IDT Vs L2 Distance for Point Set Registration

H. Alghamdi, M. Grogan and R. Dahyot

School of Computer Science and Statistics
Trinity College Dublin
Ireland

Abstract

Registration techniques have many applications such as 3D scans alignment, panoramic image mosaic creation or shape matching. This paper focuses on (2D) point cloud registration using novel iterative algorithms that are inspired by the Iterative Distribution Transfer (IDT) algorithm originally proposed to solve colour transfer [Pitié et al., 2005, Pitié et al., 2007]. We propose three variants to IDT algorithm that we compare with the standard L2 shape registration technique [Jian and Vemuri, 2011]. We show that our IDT algorithms perform well against L2 for finding correspondences between model and target shapes.

Keywords: Registration, IDT, L2 distance.

1 Introduction

We use the following notations: the datasets $\{\mathbf{u}_i\}_{i=1,\cdots,m}$ (model *moving* point cloud) and $\{\mathbf{v}_j\}_{j=1,\cdots,n}$ (target point cloud), are point clouds in $\mathbb{R}^d$. $\mathbf{e}$ is a unit vector in $\mathbb{R}^d$ to project samples on a 1D space e.g. $u_i^e = \mathbf{e}^T\mathbf{u}_i$ and $v_j^e = \mathbf{e}^T\mathbf{v}_j$ are 1D scalar values. We investigate the registration of point clouds in $\mathbb{R}^d$ using an iterative approach where at each step the problem solved is the registration of 1D datasets $\{u_i^e\}_{i=1,\cdots,m}$ on $\{v_j^e\}_{j=1,\cdots,n}$. The intuitive idea to this strategy is that if the two point clouds are aligned in all possible 1D projective spaces, then registration is also achieved in $\mathbb{R}^d$. This strategy was first proposed as part of the IDT algorithm for colour transfer [Pitié et al., 2005, Pitié et al., 2007]. They used an optimal transport solution for registration in 1D spaces, and proposed a choice for the selection of unit vector direction $\mathbf{e}$ for solving colour transfer in 3D colour spaces ($d = 3$). In this paper, we propose several IDT solutions in 2D space to solve shape registration (c.f. section 3). We then compare our algorithms with the L2 shape registration technique proposed by Jian et al [Jian and Vemuri, 2011] (section 4), and evaluate their performance in finding correspondences between model and target shapes (c.f. Figure 1 for illustration).

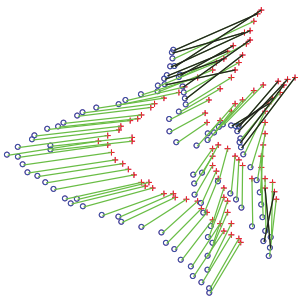

Figure 1: Shape registration (IDT-AvgD): model point set (red) and target point set (blue) with correspondences found (green for correct correspondences and black for incorrect ones).

2 State of the Art

In shape registration, many techniques have been proposed for shape registration that involve minimizing a divergence between two pdfs capturing the model point cloud $\{\mathbf{u}_i\}_{i=1,\cdots,m}$ and target point cloud $\{\mathbf{v}_j\}_{j=1,\cdots,n}$

 ISBN 978-0-9934207-2-6

[Sharp et al., 2008, Myronenko and Song, 2010, Jian and Vemuri, 2011, Ma et al., 2013]. Jian et al. propose to capture the structure of the shapes using Gaussian Mixture Models (GMMs), and estimate a parametric transformation T which registers the shapes by minimizing the L2 distance between the GMMs [Jian and Vemuri, 2011]. This technique performs very well against the state of the art and has since been extended to take into account additional information such as point correspondences or normal vectors to improve the registration result [Arellano and Dahyot, 2016, Ma et al., 2013]. Registration techniques are also important in the area of colour transfer, where the goal is to register the colour distribution of the target image to match that of the palette image. L2 registration of GMMs capturing the colour distribution of images has also been shown to give competitive results in this area [Grogan et al., 2015, Grogan and Dahyot, 2017]. Because of its performance and versatility, L2 registration [Jian and Vemuri, 2011] is used for comparison against our approach in our experimental results section.

