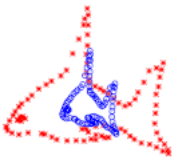


Point-set registration

In computer vision, pattern recognition, and robotics, **point-set registration**, also known as **point-cloud registration** or **scan matching**, is the process of finding a spatial transformation (*e.g.*, scaling, rotation and translation) that aligns two point clouds. The purpose of finding such a transformation includes merging multiple data sets into a globally consistent model (or coordinate frame), and mapping a new measurement to a known data set to identify features or to estimate its pose. Raw 3D point cloud data are typically obtained from Lidars and RGB-D cameras. 3D point clouds can also be generated from computer vision algorithms such as triangulation, bundle adjustment, and more recently, monocular image depth estimation using deep learning. For 2D point set registration used in image processing and feature-based image registration, a point set may be 2D pixel coordinates obtained by feature extraction from an image, for example corner detection. Point cloud registration has extensive applications in autonomous driving,^[1] motion estimation and 3D reconstruction,^[2] object detection and pose estimation,^{[3][4]} robotic manipulation,^[5] simultaneous localization and mapping (SLAM),^{[6][7]} panorama stitching,^[8] virtual and augmented reality,^[9] and medical imaging.^[10]



Point set registration is the process of aligning two point sets. Here, the blue fish is being registered to the red fish.

As a special case, registration of two point sets that only differ by a 3D rotation (*i.e.*, there is no scaling and translation), is called the Wahba Problem and also related to the orthogonal procrustes problem.

Contents

- Formulation**
- Types of algorithms**
 - Rigid registration
 - Non-rigid registration
 - Other types
- Correspondence-based registration**
 - Outlier-free registration
 - Robust registration
 - Maximum consensus
 - Outlier removal
 - M-estimation
 - Graduated non-convexity
 - Certiifiably robust registration
- Simultaneous pose and correspondence registration**
 - Iterative closest point
 - Robust point matching
 - Thin plate spline robust point matching
 - Kernel correlation
 - Gaussian mixture model
 - Coherent point drift
 - Bayesian coherent point drift (BCPD)
 - Sorting the Correspondence Space (SCS)
- See also**
- References**
- External links**

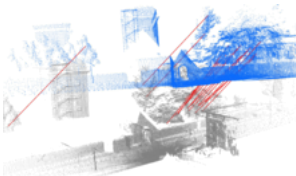
Formulation

The problem may be summarized as follows:^[11] Let $\{\mathcal{M}, \mathcal{S}\}$ be two finite size point sets in a finite-dimensional real vector space \mathbb{R}^d , which contain M and N points respectively (*e.g.*, $d = 3$ recovers the typical case of when \mathcal{M} and \mathcal{S} are 3D point sets). The problem is to find a transformation to be applied to the moving "model" point set \mathcal{M} such that the difference (typically defined in the sense of point-wise Euclidean distance) between \mathcal{M} and the static "scene" set \mathcal{S} is minimized. In other words, a mapping from \mathbb{R}^d to \mathbb{R}^d is desired which yields the best alignment between the transformed "model" set and the "scene" set. The mapping may consist of a rigid or non-rigid transformation. The transformation model may be written as T , using which the transformed, registered model point set is:

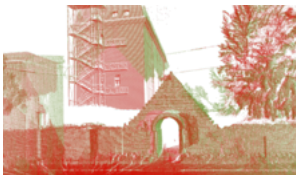
$$T(\mathcal{M})$$

(1)

The output of a point set registration algorithm is therefore the *optimal transformation* T^* such that \mathcal{M} is best aligned to \mathcal{S} , according to some defined notion of distance function $\text{dist}(\cdot, \cdot)$:



Data from two 3D scans of the same environment need to be aligned using point set registration.



Data from above, registered successfully using a variant of iterative closest point.

$$T^* = \arg \min_{T \in \mathcal{T}} \text{dist}(T(\mathcal{M}), \mathcal{S}) \quad (2)$$

where \mathcal{T} is used to denote the set of all possible transformations that the optimization tries to search for. The most popular choice of the distance function is to take the square of the Euclidean distance for every pair of points:

$$\text{dist}(T(\mathcal{M}), \mathcal{S}) = \sum_{m \in T(\mathcal{M})} \|m - s_m\|_2^2, \quad s_m = \arg \min_{s \in \mathcal{S}} \|s - m\|_2^2 \quad (3)$$

where $\|\cdot\|_2$ denotes the vector 2-norm, s_m is the *corresponding point* in set \mathcal{S} that attains the *shortest distance* to a given point m in set \mathcal{M} after transformation. Minimizing such a function in rigid registration is equivalent to solving a least squares problem.

Types of algorithms

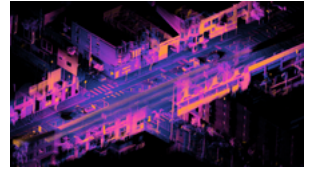
When the correspondences (i.e., $s_m \leftrightarrow m$) are given before the optimization, for example, using feature matching techniques, then the optimization only needs to estimate the transformation. This type of registration is called **correspondence-based registration**. On the other hand, if the correspondences are unknown, then the optimization is required to jointly find out the correspondences and transformation together. This type of registration is called **simultaneous pose and correspondence registration**.

Rigid registration

Given two point sets, rigid registration yields a rigid transformation which maps one point set to the other. A rigid transformation is defined as a transformation that does not change the distance between any two points. Typically such a transformation consists of translation and rotation.^[12] In rare cases, the point set may also be mirrored. In robotics and computer vision, rigid registration has the most applications.

Non-rigid registration

Given two point sets, non-rigid registration yields a non-rigid transformation which maps one point set to the other. Non-rigid transformations include affine transformations such as scaling and shear mapping. However, in the context of point set registration, non-rigid registration typically involves nonlinear transformation. If the eigenmodes of variation of the point set are known, the nonlinear transformation may be parametrized by the eigenvalues.^[13] A nonlinear transformation may also be parametrized as a thin plate spline.^{[14][13]}



Registered point cloud from a lidar mounted on a moving car.

Other types

Some approaches to point set registration use algorithms that solve the more general graph matching problem.^[11] However, the computational complexity of such methods tend to be high and they are limited to rigid registrations. In this article, we will only consider algorithms for rigid registration, where the transformation is assumed to contain 3D rotations and translations (possibly also including a uniform scaling).

The PCL (Point Cloud Library) is an open-source framework for n-dimensional point cloud and 3D geometry processing. It includes several point registration algorithms.^[15]

Correspondence-based registration

Correspondence-based methods assume the putative correspondences $m \leftrightarrow s_m$ are given for every point $m \in \mathcal{M}$. Therefore, we arrive at a setting where both point sets \mathcal{M} and \mathcal{S} have N points and the correspondences $m_i \leftrightarrow s_i, i = 1, \dots, N$ are given.

Outlier-free registration

In the simplest case, one can assume that all the correspondences are correct, meaning that the points $m_i, s_i \in \mathbb{R}^3$ are generated as follows:

$$s_i = lRm_i + t + \epsilon_i, i = 1, \dots, N \quad (\text{cb.1})$$

where $l > 0$ is a uniform scaling factor (in many cases $l = 1$ is assumed), $R \in \text{SO}(3)$ is a proper 3D rotation matrix ($\text{SO}(d)$ is the special orthogonal group of degree d), $t \in \mathbb{R}^3$ is a 3D translation vector and $\epsilon_i \in \mathbb{R}^3$ models the unknown additive noise (e.g., Gaussian noise). Specifically, if the noise ϵ_i is assumed to follow a zero-mean isotropic Gaussian distribution with standard deviation σ_i , i.e., $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2 I_3)$, then the following optimization can be shown to yield the maximum likelihood estimate for the unknown scale, rotation and translation:

$$l^*, R^*, t^* = \arg \min_{l>0, R \in \text{SO}(3), t \in \mathbb{R}^3} \sum_{i=1}^N \frac{1}{\sigma_i^2} \|s_i - lRm_i - t\|_2^2 \quad (\text{cb.2})$$

Note that when the scaling factor is 1 and the translation vector is zero, then the optimization recovers the formulation of the Wahba problem. Despite the non-convexity of the optimization (cb.2) due to non-convexity of the set $\text{SO}(3)$, seminal work by Berthold K.P. Horn showed that (cb.2) actually admits a closed-form solution, by decoupling the estimation of scale, rotation and translation.^[16] Similar results were discovered by Arun et al.^[17] In addition, in order to find a unique transformation (l, R, t) , at least $N = 3$ non-collinear points in each point set are required.

