Accurately Estimating Rigid Transformations in Registration using a Boosting-Inspired Mechanism

Yonghuai Liu[a], Honghai Liu[b], Ralph R. Martin[c], Luigi De Dominicis[d], Ran Song[e], Yitian Zhao[f*]

[a]Department of Computer Science
Aberystwyth University, Ceredigion SY23 3DB, UK
Email: yyl@aber.ac.uk

[b]School of Computing
University of Portsmouth, Portsmouth PO1 3HE
Email: honghai.liu@port.ac.uk

[c]School of Computer Science & Informatics
Cardiff University, Cardiff CF24 3AA, UK
Email: ralph@cs.cf.ac.uk

[d]Diagnostics and Metrology Laboratory UTAPRAD-DIM
00044 ENEA Frascati, Italy
Email: luigi.dedominicis@enea.it

[e]School of Computing, Engineering and Mathematics
University of Brighton, Brighton BN2 4GJ
Email: R.Song@brighton.ac.uk

[f]School of Optics and Electronics
Beijing Institute of Technology, Beijing 100081, China
Email: Yitian.Zhao@bit.edu.cn

Abstract

Feature extraction and matching provide the basis of many methods for object registration, modeling, retrieval, and recognition. However, this approach typically introduces false matches, due to lack of features, noise, occlusion, and cluttered backgrounds. In registration, these false matches lead to inaccurate estimation of the underlying transformation that brings the overlapping shapes into best possible alignment. In this paper, we propose a novel boosting-inspired method to tackle this challenging task. It includes three key steps: (i) underlying transformation estimation in

*Corresponding author. Email: yitian.zhao@bit.edu.cn

the weighted least squares sense, (ii) boosting parameter estimation and regularization via Tsallis entropy, and (iii) weight re-estimation and regularization via Shannon entropy and update with a maximum fusion rule. The process is iterated. The final optimal underlying transformation is estimated as a weighted average of the transformations estimated from the latest iterations, with weights given by the boosting parameters. A comparative study based on real shape data shows that the proposed method outperforms four other state-of-the-art methods for evaluating the established point matches, enabling more accurate and stable estimation of the underlying transformation.

Keywords: Feature extraction; Feature matching; Point match evaluation; Boosting-inspired; Rigid underlying transformation

1 Introduction

Nowadays, 3D shapes can be easily captured using laser scanners; their output is represented as sets of discrete points (see Figure 1). However, such devices have a limited field of view, and parts of the object may occlude others, so a number of scans have to be captured from different viewpoints. Where two scans cover common parts of the object, we say these two scans are *overlapping*, and the shapes in these two scans are called overlapping partial shapes. Once all scans have been captured, an important task is to analyze and fuse geometric (and possibly colour) information in these scans. Matching common points in the scans allows them to be used to register the scans. This is done by estimating the underlying transformation that best aligns the two scans. Here, we consider the underlying transformation to be rigid, involving a rotation and translation, but our method is also applicable to non-rigid registration involving more general classes of transformations, such as thin plate spline deformations [8].

Feature extraction and matching (FEM) are widely used for various tasks: object registration [41], modeling [2], retrieval, and recognition [18], as they are applicable to shapes with varying complexities of geometry, varying degrees of overlap, and varying magnitudes of transformation. The SHOT method, based on a signature of histograms of orientations [41], is one of the best methods for the extraction and matching of features from overlapping partial shapes [4, 17]. Even so, it usually unavoidably introduces mismatches amongst the established putative point matches (PPMs). In this approach, the random sample consensus (RANSAC) scheme [11] is used to reject mismatches and the unit quaternion method [7] is used to estimate the underlying transformation. However, the RANSAC scheme has a number of shortcomings, including a need to choose thresholds determining: whether a match is correct or a mismatch, when a good model has been found, and when to terminate the iterative process.

In this paper, we propose an alternative, novel, boosting-inspired method for evaluating the correctness of the established PPMs, with the aim of estimating as accurately as possible the underlying transformation. This estimate may then be used to initialize, for example, the SoftICP [23] variant of the iterative closest point (ICP) algorithm [7] for final refinement of the transformation. In particular, we

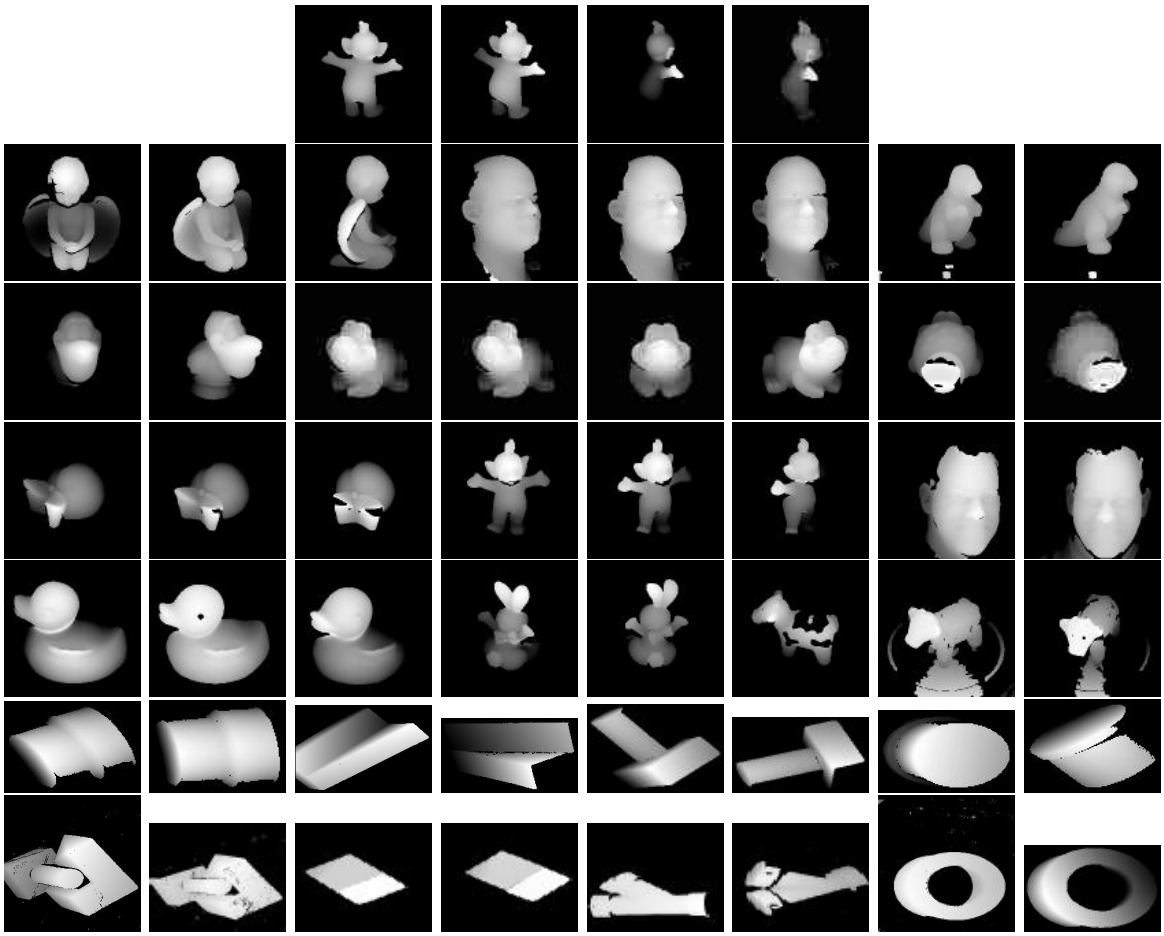

Figure 1: Real freeform shapes used in testing [32]. From left to right:
Row 1: tubby160, tubby140, tubby120, and tubby100.
Row 2: angel0, angel40, angel80, rick0, rick36, rick72, dinosaur144, and dinosaur180.
Row 3: bird60, bird100, frog0, frog20, frog40, frog80, lobster60, and lobster80.
Row 4: peach240, peach260, peach280, tubby0, tubby20, tubby40, pat108, and pat144.
Row 5: duck0, duck20, duck40, bunny0, bunny40, cow37, cow42, and cow45.
Row 6: adapter2, adapter3, block3, block5, column2, column5, cap1, and cap5.
Row 7: occl5, occl6, grnblk1, grnblk2, wye2, wye3, taperoll1, and taperoll2.

want to investigate two problems: (i) to what extent the FEM can be used to register overlapping 3D partial shapes and how accurate the estimated underlying transformations from the matched point pairs can be, and (ii) Whether our approach provides an initial estimate which is closer to the globally optimal solution than the one provided by the original method—if it does, it is more likely that the SoftICP algorithm will converge correctly to the global optimum, rather than a local optimum.

Our novel method is inspired by the widely used *adaptive boosting* learning and classification method

(AdaBoost) [15] from machine learning which combines several weak learners. The AdaBoost method has various advantages over other learning methods such as decision trees. Firstly, as long as each weak learner is better than random guessing, the boosted learner will be a stronger learner with improved performance. It is easy to find such weak learners. Secondly, the learning process concentrates on the incorrectly classified instances and increases their weights. This avoids overfitting, while ensuring that the decision when to terminate the learning process is not critical. The boosting parameter plays a crucial role in determining the extent to which the weights of the incorrectly classified instances will be increased.

Clearly, data classification as performed by Adaboost, and point match evaluation, are two different problems: the former requires training data, but such training data is not available to the latter. The main idea of our boosting-inspired method is as follows. Evaluation of the established point matches is a data fitting problem. In this case, all the established PPMs belong to the same class but are treated as having different reliabilities, represented as a real number in the unit interval $[0, 1]$; the larger the number, the more likely we believe it to be correct. The proposed method focuses on estimating and updating these reliability values iteratively. After such reliabilities or weights have been initialized or estimated, the underlying transformation is determined in a weighted least-squares sense and the weighted average e_μ and standard deviation e_σ of the errors of all the PPMs can be calculated accordingly in each iteration. Then we construct an objective function for the estimation of the boosting parameters. This objective function minimizes the weighted average of e_μ over different iterations, with the weights set to the boosting parameters. To avoid the degenerate case where all the boosting parameters are zero, they are regularized by the Tsallis entropy in the framework of entropy maximization [16]. The boosting parameters have a closed form solution. To update the weights of the PPMs, we minimize the weighted average e_c of the modified squared registration errors of all the PPMs with the weights regularized by the Shannon entropy H_s in the framework of entropy maximization again and the two terms of e_c and H_s are balanced by the boosting parameters. The weights also have a closed form solution. These weights are finally updated in each iteration by taking the maximum of the newly estimated values and the current values, so that the algorithm can learn from both accurate and inaccurate underlying transformations determined in different iterations. The whole process repeats until either the weighted average of the errors of all the PPMs is smaller than the average of the distances between the closest points in the original overlapping shapes, or some predetermined maximum number of iterations has been reached. The optimal underlying transformation is then estimated as a weighted average of the candidate transformations estimated from the last several iterations using weights given by the boosting parameters.

We use data captured by two scanners from objects with varying geometric complexities to assess our proposed regularization based adaptive boosting-inspired (RBAB) method, and compare it to four

state of the art methods. We show that, using two representative FEM methods, SHOT [41] and unique shape context (USC) [42], our method provides more accurate and robust estimates of the underlying transformation from the PPMs established.