Pitié et al. propose the Iterative Distribution Transfer (IDT) algorithm to register the 3D colour distributions of two images by iteratively projecting the 3D distributions onto several 1D subspaces before computing the registration in 1D space [Pitié et al., 2005, Pitié et al., 2007]. This dimension reduction technique reduces the computational complexity of the 3D registration problem and speeds up the process while giving good colour transfer results. However, their non-parametric transformation has been shown to cause problems when applied directly to the 3D shape registration problem [Grogan, 2017].

Algorithm 1 Iterative Distribution Transfer algorithm [Pitié et al., 2005, Pitié et al., 2007]

Require: Datasets $\{\mathbf{u}_i\}_{i=1,\cdots,m}$ and $\{\mathbf{v}_j\}_{j=1,\cdots,n}$

Require: Initialization $\mathrm{R} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4, \mathbf{e}_5, \mathbf{e}_6]$ (Eq. 1)

repeat

Compute 1D transfer functions T_1 to T_6 such that T_k is the optimal transport solution mapping the projections $\{u_i^{e_k} = \mathbf{e_k}^T \mathbf{u}_i\}$ onto the projections $\{v_j^{e_k} = \mathbf{e_k}^T \mathbf{v}_j\}$.

Compute for each point $\mathbf{u}$

$$T(\mathbf{u}) = \mathbf{u} + \mathrm{R}\begin{pmatrix} T_1(\mathbf{e_1}^T\mathbf{u}) - \mathbf{e_1}^T\mathbf{u} \\ T_2(\mathbf{e_2}^T\mathbf{u}) - \mathbf{e_2}^T\mathbf{u} \\ \vdots \\ T_6(\mathbf{e_6}^T\mathbf{u}) - \mathbf{e_6}^T\mathbf{u} \end{pmatrix} = \mathbf{u} + \mathrm{R}\begin{pmatrix} T_1(u^{e_1}) - u^{e_1} \\ T_2(u^{e_2}) - u^{e_2} \\ \vdots \\ T_6(u^{e_6}) - u^{e_6} \end{pmatrix} = \mathbf{u} + \underbrace{\sum_{k=1}^{6} \left(T_k(u^{e_k}) - u^{e_k}\right) \mathbf{e_k}}_{\text{shift to particle } \mathbf{u}}$$

Update (i.e. move) model dataset $\mathbf{u} \leftarrow T(\mathbf{u})$

$\mathrm{R} \leftarrow$ Random rotation of R

until Convergence

Algorithm 1 presents the IDT code as shared by the authors[1]. The matrix R sets 6 directions (unit vectors) for projection, and the model dataset is moved by the sum of the 6 displacements along these 6 axes. In the IDT algorithm at initialization stage, the matrix R is chosen as:

$$\mathrm{R} = \begin{pmatrix} 1 & 0 & 0 & 2/3 & 2/3 & -1/3 \\ 0 & 1 & 0 & 2/3 & -1/3 & 2/3 \\ 0 & 0 & 1 & -1/3 & 2/3 & 2/3 \end{pmatrix} \tag{1}$$

and unit vectors $\mathbf{e}_1$ to $\mathbf{e}_6$ in $\mathbb{R}^3$ correspond to the columns of R (from left to right). The transformation T_k is the Optimal Transport solution registering $\{u_i^{e_k}\}$ to $\{v_j^{e_k}\}$ computed by:

$$T_k(u) = P_v^{-1} \circ P_u(u) \tag{2}$$

where P_v and P_u are the Cumulative distribution Functions (CDFs) of v and u respectively. These CDFs are, in practice, approximated by the Empirical distribution functions using observations $\{u_i^{e_k}\}$ and $\{v_j^{e_k}\}$ [Pitié et al., 2005, Pitié et al., 2007].

[1] `https://github.com/frcs/colour-transfer`

 ISBN 978-0-9934207-2-6

The convergence of the algorithm is observed experimentally after several iterations when the overall transfer function become the identity function: $T(\mathbf{u}) \simeq Id(\mathbf{u}) = \mathbf{u}$. Convergence can also be measured experimentally by computing the Kullbach-Leibler divergence between pdfs associated with each datasets at each iteration [Pitié et al., 2007]. Following from IDT algorithm, several research contributions have also used 1D projection strategy to register distributions [Bonneel et al., 2015].

