In Search of Inliers: 3D Correspondence by Local and Global Voting

Anders Glent Buch[1] Yang Yang[2] Norbert Krüger[1] Henrik Gordon Petersen[1]
Maersk Mc-Kinney Moller Institute, University of Southern Denmark
[1]{anbu,norbert,hgp}@mmmi.sdu.dk [2]yayan13@student.sdu.dk

Abstract

We present a method for finding correspondence between 3D models. From an initial set of feature correspondences, our method uses a fast voting scheme to separate the inliers from the outliers. The novelty of our method lies in the use of a combination of local and global constraints to determine if a vote should be cast. On a local scale, we use simple, low-level geometric invariants. *On a global scale, we apply* covariant *constraints for finding compatible correspondences. We guide the sampling for collecting voters by downward dependencies on previous voting stages. All of this together results in an accurate matching procedure. We evaluate our algorithm by controlled and comparative testing on different datasets, giving superior performance compared to state of the art methods. In a final experiment, we apply our method for 3D object detection, showing potential use of our method within higher-level vision.*

1. Introduction

Consider the problem of matching two 3D point models $\mathcal{M} \subset \mathbb{R}^3$ and $\mathcal{M}' \subset \mathbb{R}^3$. For any point $p \in \mathcal{M}$, the aim is to find the matching point $p' \in \mathcal{M}'$, if such a point exists. The result of this assignment is a *correspondence*. When the full set of possible correspondences has been established, we say that $\mathcal{M}$ has been brought into correspondence with $\mathcal{M}'$. This represents a fundamental problem in computer vision and appears in *e.g.* object detection.

During local point matching or registration [3], point assignments are made by spatial proximity of p and p'. Correspondences are progressively built by estimation of the relative transformation between $\mathcal{M}$ and $\mathcal{M}'$, followed by a reassignment. In this paper, the focus is on free-form matching problems, where no prior assumption can be made on the proximity of the models. To address this problem, local invariant features have been used extensively, both in images and in 3D [2, 8, 11, 14, 15, 23]. In many practical scenarios, and especially in free-form matching problems, $\mathcal{M}$ and $\mathcal{M}'$ can be noisy and incomplete. In addition to this, either of the models can contain a significant amount of irrelevant data, or clutter. In Fig. 1, we show an example of such a scenario, in which $\mathcal{M}'$ has been captured by a sensor. Although shape features can provide many good matches, one must expect a high amount of outliers due to repetitive structures, noise, clutter and occlusions.

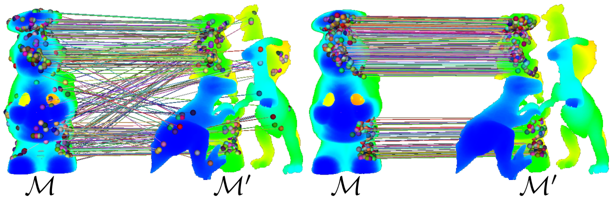


Figure 1: Matching results between a complete 3D model $\mathcal{M}$ and a captured scene $\mathcal{M}'$ with cluttering objects and 77 % occlusion of $\mathcal{M}$, taken from the dataset of [15]. Left: the 1 % highest ranked correspondences obtained by Lowe's ratio criterion. Right: the 1 % highest ranked correspondences after applying the proposed voting method.

Our contribution is a method for finding correct correspondences within a set of initial or putative feature correspondences between two 3D models, corrupted by incorrect matches. Our method employs a two-stage voting procedure for estimating the likelihood of a correspondence being correct based on different pairwise constraints. At the first stage, we use a low-level invariant distance constraint imposed on the local neighborhood of each correspondence. At the second stage, we use the highest ranked correspondences of the first stage and enforce a covariant pairwise constraint. The first stage exploits the local dependency of correspondences. The second stage provides more independent observations and utilizes the fact that correct pose hypotheses are stable on a global scale. Our method efficiently finds correct correspondences, while rejecting outliers, giving an increase in matching precision. A visualization of the two constraint types is shown in Fig. 2 on the following page.

This paper is structured as follows. Related methods are outlined in Sect. 2. Our method is presented in Sect. 3, and in Sect. 4 we provide experimental results. Finally, we draw conclusions and outline directions for future work in Sect. 5.