This work is different from our previous work [22] in the following aspects: (i) while the former uses the Tsallis entropy to regularize the potential solutions to the boosting parameters, the latter uses the shifted entropy instead; (ii) while the former minimizes the weighted average of the modified squared registration errors of the PPMs for the estimation of their weights, the latter minimizes the weighted average of the registration errors themselves of the PPMs; and (iii) while the former integrates the underlying transformations estimated at later iterations as their weighted average with the weights defined by the estimated boosting parameters and large weights given to those providing smaller errors, the latter re-estimates the underlying transformation in the weighted least squares sense from the established PPMs with the weights estimated by the proposed point match evaluation process. The rest of this paper is structured as follows: Section 2 reviews related work, Section 3 describes our boosting-inspired method, and Section 4 presents experimental results using real data. Finally, Section 5 draws conclusions and indicates future work.

2 Related work

Due to the challenging nature of registration of overlapping 3D partial shapes, many algorithms have been proposed. In the very specialised situation in which the two scans have exactly the same number of points and each point in one scan corresponds to a single point in the other, the underlying transformation can be estimated using the global scatter matrix analysis [21] or Fourier analysis [26]. However, this requirement rarely holds, in which case such methods are not applicable. A straightforward and intuitive method is to extract and match local feature points from each scan as a way to determine correspondences, from which the underlying transformation can be estimated in a least-squares sense.

2.1 Point match evaluation

Real world registration is subject to shapes having relatively featureless simple geometry, imaging noise, holes, occlusion, appearance and disappearance of points, and cluttered backgrounds, which together lead to the extracted features typically being insufficiently informative to enable points to be matched without ambiguity. The point matches established usually include many mismatches with unpredictable errors [24]. These corrupt the data, and in turn lead to inaccurate estimation of the underlying transformation. To overcome this problem, some means must be found to evaluate the reliability of each match: either to classify them as correct or false, or to characterize the extent to which they are believed to be correct. Li and Hu published the first paper [20] dedicated to the evaluation of

the established point matches; other papers have since followed [1, 28, 29]. Existing methods for point match evaluation are based on one or more of the following: (i) structural consistency, (ii) transformation consistency, (iii) robust statistics, and (iv) heuristics.

Structural consistency based methods assume that the matched points have similar local features and structures, which is also the foundation for feature extraction and matching. After the point matches have been established, those that do not have similar features should be treated as false matches. Established point matches are grouped and mismatches are rejected in [18], determined as those with low similarity of spin images, and those inconsistent with the majority of matches. If the difference between the Euclidean distances between the corresponding points from different partial shapes is larger than a threshold, then these point matches would violate the distance preservation constraint and thus are all rejected in [2]. The rigidity constraint is employed in [38] to find a set of consistent point matches from which the initial transformation is estimated. Possible correspondences between planes are refined in [34] using a set of consistency tests based on size, rotation, and translation. Initially, in [1], the Euclidean distances between points are used to measure the consistency between different matches, then the replicator equation is used to estimate their reliabilities on a global scale. In [9], the consistencies of both the descriptors of and Euclidean distances between points are first considered to determine a payoff matrix, the global weights of all the PPMs are then estimated using the infection and immunization dynamics. While invariant features have already been used to represent and match points, it remains a challenge to find other complementary and expressive invariant features for eliminating false matches.

Transformation consistency based methods assume that after an accurate transformation has been estimated and applied, all correct matches should have small and similar residuals, and point matches with large residuals will be treated as false matches. The RANSAC method [11] is one of the most widely used transformation consistency based methods [5, 6, 36, 41]. It first samples point matches, and uses the samples to estimate a candidate underlying transformation. A threshold is then used to classify point matches as correct or false. Finally, all the correct point matches are used to estimate the underlying transformation. This is repeated with differing sets of samples. In [39], after a candidate underlying transformation has been estimated from a matched point pair, if the maximum distance between the two overlapping views after transformation is below a threshold and the number of point correspondences is large enough, this underlying transformation is accepted. Otherwise, the process repeats. A Hough transform is used in [25] to eliminate mismatches in Hough space. It is assumed in [20] that the PPMs are associated via two correspondence functions: one associates points in the first shape to those in the second and the other associates the points in the second shape to those in the first, these functions are estimated using the subspace projection support vector machine regression method. If two matched points fail to satisfy any of these two functions, they will be rejected as a

mismatch. In [27, 28, 29], the established PPMs are assumed to undergo a non-rigid transformation, which is represented as a vector field and estimated using an expectation-maximization framework. The fitting residuals are then used to estimate the reliabilities of different PPMs, allowing the inlier set to be determined by thresholding these reliabilities. If the PPMs are heavily corrupted by mismatches, it is challenging for transformation consistency based methods to accurately estimate the underlying transformation, and thus for the residuals to indicate their true reliabilities. In this case, the PPMs with smaller errors are not necessarily correct, as they may happen to lie at the intersections of the transformed but misaligned shapes.

The purpose in evaluating the point matches is to estimate the underlying transformation as accurately as possible. To do so, the correct point matches should make significant contributions to the estimate and the false matches should make little contribution. Various schemes can be employed for this purpose: iteratively re-weighted least squares (IRLS) (M-estimator) [46], least median of squares (LMedS), and RANSAC [11]. IRLS is a general approach which usually includes a data dependent threshold for binary classification of the point matches. The LMedS method has a discontinuous objective function whose optimization is usually time consuming. The RANSAC scheme has a number of shortcomings: low computational efficiency, difficulty in selection of a threshold for point match classification, and difficulty in definition of the quality of a model.

Heuristics can also be used to reject suspected false matches. Points may not have matches [19] if they lie at the boundary of the sensed surface, or in the non-overlapping area; they may also have multiple potential matches. A match to a point is deemed unreliable in [25, 5, 6] if the ratio between the dissimilarity of this (best) match and that of the second best match is above a threshold. Two-way matching (from the first shape to the second and from the second shape to the first) is used in [5, 6] to reject mismatches. Such heuristics are usually data dependent and do not always work. In fact, all of the methods above use heuristics in one way or another for the purpose of point match evaluation.

In contrast with most existing methods that classify the PPMs as either correct or false, we use a real number in the unit interval to represent their reliabilities. A transformation can be estimated at each iteration, in a weighted least squares sense based on such reliabilities. Instead of selecting one transformation as the final estimate, we combine them together to improve both accuracy and stability.

After using feature extraction and matching to estimate the transformation, it is typically refined using the ICP algorithm [7] or one of its variants [23, 35, 37]. Clearly, the closer the unrefined transformation is to the global minimum, the more likely it is that refinement will succeed when using such nonlinear algorithms. On the contrary, if the initial transformation estimate is not good enough, refinement can fail catastrophically.

2.2 Boosting

Since *adaptive boosting* (AdaBoost) was originally proposed in around 1995 [12], it has received intense attention from the machine learning community. It employs a set of weak learners to obtain good solutions to a data classification and learning problem, assuming that any weak learner performs better than random guessing. The main idea is that an instance is assigned a weight, informing the weak learner whether this instance should be focused on during learning. In the process of learning, we decrease the weights for correctly classified instances and increase the weights for mis-classified instances, and thus focus on the instances with mis-classifications, in an attempt to better handle them. The iterative learning process includes four main steps: fitting the classifier to the training data using the current weights, calculating the average error of the classifier, estimating the boosting parameter, and updating the weights of each instance. Each instance is finally classified using a function of the weighted average of the classifications from the weak classifiers, with greater weight given to those providing lower errors. Different definitions have been used for computing the error of each weak learner, the boosting parameter, the weight update scheme, and the final decision rule, leading to a number of variants, such as gradient boosting [13, 14], Real AdaBoost [15], LogitBoost [15], and Gentle AdaBoost [15].

The boosting method has been employed in a number of computer vision applications: classification of handwritten data [3], real time face detection [44], object tracking [33], classification of trees and vehicles in urban scenes [45], categorization of natural scenes [31], keypoint detection and landmarking on human faces [10], and person re-identification [30].

It has not yet been used to evaluate point matches between overlapping 3D partial shapes, perhaps for several reasons. Firstly, registration is a regression problem, rather than a classification problem. In some sense, all the point matches belong to the same class but differ in how good a fit they are to the underlying transformation. Secondly, labelled training data is not available–AdaBoost is a supervised learning method. Thirdly, AdaBoost is not straightforward to adapt to new tasks.

3 Our boosting-inspired method

AdaBoost is a powerful algorithm for learning and classification. Careful analysis reveals that it possesses two key properties: (i) the final solution is estimated as the weighted average of solutions provided by all the weak learners (an additive model), with the weights defined by the boosting parameters, (ii) it provides as a byproduct an estimate of the reliability of each instance, indicating how well it fits the additive model. In this section, we adapt these properties for our registration problem, expressing it in terms of data fitting: finding the underlying transformation is a non-linear problem, the data items are usually heavily corrupted by outliers and training data are unavailable.

We now explain our novel algorithm. Given two overlapping 3D partial shapes, any FEM method

of choice (e.g. the SHOT algorithm [41]) can be employed to establish a set of PPMs $(\mathbf{p}_i, \mathbf{p}'_i)$, $i = 1, \cdots, N \geq 3$ where $\mathbf{p}_i$ comes from the first shape and $\mathbf{p}'_i$ from the second. Such point matches are typically heavily contaminated by mismatches. Weights w_i in the interval $[0, 1]$ represent the extent to which we believe match $(\mathbf{p}_i, \mathbf{p}'_i)$ to be correct, a value of 1 meaning certainty. All weights are initialized to $w_i^{(1)} = 1$, to say we initially trust each match until we have evidence it may be wrong. Our novel boosting-inspired method iteratively estimates and updates these weights, and the underlying transformation is re-estimated in each successive iteration; interation number is denoted by k. Those immediate transformations at later iterations are finally fused together to give an optimal estimate.

Our method is based on three ideas: underlying transformation estimation, boosting parameter estimation, and weight re-estimation and update; these are described in the next section. We then summarise our algorithm, and finally compare our novel method and the discrete AdaBoost method.

3.1 The main computational steps

In each iteration, given the PPMs $(\mathbf{p}_i, \mathbf{p}'_i)$, we first normalize these weights $w_i^{(k)} \leftarrow w_i^{(k)} / \sum_{j=1}^{N} w_j^{(k)}$ in order to facilitate later the calculation of the weighted average and standard deviation of the errors of all the PPMs and update of these weights. We then estimate the underlying transformation $(\mathbf{R}^{(k)}, \mathbf{t}^{(k)})$ in the weighted least squares sense by minimizing the following objective function:

$$J(\mathbf{R}^{(k)}, \mathbf{t}^{(k)}) = \sum_{i=1}^{N} w_i^{(k)} ||\mathbf{p}'_i - \mathbf{R}^{(k)}\mathbf{p}_i - \mathbf{t}^{(k)}||^2. \tag{1}$$

We used the quaternion method [7] to do so. Then the weighted average $e_\mu^{(k)}$ and standard deviation $e_\sigma^{(k)}$ of the registration errors $e_i^{(k)}$ of all the PPMs $(\mathbf{p}_i, \mathbf{p}'_i)$ are found using:

$$e_\mu^{(k)} = \sum_{i=1}^{N} w_i^{(k)} e_i^{(k)}, \qquad (e_\sigma^{(k)})^2 = \sum_{i=1}^{N} w_i^{(k)} (e_i^{(k)} - e_\mu^{(k)})^2, \tag{2}$$

where $e_i^{(k)} = ||\mathbf{p}'_i - \mathbf{R}^{(k)}\mathbf{p}_i - \mathbf{t}^{(k)}||^\alpha$ and the parameter α is a positive real number and is set as $\alpha = 1$ in this paper.