3 Alternative IDTs

We are proposing next to adapt the IDT algorithm to a 2D space for solving shape registration (see Algorithm 2). Three variants are tested using $n_{\mathbf{e}} = 6$ (noted IDT-AvgD), $n_{\mathbf{e}} = 2$ (noted IDT-orthD) and $n_{\mathbf{e}} = 1$ (noted IDT-seqD) projections per iteration of the algorithm. For all three variants, the first unit vector $\mathbf{e_1}$ is randomly generated in $\mathbb{R}^2$ as follows [Muller, 1959]:

$$\mathbf{e_1} \sim \mathcal{N}(\mathbf{0}, \mathrm{I}_2) \tag{3}$$

where $\mathcal{N}(0, \mathrm{I}_2)$ is a normal distribution in $\mathbb{R}^2$ centered on the origin in $\mathbb{R}^2$ with covariance matrix equal to the identity matrix in $\mathbb{R}^2$. Then $\mathbf{e_1}$ is normalized as follow so that $\mathbf{e_1}$ is a unit vector:

$$\mathbf{e_1} \leftarrow \frac{\mathbf{e_1}}{\|\mathbf{e_1}\|} \tag{4}$$

IDT-SeqD: sequential displacements ($n_{\mathbf{e}} = 1$). Unit vector $\mathbf{e_1}$ is the only one needed for IDT-seqD, and it is renewed at each iteration of our algorithm 2. This implies that the model dataset is moved or updated using one projected shift.

IDT-OrthD: using an orthogonal basis of $\mathbb{R}^2$ ($n_{\mathbf{e}} = 2$). In this variant of IDT, vector $\mathbf{e_1}$ is randomly generated as before and $\mathbf{e_2}$ is computed to be orthogonal to $\mathbf{e_1}$ so that $(\mathbf{e_1}, \mathbf{e_2})$ defines an orthonormal basis of $\mathbb{R}^2$. In this variant of IDT, the model dataset is updated using two cumulated projected shifts. The assumption is that the projections along the two orthogonal axis are independent, and the transfer functions T_1 and T_2 model two independent transfer functions for the marginal pdfs computed in the directions of $\mathbf{e_1}$ and $\mathbf{e_2}$. The overall shift is then computed by summing their displacements.

IDT-AvgD: average displacements $n_{\mathbf{e}} = 6$. Following Pitié et al., this variant of IDT in $\mathbb{R}^2$ considers 6 unit vectors at each iteration. While $\mathbf{e_1}$, $\mathbf{e_3}$ and $\mathbf{e_5}$ are randomly generated in $\mathbb{R}^2$ (c.f. equations 3 and 4), unit vectors $\mathbf{e_2}$, $\mathbf{e_4}$ and $\mathbf{e_6}$ are orthogonal respectively to $\mathbf{e_1}$, $\mathbf{e_3}$ and $\mathbf{e_5}$. The mapping can then be applied by taking the average of the independent 1D displacements on all axes to transform the 2D sample point, we will refer to this approach by average displacements (IDT-AvgD).

4 Experimental Results

We present experimental results on the application of our IDT variants to register 2D synthetic shapes differing by a non-rigid deformation. The data[2] consists of shapes with 5 different levels of deformation, with each level including 100 instances. All experiments are performed using MATLAB R2016b on a PC with 16 GB of RAM and an Intel Xeon E5-1620 (3.7 GHz) CPU. The goal of the point set registration is to align the model point set onto the target point set. The model point set is presented using red pluses and the target point set by blue circles in the figures below. To provide a quantitative comparison, we also report the results of the state-of-the-art algorithm GMM-TPS [Jian and Vemuri, 2011], which is implemented using publicly available code[3]. The GMM-TPS algorithm estimates a parametric non-rigid transformation (Thin Plate Spline) by minimizing the L2 distance between two GMMs capturing the shape of the point clouds. Since parametric TPS transformation is designed to be smooth, it typically maintains good point correspondences after registration and we investigate

[2]obtained at `http://www.cise.ufl.edu/anand/students/chui/research.html`

[3]obtained at `https://github.com/bing-jian/gmmreg`

 ISBN 978-0-9934207-2-6

Algorithm 2 Alternative IDT.

