ALBERT-LUDWIG UNIVERSITY FREIBURG
Institute of Computer Science
Chair for Pattern Recognition and Image Processing

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

Maja Temerinac-Ott

Technical Report 254

January 2010

Chair for Pattern Recognition and Image Processing
University of Freiburg

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

Maja Temerinac-Ott

University of Freiburg
Institute of Computer Science
Chair for Pattern Recognition and Image Processing
Georges-Koehler-Allee Geb. 052
79110 Freiburg, Germany

Telephone +49 761 203 8275, Fax +49 761 203 8262
e-mail: temerina@informatik.uni-freiburg.de

Technical Report 254

January 2010

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 three biological applications on 2D and 3D images.

1 Introduction

Registering point clouds is a task often encountered in computer vision. The point clouds can e.g. represent interest points, detected markers or laser scan data. The registration is crucial for e.g. 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 ([2, 6, 3]). If the points are distributed randomly in space, state-of-the-art methods perform poorly. This is why in [8] 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 [9] 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 describe 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 [7].

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.

2 Related Methods

There are two kinds of methods for comparing point clouds. The first kind of methods operates on the whole point set and tries to find correspondences by optimizing the distance function between the source and the moving data([6], [3]). These methods work particularly well if the point cloud has a specific shape and the optimization function has a global minimum. The optimal transformation is computed without explicitly determining the correspondences. This method fails if the two point sets differ in size and only their global distribution is matched.

The second kind of methods is based on first computing the local features for each point. Then the correspondence between the points is determined using these features. Finally, the transformation of the points is computed using the correspondences and applied to the whole point set. The most prominent of this methods is RANSAC ([2]), however it can get very slow for a large point set ($P_i > 3000$ points). Therefore it is necessary to reduce the size of points in the point set before applying RANSAC. A small set of reliable correspondences can be preselected using distances between local features. In the following section, we will present the theory for computing local features by Group integration.

3 Group Averaging on Point Sets

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

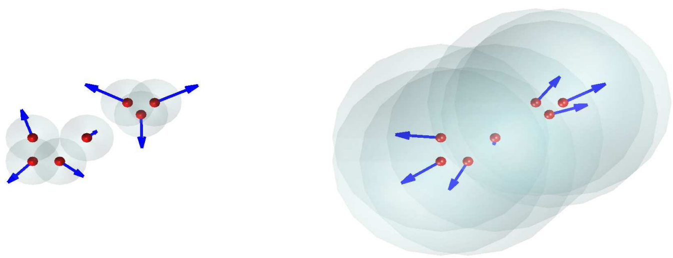
3.1 Point Gradients

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:

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

The gradient of X is then:

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



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

Figure 1: **The influence of σ on the 'point gradient'.** The points are marked in red, the gradient is marked in blue. The shape of the Gaussian is indicated in light blue.

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. From Fig. 1 we can see that σ controls the influence of points far away from the point $\mathbf{p}$ on the gradient $\nabla X(\mathbf{p})$. A small value of sigma has the effect, that only the points in the direct neighborhood of $\mathbf{p}$ influence the direction of the gradient. A large value of σ enhances the influence of points further away from $\mathbf{p}$. The gradient is always directed in the opposite direction of the point cloud of influence.

3.2 Group Averaging Algorithm

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 a orthogonal matrix. A group integration feature is obtained by integrating a kernel function f over the group of Euclidean motion $\mathcal{E}$ (see [1]):

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

In [9] 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 for each point. 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 $\Delta \in \mathbb{R}^+$ (Fig. 2). For 2D vectors two angles (α, β) and the distance vector $\Delta \in \mathbb{R}^+$ are sufficient.

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). For instance GI is performed inside hyperspheres with constant radius r positioned at centers $\mathbf{p}$. Corresponding points $\mathbf{p}_1 \in P_1$ and $\mathbf{p}_2 \in P_2$ have similar histograms.

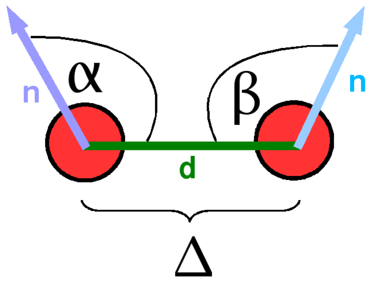


Figure 2: **The configuration of two points.** The configuration between two points in 3D is captured by three angles (α, β, γ) and the distance Δ between the points. The angle γ (not shown) is the angle between the orientation vectors n and n'.