The boosting parameter β determines the extent to which point matches with errors should be penalized in the next iteration and how the transformations $(\mathbf{R}^{(k)}, \mathbf{t}^{(k)})$ estimated in different iterations (weak estimators, corresponding to weak classifiers in AdaBoost) will be finally combined. They must thus be carefully and accurately estimated. To this end, we construct an objective function using the framework of entropy maximization [16, 8]. This function minimizes the weighted average e_b of errors $e_\mu^{(k)}$ arising from weak estimators from different iterations, where here the weights are the boosting parameters $\beta^{(k)}$. To avoid the degenerate case in which all the boosting parameters are zero, they are regularized by the

Tsallis entropy H_t:

$$J(\beta) = e_b - H_t = \sum_{k=1}^{K} \beta^{(k)} e_\mu^{(k)} - (\sum_{k=1}^{K} (\beta^{(k)})^q - 1)/(1-q), \tag{3}$$

where K is the number of iterations after which the boosting process terminates and the entropic index q is a real number ($q \neq 1$). For convenience we gather $\beta = \{\beta^{(1)}, \cdots, \beta^{(K)}\}$. The two terms of e_b and H_t make equal contributions to the objective function. To minimize this function, setting its derivative with respect to $\beta^{(k)}$ to zero leads to:

$$\beta^{(k)} = ((1-q)e_\mu^{(k)}/q)^{q-1}.$$

Since the boosting process should penalize weak estimators with a large error $e_\mu^{(k)}$, as in AdaBoost, $q \in (0, 1)$. The proposed algorithm is not sensitive to its setting, and in this paper, we set $q = 0.25$. Since $q > 0$, $1 - q > 0$ and $e_\mu^{(k)} > 0$, $\beta^{(k)}$ will be positive: $\beta^{(k)} > 0$.

In order to update the weights $w_i^{(k+1)}$ of the PPMs $(\mathbf{p}_i, \mathbf{p}_i')$, we construct another objective function again using the framework of the entropy maximization. This objective function minimizes the weighted average e_c of the modified squared registration (MSR) errors (defined below) of the point matches. This time the weights are regularized by the traditional Shannon entropy H_s:

$$J(\mathbf{W}^{(k+1)}) = e_c - \frac{1}{\beta^{(k)}} H_s = \sum_{i=1}^{N} w_i^{(k+1)} (e_i^{(k)})^2 \exp((e_i^{(k)} - e_\mu^{(k)})^2/(2(e_\sigma^{(k)})^2)) + \frac{1}{\beta^{(k)}} \sum_{i=1}^{N} w_i^{(k+1)} (\log w_i^{(k+1)} - 1) \tag{4}$$

where $\mathbf{W}^{(k+1)} = \{w_1^{(k+1)}, \cdots, w_N^{(k+1)}\}$ and the two terms of e_c and H_s are balanced by the boosting parameter $\beta^{(k)}$. This MSR error $(e_i^{(k)})^2 \exp((e_i^{(k)} - e_\mu^{(k)})^2/(2(e_\sigma^{(k)})^2))$ considers not just the errors $e_i^{(k)}$ themselves, but also how far away they are from their weighted average $e_\mu^{(k)}$. The rationale for doing so is that the former biases those PPMs with small errors without comparing them to others, while the latter takes into account the errors of all the PPMs and thus characterizes how the error of a particular point match differs from the majority of the others. Combination of the two terms can better characterize the reliabilities of the PPMs, focusing on penalizing those PPMs with errors much larger than $e_\mu^{(k)}$. To minimize this function, again setting its derivative with respect to $w_i^{(k+1)}$ to zero leads to:

$$w_i^{(k+1)} = \exp(-\beta^{(k)} (e_i^{(k)})^2 \exp((e_i^{(k)} - e_\mu^{(k)})^2/(2(e_\sigma^{(k)})^2))).$$

Since $\beta^{(k)} > 0$ and $(e_i^{(k)})^2 \exp((e_i^{(k)} - e_\mu^{(k)})^2/(2(e_\sigma^{(k)})^2)) \geq 0$, then $w_i^{(k+1)}$ will lie in the unit interval $[0, 1]$. In contrast with the main idea of the traditional AdaBoost algorithm, which boosts the weights of the misclassified instances and decreases the weights of the correctly classified instances, our algorithm penalizes all matches $(\mathbf{p}_i, \mathbf{p}_i')$ in a uniform manner according to their errors $e_i^{(k)}$, as well as how far away they are from their weighted average $e_\mu^{(k)}$. The larger and the farther away the error $e_i^{(k)}$, the more heavily the point match will be penalized.

In order to take into account both the accurate and inaccurate underlying transformations (weak estimators) estimated in the different iterations so far, the weight of each point match $(\mathbf{p}_i, \mathbf{p}'_i)$ is updated in each iteration using a maximum fusion rule:

$$w_i^{(k+1)} = \max(w_i^{(k+1)}, w_i^{(k)}). \tag{5}$$

This ensures that w_i remains in the interval $[0, 1]$, since both $w_i^{(k+1)}$ and $w_i^{(k)}$ lie in the unit interval $[0, 1]$ from their estimations above. Our experimental results below show that this maximum fusion scheme reflects the true reliabilities of the point matches better than using the product fusion scheme in the original discrete AdaBoost [15].

When $e_\mu^{(k)}$ is smaller than the average distance between the closest points in the original shapes, or a predetermined maximum number k_{max} of iterations has been reached, the whole process terminates. The underlying transformation is finally estimated as the weighted average of the transformations $(\mathbf{R}^{(k)}, \mathbf{t}^{(k)})$ estimated in different iterations using the boosting parameters $\beta^{(k)}$ as weights:

$$\mathbf{A} = \sum_{k=k_1}^{K} \beta^{(k)} \mathbf{R}^{(k)} / \sum_{k=k_1}^{K} \beta^{(k)}, \qquad \mathbf{t} = \sum_{k=k_1}^{K} \beta^{(k)} \mathbf{t}^{(k)} / \sum_{k=k_1}^{K} \beta^{(k)}. \tag{6}$$

where $k_1 \in [1, K]$ and determines what proportion of the transformations are combined to give the final estimate. Since the weighted average $\mathbf{A}$ of the orthogonal matrices $\mathbf{R}^{(k)}$ is not necessarily orthogonal, we use singular value decomposition (SVD) to estimate the rotation matrix $\mathbf{R}$ as: $\mathbf{R} = \mathbf{U}\mathbf{V}^T$ where the matrices $\mathbf{U}$ and $\mathbf{V}$ are those from the SVD of matrix $\mathbf{A}$: $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$. Since the estimated underlying transformations $(\mathbf{R}^{(k)}, \mathbf{t}^{(k)})$ and the boosting parameters $\beta^{(k)}$ from the first few iterations are not very accurate, due to inaccurate initialization and short time spent in learning, they are discarded during estimation of the final underlying transformation. In this paper, we let $k_{max} = 100$ and $k_1 = 0.25K$.

3.2 The boosting-inspired algorithm

Pulling all the previous ingredients together, our point match evaluation algorithm can be summarized as follows:

1: Read in a set of point matches $(\mathbf{p}_i, \mathbf{p}'_i)$ between two overlapping 3D partial shapes
2: Initialize all weights $w_i^{(1)} = 1$, the average s of the distances between the closest points in the original partial shapes, constants k_1 and $k_{\max}$, and iteration index $k = 0$
3: **do**
4: $\quad k \leftarrow k + 1$
5: $\quad$ Normalize the weights: $w_i^{(k)} \leftarrow w_i^{(k)} / \sum_{j=1}^{N} w_j^{(k)}$
6: $\quad$ Estimate the solution $(\mathbf{R}^{(k)}, \mathbf{t}^{(k)})$ from Equation 1
7: $\quad$ Compute the weighted average $e_\mu^{(k)}$ and variance $(e_\sigma^{(k)})^2$ of the errors of all the point matches using Equation 2

8: Estimate the boosting parameter $\beta^{(k)}$ from Equation 3

9: Re-estimate and update the weight $w_i^{(k)}$ of each point match from Equations 4 and 5 respectively

10: **while** $e_\mu^{(k)} > s$ and $k < k_{\max}$

11: Boost the underlying transformation $(\mathbf{R}, \mathbf{t})$ using Equation 6

As our proposed method applies various entropies to regularize the estimation of the parameters of interest, and the underlying transformation is estimated as a weighted average of those from different iterations (weak estimators), we call it regularization based adaptive boosting-inspired method (RBAB). As each step has a computational complexity of $O(N)$ in the number N of established point matches, the algorithm has linear computational complexity: $O(k_{\max}N)$.

3.3 Comparison of RBAB with AdaBoost

In this section, we consider the relationship between our proposed RBAB method and AdaBoost. They share the following common properties: (i) they both estimate the boosting parameters of the weak classifiers and function estimators in such a way to give large values to those producing small errors; (ii) they both boost the weak classifiers and function estimators with small errors for the weight estimation of the given data items, (iii) they both learn from the exiting data and classifiers/function estimators through fusion of the weights estimated in each iteration; and (iv) they both integrate the outputs of the algorithms from different iterations with the weights defined by the estimated boosting parameters. In this case, we believe that the point patch evaluation problem can be cast into the framework of the traditional AdaBoost algorithm for the development of novel techniques, as demonstrated by the proposed RBAB algorithm in this paper, and the boosting-inspired framework helps deepen our understanding of and is useful for solving the challenging point match evaluation problem, as demonstrated by the experimental results given below in Section 4. We also identify their difference, to further explain how the former has been adapted for accurate estimation of the underlying transformation from point matches corrupted by mismatches.

At iteration k, the classifier $f_k(x_i)$ gives each data item x_i a label which is either 1 or -1. The discrete AdaBoost algorithm [15] estimates the boosting parameter β as $\beta^{(k)} = \log((1 - e_\mu^{(k)})/e_\mu^{(k)})$ and updates the weight w_i of each data item x_i using $w_i \leftarrow w_i \exp(\beta^{(k)} 1_{y_i \neq f_k(x_i)})$. It is assumed that $e_\mu^{(k)} < 0.5$. Otherwise, the iteration is aborted. In this case, $\beta^{(k)} > 0$. If x_i has been incorrectly labeled, then $\exp(\beta^{(k)} 1_{y_i \neq f_k(x_i)})$ will be larger than 1, leading w_i to be increased. Otherwise, $\exp(\beta^{(k)} 1_{y_i \neq f_k(x_i)})$ will be exactly 1, leading w_i to be unchanged. After normalization, the weights of incorrectly labeled items will be relatively increased, but those of the correctly labeled ones will be relatively decreased.

