $\mathcal{E}_{symm}$. Each small square shows the percentage of successful trials, averaged over all pairs of bunny scans having IOU overlap greater than 20%, and over 1000 initial transformations that have the given rotation angle and translation magnitude.

With 500 iterations, $\mathcal{E}_{point}$ and $\mathcal{E}_{plane}$ have regimes in which one is slightly better ($\mathcal{E}_{point}$ for large translation and small rotation, and vice versa for $\mathcal{E}_{plane}$). The quadratic method of Mitra et al. [2004] and Fitzgibbon's LM-ICP (applied to $\mathcal{E}_{plane}$) modestly improve convergence, with $\mathcal{E}_{symm}$ comparable to those two and $\mathcal{E}_{two\text{-}plane}$ improving reliability even more. Levenberg-Marquardt applied to $\mathcal{E}_{symm}$ is both fast and reliable, achieving the widest convergence basin at 20 iterations and coming within a few percent of $\mathcal{E}_{two\text{-}plane}$ at 500 iterations. The availability of a fast and reliable ICP variant is vital for real-time applications such as interactive 3D scanning [Izadi et al. 2011], which have heretofore used point-to-plane ICP.

5 DISCUSSION AND FUTURE WORK

The symmetric objective represents a simple improvement to traditional point-to-plane ICP. At minimal additional implementation cost, it produces faster and more reliable convergence. Future work in this area consists of exploring the combination of the symmetric objective with modern approaches for denoising, surface descriptor matching, robust ℓ_p minimization, and step size control.

A further topic for future investigation is relating the symmetric objective to distance function minimization. Just as Equation 3 can be considered a linearization of the signed distance to $\mathcal{Q}$ evaluated at points on $\mathcal{P}$, Equation 4 can be considered the linearization of the sum of that function, plus the signed distance to $\mathcal{P}$ evaluated at points on $\mathcal{Q}$. While it might be possible to simplify this description (perhaps by considering samples at the midpoint between the two surfaces), even that does not readily lead to an explanation of the properties in Sections 3 and 4.1. Future analysis of the *sum* of distance transforms could lead to additional insights on $\mathcal{E}_{symm}$.

ACKNOWLEDGMENTS

Thanks to the PIXL group, especially Maciej Halber, to Misha Kazhdan, and to the maps and 3D scanning groups with whom the author interacted under the Google Visiting Faculty program. Thanks to the U. S. National Science foundation for support under grants IIS-1421435, CHS-1617236, and IIS-1815070.

REFERENCES

Dror Aiger, Niloy J. Mitra, and Daniel Cohen-Or. 2008. 4-Points Congruent Sets for Robust Pairwise Surface Registration. *ACM Trans. Graph.* 27, 3, Article 85 (Aug. 2008).

Paul J. Besl and Neil D. McKay. 1992. A Method for Registration of 3-D Shapes. *IEEE Trans. PAMI* 14, 2 (Feb. 1992), 239–256.

Sofien Bouaziz, Andrea Tagliasacchi, and Mark Pauly. 2013. Sparse Iterative Closest Point. In *Proc. SGP.*

Benedict Brown and Szymon Rusinkiewicz. 2007. Global Non-Rigid Alignment of 3-D Scans. *ACM Trans. Graph.* 26, 3, Article 21 (July 2007).

Yang Chen and Gérard Medioni. 1992. Object Modelling by Registration of Multiple Range Images. *Image and Vision Computing* 10, 3 (April 1992), 145–155.

Dmitry Chetverikov, Dmitry Stepanov, and Pavel Krsek. 2005. Robust Euclidean Alignment of 3D Point Sets: The Trimmed Iterative Closest Point Algorithm. *Image and Vision Computing* 23, 3 (March 2005), 299–309.

Brian Curless and Marc Levoy. 1996. A Volumetric Method for Building Complex Models from Range Images. In *Proc. SIGGRAPH.*

Yago Díez, Ferran Roure, Xavier Lladó, and Joaquim Salvi. 2015. A Qualitative Review on 3D Coarse Registration Methods. *ACM Comput. Surv.* 47, 3, Article 45 (Feb. 2015).

Chitra Dorai, Gang Wang, Anil K. Jain, and Carolyn Mercer. 1998. Registration and Integration of Multiple Object Views for 3D Model Construction. *IEEE Trans. PAMI* 20, 1 (Jan. 1998), 83–89.

David W. Eggert, Adele Lorusso, and Robert B. Fisher. 1997. Estimating 3-D Rigid Body Transformations: A Comparison of Four Major Algorithms. *Machine Vision and Applications* 9, 5–6 (March 1997), 272–290.