Require: Datasets $\{\mathbf{u}_i\}_{i=1,\cdots,m}$ and $\{\mathbf{v}_j\}_{j=1,\cdots,n}$

Require: $n_{\mathbf{e}}$ number of projections at each iteration

repeat

Generate random unit vectors in $\mathbb{R}^2$ to create matrix R of size $2 \times n_{\mathbf{e}}$

Compute 1D transfer functions T_1 to $T_{n_{\mathbf{e}}}$ such that T_k is the optimal transport solution mapping the projections $\{u_i^{e_k}\}$ onto the projections $\{v_j^{e_k}\}$.

Compute for each point $\mathbf{u}$

$$T(\mathbf{u}) = \mathbf{u} + \mathrm{R}\begin{pmatrix} T_1(\mathbf{e_1}^T\mathbf{u}) - \mathbf{e_1}^T\mathbf{u} \\ T_2(\mathbf{e_2}^T\mathbf{u}) - \mathbf{e_2}^T\mathbf{u} \\ \vdots \\ T_{n_{\mathbf{e}}}(\mathbf{e_{n_e}}^T\mathbf{u}) - \mathbf{e_{n_e}}^T\mathbf{u} \end{pmatrix} = \mathbf{u} + \underbrace{\sum_{k=1}^{n_{\mathbf{e}}} (T_k(u^{e_k}) - u^{e_k})\ \mathbf{e_k}}_{\text{shift to particle } \mathbf{u}}$$

Update $\mathbf{u} \leftarrow T(\mathbf{u})$ (Move)

until Convergence

how the correspondences estimated using our non-parametric IDT based algorithms compare against it. Note that to improve the convergence of GMM-TPS, we use a coarse-to-fine strategy by applying deterministic annealing on the bandwidth h.

Evaluation Criterion. For quantitative comparison, we compute the recall and precision metrics for each technique. These metrics are used to quantify the accuracy of the estimated point correspondences. The estimated correspondences are computed as the points in the target and transformed model points sets that are the closest, and that fall within a given accuracy threshold in terms of pairwise distance. Recall and precision are defined as follows:

$$Recall = \frac{TP}{TP+FN} \qquad Precision = \frac{TP}{TP+FP} \tag{5}$$

where TP denotes the number of true-positives, defined as the number of ground truth corresponding point pairs that fall within a given accuracy threshold in terms of pairwise distance after registration. FN denotes the number of false-negatives, defined as the number of ground truth corresponding point pairs that fall outside the given accuracy threshold. FP denotes the number of false-positives, defined as the number of estimated correspondences that are not ground truth.

We also use the recall-accuracy curve as in [Jian and Vemuri, 2011], under different accuracy thresholds (from 0 to 0.03) to evaluate the algorithms' ability to estimate true-positive correspondences with low errors in accuracy. To explore the convergence of the algorithms and quantify the similarity between the target point cloud and transformed model point cloud at every iteration, we choose to compute the robust L2 distance measure between the point clouds. When computing the L2 distance we use the same formulation as Jian et al and set the bandwidth to $h = 0.1$ for all methods.

Results on synthetic data. Figure 2 shows a sample of the registration results obtained by the three variants of the IDT algorithm: AvgD, OrthD and SeqD, as well as GMM-TPS, when registering shapes with different levels of deformation: the level of shape deformation in the target and model point clouds increases (0.02, 0.035, 0.05, 0.065, 0.08) from left to right. The registration result, estimated correspondences, percentage recall and percentage precision are shown from top to bottom for each algorithm. From the results, we observe that all IDT methods are able to produce almost perfect alignment for all levels of deformation. On the other hand, the ability to estimate true-positive correspondences varies. This can also be seen in Figure 3, which displays plots of the average recall curves over 100 shapes for each algorithm. We see that with moderate deformation,