Algorithm 1: GI Algorithm

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

3.3 Extensions to Group Averaging

As demonstrated in [9], the framework of Group Averaging can be extended by Spherical Harmonics or D-Wigner Matrices in order to obtain a better description of the features.

Furthermore, scaling invariance can be introduced by scaling the standard deviation of two point clouds to the same value and then applying the GI-algorithm. Affine transformation might be achieved by considering the proportional area information of the quadrangles spanned by the point cloud.

A tolerance to noise can be achieved by using fuzzy binning for the histogram computation in the GI-Algorithm.

4 Benchmark for Group Averaging

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.

4.1 Increasing the Point Density

For the computation of the GI features in the first experiment, we choose $\sigma = 10$ and the neighborhood size $r = 10$ and perform the GI algorithm on 100 randomly generated point sets of size $|P|$. By increasing the point

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

Table 1: Time consumption of the GI algorithm for increasing point set size $|P|$.

set size $|P|$, we are increasing the point density in I. From Fig. 3 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. 1).

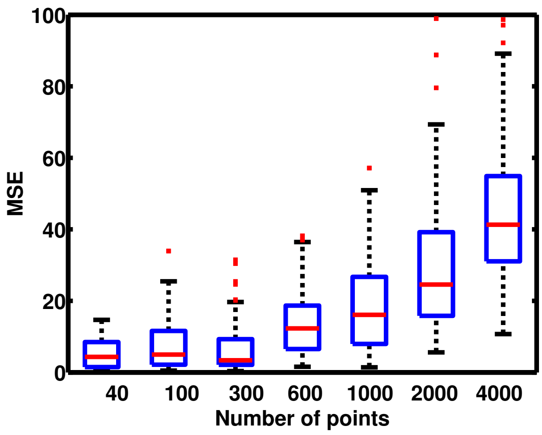


Figure 3: The influence of the point set size $|P|$ on the MSE for 100 random runs.

4.2 Influence of the Gaussian Modelling Parameter Sigma

For the second experiment (Fig. 4), we will take a set of normally distributed points of size $|P| = 500$ and consider the neighborhood $r = 10$ and modify the standard deviation of the Gaussian modelling function σ. The MSE has its minimum at $\sigma = 10$ and increases slowly for $\sigma > 10$. For $\sigma < 10$ the MSE is very large (for $\sigma = 1$ the MSE > 1000). The optimal sigma depends on the neighborhood size r. If σ is chosen too small the registration will fail.

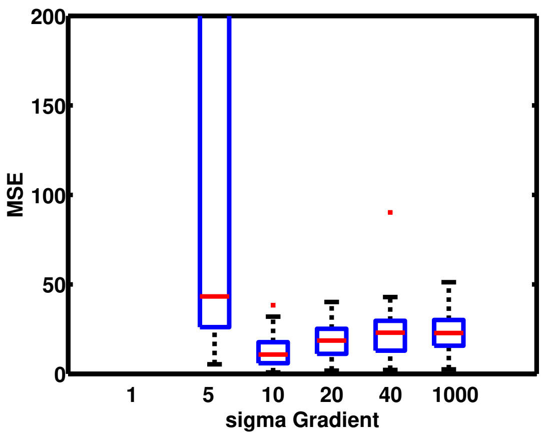


Figure 4: The influence of the Gaussian modeling parameter σ on the MSE for 100 random runs.

4.3 Influence of the Neighborhood Size

Finally, we will keep $\sigma = 10$ and $|P| = 500$ constant and modify the size of the neighborhood r. From Fig. 5, the optimal neighborhood size for this parameter set is $r = 10$. The computation time increases with the neighborhood size (Tab. 2).

