Evaluation of a New Point Clouds Registration Method based on Group Averaging Features

Maja Temerinac-Ott[1,2], Margret Keuper[1,2] and Hans Burkhardt[1,2]
[1] *Chair of Pattern Recognition and Image Processing, Institute of Computer Science, Albert-Ludwigs-University of Freiburg, Germany*
[2] *Centre for Biological Signalling Studies (bioss), Albert-Ludwigs-University of Freiburg, Germany*
Email: temerina@informatik.uni-freiburg.de

***Abstract*—Registration of point clouds is required in the processing of large biological data sets. The tradeoff between computation time and accuracy of the registration is the main challenge in this task.**

We present a novel method for registering point clouds in two and three dimensional space based on Group Averaging on the Euclidean transformation group. It is applied on a set of neighboring points whose size directly controls computing time and accuracy.

The method is evaluated regarding dependencies of the computing time and the registration accuracy versus the point density assuming their random distribution. Results are verified in two biological applications on 2D and 3D images.

I. Introduction

Registering point clouds is a task often encountered in computer vision. The point clouds can represent interest points, detected markers or laser scan data. The registration is crucial for comparing or fusing the data. Fast and precise algorithms are necessary to perform this task. Depending on the transformation at hand and the structure of the data, different algorithms are used ([1]–[3]). If the points are distributed randomly in space, state-of-the-art methods perform poorly. This is why in [4] a new method based on local descriptors was described to perform the task of registering fluorescent point markers. The drawback of this method is the relatively high computation time and the strong dependency on the k-neighbors. We have introduced a local point descriptor based on Group Integration (GI) over the Euclidean group in [5] for describing the similarity between two protein structures. This descriptor can be applied not only for protein structures but in many tasks of computer vision.

The idea of the descriptor is to capture at each point the local constellation of its neighboring points by taking into account the local density of the points. We will describe the local density of a point by defining a 'point gradient'. These gradients along with the distances between points are invariant considering the Euclidean group. We will not consider the 'point gradients' and distances themselves, but there distribution in a histogram following the idea of shape histograms [6].

Our method is particularly efficient, if the density of the points is high in some regions of the image and using the k-nearest neighbors (for small k) is not sufficient to describe the local neighborhood.

In this work we would like to evaluate how the density of the points influences the performance of the GI algorithm and in which cases the GI algorithm can be used for the registration of point sets.

II. Group Averaging on Point Sets

A discrete point set $P := \{\mathbf{p}_n \in \mathbb{R}^k | n = 1, ..., M\}$ is mapped to $\mathbb{R}$ by defining an 'intensity function' $X : \mathbb{R}^k \rightarrow \mathbb{R}$ indicating the presence of the point. We choose to represent the point set P as the sum of overlapping Gaussian distributions. The function X at point $\mathbf{p} \in P$ is defined as:

$$X(\mathbf{p}) = \sum_i \exp^{-\left(\frac{\|\mathbf{p}_i - \mathbf{p}\|}{\sigma_G}\right)^2} \quad (1)$$

The gradient of X is then:

$$\nabla X(\mathbf{p}) \quad = \quad -\frac{2}{\sigma_G^2} \sum_i (\mathbf{p}_i - \mathbf{p}) \exp^{-\left(\frac{\|\mathbf{p}_i - \mathbf{p}\|}{\sigma_G}\right)^2} \quad (2)$$

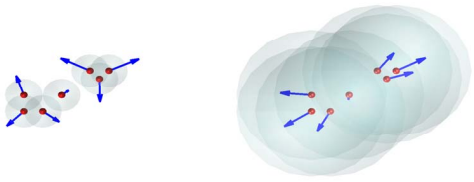
To get an intuition for the meaning of the density gradient, we show in Fig. 1 the effects of the gradient by choosing two different values for σ_G. From Fig. 1 we can see that σ_G controls the influence of points far away from the point $\mathbf{p}$ on the gradient $\nabla X(\mathbf{p})$. For instance, a small value of σ_G has the effect, that only the points in the direct neighborhood of $\mathbf{p}$ influence the direction of the gradient.