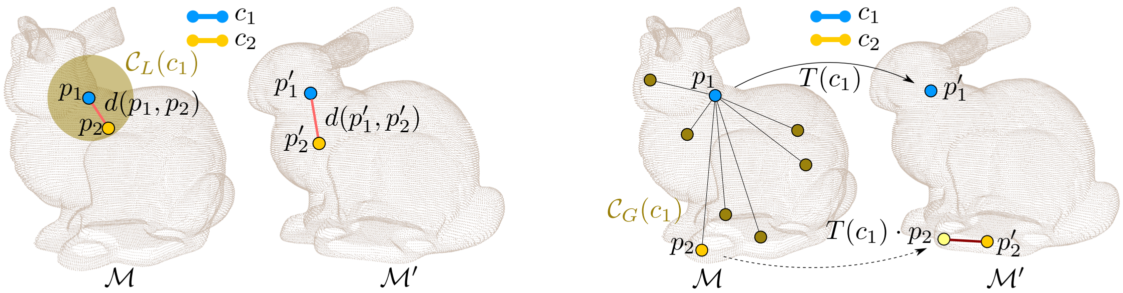


Figure 2: Schematics of the entities involved in local (left) and global (right) voting. Left: local voters are collected in the spherical neighborhood of a correspondence (dark yellow circle). The invariant pairwise compatibility υ_L is the minimum ratio of the light red distances. Right: global voters, from which c_2 is sampled, are located arbitrarily (dark yellow). The hypothesized transformation T (solid arrow) gives in this case an inaccurate alignment (dashed arrow) of p_2 (light yellow). The covariant compatibility υ_G uses the dark red distance between the hypothesized point and the assigned point p'_2.

2. Related work

Finding correspondences by voting processes has been subject to extensive investigation in the image domain. For completeness, we therefore start by outlining image-based correspondence finding methods, before describing the 3D methods used in our comparisons.

In [12, 24], the correspondence problem between sparse sets of image features is cast to a graph matching problem. The former recovers the inliers through spectral analysis of the affinity matrix of the feature matches. The latter finds a solution by minimization of an objective function, taking into account both the appearance and spatial arrangement of feature points. Common to graph matching methods is a high degree of flexibility, allowing for non-rigid matching. However, the high computational complexity makes these methods infeasible for dense matching problems with a high cardinality (several thousand features). Additionally, the results reported in [12, 24] require very high inlier rates from the feature matches, which cannot be assumed for our application, as shown in Sect. 4.

The *pyramid match kernel* [9] uses a fine to coarse matching strategy between sets of discriminative image features to achieve both robustness and discriminative power. Feature sets of unequal cardinality are allowed for, but accuracy is only justified for very sparse data (between 5 and 100 image features). In *Hough pyramid matching* (HPM) [22], local covariant image feature correspondences are cast to a tessellated transformation space. Correspondences are afterwards rejected by pyramid matching. The *Hough voting and Inverted Voting* (HVIV) [5] also uses covariant image features for casting votes, but preserves spatial locality of features before casting votes to arrive at accurate kernel densities. An additional inversion step propagates transformations on a local scale, by which an increased recall is achieved.

Lowe [14] investigated the use of a simple, yet efficient method for selecting good SIFT feature correspondences. The method assigns a penalty equal to the *ratio* of the closest to the second-closest feature distance. This measure is intrinsic to the feature space, and is immediately applicable to arbitrary feature types. On the other hand, only uniqueness is guaranteed, not necessarily discriminative power. Empirical data based on a large number of matched features suggest that this measure provides good separation of inliers and outliers. Even though the method has been used for image feature matches, it can be readily applied in 3D.

The *geometric consistency* (GC) framework groups 3D correspondences into disjoint clusters, favoring correspondences likely to produce good transformation hypotheses. Initial work was done by Johnson and Hebert [10, 11], where two oriented point pairs are grouped if they satisfy two geometric constraints. Firstly, the relative cylindrical coordinates of the point pair in first model must be compatible. Secondly, the method favors point pairs which have a large relative Euclidean distance, since this increases robustness of the subsequent transformation estimation step. Chen and Bhanu [4] relaxed the geometric constraints to only include a term which favors point pairs having similar distances. The omittance of orientation information in the grouping procedure increases robustness towards noise. Aldoma *et al.* [1] further robustified this method by applying a subsequent RANSAC [7] step to remove spurious correspondences from each cluster. The GC methods apply constraints that are extrinsic to the feature space by using point entities pertaining to $\mathbb{R}^3$.

The method presented in this paper uses Lowe's feature distance ratio for initialization. At the first voting stage, we use a constraint based on Euclidean distance ratios, as opposed to the absolute distances used by the GC methods. In the second and final voting stage, we use covariant constraints similar to HPM and HVIV, but adapted to $SE(3)$. Unlike other methods, our method combines different constraints, both on a local and global scale, and uses a voting mechanism with downward dependencies.

3. Proposed method

In this section, we describe our correspondence voting method. We start by introducing some terminology below, before describing our method in detail. To ease the readability of the following sections, we will refer to $\mathcal{M}$ as the *object* and $\mathcal{M}'$ as the *scene*.