In sharp contrast, point match evaluation is essentially a data fitting problem: the correct matches fit the underlying transformation. In this case, no training data is available and there are no data labels of 1 or -1. We cannot make any judgment whether the weak classifier f at iteration k has mapped

the data item x_i exactly to the desired label y_i. Instead, we penalize all the point matches in the same manner. Two different aspects are taken into account to compute the penalty: the error $e_i^{(k)}$ of the match $(\mathbf{p}_i, \mathbf{p}_i')$, and how far away it is from the weighted average error $e_\mu^{(k)}$. The former gives greater weight to those PPMs with small errors of fit. However, even if the shapes are mis-registered, we can still get small errors near the intersection of the two misregistered partial shapes–small errors do not necessarily mean registration is good. Thus, the latter considers the overall picture of how the errors of different point matches are distributed and whether the current error is similar to other errors. A combination of these two terms provides a better characterization of the true reliabilities of the point matches.

As the boosting process progresses, it becomes reasonable to assume that the weighted averages of the points $\bar{\mathbf{p}}' = \sum_{i=1}^N w_i^{(k)} \mathbf{p}_i'$ and $\bar{\mathbf{p}}' = \sum_{i=1}^N w_i^{(k)} \mathbf{p}_i'$ are relatively stable from one iteration to another. If so, the finally boosted transformation $(\mathbf{R}, \mathbf{t})$ is optimal, as follows. The boosting parameters $\beta^{(k)}$ minimize the weighted average e_b of the average errors $e_\mu^{(k)}$ in different iterations k,

$$e_b = \sum_{k=1}^K \beta^{(k)} e_\mu^{(k)} = \sum_{k=1}^K \beta^{(k)} \sum_{i=1}^N w_i^{(k)} ||\mathbf{p}_i' - \mathbf{R}^{(k)} \mathbf{p}_i - \mathbf{t}^{(k)}||.$$

From Jensen's inequality,

$$\begin{aligned} e_b &\geq \sum_{k=1}^K \beta^{(k)} || \sum_{i=1}^N w_i^{(k)} \mathbf{p}_i' - \mathbf{R}^{(k)} \sum_{i=1}^N w_i^{(k)} \mathbf{p}_i - \sum_{i=1}^N w_i^{(k)} \mathbf{t}^{(k)} || = \sum_{k=1}^K \beta^{(k)} ||\bar{\mathbf{p}}' - \mathbf{R}^{(k)} \bar{\mathbf{p}} - \mathbf{t}^{(k)}|| \\ &\geq \sum_{k=1}^K \beta^{(k)} \frac{\bar{\mathbf{p}}'}{||\bar{\mathbf{p}}'||} \cdot (\bar{\mathbf{p}}' - \mathbf{R}^{(k)} \bar{\mathbf{p}} - \mathbf{t}^{(k)}) = \frac{\bar{\mathbf{p}}'}{||\bar{\mathbf{p}}'||} \cdot (\sum_k^K \beta^{(k)} \bar{\mathbf{p}}' - \sum_k^K \beta^{(k)} \mathbf{R}^{(k)} \bar{\mathbf{p}} - \sum_k^K \beta^{(k)} \mathbf{t}^{(k)}) \\ &= \frac{\bar{\mathbf{p}}'}{||\bar{\mathbf{p}}'||} \cdot (\bar{\mathbf{p}}' - \mathbf{R}\bar{\mathbf{p}} - \mathbf{t}) \geq 0. \end{aligned}$$

While e_b is minimized by the boosting parameters $\beta^{(k)}$, they also minimize $\frac{\bar{\mathbf{p}}'}{||\bar{\mathbf{p}}'||} \cdot (\bar{\mathbf{p}}' - \mathbf{R}\bar{\mathbf{p}} - \mathbf{t}) \geq 0$ thus causing $\mathbf{R} = \sum_{k=1}^K \beta^{(k)} \mathbf{R}^{(k)}$ and $\mathbf{t} = \sum_{k=1}^K \beta^{(k)} \mathbf{t}^{(k)}$ to be optimized as well.

The main differences between AdaBoost and our RBAB method are as follows: (i) while the former boosts the misclassified instances for dealing with the more challenging data for their correct classification, the latter boosts to estimate the reliabilities/weights of all the point matches for their more accurate fit of the underlying transformation in the weighted least squares sense; (ii) while the former is a supervised method and requires training data, such data is not available to the latter; (iii) while the former derives the boosting parameter by minimizing the expected exponential loss on each instance, the latter derives it via entropy maximization; (iv) while the former directly applies the classification error of an instance to estimation of its weight, the latter considers not just its error, but also how far away the error is from the majority; (v) while the former multiplies the new and old weights to give new weights, the latter uses their maximum, and finally (vi) while the former considers the solutions from all weak classifiers, the latter discards the solutions from the initial, unreliable weak estimators.

4 Experimental results

Our experimental study firstly considers alternative approaches to various components of our algorithm: definition of registration error, weight accumulation scheme, and fusion of rigid transformations estimated from different iterations. Secondly, we perform a comparison between the final RBAB algorithm and other methods for evaluating point matches.

We use real data to assess the effectiveness of the proposed RBAB algorithm for evaluating N point matches $(\mathbf{p}_i, \mathbf{p}'_i)$ established using, unless otherwise stated, the SHOT method [41] and the unique shape context (USC) method [42]. Comparison is made to the following methods for evaluating matches: game theoretic matching (GTM) [1], sparse vector field consensus (SparseVFC) [29], RANSAC [11] and iteratively re-weighted least squares (IRLS) (M-estimator) [46]. These four methods are state-of-the-art representative of the three main categories discussed in Section 2.1: GTM is a structural consistency method, SparseVFC and RANSAC are transformation consistency methods, and IRLS is a robust statistical method. This study determines which method can produce the most accurate and stable estimate of the underlying transformation from point matches established by typical FEM methods.

In each pair of overlapping 3D partial shapes, we call the first the *data shape*, and the second the *reference shape*. All data in Figure 1 were downloaded from [32]; they were captured using either a Minolta Vivid 700 range camera with a fixed resolution of 200×200 or a Technical Arts 100X scanner with a varying resolution from 89×112 to 240×240. As the estimated underlying transformation from the evaluated PPMs is later used to initialize the ICP variant, SoftICP [23], for refinement, and the latter usually produces accurate results, the differences in the underlying transformation before and after ICP refinement can be used to assess the performance of the different point match evaluation methods under test [43]. Such an approach is especially useful when the ground truth is partially or completely unknown, as is the case for the data used for the experiments in this paper. The metrics used to assess the output are the relative errors $e_{\mathbf{h}}$, e_θ, and $e_{\mathbf{t}}$ in percentage of the estimated rotation axis $\hat{\mathbf{h}}$, rotation angle $\hat{\theta}$, and translation vector $\hat{\mathbf{t}}$ of the underlying transformation, without ICP refinement, from the evaluated point matches. To understand the extent to which the established point matches are corrupted by mismatches, we also estimated the percentage of correct matches as $N_c/N \times 100\%$ where N_c is the number of point matches whose errors $||\mathbf{p}'_i - \mathbf{R}\mathbf{p}_i - \mathbf{t}||$ are smaller than four times the average of the distances between the closest points in the data shape according to the transformation estimated by the proposed RBAB algorithm and refined by the SoftICP algorithm. We also consider computational times used for point match evaluation, underlying transformation estimation and performance measurement, for each approach.

The experimental results are presented in Figures 2–10, and Tables 1–5. In Figures 2, 4, 6, 8, and 10, yellow represents the data shapes after applying the underlying transformations estimated from the evaluated point matches without ICP refinement, and green represents the reference shapes. All

experiments were carried out on a PC with an Intel Xeon E5620 processor using Microsoft Visual Studio 2013 (without code optimization).

4.1 Definition of registration error

Registration error of a point match plays a crucial role in its reliability estimation and thus must be carefully defined. In Section 3, we defined it as: $e_i = ||\mathbf{p}' - \mathbf{R}\mathbf{p}_i - \mathbf{t}||^\alpha$ where $\alpha = 1$. In this section, we experimentally investigate whether this definition is optimal and thus varies its exponent α from 0.5, 1 to 2. To this end, the overlapping partial shapes angel0–40ght, angel40–80, rick0–36, rick36–72, tubby140–120, tubby120–100, peach240–260, and peach240–280 in Figure 1 were selected for the experiments; the results are presented in Figures 2 and 3 and Table 1.

They show that $\alpha = 1$ produces the most accurate and stable estimate of the underlying transformations between different overlapping partial shapes with varied complexities of geometry. Setting $\alpha = 0.5$ decreases the values of the errors, making them less expressive of the true reliabilities of the PPMs, as demonstrated by the fact that the estimated underlying transformation displaces the transformed tubby140 data shape against the reference tubby120 shape. In contrast, $\alpha = 2$ is over aggressive in taking into account the errors of the PPMs, leading some correct matches to be more heavily penalized than expected. This observation is supported by the fact that the two leaves at the end of the peach in the transformed peach240 and reference peach280 shapes are clearly displaced. The inaccurate underlying transformation is usually manifested as a superimposition of one shape onto another with much less interpenetration [40]. Overall, $\alpha = 1$ achieves the best compromise between accuracy and stability and thus justifies the definition of error given in the main body of the paper.

Table 1: The average μ and standard deviation σ of the relative errors $e_{\mathbf{h}}(\%)$, $e_\theta(\%)$, and $e_{\mathbf{t}}(\%)$ of the estimated rotation axis $\hat{\mathbf{h}}$, rotation angle $\hat{\theta}$, and translation vector $\hat{\mathbf{t}}$ of the underlying transformation, and evaluation time T in seconds, using our RBAB algorithm, applied to different overlapping partial shapes, for different values of the parameter α.

Para.	α	$e_{\mathbf{h}}$ (%)	e_θ (%)	$e_{\mathbf{t}}$ (%)	T (sec)
μ	0.5	2.96	-5.64	7.63	29.12
	1	2.65	-4.06	5.33	28.50
	2	4.48	-6.54	6.78	29.00
σ	0.5	2.59	5.52	4.23	19.57
	1	1.57	2.51	2.46	19.01
	2	6.00	4.70	4.99	18.91