 ISBN 978-0-9934207-2-6

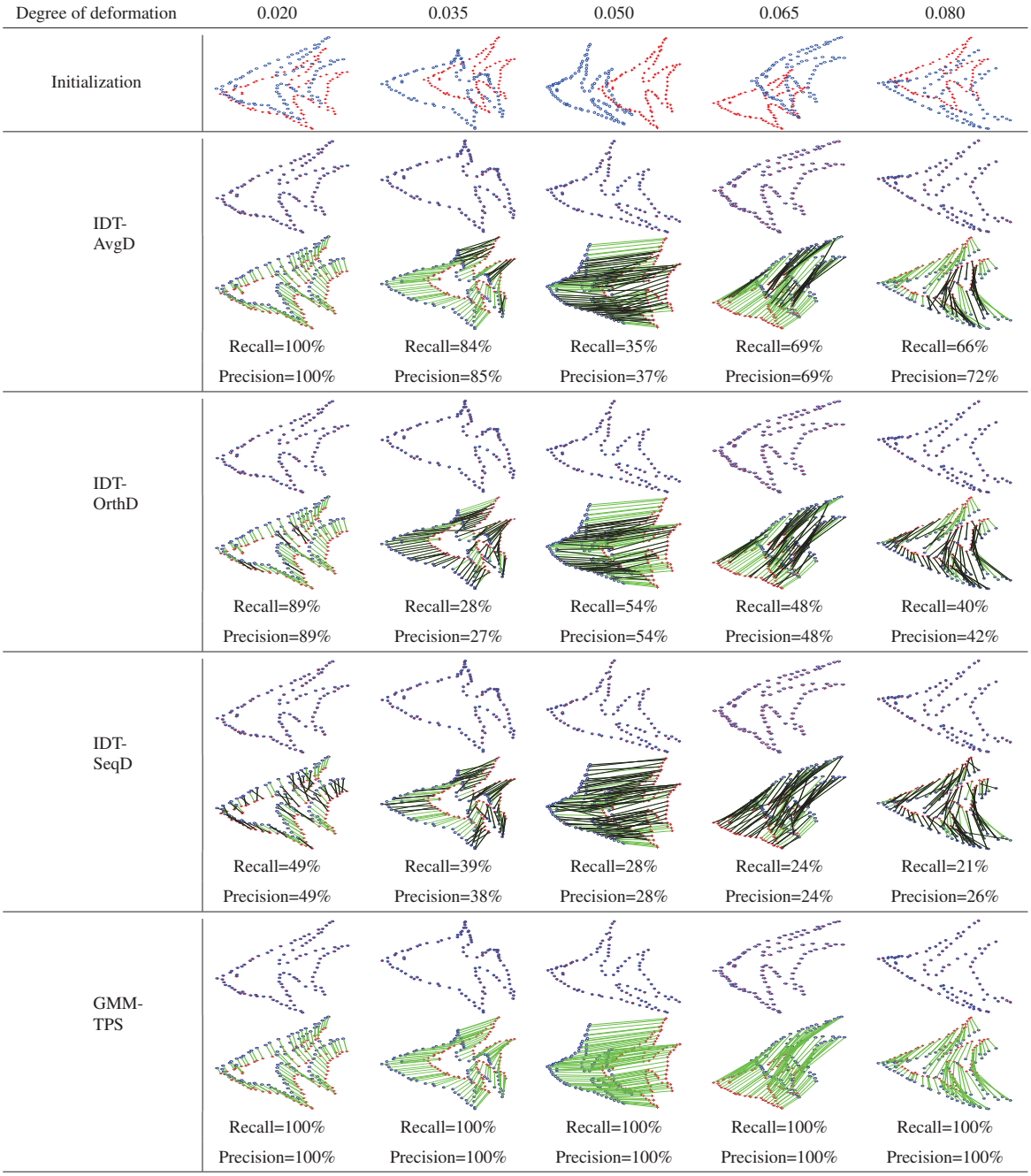

Figure 2: Comparisons of the registration results of IDT-AvgD, IDT-OrthD, IDT-SeqD and GMM-TPS with the registration result, estimated correspondences, recall and precision presented in every four rows. The first row indicates the degree of deformation, which is increasing from left to right. The second row displays the model and the target points set. The goal is to register the model point set (red pluses) onto the target point set (blue circles). In the correspondence figures, the coloured lines indicate the estimated correspondences (green = true positive, black = false positive).