In the following GI is used to obtain an invariant description of a point cloud image X with respect to the Euclidean group.

An element g of the Euclidean group $\mathcal{E}$ acts on X by $gX(\mathbf{p}) \mapsto X(R\mathbf{p}+t)$, where $k \in \{2, 3\}$, $\mathbf{p}, t \in \mathbb{R}^k$ and $R \in \mathbb{R}^{k \times k}$ is an orthogonal matrix. A group integration feature is obtained by integrating a non-linear kernel function f over the group of Euclidean motion $\mathcal{E}$ (see [7]):

$$I_f(X) = \int_{\mathcal{E}} f(gX) dg. \quad (3)$$

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.600



(a) $\sigma_G = 10$ (b) $\sigma_G = 40$

Figure 1. The influence of σ_G on the 'point gradient'.

In [5] the following kernel function was introduced:

$$f_{d,n,n'}(X) = h_n(\nabla X(0))\ h_{n'}(\nabla X(d)), \tag{4}$$

where d is the width parameter and n, n' describe normalized orientation vectors. The function h is defined by:

$$h_n(v) = |v|\delta_1\left(\frac{|v^T n|}{|v|}\right), \tag{5}$$

where δ_1 is the Delta-Distribution giving contribution if its argument is nearby 1 and otherwise zero. h_n describes a specific configuration between two vectors v, n. The integral in Eq. 3 thus sums over all possible Euclidean transformations of two points with distance d having gradient vectors parallel with n, n'. The integral is evaluated once for each parameter set d, n, n'. This specific configuration is captured for three dimensional vectors using three angles ($\alpha, \beta, \gamma \in [-1, 1]$) and one distance vector $\Delta \in \mathbb{R}^+$. For 2D vectors two angles (α, β) and the distance vector $\Delta \in \mathbb{R}^+$ are sufficient.

Algorithm 1. GI Algorithm

1: Initialize $I_\Pi = 0$ for all Π.
2: **for** $i = 1$ to M **do**
3: **for** $j = 1$ to M **do**
4: Compute:
5: $\alpha = \frac{\nabla X(\mathbf{p}_i)}{\|\nabla X(\mathbf{p}_i)\|}^T \frac{\mathbf{p}_i - \mathbf{p}_j}{\|\mathbf{p}_i - \mathbf{p}_j\|}$,
6: $\beta = \frac{\nabla X(\mathbf{p}_j)}{\|\nabla X(\mathbf{p}_j)\|}^T \frac{\mathbf{p}_i - \mathbf{p}_j}{\|\mathbf{p}_i - \mathbf{p}_j\|}$,
7: $\gamma = \frac{\nabla X(\mathbf{p}_i)}{\|\nabla X(\mathbf{p}_i)\|}^T \frac{\nabla X(\mathbf{p}_j)}{\|\nabla X(\mathbf{p}_j)\|}$, $\Delta = \|\mathbf{p}_i - \mathbf{p}_j\|$
8: Let $\Pi = \{\alpha, \beta, \gamma, \Delta\}$
9: Update $I_\Pi \rightarrow I_\Pi + \|\nabla X(\mathbf{p}_i)\| \cdot \|\nabla X(\mathbf{p}_j)\|$.
10: **end for**
11: **end for**

The resulting algorithm (Alg. 1) computes a histogram of distances and orientations of point pairs in the point set P weighted by the magnitude of their gradients. The number of histogram bins for each parameter Π is chosen according to the application.

We describe the local neighborhood $N = \{\mathbf{p}_i \in P | 0 < \|\mathbf{p} - \mathbf{p}_i\| < r\}$ of all points $\mathbf{p} \in P$ by group averaging(step 9, Alg. 1) in order to find correspondences. For instance