More recently, Briaes and Gonzalez-Jimenez have developed a semidefinite relaxation using Lagrangian duality, for the case where the model set \mathcal{M} contains different 3D primitives such as points, lines and planes (which is the case when the model \mathcal{M} is a 3D mesh).^[18] Interestingly, the semidefinite relaxation is empirically tight, i.e., a certifiably globally optimal solution can be extracted from the solution of the semidefinite relaxation.

Robust registration

The least squares formulation (cb.2) is known to perform arbitrarily bad in the presence of outliers. An outlier correspondence is a pair of measurements $s_i \leftrightarrow m_i$ that departs from the generative model (cb.1). In this case, one can consider a different generative model as follows:^[19]

$$\mathbf{s}_i = \begin{cases} \mathbf{lRm}_i + \mathbf{t} + \epsilon_i & \text{if } i - \text{th pair is an inlier} \\ \mathbf{o}_i & \text{if } i - \text{th pair is an outlier} \end{cases} \quad (\text{cb.3})$$

where if the i -th pair $\mathbf{s}_i \leftrightarrow \mathbf{m}_i$ is an inlier, then it obeys the outlier-free model (cb.1), i.e., \mathbf{s}_i is obtained from \mathbf{m}_i by a spatial transformation plus some small noise; however, if the i -th pair $\mathbf{s}_i \leftrightarrow \mathbf{m}_i$ is an outlier, then \mathbf{s}_i can be any arbitrary vector \mathbf{o}_i . Since one does not know which correspondences are outliers beforehand, robust registration under the generative model (cb.3) is of paramount importance for computer vision and robotics deployed in the real world, because current feature matching techniques tend to output highly corrupted correspondences where over **95%** of the correspondences can be outliers.^[20]

Next, we describe several common paradigms for robust registration.

Maximum consensus

Maximum consensus seeks to find the largest set of correspondences that are consistent with the generative model (cb.1) for some choice of spatial transformation $(\mathbf{l}, \mathbf{R}, \mathbf{t})$. Formally speaking, maximum consensus solves the following optimization:

$$\max_{\mathbf{l} > 0, \mathbf{R} \in \text{SO}(3), \mathbf{t} \in \mathbb{R}^3, \mathcal{I}} |\mathcal{I}|, \quad \text{subject to } \frac{1}{\sigma_i^2} \|\mathbf{s}_i - \mathbf{lRm}_i - \mathbf{t}\|_2^2 \leq \xi, \forall i \in \mathcal{I} \quad (\text{cb.4})$$

where $|\mathcal{I}|$ denotes the cardinality of the set \mathcal{I} . The constraint in (cb.4) enforces that every pair of measurements in the inlier set \mathcal{I} must have residuals smaller than a pre-defined threshold ξ . Unfortunately, recent analyses have shown that globally solving problem (cb.4) is NP-Hard, and global algorithms typically have to resort to branch-and-bound (BnB) techniques that take exponential-time complexity in the worst case.^{[21][22][23][24][25]}

Although solving consensus maximization exactly is hard, there exist efficient heuristics that perform quite well in practice. One of the most popular heuristics is the Random Sample Consensus (RANSAC) scheme.^[26] RANSAC is an iterative hypothesize-and-verify method. At each iteration, the method first randomly samples 3 out of the total number of N correspondences and computes a hypothesis $(\mathbf{l}, \mathbf{R}, \mathbf{t})$ using Horn's method,^[16] then the method evaluates the constraints in (cb.4) to count how many correspondences actually agree with such a hypothesis (i.e., it computes the residual $\|\mathbf{s}_i - \mathbf{lRm}_i - \mathbf{t}\|_2^2 / \sigma_i^2$ and compares it with the threshold ξ for each pair of measurements). The algorithm terminates either after it has found a consensus set that has enough correspondences, or after it has reached the total number of allowed iterations. RANSAC is highly efficient because the main computation of each iteration is carrying out the closed-form solution in Horn's method. However, RANSAC is non-deterministic and only works well in the low-outlier-ratio regime (e.g., below **50%**), because its runtime grows exponentially with respect to the outlier ratio.^[20]

To fill the gap between the fast but inexact RANSAC scheme and the exact but exhaustive BnB optimization, recent researches have developed deterministic approximate methods to solve consensus maximization.^{[21][22][27][23]}

Outlier removal

Outlier removal methods seek to pre-process the set of highly corrupted correspondences before estimating the spatial transformation. The motivation of outlier removal is to significantly reduce the number of outlier correspondences, while maintaining inlier correspondences, so that optimization over the transformation becomes easier and more efficient (e.g., RANSAC works poorly when the outlier ratio is above **95%** but performs quite well when outlier ratio is below **50%**).

Parra *et al.* have proposed a method called Guaranteed Outlier Removal (GORE) that uses geometric constraints to prune outlier correspondences while guaranteeing to preserve inlier correspondences.^[20] GORE has been shown to be able to drastically reduce the outlier ratio, which can significantly boost the performance of consensus maximization using RANSAC or BnB. Yang and Carlone have proposed to build pairwise translation-and-rotation-invariant measurements (TRIMs) from the original set of measurements and embed TRIMs as the edges of a graph whose nodes are the 3D points. Since inliers are pairwise consistent in terms of the scale, they must form a clique within the graph. Therefore, using efficient algorithms for computing the maximum clique of a graph can find the inliers and effectively prune the outliers.^[4] The maximum clique based outlier removal method is also shown to be quite useful in real-world point set registration problems.^[19] Similar outlier removal ideas were also proposed by Parra *et al.*^[28]

M-estimation

M-estimation replaces the least squares objective function in (cb.2) with a robust cost function that is less sensitive to outliers. Formally, M-estimation seeks to solve the following problem:

$$\mathbf{l}^*, \mathbf{R}^*, \mathbf{t}^* = \arg \min_{\mathbf{l} > 0, \mathbf{R} \in \text{SO}(3), \mathbf{t} \in \mathbb{R}^3} \sum_{i=1}^N \rho \left(\frac{1}{\sigma_i} \|\mathbf{s}_i - \mathbf{lRm}_i - \mathbf{t}\|_2 \right) \quad (\text{cb.5})$$

where $\rho(\cdot)$ represents the choice of the robust cost function. Note that choosing $\rho(\mathbf{x}) = \mathbf{x}^2$ recovers the least squares estimation in (cb.2). Popular robust cost functions include ℓ_1 -norm loss, Huber loss,^[29] Geman-McClure loss^[30] and truncated least squares loss.^{[19][8][4]} M-estimation has been one of the most popular paradigms for robust estimation in robotics and computer vision.^{[31][32]} Because robust objective functions are typically non-convex (e.g., the truncated least squares loss v.s. the least squares loss), algorithms for solving the non-convex M-estimation are typically based on local optimization, where first an initial guess is provided, following by iterative refinements of the transformation to keep decreasing the objective function. Local optimization tends to work well when the initial guess is close to the global minimum, but it is also prone to get stuck in local minima if provided with poor initialization.

Graduated non-convexity

Graduated non-convexity (GNC) is a general-purpose framework for solving non-convex optimization problems without initialization. It has achieved success in early vision and machine learning applications.^{[33][34]} The key idea behind GNC is to solve the hard non-convex problem by starting from an easy convex problem. Specifically, for a given robust cost function $\rho(\cdot)$, one can construct a surrogate function $\rho_\mu(\cdot)$ with a hyper-parameter μ , tuning which can gradually increase the non-convexity of the surrogate function $\rho_\mu(\cdot)$ until it converges to the target function $\rho(\cdot)$.^{[34][35]} Therefore, at each level of the hyper-parameter μ , the following optimization is solved:

$$\mathbf{l}_\mu^*, \mathbf{R}_\mu^*, \mathbf{t}_\mu^* = \arg \min_{\mathbf{l} > 0, \mathbf{R} \in \text{SO}(3), \mathbf{t} \in \mathbb{R}^3} \sum_{i=1}^N \rho_\mu \left(\frac{1}{\sigma_i} \|\mathbf{s}_i - \mathbf{lRm}_i - \mathbf{t}\|_2 \right) \quad (\text{cb.6})$$

Black and Rangarajan proved that the objective function of each optimization **(cb.6)** can be dualized into a sum of weighted least squares and a so-called outlier process function on the weights that determine the confidence of the optimization in each pair of measurements.^[33] Using Black-Rangarajan duality and GNC tailored for the Geman-McClure function, Zhou *et al.* developed the fast global registration algorithm that is robust against about **80%** outliers in the correspondences.^[30] More recently, Yang *et al.* showed that the joint use of GNC (tailored to the Geman-McClure function and the truncated least squares function) and Black-Rangarajan duality can lead to a general-purpose solver for robust registration problems, including point clouds and mesh registration.^[35]

Certifiably robust registration

Almost none of the robust registration algorithms mentioned above (except the BnB algorithm that runs in exponential-time in the worst case) comes with **performance guarantees**, which means that these algorithms can return completely incorrect estimates without notice. Therefore, these algorithms are undesirable for safety-critical applications like autonomous driving.