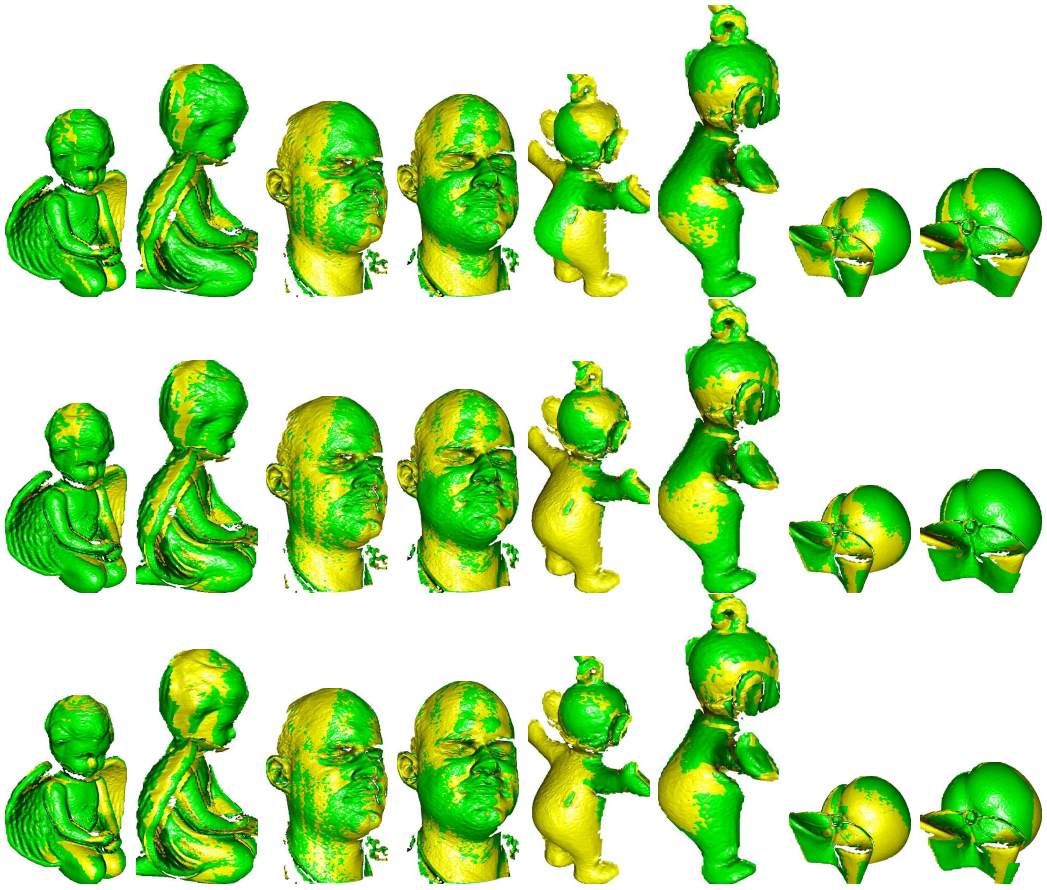

Figure 2: Final registration results for our RBAB method, with the parameter α taking different values. From left to right: angel0–40, angel40–80, rick0–36, rick36–72, tubby140–120, tubby120–100, peach240–260, and peach240–280. Top row: $\alpha = 0.5$. Middle: $\alpha = 1$. Bottom: $\alpha = 2$.

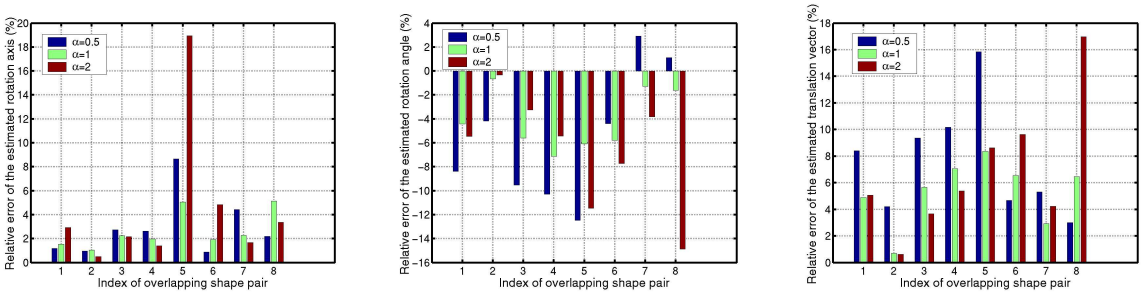

Figure 3: Relative errors in the estimated rotation axis (Left), rotation angle (Middle), and translational vector (Right) of the proposed RBAB algorithm with the parameter α taking different values. The overlapping shape pair index from 1 to 8 refers to: angel0–40, angel40–80, rick0–36, rick36–72, tubby140–120, tubby120–100, peach240–260, and peach240–280 respectively.

4.2 Weight update

It is important to learn from the weights estimated in different iterations, so that more accurate weights can finally be obtained. In this section, we consider alternative approaches to fusing the newly estimated

weight $w_i^{(k+1)}$ of each matched point pair $(\mathbf{p}_i, \mathbf{p}'_i)$ with the current weight $w_i^{(k)}$. We compare two schemes: our proposed maximum (MAX) rule $w_i^{(k+1)} \leftarrow \max(w_i^{(k)}, w_i^{(k+1)})$ and the product (PROD) rule of the discrete AdaBoost method [15]: $w_i^{(k+1)} \leftarrow w_i^{(k)} w_i^{(k+1)}$. To do so, the overlapping partial angel0–40, angel40–80, rick0–36, rick36–72, tubby140–120, tubby120–100, peach240–260, and peach240–280 shapes used in the last section were selected again for the experiments; the results are presented in Figures 4 and 5 and Table 2.

They show that the PROD scheme is not as stable as the MAX scheme for the update of the weights of point matches. The instability is manifested by the estimated underlying transformations failing to bring the transformed angle0, angel40 and peach240 data shapes into proper alignment in 3D space with the reference angel40, angel80 and peach280 shapes respectively as the heads and wings of the angel, and the two leaves of the peach are clearly displaced. These results show that the reliabilities of the PPMs can be more accurately characterized by the maximum scheme than the product. This is because the former can benefit from the accurate estimation of the reliability from any iteration, but the latter is affected by both the accurate and the inaccurate estimates from all iterations. This validates our choice of the maximum scheme for weight updates.

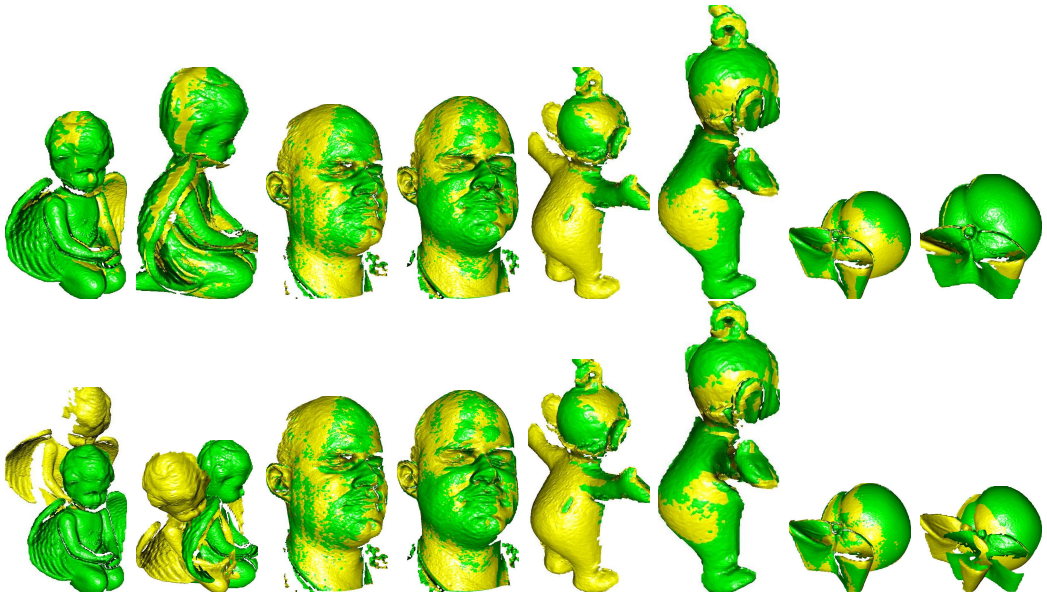

Figure 4: Final registration results for our RBAB method, with different methods used for combining weights from different iterations. From left to right: angel0–40, angel40–80, rick0–36, rick36–72, tubby140–120, tubby120–100, peach240–260, and peach240–280 respectively. Top row: MAX; Bottom: PROD.

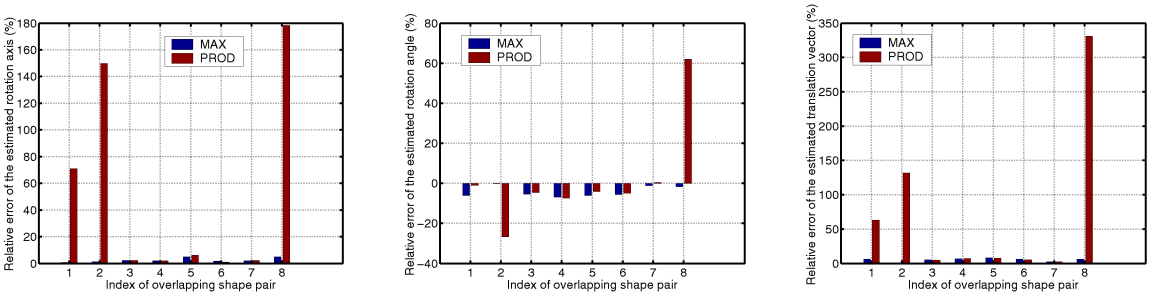

Figure 5: Relative errors of the estimated rotation axis (Left), rotation angle (Middle), and translation vector (Right) of our RBAB method, with different methods used for combining weights from different iterations. The overlapping shape pair index from 1 to 8 refers to angel0–40, angel40–80, rick0–36, rick36–72, tubby140–120, tubby120–100, peach240–260, and peach240–280 respectively.

Table 2: The average μ and standard deviation σ of relative errors $e_{\mathbf{h}}(\%)$, $e_\theta(\%)$, and $e_{\mathbf{t}}(\%)$ of the estimated rotation axis $\hat{\mathbf{h}}$, rotation angle $\hat{\theta}$, and translation vector $\hat{\mathbf{t}}$ of the underlying transformation, and evaluation time T in seconds, using our RBAB algorithm, applied to different overlapping partial shapes, with different schemes used to fuse the weights from different iterations.

Para.	Scheme	$e_{\mathbf{h}}$ (%)	e_θ (%)	$e_{\mathbf{t}}$ (%)	T (sec)
μ	MAX	2.65	-4.06	5.33	28.50
	PROD	51.70	1.75	69.29	45.12
σ	MAX	1.57	2.51	2.46	19.02
	PROD	68.95	24.13	107.69	33.65

4.3 Fusion of rigid transformations from different iterations

Finally, we consider how to combine the estimated underlying transformations from different iterations to give the final transformation estimate. To this end, three possible values of k_1 were investigated: $k_1 = 1$, $k_1 = 25\%K$ and $k_1 = K$. $k_1 = 1$ means that *all* the estimated transformations are used during combination, so we refer to this as *complete combination*; $k_1 = 25\%K$ means that the initial 25% underlying transformations are discarded and the latest 75% only are used for combination, which we refer to as *partial combination*; and $k_1 = K$ means that the underlying transformation estimated from the last iteration is selected without any combination and is thus called the last solution in the following. Such investigation will be useful to reveal how the transformations from different iterations can be combined for more accurate estimate and whether the last estimate is the best. While the last two subsections used the same data for the experiments, they have a shortcoming of repeating the same results of the proposed technique for a close comparison with those produced by its variants in

different sections. To avoid such repetition, we thus use different shapes in the following subsections for the experiments instead since: (i) They can show that the proposed technique is able to evaluate the PPMs established between different shapes with varied geometries and complexities; and (ii) They are more likely to reveal different behaviors of the proposed technique. The partially overlapping duck0–40, frog40–80, dinosaur144–180, tubby120–160, bird100–60, cow42–45, peach280–240, and lobster60–80 shapes in Figure 1 were selected for the experiments; the results are presented in Figures 6 and 7 and Table 3.