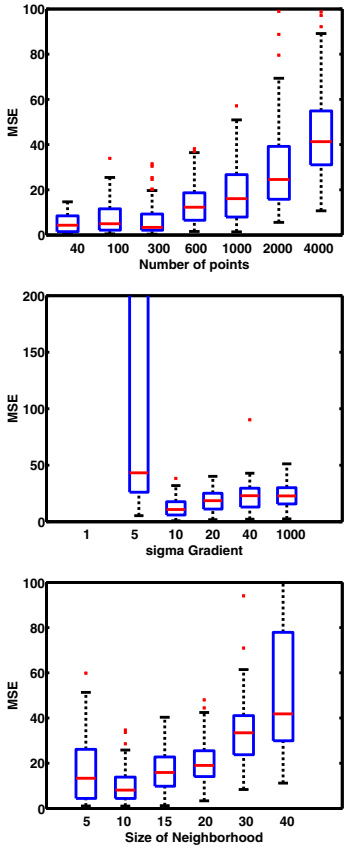


Figure 2. The influence of the point set size $|P|$ (a), the Gaussian modeling parameter σ_G (b) and the size of the neighborhood r (c) on the MSE for 100 random runs.

GI is performed inside hyperspheres with constant radius r positioned at centers $\mathbf{p}$. Corresponding points $\mathbf{p}_1 \in P_1$ and $\mathbf{p}_2 \in P_2$ are identified by histogram comparison.

III. Benchmark for the GI-Algorithm

In the following we would like to explore the stability of the GI features on a 2D data set. Our experimental setup is an image space I of size 200×250 pixel and a set of randomly normally distributed points of size $|P|$. The point set is rotated by the angle $\phi = 5$ around the center of the image and translated by $t = [20, -10]$. The transformation between the two point sets is computed by first finding correspondences using the GI features and then computing the transformation using the first $K = 20$ correspondences with the most similar GI feature vectors. True correspondences are determined with the RANSAC algorithm. The histogram bin size is $\#\text{bin}_\alpha = 4, \#\text{bin}_\beta = 4$ and $\#\text{bin}_\Delta = 10$. The implementation was performed in Matlab R2009a on a Intel Core 2 Duo processor with 3GHz and 8GB RAM.

For the computation of the GI features in the first experiment, we choose $\sigma_G = 10$ and the neighborhood size $r = 10$

$\|P\|$	40	300	600	1000	2000	4000
sec	0.03	0.57	2.66	5.24	10.65	19.10

Table I
TIME CONSUMPTION OF THE GI ALGORITHM FOR INCREASING POINT SET SIZE $|P|$.

r	5	10	15	20	30	40
sec	0.45	1.82	2.72	2.80	2.88	2.90

Table II
TIME CONSUMPTION OF THE GI ALGORITHM FOR INCREASING NEIGHBORHOOD SIZE r.

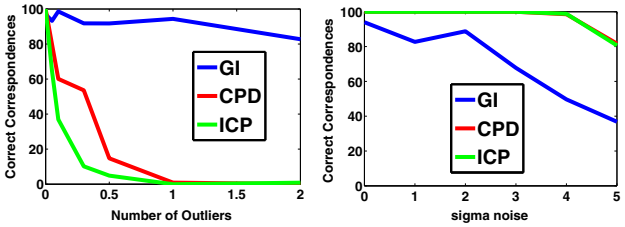
and perform the GI algorithm on 100 randomly generated point sets of size $|P|$. By increasing the point set size $|P|$, we are increasing the point density in I. From Fig. 2(a) we can see that for the given parameter set, the performance is best for $|P| = 300$. For $|P| > 300$, the mean square error (MSE) and its standard deviation increase. The time consumption increases almost linearly in relation to the point set size $|P|$ (Tab. I).

For the second experiment (Fig. 2(b)), we will take a set of randomly distributed points of size $|P| = 500$ and consider the neighborhood $r = 10$ and modify the standard deviation of the Gaussian modelling function σ_G. The MSE has its minimum at $\sigma_G = 10$ and increases slowly for $\sigma_G > 10$. For $\sigma_G < 10$ the MSE is very large. The optimal σ_G depends on the neighborhood size r. If σ_G is chosen too small the registration will fail.

Finally, we will keep $\sigma_G = 10$ and $|P| = 500$ constant and modify the size of the neighborhood r. From Fig. 2 (c), the optimal neighborhood size for this parameter set is $r = 10$. The computation time increases with the neighborhood size (Tab. II).