Very recently, Yang *et al.* has developed the first certifiably robust registration algorithm, named *Truncated least squares Estimation And SEmidefinite Relaxation* (TEASER).^[19] For point cloud registration, TEASER not only outputs an estimate of the transformation, but also quantifies the optimality of the given estimate. TEASER adopts the following truncated least squares (TLS) estimator:

$$\mathbf{t}^*, \mathbf{R}^*, \mathbf{t}^* = \arg \min_{\mathbf{t} > 0, \mathbf{R} \in \text{SO}(3), \mathbf{t} \in \mathbb{R}^3} \sum_{i=1}^N \min \left(\frac{1}{\sigma_i^2} \|\mathbf{s}_i - \mathbf{R}\mathbf{m}_i - \mathbf{t}\|_2^2, \bar{c}^2 \right) \quad (\text{cb.7})$$

which is obtained by choosing the TLS robust cost function $\rho(\mathbf{x}) = \min(\mathbf{x}^2, \bar{c}^2)$, where \bar{c}^2 is a pre-defined constant that determines the maximum allowed residuals to be considered inliers. The TLS objective function has the property that for inlier correspondences ($\|\mathbf{s}_i - \mathbf{R}\mathbf{m}_i - \mathbf{t}\|_2^2 / \sigma_i^2 < \bar{c}^2$), the usual least square penalty is applied; while for outlier correspondences ($\|\mathbf{s}_i - \mathbf{R}\mathbf{m}_i - \mathbf{t}\|_2^2 / \sigma_i^2 > \bar{c}^2$), no penalty is applied and the outliers are discarded. If the TLS optimization **(cb.7)** is solved to global optimality, then it is equivalent to running Horn's method on only the inlier correspondences.

However, solving **(cb.7)** is quite challenging due to its combinatorial nature. TEASER solves **(cb.7)** as follows : (i) It builds invariant measurements such that the estimation of scale, rotation and translation can be decoupled and solved separately, a strategy that is inspired by the original Horn's method; (ii) The same TLS estimation is applied for each of the three sub-problems, where the scale TLS problem can be solved exactly using an algorithm called adaptive voting, the rotation TLS problem can be relaxed to a semidefinite program (SDP) where the relaxation is exact in practice,^[8] even with large amount of outliers; the translation TLS problem can be solved using component-wise adaptive voting. A fast implementation leveraging GNC is open-sourced here (<https://github.com/MIT-SPARK/TEASER-plusplus>). In practice, TEASER can tolerate more than **99%** outlier correspondences and runs in milliseconds.

In addition to developing TEASER, Yang *et al.* also prove that, under some mild conditions on the point cloud data, TEASER's estimated transformation has bounded errors from the ground-truth transformation.^[19]

Simultaneous pose and correspondence registration

Iterative closest point

The iterative closest point (ICP) algorithm was introduced by Besl and McKay.^[36] The algorithm performs rigid registration in an iterative fashion by alternating in (i) given the transformation, finding the closest point in \mathcal{S} for every point in \mathcal{M} ; and (ii) given the correspondences, finding the best rigid transformation by solving the least squares problem **(cb.2)**. As such, it works best if the initial pose of \mathcal{M} is sufficiently close to \mathcal{S} . In pseudocode, the basic algorithm is implemented as follows:

```

algorithm ICP( $\mathcal{M}$ ,  $\mathcal{S}$ )
   $\theta := \theta_0$ 
  while not registered:
     $X := \emptyset$ 
    for  $m_i \in \mathcal{M}$ :
       $\hat{s}_j :=$  closest point in  $\mathcal{S}$  to  $m_i$ 
       $X := X \cup \{m_i, \hat{s}_j\}$ 
     $\theta := \text{least\_squares}(X)$ 
  return  $\theta$ 

```

Here, the function `least_squares` performs least squares optimization to minimize the distance in each of the $\langle m_i, \hat{s}_j \rangle$ pairs, using the closed-form solutions by Horn^[16] and Arun.^[17]

Because the cost function of registration depends on finding the closest point in \mathcal{S} to every point in \mathcal{M} , it can change as the algorithm is running. As such, it is difficult to prove that ICP will in fact converge exactly to the local optimum.^[37] In fact, empirically, ICP and EM-ICP do not converge to the local minimum of the cost function.^[37] Nonetheless, because ICP is intuitive to understand and straightforward to implement, it remains the most commonly used point set registration algorithm.^[37] Many variants of ICP have been proposed, affecting all phases of the algorithm from the selection and matching of points to the minimization strategy.^{[13][38]} For example, the expectation maximization algorithm is applied to the ICP algorithm to form the EM-ICP method, and the Levenberg-Marquardt algorithm is applied to the ICP algorithm to form the LM-ICP method.^[12]

Robust point matching

Robust point matching (RPM) was introduced by Gold et al.^[39] The method performs registration using deterministic annealing and soft assignment of correspondences between point sets. Whereas in ICP the correspondence generated by the nearest-neighbour heuristic is binary, RPM uses a *soft* correspondence where the correspondence between any two points can be anywhere from 0 to 1, although it ultimately converges to either 0 or 1. The correspondences found in RPM is always one-to-one, which is not always the case in ICP.^[14] Let m_i be the i th point in \mathcal{M} and s_j be the j th point in \mathcal{S} . The *match matrix* μ is defined as such:

$$\mu_{ij} = \begin{cases} 1 & \text{if point } m_i \text{ corresponds to point } s_j \\ 0 & \text{otherwise} \end{cases} \quad (\text{rpm.1})$$

The problem is then defined as: Given two point sets \mathcal{M} and \mathcal{S} find the Affine transformation \mathbf{T} and the match matrix μ that best relates them.^[39] Knowing the optimal transformation makes it easy to determine the match matrix, and vice versa. However, the RPM algorithm determines both simultaneously. The transformation may be decomposed into a translation vector and a transformation matrix:

$$\mathbf{T}(\mathbf{m}) = \mathbf{A}\mathbf{m} + \mathbf{t}$$

The matrix \mathbf{A} in 2D is composed of four separate parameters $\{\mathbf{a}, \theta, \mathbf{b}, \mathbf{c}\}$, which are scale, rotation, and the vertical and horizontal shear components respectively. The cost function is then:

$$\text{cost} = \sum_{j=1}^N \sum_{i=1}^M \mu_{ij} \|\mathbf{s}_j - \mathbf{t} - \mathbf{A}\mathbf{m}_i\|^2 + g(\mathbf{A}) - \alpha \sum_{j=1}^N \sum_{i=1}^M \mu_{ij} \quad (\text{rpm.2})$$

subject to $\forall j \sum_{i=1}^M \mu_{ij} \leq 1, \forall i \sum_{j=1}^N \mu_{ij} \leq 1, \forall ij \mu_{ij} \in \{0, 1\}$. The α term biases the objective towards stronger correlation by decreasing the cost if the match matrix has more ones in it. The function $g(\mathbf{A})$ serves to regularize the Affine transformation by penalizing large values of the scale and shear components:

$$g(\mathbf{A}(\mathbf{a}, \theta, \mathbf{b}, \mathbf{c})) = \gamma(\mathbf{a}^2 + \mathbf{b}^2 + \mathbf{c}^2)$$

for some regularization parameter γ .