3.1. Terminology

In the following, a correspondence c between the object and scene models $\mathcal{M}$ and $\mathcal{M}'$ is parameterized by two matched points $p \in \mathcal{M}$, $p' \in \mathcal{M}'$ and a real-valued matching score s:

$$c = (p, p', s) \tag{1}$$

Denote the feature space associated with $\mathcal{M}$ and $\mathcal{M}'$ as $\mathcal{F}$ and $\mathcal{F}'$, respectively. A feature is computed for each point, *i.e.* the feature sets are equivalent to the model point sets and have the same order. A feature-based correspondence is obtained by matching an invariant feature vector $f \in \mathcal{F}$ with the nearest matching feature $f' \in \mathcal{F}'$. Denote the general n-dimensional Euclidean L_2 distance metric as d:

$$d(a, b) \coloneqq \|a - b\|_{L_2} \qquad a, b \in \mathbb{R}^n \tag{2}$$

Associating a score to the feature match can now be done by the negative of the matching distance:

$$s_{\mathcal{F}}(c) \coloneqq -d(f, f') \tag{3}$$

The set of all correspondences is denoted $\mathcal{C} = \mathcal{M} \times \mathcal{M}' \times \mathbb{R}$. For free-form correspondence matching problems, there exists a unique subset of *correct* correspondences that brings $\mathcal{M}$ into correspondence with $\mathcal{M}'$:

$$\mathcal{C}_{Correct} \subset \mathcal{C} \tag{4}$$

which represents the objective of the matching process.

In the case of occlusions (as in Fig. 1), some points do not have a correspondence, and the scene $\mathcal{M}'$ contains an incomplete instance of $\mathcal{M}$. However, when performing dense feature matching—as we do in this work—all features in $\mathcal{M}$ will be assigned a *putative* feature match in $\mathcal{M}'$ by the feature matching process. We denote this initial set $\mathcal{C}_{\mathcal{F}} \subset \mathcal{C}$, and this serves as the input to our method. The problem is now to find the correspondences in $\mathcal{C}_{\mathcal{F}}$ that are also part of $\mathcal{C}_{Correct}$ (the inliers), while rejecting those that are not (the outliers). This makes the problem of finding correspondence a binary classification problem.

The *recall* of a matching method is defined as the ratio of correctly accepted correspondences to the number of inliers in $\mathcal{C}_{\mathcal{F}}$. The *precision* of a method is the ratio of correctly accepted correspondences to the total number of accepted correspondences. The initial *inlier fraction* in $\mathcal{C}_{\mathcal{F}}$ is thus the precision of the feature matching. An accurate matching method should therefore accept as many of the inliers as possible, while rejecting as many outliers as possible, giving an increase in precision.

3.2. Overview

The basic assumption behind our approach is that within the complete set of input feature correspondences $\mathcal{C}_{\mathcal{F}}$, the inliers should systematically satisfy certain geometric constraints, while the outliers should only do so randomly. We enforce these constraints in a voting framework where each correspondence is paired with a number of voter correspondences. Each positive vote increases the likelihood or ranking score of a correspondence. We bootstrap the process by an initialization step based on the feature distance ratio, and then perform two voting stages. At the first stage, we use invariant distance constraints on a local scale by collecting voters in the immediate neighborhood of each correspondence. The fraction of positive votes gives a crude ranking of each correspondence. In the second voting stage, we find voters on a global scale, based on the first stage ranks, and enforce a covariant constraint. We introduce an additional dependency between the stages by accumulating all votes.

3.3. Initialization

Our method requires an initial ranking of the input correspondences. We start by ranking all input correspondences by the feature distance ratio, which has proven more discriminative than the closest feature distance. Lowe's ratio penalizes correspondences by the ratio of the closest to the second-closest matching feature distance. Since d is a metric, this ratio will always lie in the interval $[0, 1]$, and we can define the ranking score of a feature correspondence as:

$$s_{Ratio}(c) \coloneqq 1 - \frac{d(f, f'_1)}{d(f, f'_2)} \tag{5}$$

where f'_1 and f'_2 are used for denoting the closest and second-closest feature match of f, respectively. The ratio method then performs hard thresholding as follows:

$$\mathcal{C}_{Ratio} = \{c \in \mathcal{C}_{\mathcal{F}} \ : \ s_{Ratio}(c) \geq t_{Ratio}\} \tag{6}$$

In the original work, an upper threshold for the penalty of 0.8 was determined using empirical data, giving a lower threshold on the score of $t_{Ratio} = 0.2$.