IV. COMPARISON TO STATE OF THE ART

In the following we conducted experiments in order to compare our method with state of the art methods ICP ([3]) and CPD ([2]). For this task a data set Q of 3D model points is generated by randomly selecting n_i^Q inlier in a given region of the space. Then the corresponding inlier in P are obtained by disturbing the n_i^Q points from Q with white gaussian noise $N(0, \sigma)$ and then rotate and translate the whole data set Q with a random rotation and translation. Next we add n_Q^o and n_o^P outliers in Q and P, respectively, by randomly selecting points in the same region as the inlier from Q and P, respectively, from the same random uniform distribution over the x-y-z coordinates. We enforce a constant density of 100 points over a $64 \times 64 \times 64$ volume. The total number of points in Q and P are $n_Q = n_Q^i + n_Q^o$ and $n_P = n_P^i + n_P^o$. The parameter σ controls the level of deformation between two point sets, while n_P^o and n_Q^o control the numbers of outliers in P and Q, respectively.



(a) Influence of outliers (b) Influence of noise σ

Figure 3. Comparison of the different point registration methods: On the y-Axis the number of correct matches is displayed. The x-axis displays the number of outliers in (a) and the level of noise (σ) in (b).

Figure 3 compares the performance of the different methods depending on (a) the percentage of outliers (without noise) and (b) the level of noise (without outliers). All algorithms ran on the same data sets over 30 trials for each value of the varying parameter, and the mean performance curves are plotted. The performance is measured by counting how many matches agree with the ground truth.

Figure 3 (a) shows that GI features can cope with a large number of outliers. For instance even if 200% outliers are added to P and to Q, 80% of the correct correspondences can be retrieved. On contrary, iterative methods fail to retrieve 80% of the matches if the percentage of outliers exceeds 5%. Iterative methods use the global shape of the point cloud and have difficulties if the outliers are evenly distributed and thus influence the global shape.

On the other hand, methods based on local descriptors have difficulties if noise is present. When the noise level $\sigma > 2$ the number of the retrieved matches drops rapidly. The iterative methods however, manage to retrieve 100% of the correct matches even if $\sigma = 4$. Thus when dealing with very noisy data, iterative methods should be preferred.

V. APPLICATIONS

In the following, we will present some successful applications of the GI-Algorithm to 2D and 3D data. For further applications and high resolution images please refer to [8].

A. Tracking and Identifying 2D Point Patterns

Marine biologists are interested in tracking and comparing bluespotted ribbontail rays (Taeniura lymma) according to there characteristic point pattern (Fig. 4). The GI-Algorithm can be applied to either track one individual ray in a video sequence or compare patterns of two different rays. The movement vector is computed by determining the point correspondences in two video frames. For the registration of two frames (Fig.4) with c.a. 70 points the GI-Algorithm requires 0.7 sec with average bead registration error of 1.52 px (min. 0.17 px max 2.95 px) on 52 correspondences. The parameters are $(\#\text{bin}_\alpha, \#\text{bin}_\beta, \#\text{bin}_\Delta) = (4, 4, 10)$, $\sigma_G = 10$ and $r = 20$. Since the videos are recorded in 2D, projective mappings can also occur. If the transform can be

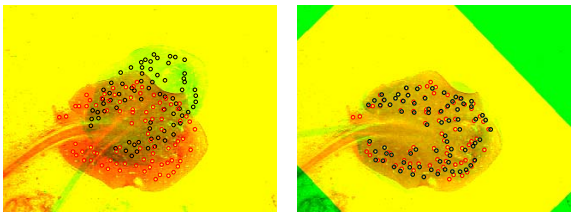
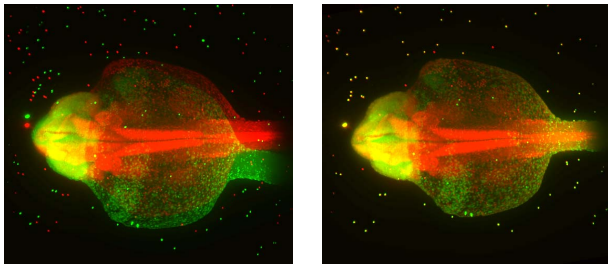


Figure 4. Registration of frame 6 to frame 12 of a ray video with the GI-Algorithm.