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

Table 2: **Time consumption of the GI algorithm for increasing neighborhood size r.**

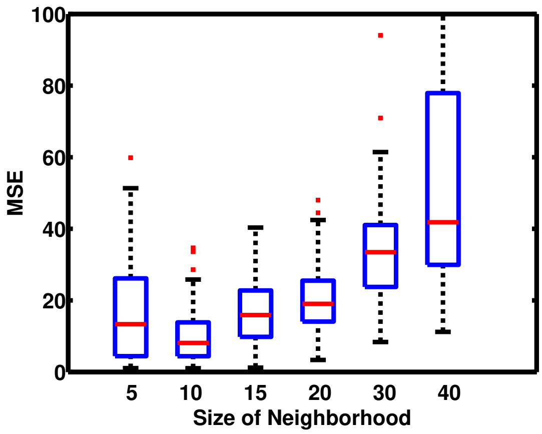


Figure 5: The influence of the the size of the neighborhood r on the MSE for 100 random runs.

5 Properties of the GI Algorithm

From the experiments above we conclude that a careful choice of the neighborhood size and the parameter sigma is crucial for the success of the algorithm.

The introduction of the gradient is crucial if we want to compare the local constellations of the points. By using only the distance between the points, we will get an incomplete descriptor. In Fig. 6 the three neighbors have the same distance to the reference point, however there is no Euclidean transform which can register the two points. Thus after defining the correspondence, the transformation between the two points must be computed explicitly in order to verify the correspondence. This verification step is not necessary after computing the descriptor by group averaging. The point gradients can resolve the ambiguity introduced by the k-nearest neighbors (Fig. 7).

Further we would like to emphasize, that the GI-Algorithm has two interesting properties. The first property is its stability to outliers: Since only local correspondences are considered, additional points (e.g. points from out of the field of view) will not hinder the success of the algorithm if there are enough local correspondences to be retrieved.

The second property is its ability to describe the neighborhood of a point in a qualitative way thus allowing for missing neighbors (e.g. due to occlusions) which is not possible when using the k-nearest neighbor descriptor. When using the GI-algorithm, we do not need to define the number of neighbors of one point, we can only consider a volume around the point. A large number of neighbors occurs already in a small volume, if the density of the point cloud is high.

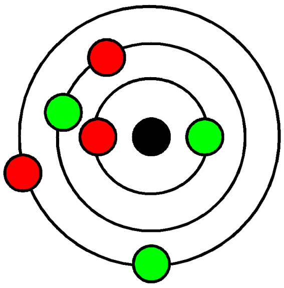
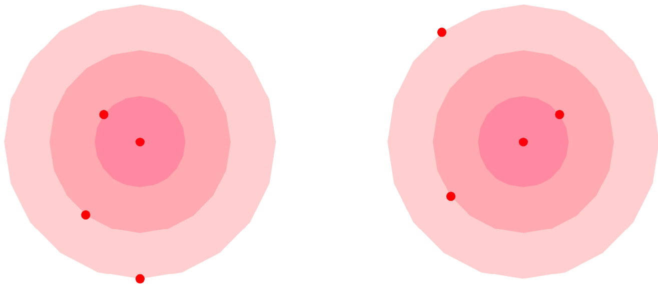
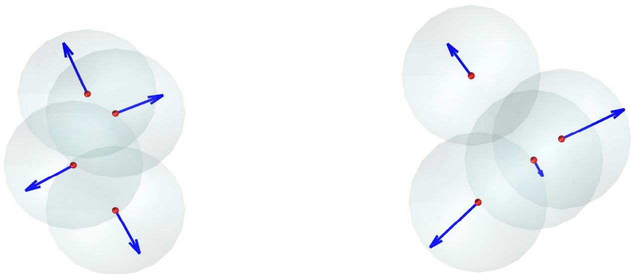


Figure 6: **Ambiguity of the K-nearest neighbor descriptor.** The red points with respect to the black point and the green points with respect to the black point have the same k-nearest neighbor descriptor. They can however not be transformed by Euclidean motion to the same reference frame.



(a) Points on concentric circles around the center point have the same distance. Two constellations with same distance to the origin are shown.



(b) The gradients are plotted for the points shown above. The gradients differ significantly, although the distances to the center point are equal.