The RPM method optimizes the cost function using the Softassign algorithm. The 1D case will be derived here. Given a set of variables $\{\mathbf{Q}_j\}$ where $\mathbf{Q}_j \in \mathbb{R}^1$. A variable μ_j is associated with each \mathbf{Q}_j such that $\sum_{j=1}^J \mu_j = 1$. The goal is to find μ that maximizes $\sum_{j=1}^J \mu_j \mathbf{Q}_j$. This can be formulated as a continuous problem by introducing a control parameter $\beta > 0$. In the deterministic annealing method, the control parameter β is slowly increased as the algorithm runs. Let μ be:

$$\mu_j = \frac{\exp(\beta \mathbf{Q}_j)}{\sum_{j=1}^J \exp(\beta \mathbf{Q}_j)} \quad (\text{rpm.3})$$

this is known as the softmax function. As β increases, it approaches a binary value as desired in Equation (rpm.1). The problem may now be generalized to the 2D case, where instead of maximizing $\sum_{j=1}^J \mu_j \mathbf{Q}_j$, the following is maximized:

$$E(\mu) = \sum_{j=1}^N \sum_{i=0}^M \mu_{ij} \mathbf{Q}_{ij} \quad (\text{rpm.4})$$

where

$$\mathbf{Q}_{ij} = -(\|\mathbf{s}_j - \mathbf{t} - \mathbf{A}\mathbf{m}_i\|^2 - \alpha) = -\frac{\partial \text{cost}}{\partial \mu_{ij}}$$

This is straightforward, except that now the constraints on μ are doubly stochastic matrix constraints: $\forall j \sum_{i=1}^M \mu_{ij} = 1$ and $\forall i \sum_{j=1}^N \mu_{ij} = 1$. As such the denominator from Equation (rpm.3) cannot be expressed for the 2D case simply. To satisfy the constraints, it is possible to use a result due to Sinkhorn,^[39] which states that a doubly stochastic matrix is obtained from any square matrix with all positive entries by the iterative process of alternating row and column normalizations. Thus the algorithm is written as such:^[39]

```

algorithm RPM2D( $\mathcal{M}, \mathcal{S}$ )
   $\mathbf{t} := 0$ 
   $\mathbf{a}, \theta, \mathbf{b}, \mathbf{c} := 0$ 
   $\beta := \beta_0$ 
   $\hat{\mu}_{ij} := 1 + \epsilon$ 
  while  $\beta < \beta_f$ :
    while  $\mu$  has not converged:
      // update correspondence parameters by softassign
       $\mathbf{Q}_{ij} := -\frac{\partial \text{cost}}{\partial \mu_{ij}}$ 
       $\mu_{ij}^0 := \exp(\beta \mathbf{Q}_{ij})$ 
      // apply Sinkhorn's method
      while  $\hat{\mu}$  has not converged:
        // update  $\hat{\mu}$  by normalizing across all rows:
         $\hat{\mu}_{ij}^1 := \frac{\mu_{ij}^0}{\sum_{i=1}^{M+1} \mu_{ij}^0}$ 
        // update  $\hat{\mu}$  by normalizing across all columns:
         $\hat{\mu}_{ij}^0 := \frac{\hat{\mu}_{ij}^1}{\sum_{j=1}^{N+1} \hat{\mu}_{ij}^1}$ 
      // update pose parameters by coordinate descent
      update  $\theta$  using analytical solution
      update  $\mathbf{t}$  using analytical solution
      update  $\mathbf{a}, \mathbf{b}, \mathbf{c}$  using Newton's method
     $\beta := \beta_r \beta$ 
     $\gamma := \frac{\gamma}{\beta_r}$ 
  return  $\mathbf{a}, \mathbf{b}, \mathbf{c}, \theta$  and  $\mathbf{t}$ 

```

where the deterministic annealing control parameter β is initially set to β_0 and increases by factor β_r until it reaches the maximum value β_f . The summations in the normalization steps sum to $M + 1$ and $N + 1$ instead of just M and N because the constraints on μ are inequalities. As such the $M + 1$ th and $N + 1$ th elements are slack variables.

The algorithm can also be extended for point sets in 3D or higher dimensions. The constraints on the correspondence matrix μ are the same in the 3D case as in the 2D case. Hence the structure of the algorithm remains unchanged, with the main difference being how the rotation and translation matrices are solved.^[39]

Thin plate spline robust point matching

The thin plate spline robust point matching (TPS-RPM) algorithm by Chui and Rangarajan augments the RPM method to perform non-rigid registration by parametrizing the transformation as a thin plate spline.^[14] However, because the thin plate spline parametrization only exists in three dimensions, the method cannot be extended to problems involving four or more dimensions.

Kernel correlation

The kernel correlation (KC) approach of point set registration was introduced by Tsin and Kanade.^[37] Compared with ICP, the KC algorithm is more robust against noisy data. Unlike ICP, where, for every model point, only the closest scene point is considered, here every scene point affects every model point.^[37] As such this is a *multiply-linked* registration algorithm. For some kernel function K , the kernel correlation KC of two points $\mathbf{x}_i, \mathbf{x}_j$ is defined thus:^[37]

$$KC(\mathbf{x}_i, \mathbf{x}_j) = \int K(\mathbf{x}, \mathbf{x}_i) \cdot K(\mathbf{x}, \mathbf{x}_j) d\mathbf{x} \quad (\text{kc.1})$$

The kernel function K chosen for point set registration is typically symmetric and non-negative kernel, similar to the ones used in the Parzen window density estimation. The Gaussian kernel typically used for its simplicity, although other ones like the Epanechnikov kernel and the tricube kernel may be substituted.^[37] The kernel correlation of an entire point set \mathcal{X} is defined as the sum of the kernel correlations of every point in the set to every other point in the set:^[37]

$$KC(\mathcal{X}) = \sum_{i \neq j} KC(\mathbf{x}_i, \mathbf{x}_j) = 2 \sum_{i < j} KC(\mathbf{x}_i, \mathbf{x}_j) \quad (\text{kc.2})$$

The logarithm of KC of a point set is proportional, within a constant factor, to the information entropy. Observe that the KC is a measure of a "compactness" of the point set—trivially, if all points in the point set were at the same location, the KC would evaluate to a large value. The cost function of the point set registration algorithm for some transformation parameter θ is defined thus:

$$\text{cost}(\mathcal{S}, \mathcal{M}, \theta) = - \sum_{\mathbf{m} \in \mathcal{M}} \sum_{\mathbf{s} \in \mathcal{S}} KC(\mathbf{s}, T(\mathbf{m}, \theta)) \quad (\text{kc.3})$$

Some algebraic manipulation yields:

$$KC(\mathcal{S} \cup T(\mathcal{M}, \theta)) = KC(\mathcal{S}) + KC(T(\mathcal{M}, \theta)) - 2 \text{cost}(\mathcal{S}, \mathcal{M}, \theta) \quad (\text{kc.4})$$

The expression is simplified by observing that $KC(\mathcal{S})$ is independent of θ . Furthermore, assuming rigid registration, $KC(T(\mathcal{M}, \theta))$ is invariant when θ is changed because the Euclidean distance between every pair of points stays the same under rigid transformation. So the above equation may be rewritten as:

$$KC(\mathcal{S} \cup T(\mathcal{M}, \theta)) = C - 2 \text{cost}(\mathcal{S}, \mathcal{M}, \theta) \quad (\text{kc.5})$$

The kernel density estimates are defined as:

$$P_{\mathcal{M}}(\mathbf{x}, \theta) = \frac{1}{M} \sum_{\mathbf{m} \in \mathcal{M}} K(\mathbf{x}, T(\mathbf{m}, \theta))$$

$$P_{\mathcal{S}}(\mathbf{x}) = \frac{1}{N} \sum_{\mathbf{s} \in \mathcal{S}} K(\mathbf{x}, \mathbf{s})$$

The cost function can then be shown to be the correlation of the two kernel density estimates:

$$\text{cost}(\mathcal{S}, \mathcal{M}, \theta) = -N^2 \int_{\mathbf{x}} P_{\mathcal{M}} \cdot P_{\mathcal{S}} d\mathbf{x} \quad (\text{kc.6})$$

Having established the cost function, the algorithm simply uses gradient descent to find the optimal transformation. It is computationally expensive to compute the cost function from scratch on every iteration, so a discrete version of the cost function Equation (kc.6) is used. The kernel density estimates $P_{\mathcal{M}}, P_{\mathcal{S}}$ can be evaluated at grid points and stored in a lookup table. Unlike the ICP and related methods, it is not necessary to find the nearest neighbour, which allows the KC algorithm to be comparatively simple in implementation.

Compared to ICP and EM-ICP for noisy 2D and 3D point sets, the KC algorithm is less sensitive to noise and results in correct registration more often.^[37]

Gaussian mixture model

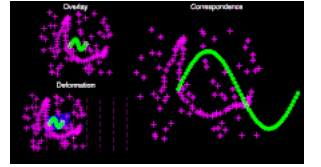
The kernel density estimates are sums of Gaussians and may therefore be represented as Gaussian mixture models (GMM).^[40] Jian and Vemuri use the GMM version of the KC registration algorithm to perform non-rigid registration parametrized by thin plate splines.