3.4. First voting stage: local invariants

At the first voting stage, we locate the κ-nearest Euclidean neighbors $\mathcal{N}$ of each correspondence on the object. For each correspondence c, we thus get a subset $\mathcal{N}(c) \subset \mathcal{C}_{\mathcal{F}}$ with $|\mathcal{N}(c)| = \kappa$. The number of neighbors κ is a free parameter of our method, and specifies the *sample size*. We can now collect local *voters* $\mathcal{C}_L$ as the subset of neighbors that satisfy the ratio threshold (6):

$$\mathcal{C}_L(c) = \{\mathcal{N}(c) \cap \mathcal{C}_{Ratio}\} \tag{7}$$

We pair the correspondence with each voter neighbor and measure their compatibility v_L by the minimum ratio of

Euclidean distances between the object points and the corresponding scene points (see Fig. 2, left):

$$\upsilon_L(c_1, c_2) := \min\left(\frac{d(p_1, p_2)}{d(p_1', p_2')}, \frac{d(p_1', p_2')}{d(p_1, p_2)}\right) \tag{8}$$

where the minimum of the two possible ratios is taken to get a result in $[0, 1]$. By using a relative distance ratio in υ_L, the compatibility function becomes invariant to the absolute sizes of the involved distance pairs.

The set of positive local *votes* Υ_L is the subset of local voters with a high compatibility:

$$\Upsilon_L(c) = \{c_L \in \mathcal{C}_L(c) \ : \ \upsilon_L(c, c_L) > \varsigma\} \tag{9}$$

where $\varsigma \in [0, 1[$ is the lower *similarity* and is the second free parameter of our method. Larger values make the method more restrictive (giving fewer votes), but this also requires a more accurate representation of the surfaces to be reliable.

Finally, the likelihood, or estimated local score, s_L, giving evidence of a correspondence under the local constraint, is calculated as the ratio of votes to the number of voters:

$$s_L(c) = \frac{|\Upsilon_L(c)|}{|\mathcal{C}_L(c)|} \tag{10}$$

It is worth noting that the local voter collection process (7) in some cases returns very small ($\lll \kappa$) or even empty sets, depending on how many neighbor features pass the ratio test. Although this rarely happens in our experience, we must handle this by setting $s_L = 0$ in the case of an empty set. The second voting stage explicitly handles the case of small voter sets by accumulating votes, as described in the following.

3.5. Second voting stage: covariant surface points

As previously mentioned, the first voting stage produces a crude estimate of each correspondence being correct using local neighbors in $\mathcal{M}$. This is justified by other studies, which have verified that correspondences exhibit local dependencies [5, 25], meaning that correct correspondences often occur together. However, this also implies that inliers occurring near outliers passing the ratio test will get a low local score. We address this issue by introducing a second voting stage where the κ globally highest ranked correspondences of the first stage are used.

We start by reordering $\mathcal{C}_\mathcal{F}$ according to (10) to get a monotonically decreasing sequence in s_L, denoted $\mathcal{C}_{s_L}$. We take out the κ top ranked correspondences, and arrive at a set of feasible voter correspondences for use in the global stage:

$$\mathcal{C}_G = \{c_i \in \mathcal{C}_{s_L}\}_{i=1}^{\kappa} \tag{11}$$

Since sampling is now based on s_L, the voters are collected globally on $\mathcal{M}$. We have also tested using a different number of global voters than κ at this stage, but found that best performance was achieved by reusing κ. Unlike the local stage, all correspondences share the same voters, and we always have $|\mathcal{C}_G| = \kappa$. We now compute a hypothesis transformation $T \in SE(3)$ for each input correspondence in $\mathcal{C}_\mathcal{F}$ using the reference frame (RF) associated to each feature point:

$$T(c) = T(p')^{-1} \cdot T(p) \tag{12}$$

The use of RFs is common in images [2, 13, 14, 20], where the local RF consists of a pixel position, an orientation angle and a scale. Recently, methods for finding repeatable RFs for 3D shape features have emerged [15, 23], and we require this information to be available.

The transformation T gives a hypothesis pose for bringing $\mathcal{M}$ into correspondence with $\mathcal{M}'$. Two correspondences c_1 and c_2 are compatible if c_2 *covaries* with the transformation hypothesized by c_1. We thus arrive at the following global compatibility function υ_G (see Fig. 2, right):

$$\upsilon_G(c_1, c_2) := d\left(T(c_1) \cdot p_2, p_2'\right) \tag{13}$$

We now find global votes Υ_G by applying both the local and the global constraint to the global voters $\mathcal{C}_G$:

$$\Upsilon_G(c) = \{c_G \in \mathcal{C}_G \ : \ \upsilon_L(c, c_G) > \varsigma \wedge \upsilon_G(c, c_G) < \delta\} \tag{14}$$

where δ is a Euclidean distance tolerance. To compensate for noise and inaccuracies in the RF rotation estimation, we set this tolerance to five times the point cloud resolution. If the resolution is not known a priori, it is estimated as the median distance between any model point and its nearest Euclidean neighbor. The local constraint υ_L is enforced on the global voters for two reasons. Firstly, the distance ratio constraint should be satisfied for rigid objects, no matter if correspondences are paired locally or globally. Secondly, υ_L is computationally cheap, and thus serves as a prerejection step to the more expensive υ_G.