IDT methods are able to achieve satisfactory results, in particular IDT-AvgD. Moreover, IDT-AvgD maintains better performance compared with the other IDT approaches as the deformation level increases. However,

 ISBN 978-0-9934207-2-6

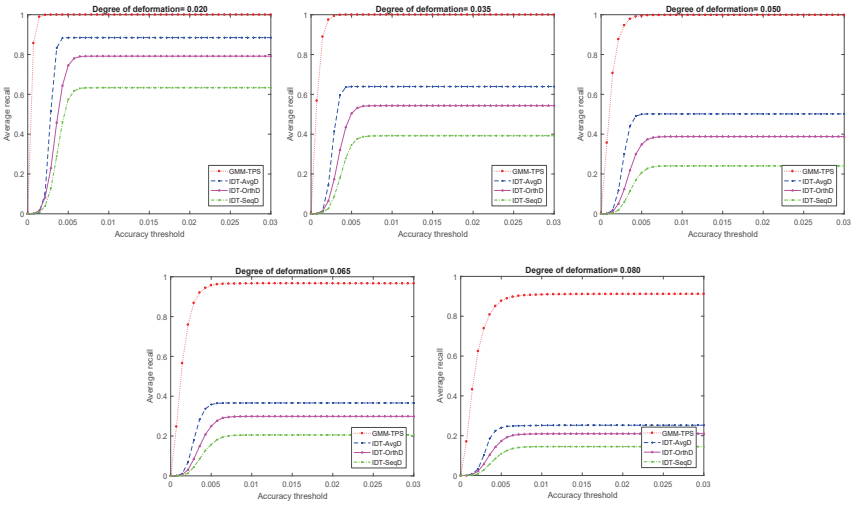

Figure 3: Average recall-accuracy curves over 100 shapes are used to evaluate IDT-AvgD, IDT-Orth, IDT-Seq and GMM-TPS under the accuracy threshold from 0 to 0.03.

the overall performance of the IDT algorithms degrades gradually as the degree of deformation in the data increases. On the other hand, the parametric transformation used by GMM-TPS generates good results, and maintains accurate point correspondences even as the level of deformation increases.

Figure 4 shows plots obtained by averaging the L2 distance, computed between the transformed model and target point sets, over 100 different shapes at each iteration in 2D. Note that we generate a sequence of projection axes and fixed these axes for all IDT algorithms to be the same. From left to right, the deformation level increases, and from top to bottom, the results for IDT-AvgD, IDT-OrthD and IDT-SeqD are presented. While all algorithms converge, we can see that the convergence behaviour of IDT-SeqD is less smooth than that of IDT-OrthD and IDT-AvgD, indicating that combining information from orthogonal bases improves the convergence of the algorithm. We can also see that although IDT-AvgD and IDT-OrthD converge to similar values by the final iteration, at earlier iterations IDT-OrthD obtains lower L2 values than IDT-AvgD. Therefore IDT-OrthD may be the preferred algorithm for convergence when considering fewer number of projection axes.

In Figure 5, for each algorithm we plot a box plot of the registration results for all 100 shapes at each level of deformation. The purpose is to show how close the IDT methods are to the state-of-the-art algorithm GMM-TPS in minimizing L2 distance between the transformed model and target point sets. We compare using the median instead of the mean to avoid the effect of outliers on the mean value. We see from the figure that the parametric GMM-TPS algorithm creates results with the lowest L2 distance, while the IDT-SeqD has the largest median L2 distance value. The median for GMM-TPS increases slightly when the degree of deformation increases, while the IDT methods maintain similar medians across all levels of deformation. The overall variability of the performance for all algorithms increases when the degree of the deformation increases. IDT-AvgD is the best of the IDT variations in term of minimizing L2 distance.