Coherent point drift

Coherent point drift (CPD) was introduced by Myronenko and Song.^{[13][41]} The algorithm takes a probabilistic approach to aligning point sets, similar to the GMM KC method. Unlike earlier approaches to non-rigid registration which assume a thin plate spline transformation model, CPD is agnostic with regard to the transformation model used. The point set \mathcal{M} represents the Gaussian mixture model (GMM) centroids. When the two point sets are optimally aligned, the correspondence is the maximum of the GMM posterior probability for a given data point. To preserve the topological structure of the point sets, the GMM centroids are forced to move coherently as a group. The expectation maximization algorithm is used to optimize the cost function.^[13]

Let there be M points in \mathcal{M} and N points in \mathcal{S} . The GMM probability density function for a point \mathbf{s} is:

$$p(\mathbf{s}) = \sum_{i=1}^{M+1} P(i)p(\mathbf{s}|i) \quad (\text{cpd.1})$$

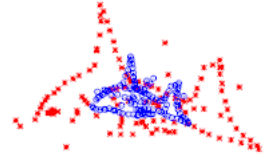


Animation of 2D non-rigid registration of the green point set \mathcal{M} to the magenta point set \mathcal{S} corrupted with noisy outliers. The size of the blue circles is inversely related to the control parameter β . The yellow lines indicate correspondence.

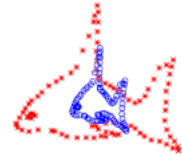
where, in D dimensions, $p(s|i)$ is the [Gaussian distribution](#) centered on point $m_i \in \mathcal{M}$.

$$p(s|i) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left(-\frac{\|s - m_i\|^2}{2\sigma^2}\right)$$

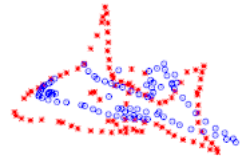
The membership probabilities $P(i) = \frac{1}{M}$ is equal for all GMM components. The weight of the uniform distribution is denoted as $w \in [0, 1]$. The mixture model is then:



Rigid (with the addition of scaling) registration of a blue point set \mathcal{M} to the red point set \mathcal{S} using the Coherent Point Drift algorithm. Both point sets have been corrupted with removed points and random spurious outlier points.



Affine registration of a blue point set \mathcal{M} to the red point set \mathcal{S} using the Coherent Point Drift algorithm.



Non-rigid registration of a blue point set \mathcal{M} to the red point set \mathcal{S} using the Coherent Point Drift algorithm. Both point sets have been corrupted with removed points and random spurious outlier points.

$$p(s) = w \frac{1}{N} + (1 - w) \sum_{i=1}^M \frac{1}{M} p(s|i) \quad (\text{cpd.2})$$

The GMM centroids are re-parametrized by a set of parameters θ estimated by maximizing the likelihood. This is equivalent to minimizing the negative [log-likelihood function](#):

$$E(\theta, \sigma^2) = - \sum_{j=1}^N \log \sum_{i=1}^{M+1} P(i) p(s_j|i) \quad (\text{cpd.3})$$

where it is assumed that the data is [independent and identically distributed](#). The correspondence probability between two points m_i and s_j is defined as the [posterior probability](#) of the GMM centroid given the data point:

$$P(i|s_j) = \frac{P(i)p(s_j|i)}{p(s_j)}$$

The [expectation maximization](#) (EM) algorithm is used to find θ and σ^2 . The EM algorithm consists of two steps. First, in the E-step or *estimation* step, it guesses the values of parameters ("old" parameter values) and then uses [Bayes' theorem](#) to compute the posterior probability distributions $P^{\text{old}}(i, s_j)$ of mixture components. Second, in the M-step or *maximization* step, the "new" parameter values are then found by minimizing the expectation of the complete negative log-likelihood function, i.e. the cost function:

$$\text{cost} = - \sum_{j=1}^N \sum_{i=1}^{M+1} P^{\text{old}}(i|s_j) \log(P^{\text{new}}(i) p^{\text{new}}(s_j|i)) \quad (\text{cpd.4})$$

Ignoring constants independent of θ and σ , Equation (cpd.4) can be expressed thus:

$$\text{cost}(\theta, \sigma^2) = \frac{1}{2\sigma^2} \sum_{j=1}^N \sum_{i=1}^{M+1} P^{\text{old}}(i|s_j) \|s_j - T(m_i, \theta)\|^2 + \frac{N_F D}{2} \log \sigma^2 \quad (\text{cpd.5})$$

where

$$N_{\mathbf{P}} = \sum_{j=0}^N \sum_{i=0}^M P^{\text{old}}(i|s_j) \leq N$$

with $N = N_{\mathbf{P}}$ only if $\mathbf{w} = \mathbf{0}$. The posterior probabilities of GMM components computed using previous parameter values \mathbf{P}^{old} is:

$$P^{\text{old}}(i|s_j) = \frac{\exp\left(-\frac{1}{2\sigma^{\text{old}2}}\|s_j - T(\mathbf{m}_i, \theta^{\text{old}})\|^2\right)}{\sum_{k=1}^M \exp\left(-\frac{1}{2\sigma^{\text{old}2}}\|s_j - T(\mathbf{m}_k, \theta^{\text{old}})\|^2\right) + (2\pi\sigma^2)^{\frac{D}{2}} \frac{w}{1-w} \frac{M}{N}} \quad (\text{c})$$

Minimizing the cost function in Equation (cpd.5) necessarily decreases the negative log-likelihood function E in Equation (cpd.3) unless it is already at a local minimum.^[13] Thus, the algorithm can be expressed using the following pseudocode, where the point sets \mathcal{M} and \mathcal{S} are represented as $M \times D$ and $N \times D$ matrices \mathbf{M} and \mathbf{S} respectively:^[13]

```

algorithm CPD( $\mathcal{M}, \mathcal{S}$ )
   $\theta := \theta_0$ 
  initialize  $0 \leq w \leq 1$ 
   $\sigma^2 := \frac{1}{DNM} \sum_{j=1}^N \sum_{i=1}^M \|s_j - m_i\|^2$ 
  while not registered:
    // E-step, compute  $P$ 
    for  $i \in [1, M]$  and  $j \in [1, N]$ :
       $p_{ij} := \frac{\exp\left(-\frac{1}{2\sigma^2}\|s_j - T(\mathbf{m}_i, \theta)\|^2\right)}{\sum_{k=1}^M \exp\left(-\frac{1}{2\sigma^2}\|s_j - T(\mathbf{m}_k, \theta)\|^2\right) + (2\pi\sigma^2)^{\frac{D}{2}} \frac{w}{1-w} \frac{M}{N}}$ 
    // M-step, solve for optimal transformation
     $\{\theta, \sigma^2\} := \text{solve}(\mathbf{S}, \mathbf{M}, \mathbf{P})$ 
  return  $\theta$ 

```

where the vector $\mathbf{1}$ is a column vector of ones. The **solve** function differs by the type of registration performed. For example, in rigid registration, the output is a scale a , a rotation matrix \mathbf{R} , and a translation vector \mathbf{t} . The parameter θ can be written as a tuple of these:

$$\theta = \{a, \mathbf{R}, \mathbf{t}\}$$

which is initialized to one, the identity matrix, and a column vector of zeroes:

$$\theta_0 = \{1, \mathbf{I}, \mathbf{0}\}$$

The aligned point set is:

$$T(\mathbf{M}) = a\mathbf{MR}^T + \mathbf{1t}^T$$

The **solve_rigid** function for rigid registration can then be written as follows, with derivation of the algebra explained in Myronenko's 2010 paper.^[13]

```

solve_rigid( $\mathbf{S}, \mathbf{M}, \mathbf{P}$ )
   $N_{\mathbf{P}} := \mathbf{1}^T \mathbf{P} \mathbf{1}$ 
   $\mu_s := \frac{1}{N_{\mathbf{P}}} \mathbf{S}^T \mathbf{P}^T \mathbf{1}$ 
   $\mu_m := \frac{1}{N_{\mathbf{P}}} \mathbf{M}^T \mathbf{P} \mathbf{1}$ 
   $\hat{\mathbf{S}} := \mathbf{S} - \mathbf{1}\mu_s^T$ 
   $\hat{\mathbf{M}} := \mathbf{M} - \mathbf{1}\mu_m^T$ 
   $\mathbf{A} := \mathbf{S}^T \mathbf{P}^T \hat{\mathbf{M}}$ 
   $\mathbf{U}, \mathbf{V} := \text{svd}(\mathbf{A})$  // the singular value decomposition of  $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ 
   $\mathbf{C} := \text{diag}(1, \dots, 1, \det(\mathbf{UV}^T))$  //  $\text{diag}(\xi)$  is the diagonal matrix formed from vector  $\xi$ 
   $\mathbf{R} := \mathbf{UCV}^T$ 
   $a := \frac{\text{tr}(\mathbf{A}^T \mathbf{R})}{\text{tr}(\hat{\mathbf{M}}^T \text{diag}(\mathbf{P} \mathbf{1}) \hat{\mathbf{M}})}$  //  $\text{tr}$  is the trace of a matrix
   $\mathbf{t} := \mu_s - a\mathbf{R}\mu_m$ 
   $\sigma^2 := \frac{1}{N_{\mathbf{P}} D} (\text{tr}(\hat{\mathbf{S}}^T \text{diag}(\mathbf{P}^T \mathbf{1}) \hat{\mathbf{S}}) - a \text{tr}(\mathbf{A}^T \mathbf{R}))$ 
  return  $\{a, \mathbf{R}, \mathbf{t}\}, \sigma^2$ 