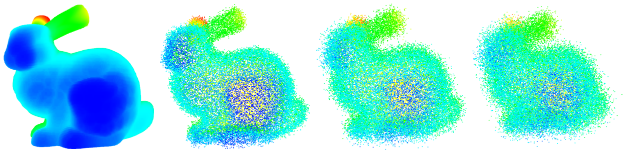
We integrate all votes and arrive at the final score function s as the likelihood computed by accumulating both local and global votes:

$$s(c) = \frac{|\Upsilon_L(c)| + |\Upsilon_G(c)|}{|\mathcal{C}_L(c)| + |\mathcal{C}_G(c)|} \tag{15}$$

which also makes it clear how small local voter sets is handled: smaller number of local voters gives higher relative importance to the global voters, which is a desirable effect as it reduces the bias from the small local sample size. We stress that in both stages the computed likelihoods have a downward dependency on the previous stage, introduced by the voter selection processes (7) and (11). This guided sampling increases precision, as we will demonstrate in Sect. 4.

3.6. Thresholding

Here we shortly describe how we perform the final thresholding to separate the inliers from the outliers based



$\sigma = 2.5\,\mathrm{mm}$ $\sigma = 5.0\,\mathrm{mm}$ $\sigma = 7.5\,\mathrm{mm}$

Figure 3: Original version of the Stanford *Bunny* model (leftmost) and three of the test models used in the controlled experiment, color rendered by depth value.

on the computed scores. The function $s(c)$ is real-valued, so the problem is now to calculate a decision threshold $t \in [0, 1]$, based on some optimality criterion.

To address this, we apply a well-known method from the image processing domain, namely Otsu's adaptive thresholding method [18], which is non-parametric and finds the optimal decision threshold in a sampled univariate distribution under the assumption of bimodality. The method estimates the probability density function of the data by a histogram, and then exhaustively searches for the threshold which maximizes the between-class variance. In Sect. 4 we give experimental justification for the use of this method.

3.7. Computational considerations and complexity

We end the description of our approach by considering its time complexity. The feature estimation process requires local neighbors, and it is often possible to reuse these point neighbors at the local stage of our method. If not, neighbors can be found in logarithmic time by spatial indexing, such as k-d trees. In all experiments presented below, we have reused the feature neighbors for collecting local voters.

Since we use a fixed sample size κ, and these are collected on the object, our algorithm is linear in the number of object points. The three components of our algorithm (initialization, local and global voting) each require a loop over all input correspondences, where the local and global stages each have an upper operation count of κ per correspondence. We thus get a final time complexity of $O(|\mathcal{M}| + \kappa \cdot |\mathcal{M}| + \kappa \cdot |\mathcal{M}|) = O(|\mathcal{M}|)$.

4. Experiments

We have performed both controlled and comparative experiments to evaluate our method. The standard evaluation procedure for feature matching is recall *vs*. 1-precision [16], and we adopt these measures here. All methods are evaluated by varying the threshold t on the score associated with the method. In addition, we compute maximum F_1 scores, giving a conservative estimate of the overall accuracy.

With regard to the feature estimation, there are many variabilities, *e.g*. support radius, matching distance metric *etc*. We found that changing the feature or the radius has minor impact on relative performances. Like similar studies in the image domain [9, 22], we use the same feature in all tests, the SHOT feature [23], which can be regarded as a SIFT-like shape feature. The SHOT features provide RFs for use in our global voting stage. The radius is set to $0.015\,\mathrm{m}$ for a good trade-off between robustness and discrimination. For neighbor search, we use k-d trees [17] to locate both point and feature neighbors by the L_2 metric.

All experiments have been performed in a single-threaded C++ application using a laptop computer equipped with a $2.2\,\mathrm{GHz}$ processor and $8\,\mathrm{GiB}$ memory.

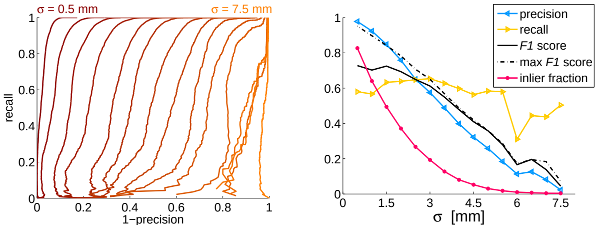


Figure 4: Performance measures for the *Bunny* experiment for increasing noise. Left: recall *vs*. 1-precision curves for all 15 noise levels (dark red to orange, left to right). Right: Precision, recall and F_1 score at decision threshold, maximum possible F_1 score and inlier fraction.

4.1. Controlled experiment

We start with a controlled experiment in order to test the effect of both noise and parameter changes on our method. All experiments described here are performed using the full Stanford *Bunny* model,[1] which contains 35947 vertices. We use the vertices and normals of the original mesh and add isotropic Gaussian point noise of increasing standard deviation $\sigma = \{0.5, 1.0, \ldots, 7.5\}$ mm, before computing features. For the high noise values, all local structures are severely distorted, rendering feature matching very challenging. See Fig. 3 for an illustration.