Andrew W. Fitzgibbon. 2001. Robust Registration of 2D and 3D Point Sets. In *Proc. BMVC.*

Natasha Gelfand, Leslie Ikemoto, Szymon Rusinkiewicz, and Marc Levoy. 2003. Geometrically Stable Sampling for the ICP Algorithm. In *Proc. 3DIM.*

Natasha Gelfand, Niloy J. Mitra, Leonidas Guibas, and Helmut Pottmann. 2005. Robust Global Registration. In *Proc. SGP.*

Maciej Halber and Thomas Funkhouser. 2017. Fine-to-Coarse Global Registration of RGB-D Scans. In *Proc. CVPR.*

Daniel F. Huber and Martial Hebert. 2003. Fully Automatic Registration of Multiple 3D Data Sets. *Image and Vision Computing* 21, 7 (July 2003), 637–650.

Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. 2011. KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. In *Proc. UIST.*

Peter Kovesi. 2015. Good Colour Maps: How to Design Them. arXiv:1509.03700.

Stefan Leopoldseder, Helmut Pottmann, and Hongkai Zhao. 2003. *The d^2-Tree: A Hierarchical Representation of the Squared Distance Function.* Technical Report 101. Institute of Geometry, Vienna University of Technology.

Hiêp Quang Luong, Michiel Vlaminck, Werner Goeman, and Wilfried Philips. 2016. Consistent ICP for the Registration of Sparse and Inhomogeneous Point Clouds. In *Proc. Int. Conf. on Communications and Electronics (ICCE).*

Takeshi Masuda, Katsuhiko Sakaue, and Naokazu Yokoya. 1996. Registration and Integration of Multiple Range Images for 3-D Model Construction. In *Proc. ICPR.*

Niloy J. Mitra, Natasha Gelfand, Helmut Pottmann, and Leonidas Guibas. 2004. Registration of Point Cloud Data from a Geometric Optimization Perspective. In *Proc. SGP.*

Tomáš Pajdla and Luc Van Gool. 1995. Matching of 3-D Curves Using Semi-Differential Invariants. In *Proc. ICCV.*

Fran cois Pomerleau, Francis Colas, and Roland Siegwart. 2015. A Review of Point Cloud Registration Algorithms for Mobile Robotics. *Foundations and Trends in Robotics* 4, 1 (May 2015), 1–104.

Helmut Pottmann, Qi-Xing Huang, Yong-Liang Yang, and Shi-Min Hu. 2006. Geometry and Convergence Analysis of Algorithms for Registration of 3D Shapes. *IJCV* 67, 3 (May 2006), 277–296.

Kari Pulli. 1999. Multiview Registration for Large Data Sets. In *Proc. 3DIM.*

Szymon Rusinkiewicz and Marc Levoy. 2001. Efficient Variants of the ICP Algorithm. In *Proc. 3DIM.*

Aleksandr V. Segal, Dirk Haehnel, and Sebastian Thrun. 2009. Generalized-ICP. In *Proc. RSS.*

Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. 2012. A Benchmark for the Evaluation of RGB-D SLAM Systems. In *Proc. IROS.*

Andrea Tagliasacchi, Matthias Schroeder, Anastasia Tkach, Sofien Bouaziz, Mario Botsch, and Mark Pauly. 2015. Robust Articulated-ICP for Real-Time Hand Tracking. In *Proc. SGP.*

Anastasia Tkach, Mark Pauly, and Andrea Tagliasacchi. 2016. Sphere-Meshes for Real-Time Hand Modeling and Tracking. *ACM Trans. Graph.* 35, 6, Article 222 (Nov. 2016).

Greg Turk and Marc Levoy. 1994. Zippered Polygon Meshes from Range Images. In *Proc. SIGGRAPH.*

Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. 2016. Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration. *IEEE Trans. PAMI* 38, 11 (Nov. 2016), 2241–2254.

Translation Magnitude

Rotation Angle

	0%	10%	20%	30%	40%	50%
0°	100	43	20	13	10	8
20°	29	22	14	10	8	7
40°	8	7	7	6	5	4
60°	3	2	2	2	2	2

	0%	10%	20%	30%	40%	50%
0°	100	99	90	80	74	71
20°	98	92	83	76	71	67
40°	71	70	66	63	61	59
60°	49	49	47	46	45	43

	0%	10%	20%	30%	40%	50%
0°	100	100	99	95	92	90
20°	100	99	96	93	90	88
40°	93	91	89	87	85	83
60°	78	77	76	75	73	71

Point-to-point

	0%	10%	20%	30%	40%	50%
0°	100	100	98	88	79	72
20°	100	98	94	84	75	69
40°	85	83	77	70	64	58
60°	54	53	50	47	43	40