```

For affine registration, where the goal is to find an affine transformation instead of a rigid one, the output is an affine transformation matrix \mathbf{B} and a translation \mathbf{t} such that the aligned point set is:

$$T(\mathbf{M}) = \mathbf{MB}^T + \mathbf{1t}^T$$

The **solve_affine** function for rigid registration can then be written as follows, with derivation of the algebra explained in Myronenko's 2010 paper.^[13]

```

solve_affine( $\mathbf{S}, \mathbf{M}, \mathbf{P}$ )
   $N_{\mathbf{P}} := \mathbf{1}^T \mathbf{P} \mathbf{1}$ 
   $\mu_s := \frac{1}{N_{\mathbf{P}}} \mathbf{S}^T \mathbf{P}^T \mathbf{1}$ 
   $\mu_m := \frac{1}{N_{\mathbf{P}}} \mathbf{M}^T \mathbf{P} \mathbf{1}$ 
   $\hat{\mathbf{S}} := \mathbf{S} - \mathbf{1}\mu_s^T$ 
   $\hat{\mathbf{M}} := \mathbf{M} - \mathbf{1}\mu_m^T$ 
   $\mathbf{B} := (\hat{\mathbf{S}}^T \mathbf{P}^T \hat{\mathbf{M}})(\hat{\mathbf{M}}^T \text{diag}(\mathbf{P} \mathbf{1}) \hat{\mathbf{M}})^{-1}$ 
   $\mathbf{t} := \mu_s - \mathbf{B}\mu_m$ 
   $\sigma^2 := \frac{1}{N_{\mathbf{P}} D} (\text{tr}(\hat{\mathbf{S}}^T \text{diag}(\mathbf{P}^T \mathbf{1}) \hat{\mathbf{S}}) - \text{tr}(\hat{\mathbf{S}}^T \mathbf{P}^T \hat{\mathbf{M}} \mathbf{B}^T))$ 
  return  $\{\mathbf{B}, \mathbf{t}\}, \sigma^2$ 

```


It is also possible to use CPD with non-rigid registration using a parametrization derived using [calculus of variations](#).^[13]

Sums of Gaussian distributions can be computed in [linear time](#) using the [fast Gauss transform](#) (FGT).^[13] Consequently, the [time complexity](#) of CPD is $O(M + N)$, which is asymptotically much faster than $O(MN)$ methods.^[13]

Bayesian coherent point drift (BCPD)

A variant of coherent point drift, called Bayesian coherent point drift (BCPD), was derived through a Bayesian formulation of point set registration.^[42] BCPD has several advantages over CPD, e.g., (1) nonrigid and rigid registrations can be performed in a single algorithm, (2) the algorithm can be accelerated regardless of the Gaussianity of a Gram matrix to define motion coherence, (3) the algorithm is more robust against outliers because of a more reasonable definition of an outlier distribution. Additionally, in the Bayesian formulation, motion coherence was introduced through a prior distribution of displacement vectors, providing a clear difference between tuning parameters that control motion coherence.

Sorting the Correspondence Space (SCS)

This algorithm was introduced in 2013 by H. Assalih to accommodate sonar image registration.^[43] These types of images tend to have high amounts of noise, so it is expected to have many outliers in the point sets to match. SCS delivers high robustness against outliers and can surpass ICP and CPD performance in the presence of outliers. SCS doesn't use iterative optimization in high dimensional space and is neither probabilistic nor spectral. SCS can match rigid and non-rigid transformations, and performs best when the target transformation is between three and six [degrees of freedom](#).

See also

- Point feature matching
- Point-set triangulation

References

- Zhang, Ji; Singh, Sanjiv (May 2015). "Visual-lidar odometry and mapping: low-drift, robust, and fast". *2015 IEEE International Conference on Robotics and Automation (ICRA)*: 2174–2181. doi:10.1109/ICRA.2015.7139486 (https://doi.org/10.1109%2FICRA.2015.7139486). ISBN 978-1-4799-6923-4. S2CID 6054487 (https://api.semanticscholar.org/CorpusID:6054487).
- Choi, Sungjoon; Zhou, Qian-Yi; Koltun, Vladlen (2015). "Robust reconstruction of indoor scenes" (https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Choi_Robust_Reconstruction_of_2015_CVPR_paper.pdf) (PDF). *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*: 5556–5565.
- Lai, Kevin; Bo, Liefeng; Ren, Xiaofeng; Fox, Dieter (May 2011). "A large-scale hierarchical multi-view RGB-D object dataset". *2011 IEEE International Conference on Robotics and Automation*: 1817–1824. CiteSeerX 10.1.1.190.1598 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.190.1598). doi:10.1109/ICRA.2011.5980382 (https://doi.org/10.1109%2FICRA.2011.5980382). ISBN 978-1-61284-386-5. S2CID 14986048 (https://api.semanticscholar.org/CorpusID:14986048).
- Yang, Heng; Carlone, Luca (2019). "A polynomial-time solution for robust registration with extreme outlier rates". *Robotics: Science and Systems*. arXiv:1903.08588 (https://arxiv.org/abs/1903.08588). doi:10.15607/RSS.2019.XV.003 (https://doi.org/10.15607%2FRSS.2019.XV.003). ISBN 978-0-9923747-5-4. S2CID 84186750 (https://api.semanticscholar.org/CorpusID:84186750).
- Calli, Berk; Singh, Arjun; Bruce, James; Walsman, Aaron; Konolige, Kurt; Srinivasa, Siddhartha; Abbeel, Pieter; Dollar, Aaron M (2017-03-01). "Yale-CMU-Berkeley dataset for robotic manipulation research". *The International Journal of Robotics Research*. **36** (3): 261–268. doi:10.1177/0278364917700714 (https://doi.org/10.1177%2F0278364917700714). ISSN 0278-3649 (https://www.worldcat.org/issn/0278-3649). S2CID 6522002 (https://api.semanticscholar.org/CorpusID:6522002).
- Cadena, Cesar; Carlone, Luca; Carrillo, Henry; Latif, Yasir; Scaramuzza, Davide; Neira, José; Reid, Ian; Leonard, John J. (December 2016). "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age". *IEEE Transactions on Robotics*. **32** (6): 1309–1332. arXiv:1606.05830 (https://arxiv.org/abs/1606.05830). Bibcode:2016arXiv160605830C (https://ui.adsabs.harvard.edu/abs/2016arXiv160605830C). doi:10.1109/TRO.2016.2624754 (https://doi.org/10.1109%2FTRO.2016.2624754). ISSN 1941-0468 (https://www.worldcat.org/issn/1941-0468). S2CID 2596787 (https://api.semanticscholar.org/CorpusID:2596787).
- Mur-Artal, Raúl; Montiel, J. M. M.; Tardós, Juan D. (October 2015). "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". *IEEE Transactions on Robotics*. **31** (5): 1147–1163. arXiv:1502.00956 (https://arxiv.org/abs/1502.00956). Bibcode:2015arXiv150200956M (https://ui.adsabs.harvard.edu/abs/2015arXiv150200956M). doi:10.1109/TRO.2015.2463671 (https://doi.org/10.1109%2FTRO.2015.2463671). ISSN 1941-0468 (https://www.worldcat.org/issn/1941-0468). S2CID 206775100 (https://api.semanticscholar.org/CorpusID:206775100).
- Yang, Heng; Carlone, Luca (2019). "A Quaternion-based Certifiably Optimal Solution to the Wahba Problem with Outliers" (http://openaccess.thecvf.com/content_ICCV_2019/papers/Yang_A_Quaternion-Based_Certifiably_Optimal_Solution_to_the_Wahba_Problem_With_ICC_V_2019_paper.pdf) (PDF). *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*: 1665–1674. arXiv:1905.12536 (https://arxiv.org/abs/1905.12536). Bibcode:2019arXiv190512536Y (https://ui.adsabs.harvard.edu/abs/2019arXiv190512536Y).
- Newcombe, Richard A.; Izadi, Shahram; Hilliges, Otmar; Molyneaux, David; Kim, David; Davison, Andrew J.; Kohi, Pushmeet; Shotton, Jamie; Hodges, Steve; Fitzgibbon, Andrew (October 2011). "KinectFusion: Real-time dense surface mapping and tracking". *2011 10th IEEE International Symposium on Mixed and Augmented Reality*: 127–136. CiteSeerX 10.1.1.453.53 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.453.53). doi:10.1109/ISMAR.2011.6092378 (https://doi.org/10.1109%2FISMAR.2011.6092378). ISBN 978-1-4577-2183-0. S2CID 11830123 (https://api.semanticscholar.org/CorpusID:11830123).
- Audette, Michel A.; Ferrie, Frank P.; Peters, Terry M. (2000-09-01). "An algorithmic overview of surface registration techniques for medical imaging". *Medical Image Analysis*. **4** (3): 201–217. doi:10.1016/S1361-8415(00)00014-1 (https://doi.org/10.1016%2FS1361-8415(00)00014-1). ISSN 1361-8415 (https://www.worldcat.org/issn/1361-8415). PMID 11145309 (https://pubmed.ncbi.nlm.nih.gov/11145309).
- Jian, Bing; Vemuri, Baba C. (2011). "Robust Point Set Registration Using Gaussian Mixture Models". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **33** (8): 1633–1645. doi:10.1109/tpami.2010.223 (https://doi.org/10.1109%2Ftpami.2010.223). PMID 21173443 (https://pubmed.ncbi.nlm.nih.gov/21173443). S2CID 10923565 (https://api.semanticscholar.org/CorpusID:10923565).
- Fitzgibbon, Andrew W. (2003). "Robust registration of 2D and 3D point sets". *Image and Vision Computing*. **21** (13): 1145–1153. CiteSeerX 10.1.1.335.116 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.335.116). doi:10.1016/j.imavis.2003.09.004 (https://doi.org/10.1016%2Fj.imavis.2003.09.004).