The original noise-free model is paired with each of the 15 noisy versions, and we evaluate our method on all 15 shape pairs in order to measure the degradation in performance as a result of noise. The results of this experiment can be seen in Fig. 4, showing both recall *vs*. 1-precision curves for each noise level and performance measures at the decision threshold determined by our method.

The results in Fig. 4 show a close to linear drop in precision with increasing noise, while achieving an almost constant recall, even though there is a rapid, sublinear drop in the inlier fraction. The F_1 score at the decision threshold t is close to the optimal value, which indicates good classification accuracy of the thresholding method. The matching destabilizes for $\sigma \geq 6.0\,\mathrm{mm}$, where the inlier fraction is close to zero.

We also tested the influence of parameter changes at a

[1] http://graphics.stanford.edu/data/3Dscanrep

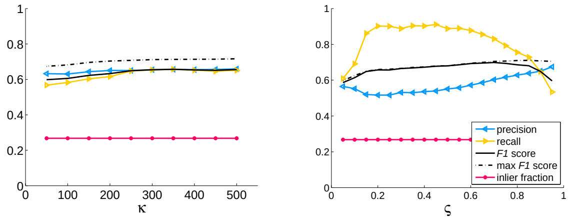


Figure 5: Precision, recall and F_1 score at the decision threshold, maximum possible F_1 score and inlier fraction for a fixed noise level of $\sigma = 2.5\,\text{mm}$. Left: performance for varying sample counts κ ($\varsigma = 0.9$). Right: performance for varying similarity thresholds ς ($\kappa = 250$).

fixed noise level of $\sigma = 2.5\,\text{mm}$ and performed the matching with a linear change in the sampling size κ and the similarity ς while fixing the other parameter. We show the results in Fig. 5, equivalent to the right part of Fig. 4.

Interestingly, the sample size κ has little influence on the results, whereas the similarity ς is more crucial to performance. The leftmost plot shows the convergent state of the algorithm for high sample sizes, which we have verified by testing even larger sample sizes. The interpretation of the rightmost plot in Fig. 5 is that ς represents a trade-off between precision and recall. When this value is set high, the method becomes more selective, leading to fewer votes for all correspondences. This gives an increased precision, since the few accepted correspondences are more reliable, but at the expense of recall. Both plots confirm the accuracy of the thresholding method; indeed the F_1 scores at the all decision thresholds are very close to the optimal value.

Based on these results, we use $\kappa = 250$ and $\varsigma = 0.9$ for a good trade-off between speed and accuracy in all the following experiments.

4.2. Comparative experiments

In this section, we present comparative experiments carried out on two different datasets. The methods used in the comparison are shortly described below.

L_2 distance: The baseline feature distance ranking simply uses the negative of the L_2 feature distance (3) for ranking correspondences. As noted before, this method is expected to be highly sensitive to *e.g.* repetitive structures.

Ratio: The ratio method [14] ranks each correspondence by the negative of Lowe's ratio penalty (5). Non-unique feature matches are now removed, but correctness of the remaining matches is not guaranteed.

Geometric consistency: The GC method [4] clusters correspondences by imposing an absolute pairwise distance constraint equal to the Euclidean distance between the feature points. The algorithm initializes a cluster with a seed correspondence and adds all correspondences that are compatible with the seed to the cluster. The clustered correspondences are then marked as visited, and the seed growing repeats until all correspondences have been visited. As an additional step, [1] applies RANSAC to each cluster to remove spurious correspondences for increased precision.

For our evaluations, we must apply a proper ranking to the correspondences output by GC. We found that the relative size of the containing cluster to the total input size performs better than *e.g.* the ratio or the RMSE reported by RANSAC. This is intuitive since the cluster size can be seen as an estimate of the inlier fraction, conditioned that a cluster contains inliers.

4.2.1 Results

We run comparative tests on two datasets, both synthetic and real. The first dataset of Tombari *et al.* [23] consists of 45 synthetic scenes containing between three and five instances of the Stanford models *Armadillo*, *Asian Dragon*, *Bunny*, *Dragon*, *Happy Buddha* and *Thai Statue*. All scenes are contaminated by isotropic Gaussian noise of $10\,\%$ of the spatial resolution, followed by a downsampling to half resolution. The protocol for this dataset is to sample 1000 keypoints per object, which allows us to also test the influence of sparse feature matching on our method. The second dataset by Mian *et al.* [15] contains four complete 3D models (*Chef*, *Para*, *T-rex* and *Chicken*) and 50 real scenes captured with a laser scanner (see Fig. 1). Prior to feature extraction, all models in the laser scanner dataset are downsampled to $2\,\text{mm}$, followed by surface normal estimation [21]. For all tests, we extract SHOT features and find ground truth inliers using the ground truth poses provided by the datasets, by requiring that matched points must be closer than two resolution units. The mean inlier fraction over all scenes ranges from $1.7\,\%$ (*T-rex*) to $4.0\,\%$ (*Chef*), making the task of finding the inliers very challenging.