The results show that the complete combination usually produces the worst estimates of the underlying transformation: see in particular that the transformed tubby120 data shape and the reference tubby160 shape are clearly displaced. Even though the partial combination and the last solution produce visually similar results in Figure 6, both Figure 7 and Table 3 show that the partial combination is more accurate than using the last solution, decreasing the relative error of the estimated rotation axis and translation vector by as much as 23.09% and 21.89% respectively. This is because the underlying transformations and the boosting parameters estimated in the initial iterations are not accurate due to inaccurate initialization of the weights and the short time spent in learning. While the proposed RBAB algorithm minimizes the weighted average e_b of the errors $e_\mu^{(k)}$ of the PPMs in different iterations for the estimation of the boosting parameters, it does not necessarily minimize the weighted average $e_\mu^{(k)}$ of the PPMs in each iteration and produce the most accurate underlying transformation estimate in the last iteration. These results validate our decision to discard the initial transformations when computing the final transformation.

Table 3: The average μ and standard deviation σ of relative errors $e_{\mathbf{h}}(\%)$, $e_\theta(\%)$, and $e_{\mathbf{t}}(\%)$ of the estimated rotation axis $\hat{\mathbf{h}}$, rotation angle $\hat{\theta}$, and translation vector $\hat{\mathbf{t}}$ of the underlying transformation, and evaluation time T in seconds, using our RBAB algorithm, applied to different overlapping partial shapes, with different schemes for the fusion of the results from different weak estimators.

Para.	Scheme	$e_{\mathbf{h}}$ (%)	e_θ (%)	$e_{\mathbf{ti}}$ (%)	T (sec)
μ	Complete	18.40	-7.44	27.93	28.87
	Partial	3.93	-2.57	4.78	26.12
	Last Solution	5.11	-0.93	6.12	27.12
σ	Complete	36.74	8.47	56.07	18.41
	Partial	2.10	4.58	2.92	20.19
	Last solution	2.47	4.80	3.42	20.95

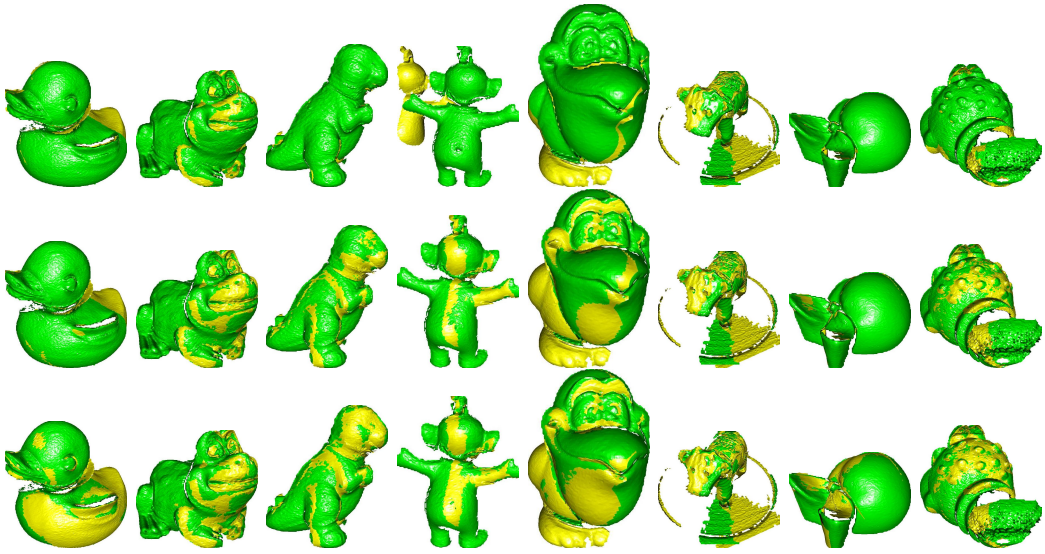

Figure 6: Final registration results for our RBAB method using different schemes for the fusion of the underlying transformations estimated by different weak estimators. From left to right: duck0–40, frog40–80, dinosaur144–180, tubby120–160, bird100–60, cow42–45, peach280–240, and lobster60–80. Top row: Complete; Middle: Partial; Bottom: Last solution.

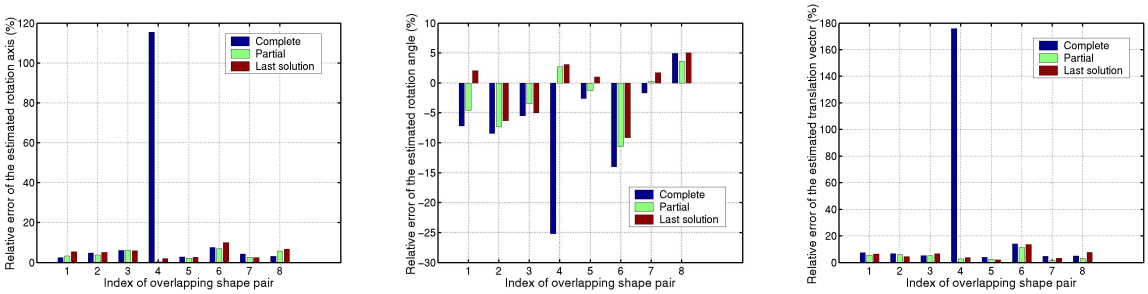

Figure 7: Relative errors of the estimated rotation axis (Left), rotation angle (Middle), and translational vector (Right) of our RBAB method using different schemes for the fusion of the underlying transformations estimated by different weak estimators. The overlapping shape pair index from 1 to 8 refers to duck0–40, frog40–80, dinosaur144–180, tubby120–160, bird100–60, cow42–45, peach280–240, and lobster60–80 respectively.

4.4 Algorithm comparison, Minolta Vivid 710 data

In this section, we present results of a comparative study of different techniques for evaluating point matches: GTM [1], SparseVFC [29], RANSAC, iteratively re-weighted least squares (IRLS) (M-estimator), and our RBAB method. To do so, the overlapping partial shapes tubby0–20, tubby20–40, frog0–40, frog40–80, duck0–20, pat108–144, bunny0–40 and cow37–45 in Figure 1 were selected. The experimental results are presented in Figures 8 and 9 and Table 4. The percentages of correct matches among

the established matches between these shapes are 63%, 61%, 11%, 14%, 27%, 31%, 17%, and 5% respectively.

They show that the IRLS method failed to estimate an accurate underlying transformation from the point matches established between the partial overlapping tubby20–40 shapes, the GTM and SparseVFC methods failed to reliably register the frog0–40 and frog40–80 shapes, the GTM method failed to register the pat108-144 shapes properly, and the SparseVFC and RANSAC methods also failed to accurately evaluate the point matches between the cow37–45 shapes. The failure of point match evaluation is demonstrated by the fact that the transformed data tubby20, frog0, frog40, pat108, and cow37 shapes intersect the reference tubby40, frog40, frog80, pat144, and cow45 shapes respectively in 3D space. The RANSAC method also inaccurately estimated the underlying transformations between the frog0–40 shapes, the SparseVFC method inaccurately down-weighted point matches between the duck0–20 and pat108–144 shapes, and the GTM, SparseVFC, and RANSAC methods also all estimated the underlying transformation inaccurately between the bunny0-40 shapes. The inaccurate estimation of the underlying transformation is manifested as the fact that the transformed frog0, duck0, pat108, and buny0 data shapes are superimposed onto the reference frog40, duck20, pat144, and bunny40 shapes respectively with low inter-penetration. In sharp contrast, the proposed RBAB method estimated the underlying transformations accurately and robustly for all eight pairs of overlapping partial shapes, with relative errors as small as 5% in the estimated rotation axis, rotation angle, and translation vector, relative to those refined by SoftICP. It is difficult for the GTM method to define a payoff matrix which represents the consistency of different PPMs. The SparseVFC method uses 3N degrees of freedom (DOF) to represent the underlying transformation, which is much larger than the 6 DOFs required to represent a rigid transformation. It imposes few constraints for the estimation of the underlying transformation and thus is likely to overfit. The RANSAC method has difficulties in choice of thresholds for the classification of the PPMs into correct matches and mismatches, and for determining how good an underlying transformation is. The IRLS algorithm has difficulty in setting up a threshold for the classification of the PPMs into two categories with different weights. In contrast, the Tsallis and Shannon entropies are powerful regularization tools in the estimation of the boosting parameter and in weighting the PPMs from different iterations, leading to more representative and stable estimation of the underlying transformations.

All five methods are similar in terms of computational time, especially when their initialization of the underlying transformation is good enough for the SoftICP algorithm to refine. Poor initialisation causes the SoftICP algorithm to take longer to converge.

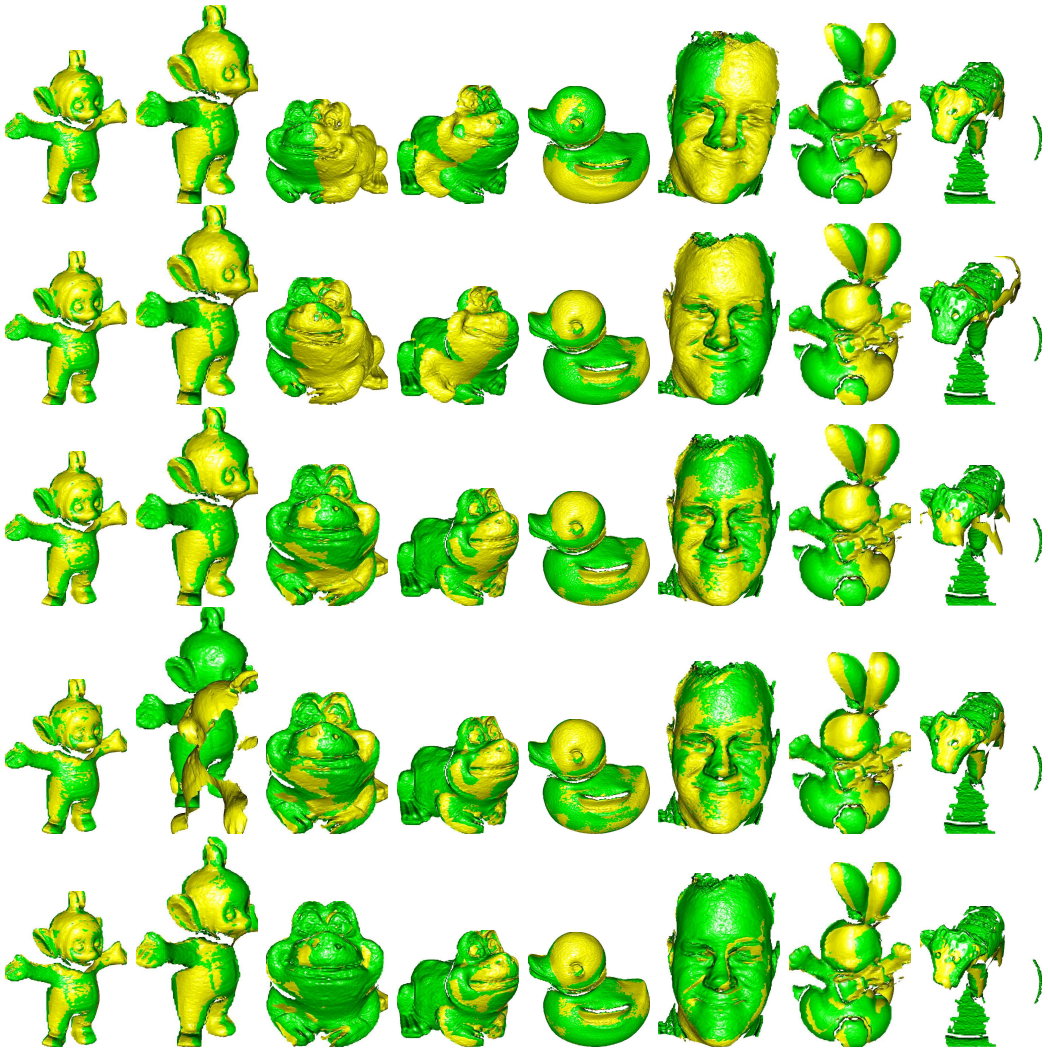

Figure 8: Registration results of different overlapping partial shapes for different evaluation algorithms. From left to right: tubby0–20, tubby20–40, frog0–40, frog40–80, duck0–20, pat108–144, bunny0–40 and cow37–45. Top row: GTM; Second: SparseVFC; Third: RANSAC; Fourth: IRLS; Bottom: RBAB.