13. Myronenko, Andriy; Song, Xubo (2010). "Point set registration: Coherent Point drift". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **32** (2): 2262–2275. arXiv:0905.2635 (https://arxiv.org/abs/0905.2635). doi:10.1109/tpami.2010.46 (https://doi.org/10.1109/2Ftpami.2010.46). PMID 20975122 (https://pubmed.ncbi.nlm.nih.gov/20975122). S2CID 10809031 (https://api.semanticscholar.org/CorpusID:10809031).
14. Chui, Haili; Rangarajan, Anand (2003). "A new point matching algorithm for non-rigid registration". *Computer Vision and Image Understanding*. **89** (2): 114–141. CiteSeerX 10.1.1.7.4365 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.7.4365). doi:10.1016/S1077-3142(03)00009-2 (https://doi.org/10.1016%2FS1077-3142%2803%2900009-2).
15. Holz, Dirk; Ichim, Alexandru E.; Tombari, Federico; Rusu, Radu B.; Behnke, Sven (2015). "Registration with the Point Cloud Library: A Modular Framework for Aligning in 3-D" (https://www.researchgate.net/publication/283198426). *IEEE Robotics Automation Magazine*. **22** (4): 110–124. doi:10.1109/MRA.2015.2432331 (https://doi.org/10.1109%2FMRA.2015.2432331). S2CID 2621807 (https://api.semanticscholar.org/CorpusID:2621807).
16. Horn, Berthold K. P. (1987-04-01). "Closed-form solution of absolute orientation using unit quaternions". *JOSA A*. **4** (4): 629–642. Bibcode:1987JOSAA...4..629H (https://ui.adsabs.harvard.edu/abs/1987JOSAA...4..629H). doi:10.1364/JOSAA.4.000629 (https://doi.org/10.1364%2FJOSAA.4.000629). ISSN 1520-8532 (https://www.worldcat.org/issn/1520-8532).
17. Arun, K. S.; Huang, T. S.; Blostein, S. D. (September 1987). "Least-Squares Fitting of Two 3-D Point Sets". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. PAMI-9 (5): 698–700. doi:10.1109/TPAMI.1987.4767965 (https://doi.org/10.1109%2FTPAMI.1987.4767965). ISSN 1939-3539 (https://www.worldcat.org/issn/1939-3539). PMID 21869429 (https://pubmed.ncbi.nlm.nih.gov/21869429). S2CID 8724100 (https://api.semanticscholar.org/CorpusID:8724100).
18. Briaes, Jesus; Gonzalez-Jimenez, Javier (July 2017). "Convex Global 3D Registration with Lagrangian Duality". *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*: 5612–5621. doi:10.1109/CVPR.2017.595 (https://doi.org/10.1109%2FCVPR.2017.595). hdl:10630/14599 (https://hdl.handle.net/10630%2F14599). ISBN 978-1-5386-0457-1. S2CID 11549421 (https://api.semanticscholar.org/CorpusID:11549421).
19. Yang, Heng; Shi, Jingnan; Carlone, Luca (2020-01-21). "TEASER: Fast and Certifiable Point Cloud Registration". arXiv:2001.07715 (https://arxiv.org/abs/2001.07715) [cs.RO (https://arxiv.org/archive/cs.RO)].
20. Parra Bustos, Álvaro; Chin, Tat-Jun (December 2018). "Guaranteed Outlier Removal for Point Cloud Registration with Correspondences". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **40** (12): 2868–2882. arXiv:1711.10209 (https://arxiv.org/abs/1711.10209). doi:10.1109/TPAMI.2017.2773482 (https://doi.org/10.1109%2FTPAMI.2017.2773482). ISSN 1939-3539 (https://www.worldcat.org/issn/1939-3539). PMID 29990122 (https://pubmed.ncbi.nlm.nih.gov/29990122). S2CID 3331003 (https://api.semanticscholar.org/CorpusID:3331003).
21. Chin, Tat-Jun; Suter, David (2017-02-27). "The Maximum Consensus Problem: Recent Algorithmic Advances". *Synthesis Lectures on Computer Vision*. **7** (2): 1–194. doi:10.2200/s00757ed1v01y201702cov011 (https://doi.org/10.2200%2FS00757ed1v01y201702cov011). ISSN 2153-1056 (https://www.worldcat.org/issn/2153-1056).
22. Wen, Fei; Ying, Rendong; Gong, Zheng; Liu, Peilin (February 2020). "Efficient Algorithms for Maximum Consensus Robust Fitting". *IEEE Transactions on Robotics*. **36** (1): 92–106. doi:10.1109/TRO.2019.2943061 (https://doi.org/10.1109%2FTRO.2019.2943061). ISSN 1941-0468 (https://www.worldcat.org/issn/1941-0468). S2CID 209976632 (https://api.semanticscholar.org/CorpusID:209976632).
23. Cai, Zhipeng; Chin, Tat-Jun; Koltun, Vladlen (2019). "Consensus Maximization Tree Search Revisited" (http://openaccess.thecvf.com/content_ICCV_2019/html/Cai_Consensus_Maximization_Tree_Search_Revisited_ICCV_2019_paper.html). *Proceedings of IEEE International Conference on Computer Vision (ICCV)*: 1637–1645. arXiv:1908.02021 (https://arxiv.org/abs/1908.02021). Bibcode:2019arXiv190802021C (https://ui.adsabs.harvard.edu/abs/2019arXiv190802021C).
24. Bazin, Jean-Charles; Seo, Yongduek; Pollefeys, Marc (2013). Lee, Kyoung Mu; Matsushita, Yasuyuki; Rehg, James M.; Hu, Zhanyi (eds.). "Globally Optimal Consensus Set Maximization through Rotation Search". *Computer Vision – ACCV 2012. Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer. **7725**: 539–551. doi:10.1007/978-3-642-37444-9_42 (https://doi.org/10.1007%2F978-3-642-37444-9_42). ISBN 978-3-642-37444-9.
25. Hartley, Richard I.; Kahl, Fredrik (2009-04-01). "Global Optimization through Rotation Space Search". *International Journal of Computer Vision*. **82** (1): 64–79. doi:10.1007/s11263-008-0186-9 (https://doi.org/10.1007%2Fs11263-008-0186-9). hdl:1885/50831 (https://hdl.handle.net/1885%2F50831). ISSN 1573-1405 (https://www.worldcat.org/issn/1573-1405). S2CID 509788 (https://api.semanticscholar.org/CorpusID:509788).
26. Fischler, Martin; Bolles, Robert (1981). "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". *Communications of the ACM*. **24** (6): 381–395. doi:10.1145/358669.358692 (https://doi.org/10.1145%2F358669.358692). S2CID 972888 (https://api.semanticscholar.org/CorpusID:972888).
27. Le, Huu Minh; Chin, Tat-Jun; Eriksson, Anders; Do, Thanh-Toan; Suter, David (2019). "Deterministic Approximate Methods for Maximum Consensus Robust Fitting". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **43** (3): 842–857. arXiv:1710.10003 (https://arxiv.org/abs/1710.10003). doi:10.1109/TPAMI.2019.2939307 (https://doi.org/10.1109%2FTPAMI.2019.2939307). ISSN 1939-3539 (https://www.worldcat.org/issn/1939-3539). PMID 31494545 (https://pubmed.ncbi.nlm.nih.gov/31494545). S2CID 29346470 (https://api.semanticscholar.org/CorpusID:29346470).
28. Bustos, Alvaro Parra; Chin, Tat-Jun; Neumann, Frank; Friedrich, Tobias; Katzmann, Maximilian (2019-02-04). "A Practical Maximum Clique Algorithm for Matching with Pairwise Constraints". arXiv:1902.01534 (https://arxiv.org/abs/1902.01534) [cs.CV (https://arxiv.org/archive/cs.CV)].
29. Huber, Peter J.; Ronchetti, Elvezio M. (2009-01-29). *Robust Statistics*. Wiley Series in Probability and Statistics. Hoboken, NJ, USA: John Wiley & Sons, Inc. doi:10.1002/9780470434697 (https://doi.org/10.1002%2F9780470434697). ISBN 978-0-470-43469-7.
30. Zhou, Qian-Yi; Park, Jaesik; Koltun, Vladlen (2016). Leibe, Bastian; Matas, Jiri; Sebe, Nicu; Welling, Max (eds.). "Fast Global Registration". *Computer Vision – ECCV 2016. Lecture Notes in Computer Science*. Cham: Springer International Publishing. **9906**: 766–782. doi:10.1007/978-3-319-46475-6_47 (https://doi.org/10.1007%2F978-3-319-46475-6_47). ISBN 978-3-319-46475-6.
31. MacTavish, Kirk; Barfoot, Timothy D. (2015). "At all Costs: A Comparison of Robust Cost Functions for Camera Correspondence Outliers". *2015 12th Conference on Computer and Robot Vision*: 62–69. doi:10.1109/CRV.2015.52 (https://doi.org/10.1109%2FCRV.2015.52). ISBN 978-1-4799-1986-4. S2CID 9305263 (https://api.semanticscholar.org/CorpusID:9305263).
32. Bosse, Michael; Agamennoni, Gabriel; Gilitschenski, Igor (2016). "Robust Estimation and Applications in Robotics" (https://ieeexplore.ieee.org/document/8187472). *Foundations and Trends in Robotics*. now. **4** (4): 225–269. doi:10.1561/23000000047 (https://doi.org/10.1561%2F23000000047).
33. Black, Michael J.; Rangarajan, Anand (1996-07-01). "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision". *International Journal of Computer Vision*. **19** (1): 57–91. doi:10.1007/BF00131148 (https://doi.org/10.1007%2FBF00131148). ISSN 1573-1405 (https://www.worldcat.org/issn/1573-1405). S2CID 7510079 (https://api.semanticscholar.org/CorpusID:7510079).
34. Blake, Andrew; Zisserman, Andrew (1987). *Visual reconstruction* (https://mitpress.mit.edu/books/visual-reconstruction). The MIT Press. ISBN 9780262524063.
35. Yang, Heng; Antonante, Pasquale; Tzoumas, Vasileios; Carlone, Luca (2020). "Graduated Non-Convexity for Robust Spatial Perception: From Non-Minimal Solvers to Global Outlier Rejection". *IEEE Robotics and Automation Letters*. **5** (2): 1127–1134. arXiv:1909.08605 (https://arxiv.org/abs/1909.08605). doi:10.1109/LRA.2020.2965893 (https://doi.org/10.1109%2FLRA.2020.2965893). ISSN 2377-3774 (https://www.worldcat.org/issn/2377-3774). S2CID 202660784 (https://api.semanticscholar.org/CorpusID:202660784).
36. Besl, Paul; McKay, Neil (1992). "A Method for Registration of 3-D Shapes" (https://www.researchgate.net/publication/3191994). *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **14** (2): 239–256. Bibcode:1992SPIE.1611..586B (https://ui.adsabs.harvard.edu/abs/1992SPIE.1611..586B). doi:10.1109/34.121791 (https://doi.org/10.1109%2F34.121791).