(a) preregistered data (b) after registration

Figure 5. Overlay of maximum intensity projections of two SPIM-views (angle 0° and angle 180°) before (a) and after applying the GI-Algorithm (b) on the surrounding beads.

approximated by rigid motion (as in the video at hand), the GI-Algorithm can be applied.

B. Registration of 3D image stacks

Three-dimensional zebrafish images are recorded from six different views using the Single Plane Illumination Microscopy (SPIM) ([9]). The SPIM obtains different views by rotating the probe around the y-axis for a specified angle (in this case 60°). A pre-registration of the data can be achieved by rotating all probes back by the inverse angle (Fig. 5(a)). However, a coarse registration is needed. Fluorescent point markers (called beads) are embedded in a surrounding medium for the purpose of coarse registration. The coarse registration is performed with the GI-Algorithm (Fig. 5(b)) with precision 0.91 px (min. 0.73 px max 1.72 px) on 61 correspondences. The parameters are $(\#\text{bin}_\alpha, \#\text{bin}_\beta, \#\text{bin}_\gamma, \#\text{bin}_\Delta) = (2, 2, 2, 10)$, $\sigma_G = 40$ and $r = 60$. For the registration of two SPIM images with c.a. 100 beads the GI-Algorithm requires 2 sec. In Fig. 5(b) the registration of two views (angle 0° and angle 180°) of a 24h old zebrafish is depicted. By first finding pair-wise correspondences using the GI-Algorithm and then applying groupwize registration, all views can be registered.

VI. Conclusion

A new method for the registration of point clouds is presented in this work. It is based on Group Averaging applied on a set of neighboring points. The number of points and their flexible choice make this method more efficient and reliable, even in cases when the standard methods fail.

The computation time and the quality of the GI-registration directly increase with the neighborhood size r. However, it is shown that an upper bound for r exists, when the quality of the registration begins to decrease. Therefore it is possible to find an optimal r e.g. by using point density statistics.

The GI-Algorithm has two interesting properties: 1. a stability to outliers and 2. the ability to describe the neighborhood of a point in a qualitative way. These properties will be backed up by further experiments in the future.

Acknowledgment

This study was supported by the Excellence Initiative of the German Federal Governments (EXC 294) and the SFB 592.

References

[1] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, pp. 381–395, 1981.

[2] A. Myronenko and X. Song, "Point-set registration: Coherent point drift," Oregon Health and Science University, Tech. Rep., May 2009.

[3] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, 1992.

[4] S. Preibisch, S. Saalfeld, T. Rohlfing, and P. Tomancak, "Bead-based mosaicing of single plane illumination microscopy images using geometric local descriptor matching," *Proceedings of SPIE*, no. 72592S, pp. 1–10, 2009.

[5] M. Temerinac, M. Reisert, and H. Burkhardt, "Invariant features for searching in protein fold databases," *International Journal on Computer Mathematics*, vol. 84, no. 5, pp. 635–651, May 2007.

[6] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Matching 3d models with shape distribution," *Proceedings Shape Modeling International*, 2001.

[7] H. Burkhardt and S. Siggelkow, "Invariant features in pattern recognition – fundamentals and applications," in *Nonlinear Model-Based Image/Video Processing and Analysis*, C. Kotropoulos and I. Pitas, Eds. John Wiley & Sons, 2001, pp. 269–307.

[8] M. Temerinac-Ott, "Point clouds registration using group averaging features," Technical Report 254, Jan 2010, available at http://lmb.informatik.uni-freiburg.de/papers/index.en.html.

[9] J. Huisken, J. Swoger, F. D. Bene, J. Wittbrodt, and E. H. K. Stelzer, "Optical sectioning deep inside live embryos by selective plane illumination microscopy," *SCIENCE*, vol. 305, pp. 1007–1009, August 2004.