5 Conclusion & Future Work

This paper has investigated the registration of point clouds in $\mathbb{R}^2$ using a non-parametric iterative approach where at each step the problem solved is shape registration of 1D datasets. Overall our IDT based algorithms have a good performance while L2 remains the best. Note that IDTs solve iteratively the problem in 1D projective spaces with an unconstrained non parametric transformation while L2 solves it directly in 2D considering a smooth parametric transformation (TPS). TPS does not scale well in high dimensional spaces, but IDT ap-

 ISBN 978-0-9934207-2-6

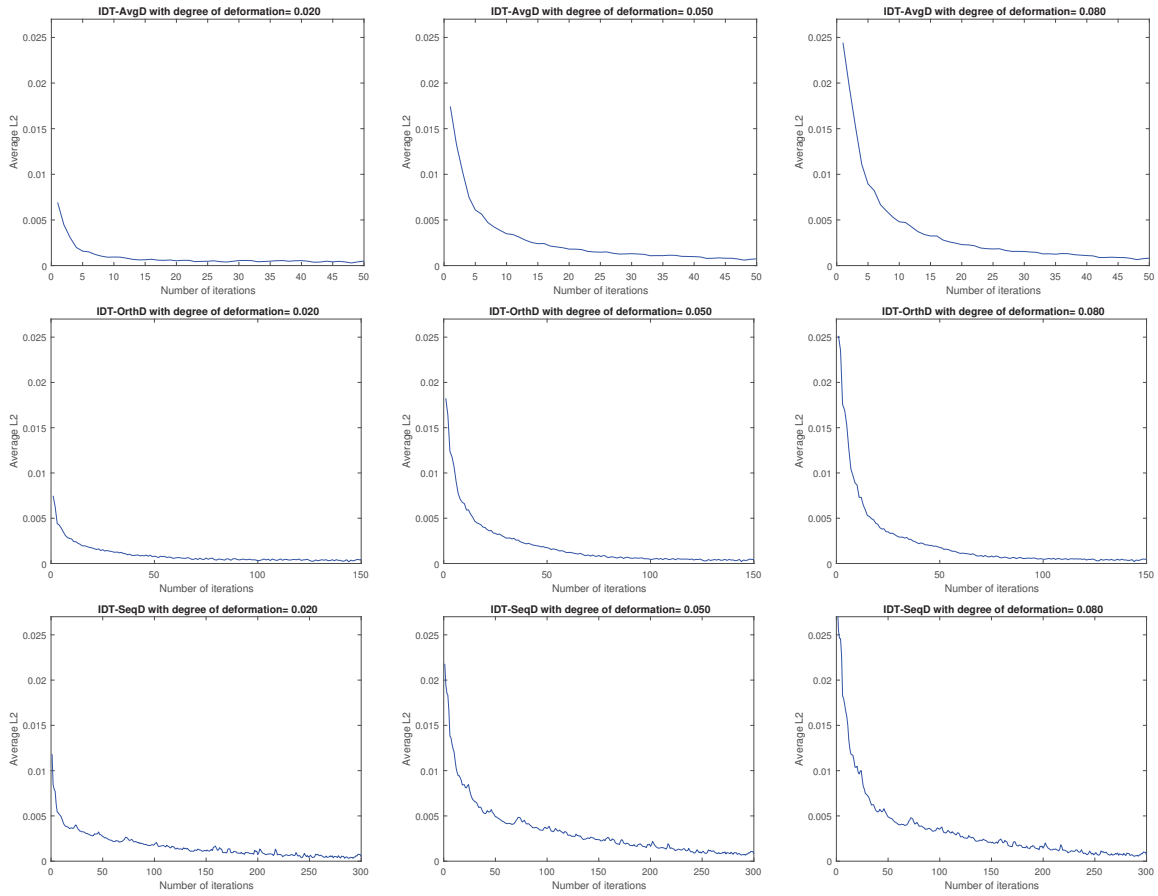

Figure 4: Evolution of the average L2 distance in 2D for 100 simulations. The abscissa indicates the number of iterations of our IDT algorithms: IDT-AvgD (top) at 20 iterations has used the same successive 20×6 projections as IDT-OrthD (middle) at 60 iterations and IDT-SeqD (bottom) at 120 iterations.

proach that considered 1D projective space has the potential to adapt well in higher dimensions, and it is also suitable for parallel optimization. Future work will then aim at assessing if a projective approach to registration can ease efficiently the computational load for registration of datasets in these cases.