37. Tsin, Yanghai; Kanade, Takeo (2004). *A Correlation-Based Approach to Robust Point Set Registration*. *Computer Vision ECCV*. Lecture Notes in Computer Science. Vol. 3023. Springer Berlin Heidelberg. pp. 558–569. CiteSeerX 10.1.1.156.6729 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.156.6729). doi:10.1007/978-3-540-24672-5_44 (https://doi.org/10.1007%2F978-3-540-24672-5_44). ISBN 978-3-540-21982-8.
38. Rusinkiewicz, Szymon; Levoy, Marc (2001). *Efficient variants of the ICP algorithm*. Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling, 2001. IEEE. pp. 145–152. doi:10.1109/IM.2001.924423 (https://doi.org/10.1109%2FIM.2001.924423).
39. Gold, Steven; Rangarajan, Anand; Lu, Chien-Ping; Suguna, Pappu; Mjolsness, Eric (1998). "New algorithms for 2d and 3d point matching:: pose estimation and correspondence". *Pattern Recognition*. **38** (8): 1019–1031. Bibcode:1998PatRe..31.1019G (http://ui.adsabs.harvard.edu/abs/1998PatRe..31.1019G) (http://ui.adsabs.harvard.edu/abs/1998PatRe..31.1019G). doi:10.1016/S0031-3203(98)80010-1 (https://doi.org/10.1016%2FS0031-3203%2898%2980010-1).
40. Jian, Bing; Vemuri, Baba C. (2005). *A robust algorithm for point set registration using mixture of Gaussians*. Tenth IEEE International Conference on Computer Vision 2005. Vol. 2. pp. 1246–1251.
41. Myronenko, Andriy; Song, Xubo; Carriera-Perpinán, Miguel A. (2006). "Non-rigid point set registration: Coherent point drift" (http://papers.nips.cc/paper/2962-non-rigid-point-set-registration-coherent-point-drift). *Advances in Neural Information Processing Systems*. **19**: 1009–1016. Retrieved 31 May 2014.
42. Hirose, Osamu (2020). "A Bayesian formulation of coherent point drift" (https://doi.org/10.1109%2FTPAMI.2020.2971687). *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Early access (7): 1–18. doi:10.1109/TPAMI.2020.2971687 (https://doi.org/10.1109%2FTPAMI.2020.2971687). PMID 32031931 (https://pubmed.ncbi.nlm.nih.gov/32031931).
43. Assalih, Hassan. (2013). "Chapter 6: Sorting the Correspondence Space" (http://www.ros.hw.ac.uk/bitstream/handle/10399/2647/AssalihH_1013_eps.pdf) (PDF). *3D reconstruction and motion estimation using forward looking sonar* (Ph.D.). Heriot-Watt University.

External links

- Reference implementation of thin plate spline robust point matching (http://www.cise.ufl.edu/~anand/students/chui/research.html)
- Reference implementation of kernel correlation point set registration (https://www.cs.cmu.edu/~ytsin/KCReg/)
- Reference implementation of coherent point drift (https://sites.google.com/site/myronenko/research/cpd)
- Reference implementation of ICP variants (https://github.com/ethz-asl/libpointmatcher)
- Reference implementation of Bayesian coherent point drift (https://github.com/ohirose/bcpd)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Point-set_registration&oldid=1068846219"

This page was last edited on 30 January 2022, at 12:56 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.