Mean recall *vs.* 1-precision curves for both datasets are reported in Fig. 6. As expected, the L_2 distance matching shows poor performance. We explain this by the fact that feature distances are very sensitive to repetitive structures. The ratio method shows quite good performance for the synthetic dataset, but quickly degrades to the performance of the distance matching for the real scenes. Surprisingly, GC has better overall performance than GC+RANSAC. However, GC+RANSAC—being more selective—shows a higher initial precision, making it more suitable for algorithms requiring few inliers, *e.g.* pose estimation. The proposed method performs significantly better than all other methods, which we believe comes from the benefit of using different kinds of pairwise geometric constraints.

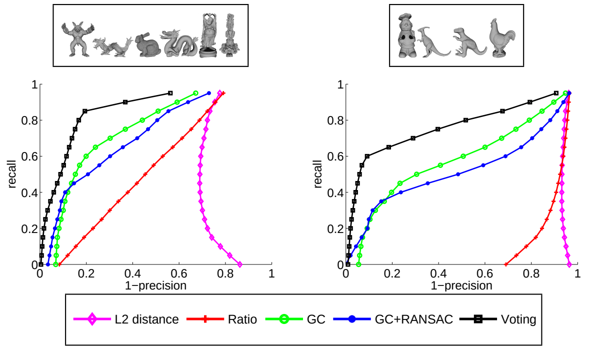


Figure 6: Overall results for the synthetic feature matching benchmark (left) and the real laser scanner dataset (right).

Additional performance measures are reported in Tab. 1. The feature distance is the baseline required for all methods, and marks the temporal starting point. Since the ratio score only involves a floating point division followed by a subtraction, it is very fast. Compared to the GC methods, our method provides several magnitudes of speedup.

Object	Method	Max F_1	Run time [s]
Chef	L_2 distance	0.13	-
(28940)	Ratio	0.22	0.012
	GC	0.67	1.7
	GC+RANSAC	0.64	1.9
	Voting	0.85	0.33
Para	L_2 distance	0.11	-
(16732)	Ratio	0.15	0.0058
	GC	0.57	0.64
	GC+RANSAC	0.54	0.77
	Voting	0.71	0.16
T-rex	L_2 distance	0.10	-
(15851)	Ratio	0.11	0.0051
	GC	0.56	0.56
	GC+RANSAC	0.47	0.65
	Voting	0.78	0.13
Chicken	L_2 distance	0.14	-
(12324)	Ratio	0.18	0.0039
	GC	0.59	0.35
	GC+RANSAC	0.56	0.41
	Voting	0.80	0.11

Table 1: Performance measures for the laser scanner dataset. The number below each object is the vertex count. Maximum F_1 scores are computed along the mean curves in Fig. 6, right. Run times are for full scenes, containing between approx. 16000 and 23000 vertices.

4.3. Application: object detection

In two final experiments, we demonstrate the power of our method by applying it for object detection. We embed our correspondence matching procedure in a naive detection system as follows. For each object model in the dataset, we input the full set of calculated scene features and calculate feature correspondences $\mathcal{C}_{\mathcal{F}}$, now with an increased radius of 0.03 m for better initial feature matching. Calculating scene features is done once per scene, and takes on average approx. 1 s. We run our algorithm and take the pose hypothesis of the single top ranked correspondence in the output. We run 10 ICP iterations [3] to refine the result and accept the detection if the aligned object model is covered at least 5 % by the scene data. We deliberately avoid using sophisticated methods for hypothesis segmentation or cross-verification in order to evaluate the strength of our method alone.

4.3.1 Results for laser scanner scenes

We applied the detection method to the real laser scanner dataset, which has been used for object detection comparisons in several works. As shown in Tab. 2, even with our simplistic system, we achieve good detection performance. Since there are no false positives, precision is 100 % for all objects. We encourage the reader to compare detection rates, and especially timings, with state of the art recognition systems such as [1, 6, 15, 19]. We believe this demonstrates high potential for the use of our method for higher-level matching tasks such as object detection.

Object	Recall [%]	Time [s]
Chef	100	0.39
Para	100	0.20
T-rex	100	0.18
Chicken	90	0.15

Table 2: Detection rates and mean timings for the laser scanner dataset. Timings include both correspondence voting and pose refinement.