Acknowledgments: The first author would like to thank Umm Al-Qura University, Saudi Arabia for funding this work as part of her PhD scholarship Programme. This work is also partly supported by the ADAPT Centre for Digital Content Technology (`www.adaptcentre.ie`) that is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

References

[Arellano and Dahyot, 2016] Arellano, C. and Dahyot, R. (2016). Robust ellipse detection with gaussian mixture models. *Pattern Recognition*, 58.

[Bonneel et al., 2015] Bonneel, N., Rabin, J., Peyré, G., and Pfister, H. (2015). Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45.

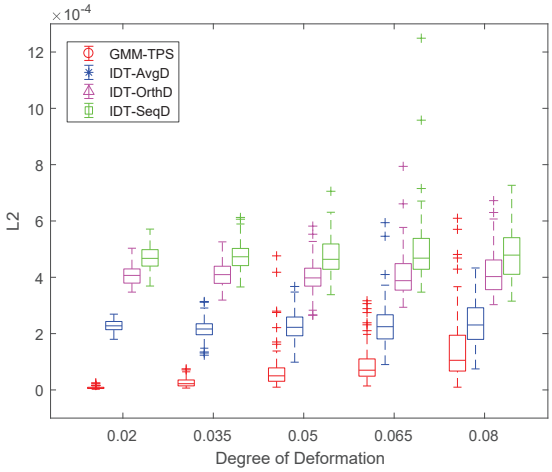

Figure 5: Box plots are used to compare the performance of IDT-AvgD, IDT-OrthD, IDT-SeqD and GMM-TPS in minimizing L2 distance. The median and the dispersion of 100 shapes are compared as the deformation degree increases.

[Grogan, 2017] Grogan, M. (2017). *Colour Transfer and Shape Registration using Functional Data Representations*. PhD thesis, Trinity College Dublin.

[Grogan and Dahyot, 2017] Grogan, M. and Dahyot, R. (2017). Robust registration of gaussian mixtures for colour transfer. Technical report, `https://arxiv.org/abs/1705.06091`.

[Grogan et al., 2015] Grogan, M., Prasad, M., and Dahyot, R. (2015). L2 registration for colour transfer. In *European Signal Processing Conference (Eusipco)*, Nice France.

[Jian and Vemuri, 2011] Jian, B. and Vemuri, B. (2011). Robust point set registration using gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1633 – 1645.

[Ma et al., 2013] Ma, J., Zhao, J., Tian, J., Tu, Z., and Yuille, A. L. (2013). Robust estimation of nonrigid transformation for point set registration. In *IEEE Conference on Computer Vision and Pattern Recognition*, Portland, OR, USA USA.

[Muller, 1959] Muller, M. E. (1959). A note on a method for generating points uniformly on n-dimensional spheres. *Communications of the ACM*, 2(4):19–20.

[Myronenko and Song, 2010] Myronenko, A. and Song, X. (2010). Point set registration: Coherent point drift. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(12):2262–2275.

[Pitié et al., 2005] Pitié, F., Kokaram, A. C., and Dahyot, R. (2005). N-dimensional probability density function transfer and its application to color transfer. In *Tenth IEEE International Conference on Computer Vision (ICCV 2005)*, volume 2, pages 1434 – 1439.

[Pitié et al., 2007] Pitié, F., Kokaram, A. C., and Dahyot, R. (2007). Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding journal (Special Issue on Color Image Processing)*.

[Sharp et al., 2008] Sharp, G. C., Lee, S. W., and Wehe, D. K. (2008). Maximum-likelihood registration of range images with missing data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(1):120–130.

 ISBN 978-0-9934207-2-6