	0%	10%	20%	30%	40%	50%
0°	100	100	99	96	93	90
20°	100	99	97	94	91	89
40°	95	94	91	89	87	84
60°	81	80	78	76	74	71

	0%	10%	20%	30%	40%	50%
0°	100	100	99	96	93	90
20°	100	99	97	94	91	89
40°	95	95	92	89	87	85
60°	83	82	80	78	75	73

Quadratic

	0%	10%	20%	30%	40%	50%
0°	100	100	98	89	78	69
20°	100	99	94	84	74	66
40°	89	87	81	72	64	58
60°	64	62	58	53	48	44

	0%	10%	20%	30%	40%	50%
0°	100	100	98	93	88	83
20°	100	99	96	91	87	82
40°	95	94	91	86	82	78
60°	82	81	79	75	71	68

	0%	10%	20%	30%	40%	50%
0°	100	100	98	93	88	83
20°	100	99	96	91	87	83
40°	95	94	91	87	82	79
60°	83	82	80	76	72	69

Point-to-plane

	0%	10%	20%	30%	40%	50%
0°	100	100	98	91	83	76
20°	100	99	95	87	80	73
40°	91	89	84	77	71	65
60°	66	65	62	58	54	51

	0%	10%	20%	30%	40%	50%
0°	100	100	99	96	94	92
20°	100	100	98	95	93	91
40°	96	95	94	91	89	87
60°	86	86	85	82	80	78

	0%	10%	20%	30%	40%	50%
0°	100	100	99	97	95	93
20°	100	100	98	96	93	91
40°	96	96	94	92	90	88
60°	88	88	86	84	82	80

Two-plane

	0%	10%	20%	30%	40%	50%
0°	100	100	99	94	89	83
20°	100	99	97	92	86	81
40°	94	93	89	84	79	73
60°	79	77	74	69	63	59

	0%	10%	20%	30%	40%	50%
0°	100	100	99	95	91	87
20°	100	99	97	93	89	85
40°	95	94	91	87	83	79
60°	83	82	80	76	72	68

	0%	10%	20%	30%	40%	50%
0°	100	100	99	95	91	87
20°	100	99	97	93	89	85
40°	95	94	91	87	83	79
60°	84	83	80	76	72	69

Symmetric-RN

	0%	10%	20%	30%	40%	50%
0°	100	100	99	94	88	83
20°	100	99	97	92	86	81
40°	95	94	91	86	81	75
60°	83	82	79	74	69	65

	0%	10%	20%	30%	40%	50%
0°	100	100	99	95	90	87
20°	100	99	97	93	89	85
40°	95	95	93	89	85	81
60°	87	86	84	81	77	73

	0%	10%	20%	30%	40%	50%
0°	100	100	99	95	90	87
20°	100	99	97	93	89	85
40°	95	95	93	89	85	82
60°	87	87	84	81	77	74

Symmetric

	0%	10%	20%	30%	40%	50%
0°	100	100	99	94	89	83
20°	100	99	96	90	84	78
40°	86	84	81	75	70	66
60°	58	58	55	52	50	47

	0%	10%	20%	30%	40%	50%
0°	100	100	100	96	93	90
20°	100	99	97	94	91	89
40°	94	93	91	89	86	84
60°	81	80	78	76	74	73

	0%	10%	20%	30%	40%	50%
0°	100	100	100	96	93	90
20°	100	99	97	94	91	89
40°	94	94	92	89	86	84
60°	82	81	80	77	76	74

LM-Point-to-plane

	0%	10%	20%	30%	40%	50%
0°	100	100	100	97	93	91
20°	100	99	98	95	92	89
40°	93	93	91	89	86	83
60°	79	79	78	75	73	70

	0%	10%	20%	30%	40%	50%
0°	100	100	100	97	94	92
20°	100	99	98	96	93	91
40°	95	94	93	91	89	87
60°	85	85	84	82	80	78

	0%	10%	20%	30%	40%	50%
0°	100	100	100	97	94	92
20°	100	99	98	96	93	91
40°	95	94	93	91	89	87
60°	86	86	85	83	81	79

LM-Symmetric

20 iterations 100 iterations 500 iterations

Fig. 5. Convergence basins of different ICP variants, illustrated as percentage of the time that ICP succeeded within 20, 100, and 500 iterations (left, center, and right columns, respectively). Each small square represents an average over all overlapping pairs of scans in the bunny dataset, and over 1000 random initial transformations that have the given rotation angle and translation magnitude (relative to mesh size). Colormap: CET-L19 by Kovesi [2015].