4.3.2 Qualitative result from real experiment

We finally present a qualitative result from our own experimental setup, consisting of three calibrated stereo cameras. We project texture to the scene before extracting images and performing dense stereo matching. The full point cloud of the scene is obtained by aligning the reconstructed point clouds from the three views, followed by a 2 mm downsampling. The application is robotic (dis)assembly of three very similar pegs, which need to be automatically detected. We perform no scene preprocessing, such as *e.g.* ground plane removal, and input the full point cloud when detecting each object. To allow for multiple instances, we now use the top 100 ranked correspondences per object, and accept the refined pose if it does not overlap with a previous detection of the same object by more than 10 %. As shown in Fig. 7 on the next page, our method localizes the parts, even when multiple instances are present. The total detection time for this scene, including pose refinements, is 1.9 s.

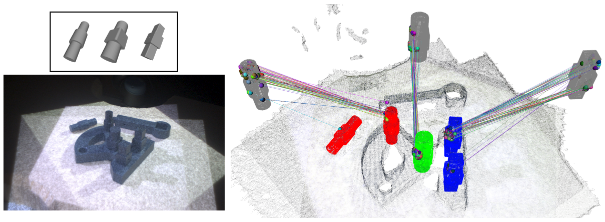


Figure 7: Left: objects (top) and left frame of one of three texture-projected stereo views used in our setup (bottom). Right: top 100 ranked correspondences and final detections within the point cloud for each object (red: round peg, green: round peg with square handle, blue: square peg).

5. Conclusions and future work

The method described in this paper allows for efficient and accurate retrieval of correspondences between 3D models based on putative matches obtained by feature matching. Evaluated on different datasets, the proposed method gives an increase in both speed and accuracy by up to several orders of magnitude compared to other methods. We have justified the use of our method for real-life vision problems by testing it for object detection, leading to promising results.

An extension of the method, which we plan to pursue in the future, is multi-instance correspondence voting including several object models. We expect this to achieve sublinear runtime increase in the number of models, which is essential for scalability. By such an extension, we intend to integrate our method into an object recognition framework, allowing for efficient detection of multiple 3D objects.

Acknowledgments: This work has been supported by the EC project IntellAct (FP7-ICT-269959). The financial support from the The Danish Council for Strategic Research through the project Carmen is gratefully acknowledged.

References

[1] A. Aldoma, F. Tombari, L. Stefano, and M. Vincze. A global hypotheses verification method for 3d object recognition. In *ECCV*, pages 511–524. 2012. 2, 6, 7

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *CVIU*, 110(3):346–359, 2008. 1, 4

[3] P. Besl and N. D. McKay. A method for registration of 3-d shapes. *TPAMI*, 14(2):239–256, 1992. 1, 7

[4] H. Chen and B. Bhanu. 3d free-form object recognition in range images using local surface patches. *Pattern Recognition Letters*, 28(10):1252–1262, 2007. 2, 6

[5] H.-Y. Chen, Y.-Y. Lin, and B.-Y. Chen. Robust feature matching with alternate hough and inverted hough transforms. In *CVPR*, pages 2762–2769, 2013. 2, 4

[6] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *CVPR*, pages 998–1005, 2010. 7

[7] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2

[8] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *ECCV*, pages 224–237. 2004. 1

[9] K. Grauman and T. Darrell. The pyramid match kernel: discriminative classification with sets of image features. In *ICCV*, volume 2, pages 1458–1465, 2005. 2, 5

[10] A. E. Johnson and M. Hebert. Surface matching for object recognition in complex three-dimensional scenes. *Image and Vision Computing*, 16(9):635–651, 1998. 2

[11] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *TPAMI*, 21(5):433–449, 1999. 1, 2

[12] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, volume 2, pages 1482–1489, 2005. 2

[13] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *ICCV*, pages 2548–2555, 2011. 4

[14] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1, 2, 4, 6

[15] A. Mian, M. Bennamoun, and R. Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *TPAMI*, 28(10):1584–1601, 2006. 1, 4, 6, 7

[16] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *TPAMI*, 27(10):1615–1630, 2005. 5

[17] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP*, pages 331–340, 2009. 5

[18] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979. 5

[19] C. Papazov and D. Burschka. An efficient ransac for 3d object recognition in noisy and occluded scenes. In *ACCV*, pages 135–148. 2011. 7

[20] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: an efficient alternative to sift or surf. In *ICCV*, pages 2564–2571, 2011. 4

[21] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *ICRA*. 2011. 6

[22] G. Tolias and Y. Avrithis. Speeded-up, relaxed spatial matching. In *ICCV*, pages 1653–1660, 2011. 2, 5

[23] F. Tombari, S. Salti, and L. Stefano. Unique signatures of histograms for local surface description. In *ECCV*, pages 356–369. 2010. 1, 4, 5, 6

[24] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. In *ECCV*, pages 596–609, 2008. 2

[25] P. Yarlagaddam, A. Monroy, and B. Ommer. Voting by grouping dependent parts. In *ECCV*, volume 5, pages 197–210. 2010. 4