4.5 Algorithm comparison, Technical Arts 100X data

We now give another comparative study of these techniques using data captured by another scanner Technical Arts 100X, with the PPMs established by another FEM method, USC [42], to verify whether our RBAB method is robust with respect to changes in data source, and features used for determining point matches. To this end, the overlapping partial adapter2–3, block3–5, column2–5, cap1–5, occl5–6, grnblk1-2, wye2–3 and tapetoll1–2 shapes in Figure 1 were selected. The experimental results are presented in Figures 10 and 11 and Table 5. The percentages of correct marches among the matches established between these shapes are 52%, 3%, 17%, 2%, 19%, 16%, 3%, and 32% respectively.

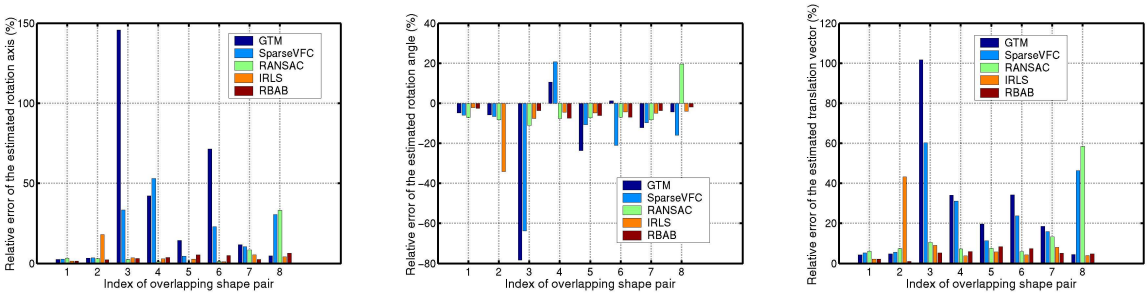

Figure 9: Relative errors of the estimated rotation axis (Left), rotation angle (Middle), and translational vector (Right) of different algorithms applied to different overlapping partial shapes. The overlapping shape pair index from 1 to 8 refers to tubby0–20, tubby20–40, frog0–40, frog40–80, duck0–20, pat108–144, bunny0–40 and cow37–45 respectively.

Table 4: The average μ and standard deviation σ of the relative errors $e_{\mathbf{h}}(\%)$, $e_{\theta}(\%)$, and $e_{\mathbf{t}}(\%)$ of the estimated rotation axis $\hat{\mathbf{h}}$, rotation angle $\hat{\theta}$, and translation vector $\hat{\mathbf{t}}$ for the underlying transformation, and evaluation time T in seconds, for different algorithms applied to different overlapping partial shapes.

Para.	Algo.	$e_{\mathbf{h}}$ (%)	e_{θ} (%)	$e_{\mathbf{t}}$ (%)	T (sec)
μ	GTM	37.05	-14.64	27.77	50.00
	SparseVFC	20.23	-14.08	25.03	43.50
	RANSAC	6.72	-4.59	14.57	31.25
	IRLS	4.96	-8.27	10.07	37.75
	RBAB	3.80	-3.98	5.04	30.75
σ	GTM	50.08	27.53	32.33	80.59
	SparseVFC	18.07	23.53	19.93	65.13
	RANSAC	11.00	9.88	17.91	36.02
	IRLS	5.51	10.54	13.62	35.81
	RBAB	1.69	2.59	2.43	34.36

The results show that the different algorithms exhibit similar behavior to those observed in the last section. The SparseVFC and RANSAC algorithms failed to accurately classify matches as correct and mismatches for the block3-5 shapes, the GTM, SparseVFC, and RANSAC algorithms failed to accurately estimate the underlying transformation from the evaluated PPMs for the cap1–5 shapes, the SparseVFC and IRLS methods failed to reliably classify the PPMs as correct or mismatches for the grnblk1-2 shapes, and the GTM, SparseVFC, and IRLS algorithms also failed to accurately down-weight the point matches between the wye2–3 shapes, leading to inaccurate estimation of the underlying transformation. Failure

of point match evaluation is manifested by the fact that the transformed block3, cap1, grnblk1, and wye2 data shapes intersect the reference block5, cap5, grnblk2, and wye3 shapes respectively in 3D space. The underlying transformation estimated by the IRLS algorithm from the evaluated PPMs between the block3-5 shapes is inaccurate, the GTM and SparseVFC algorithms inaccurately estimate the underlying transformation for the column2–5 shapes, and the RANSAC algorithm gives estimates for the underlying transformations for the occl5–6 and wye2–3 shapes that are significantly different from those refined by the SoftICP algorithm. Inaccurate estimation of the underlying transformation can be clearly seen from the superimposition, rather than inter-penetration, of the transformed block3, column2, occl5, and wye2 data shapes with respect to the reference block5, column5, occl6, and wye3 shapes respectively. In sharp contrast, the underlying transformations estimated by our RBAB algorithm brings all eight pairs of overlapping partial shapes into accurate alignment. Again, the various algorithms took similar time, unless they failed to produce an accurate enough initial underlying transformation for the SoftICP algorithm to refine. These superior results show that our proposed RBAB algorithm can handle data captured by different scanners, and matches provided using different features.

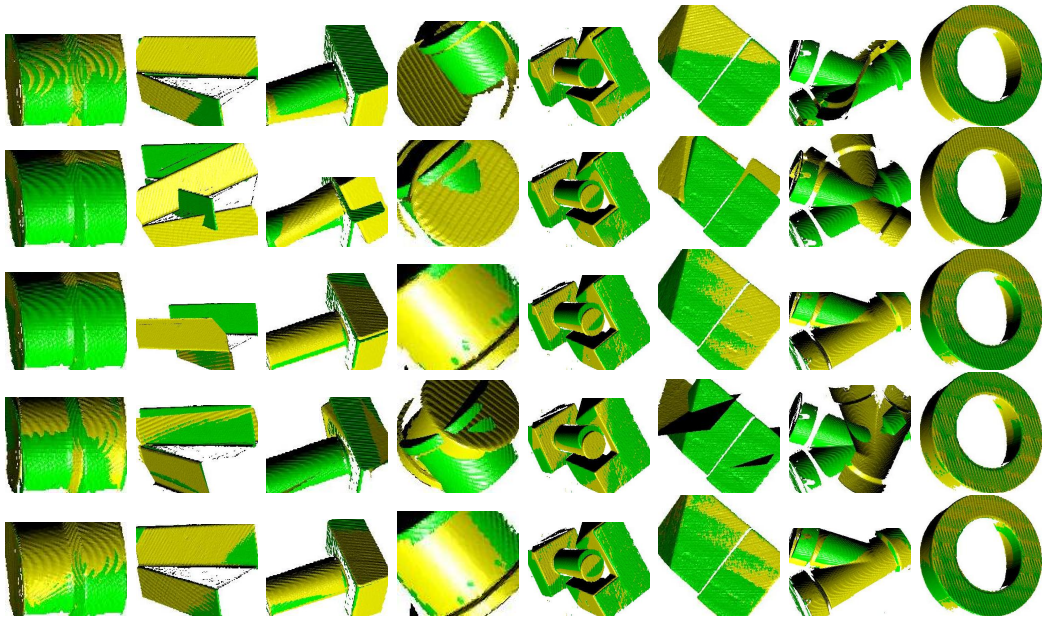

Figure 10: Registration results of different overlapping partial shapes for different evaluation algorithms. From left to right: adapter2–3, block3–5, column2–5, cap1–5, occl5–6, grnblk1-2, wye2–3 and tapetoll1–2. Top row: GTM; Second: SparseVFC; Third: RANSAC. Fourth: IRLS; Bottom: RBAB.

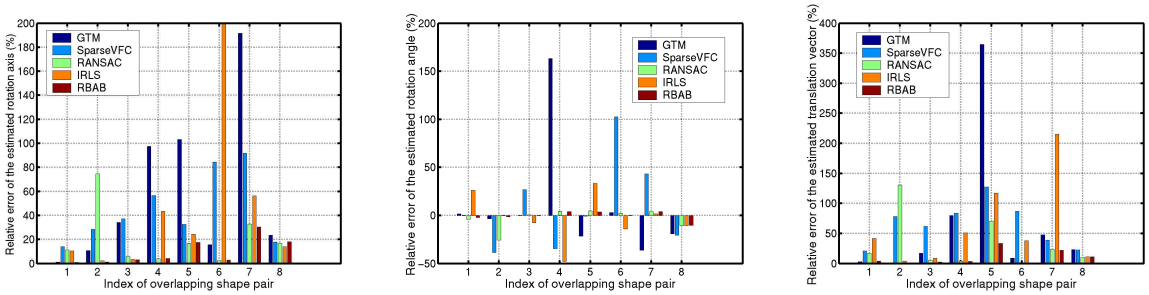

Figure 11: Relative errors of the estimated rotation axis (Left), rotation angle (Middle), and translational vector (Right) of different algorithms applied to different overlapping partial shapes. The overlapping shape pair index from 1 to 8 refers to adapter2–3, block3–5, column2–5, cap1–5, occl5–6, grnblk1-2, wye2–3 and taperoll1–2 respectively.

Table 5: The average μ and standard deviation σ of the relative errors $e_{\mathbf{h}}(\%)$, $e_{\theta}(\%)$, and $e_{\mathbf{t}}(\%)$ of the estimated rotation axis $\hat{\mathbf{h}}$, rotation angle $\hat{\theta}$, and translation vector $\hat{\mathbf{t}}$ for the underlying transformation, and evaluation time T in seconds, for different algorithms applied to different overlapping partial shapes.

Para.	Algo.	$e_{\mathbf{h}}$ (%)	e_{θ} (%)	$e_{\mathbf{t}}$ (%)	T (sec)
μ	GTM	59.72	10.92	68.25	58.25
	SparseVFC	45.37	9.84	65.12	69.25
	RANSAC	20.61	-3.02	32.84	36.62
	IRLS	44.34	-2.28	60.91	102.37
	RBAB	9.83	-0.28	9.92	37.00
σ	GTM	65.93	63.06	122.73	38.90
	SparseVFC	29.45	47.01	36.49	58.66
	RANSAC	23.94	10.56	45.46	32.28
	IRLS	65.75	25.12	72.16	110.15
	RBAB	10.92	4.83	11.75	33.01

5 Conclusions

Feature extraction and matching are widely used for registering overlapping 3D partial shapes. Unfortunately, up to 98% of the putative point matches established can be mismatches with usually unpredictable errors, and are thus challenging to evaluate. While RANSAC is popular and applicable, it requires choices of the maximum number of iterations for termination, quantifying the quality of the underlying transformation for its optimal selection, and for a threshold for determining whether a match is correct or a mismatch. This paper proposes a novel boosting-inspired method for this task.

It is applicable to matches with as many as 98% being mismatches, representing various complexities of geometry. The smaller this proportion, the easier that point match evaluation becomes. Our new method estimates a boosting parameter and updates the weights of the PPMs by regularizing them using an entropy maximization framework [16]. The PPMs are penalized by considering both how large their errors are, and how far away they are from the weighted average error. This has the advantage of overcoming the bias introduced by using individual errors alone, without considering how other errors are distributed. The underlying transformations from different iterations are combined by weighted averaging, after discarding the initial few transformations, to provide an optimal estimate; the weights are defined by the boosting parameters.

Our contributions can be summarized as follows. Firstly, we have provided a novel boosting-inspired method for evaluating point matches between two overlapping 3D partial shapes which have already been established by any chosen FEM method. Point matches are treated as having different reliabilities, and are accordingly penalized both when re-estimating these reliabilities, and when using the point matches to estimate the underlying transformation. Our method learns to characterize the reliabilities of both good and poor matches over different iterations, and to estimate underlying transformations that minimize the weighted average of the errors of these PPMs. This appears to be the first time that a method inspired by boosting has been suggested for this task.

Secondly, our comparative study based on a variety of real data shows that the proposed method outperforms other state-of-the-art approaches. It improves upon the coarse pose estimate typically provided by FEM methods, which is often used to initialise fine registration using the ICP algorithm or some variant. The result is that the latter is more likely to find a correct global minimum, leading to more accurate and robust registration.

Thirdly, provided that the weighted averages of the points are relatively stable from one iteration to another, it can be shown that the finally estimated underlying transformation is optimal. This property helps explain why the proposed RBAB algorithm produces more accurate and stable results than other methods.

Finally, our results also show that even when up to 98% of the established point matches are mismatches, the underlying transformation can still be recovered with an error as small as 5%. As long as a few good matches are present, our method can successfully determine them from amongst many mismatches.

While much attention has been paid to alternative feature descriptors, our work shows the importance of evaluating matches after putative matches have been found, and indeed, that a successful evaluation strategy can overcome weaknesses in feature descriptor and matching methods. The latter problems have attracted far more attention, but match evaluation is a relatively unexplored avenue for more accurate registration of overlapping partial shapes.

More work is needed to decide whether the proposed RBAB algorithm has limited errors after conver-

gence, like AdaBoost. Incorporating a rigidity constraint is also likely to further improve the results provided by our method. Applying the proposed technique for the evaluation of the point matches between projective images will also be one of our future research topics.

Acknowledgments

We feel really grateful to the anonymous reviewers for their insightful comments that have improved the clarity and readability of this paper.

References

[1] A. Albarelli, E. Rodola, A. Torsello. Fast and accurate surface alignment through an isometry-enforcing game. *Pattern Recognition* 48(2015) 2209–2226.

[2] M. Andreetto, N. Brusco, G.M. Cortelazzo. Automatic 3-D modeling of textured cultural heritage objects. *IEEE Trans. Image Processing* 13(2004) 354–369.

[3] T.R. Babu, M.N. Murty, V.K. Agrawal. Adaptive boosting with leader based learners for classification of large handwritten data. *Proc. Four Int. Conf. on Hybrid Intelligent Systems (HIS'04)*, pp. 326–331, 2004.

[4] J. Behley, V. Steinhage, A.B. Cremers. Performance of histogram descriptors for the classification of 3D laser range data in urban environments. *Proc. ICRA*, pp. 4391–4398, 2012.

[5] S. Berretti, N. Werghi, A.del Bimbo, P. Pala. Matching 3D face scans using interest points and local histogram descriptors. *Computers & Graphics* 37(2013) 509–525.

[6] S. Berretti, N. Werghi, A. del Bimbo, P. Pala. Selecting stable keypoints and local descriptors for person identification using 3D face scans. *The Visual Computer* 30(2014) 1275–1292.

[7] P.J. Besl, N.D. McKay. A method for registration of 3D shapes. *IEEE Trans. PAMI*, 14(1992) 239–256.

[8] H. Chui, A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding* 89(2003): 114–141.

[9] P. Cirujeda, Y.D. Cid, X. Mateo, X. Binefa. A 3D registration method via covariance descriptors and an evolutionary stable strategy game theory solver. *International Journal of Computer Vision* 115(2015) 306-329.

[10] C. Creusot, N. Pears, J. Austin. A machine learning approach to keypoint detection and landmarking on 3D meshes. *IJCV* 102(2013) 146–179.

[11] M.A. Fishler, R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of ACM*, 24(6) 381–295, 1981.

[12] Y. Freund, R.E. Schapire. A decision-theoretic generation of on-line learning and application to boosting. *Journal of Computer and System Science* 55(1997) 119–139.

[13] J.H. Friedman. Greedy function approximation: a gradient boosting machine. Technical report, Department of Statistics, Standford University, 1999.

[14] J.H. Friedman. Stochastic gradient boosting. Technical report, Department of Statistics, Standford University, 1999.

[15] J.H. Friedman, T. Hastie, R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Ann. Statist.* 28(2000) 337–407.

[16] S. Gold, A. Rangarajan, et al. New algorithms for 2-D and 3-D point matching: pose estimation and correspondence. *Pattern Recognition* 31(1998) 1019–1031.

[17] Y. Guo, M. Bennamoun, F. Sohel, M. Li, J. Wan, J. Zhang. Performance evaluation of 3D local feature descriptors. *Proc. ACCV*, 2014, pp. 178–194.

[18] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21(1999) 433–449.

[19] W. Li, Z. Yin, Y. Huang, Y. Xiong. Automatic registration for 3D shapes using hybrid dimensionality-reduction shape descriptions. *Pattern Recognition* 44(2011) 2926–2943.

[20] X. Li and Z. Hu. Rejecting mismatches by correspondence function. *Int. J. Comput. Vis.*, 89(2010) 117.

[21] Z. Lin, H. Lee and T. Huang. Finding 3-D point correspondences in motion estimation. *Proc. 8th Int. Conf. on Pattern Recognition*, 1986, pp. 303–305,

[22] Y. Liu, L. De Dominicis, B. Wei, L. Chen, R.R. Martin. Regularization based iterative point match weighting for accurate rigid transformation estimation. *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 9, pp. 1058-1071, 2015.

[23] Y. Liu. Automatic registration of overlapping 3D point clouds using closest points. *Image and Vision Computing*, 24(2006) 762-781. Y. Liu. Automatic 3d free form shape matching using the graduated assignment algorithm. *Pattern Recognition*, 38(2005) 1615–1631.

[24] Y. Liu. Constraints for closest point finding. *Pattern Recognition Letters*, 29(2008) 841–851.

[25] T.R. Lo, and J.P. Siebert. Local feature extraction and matching on range images: 2.5D SIFT. *Computer Vision and Image Understanding*, 113 (2009) 1235-1250.

[26] L. Lucchese, G. Dorette, G.M. Cortelazzo. A frequency domain technique for range data registration. *IEEE Trans. PAMI*, 24(2002) 1468–1484.

[27] J. Ma, J. Zhao, A.L. Yuille. Non-rigid point set registration by preserving global and local structures. *IEEE Trans. Image Processing*, vol. 25, no. 1, pp. 53-64, 2016.

[28] J. Ma, J. Zhao, J. Tian, A.L. Yuille, Z. Tu. Robust point matching via vector field consensus. *IEEE Trans. Image Processing*, 23(2014) 1706–1721.

[29] J. Ma, J. Zhao, J. Tian, X. Bai, Z. Tu. Regularized vector field learning with sparse approximation for mismatch removal. *Pattern Recognition*, 46(2013) 3519–3532.

[30] S. Messelodi, C.M. Modena. Boosting Fisher vector based scoring functions for presen re-identification. *Image and Vision Computing* 44(2015) 44-58.

[31] R. Nock, P. Piro, F. Nielsen, W.B.H. Ali, M. Barlaud. Boosting k-NN for categorization of natural scenes. *International Journal of Computer Vision* 100(2012) 294–314.

[32] OSU(MSU/WSU) range image database. http://sampl.ece.ohio-state.edu/data/3DDB/RID/index.htm.

[33] T. Parag, F. Porikli, A. Elgammal. Boosting adaptive linear weak classifiers for online learning and tracking. *Proc. CVPR*, pp. 1–8, 2008.

[34] K. Pathak, A. Birk, N. Vaskevicius, J. Popinga. Fast registration based on noisy planes with unknown correspondences for 3-D mapping. *IEEE Trans. PAMI*, 26(2010) 424–441.

[35] J.M. Phillips, R. Liu, C. Tomasi. Outlier Robust ICP for Minimizing Fractional RMSD. *Proc. 3DIM*, 2007, pp. 427–434.

[36] S. Ramalingam, Y. Taguchi. A theory of minimal 3D point to 3D plane registration and its generalization. *International Journal of Computer Vision*, 102(2013) 73–90.

[37] S. Rusinkiewicz, M. Levoy. Efficient Variants of the ICP Algorithm. *Proc. International Conference on 3D Digital Imaging and Modeling (3DIM)*, 2001.

[38] R.B. Rusu, N. Blodow, Z.C. Marton, M. Beetz. Aligning point cloud views using persistent feature histograms. *Proc. IROS*, pp. 3384-3391, 2008.

[39] S.A.A. Shah, M. Bennamoun, F. Boussaid. A novel 3D vorticity based approach for automatic registration of low resolution range images. *Patter Recognition*, 48(2015) 2859–2871.

[40] L. Silva, Olga R.P. Bellon, and K.L. Boyer. Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. *IEEE Trans. PAMI*, 27(2005) 762–776.

[41] F. Tombari, S. Salti, L.D. Stefano. Unique signatures of histograms for local surface description. *Proc. ECCV*, 2010, pp. 347–360.

[42] F. Tombari, S. Salti, L.D. Stefano. Unique shape context for 3D data description. *Proc. 3DOR'10*, 2010, pp. 57–62.

[43] C. Torre-Ferrero, J.R. Llata, S. Robla, E.G. Sarabia. A similarity measure for 3D rigid registration of point clouds using image-based descriptor with low overlap. *Proc. IEEE 12th Int. Conf. on Computer Vision Workshops*, 2009, pp. 71–78.

[44] P. Viola, M.J. Jones. Robust real time face detection. *International Journal of Computer Vision*, 57(2004) 137–154.

[45] Y. Wei, W. Yao, J. Wu, M. Schmitt, U. Stilla. Adaboost-based feature relevance assessment in fusing LIDAR and image data for classification of trees and vehicles in urban scenes. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-7(2012) 323–328.

[46] Z. Zhang. M-estimators. http://research.microsoft.com/en-us/um/people/zhang/inria/publis/tutorial-estim/node24.html, 1996.