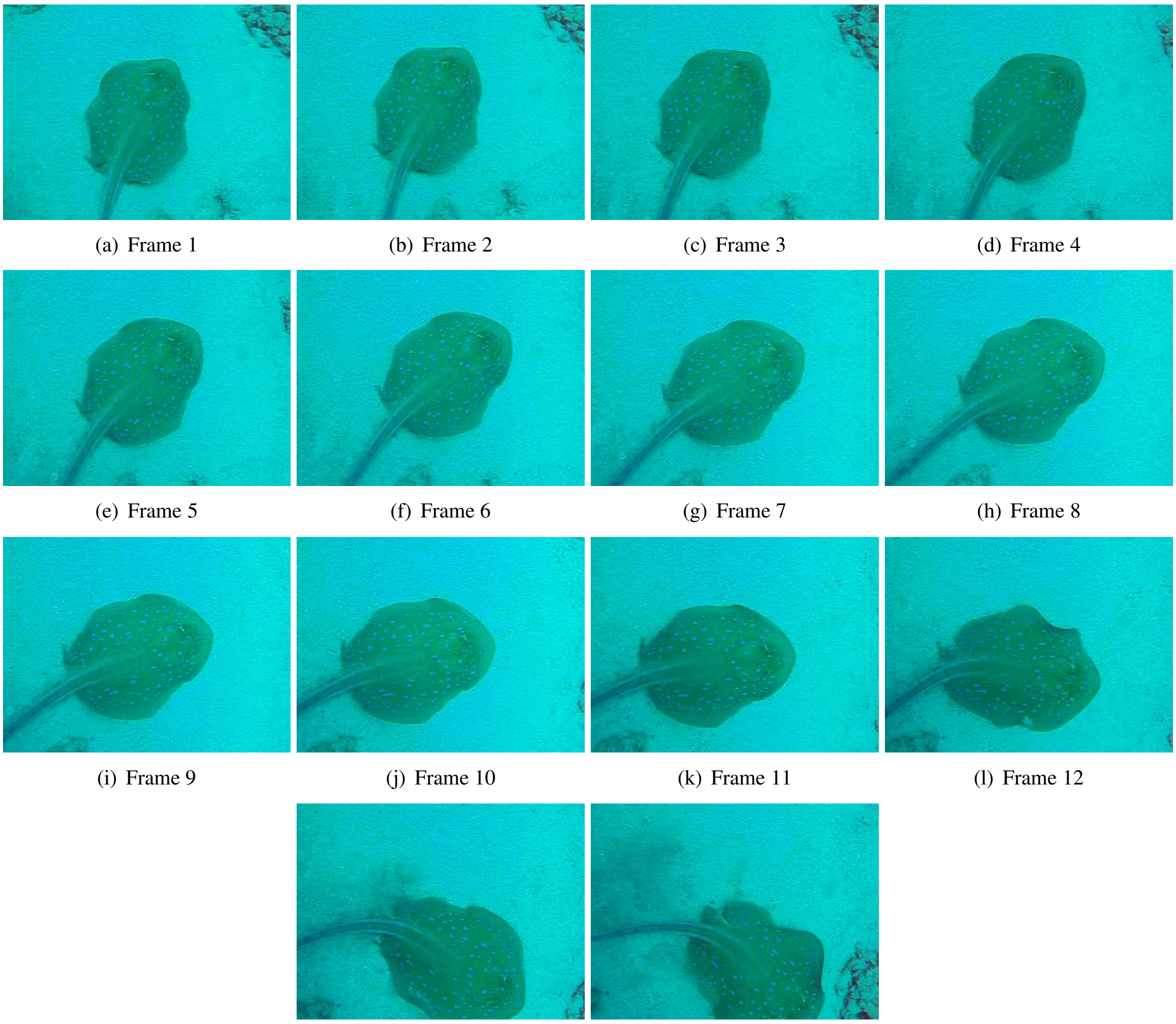
Figure 7: **Ambiguity solved with GI descriptor.** The points described here can be differentiated well with the GI-descriptor.

6 Applications

In the following, we will present some successful applications of the GI-Algorithm to 2D and 3D data.

6.1 Tracking and Identification of 2D Point Patterns

Marine biologists are interested in tracking and comparing bluespotted ribbontail rays (Taeniura lymma) according to there characteristic point pattern. They record videos of the rays and are interested in the evaluation of their data (Fig. 9). The coordinates of the points in the point pattern where manually extracted and vary between 67 and 83 points.



(a) Frame 1 (b) Frame 2 (c) Frame 3 (d) Frame 4

(e) Frame 5 (f) Frame 6 (g) Frame 7 (h) Frame 8

(i) Frame 9 (j) Frame 10 (k) Frame 11 (l) Frame 12

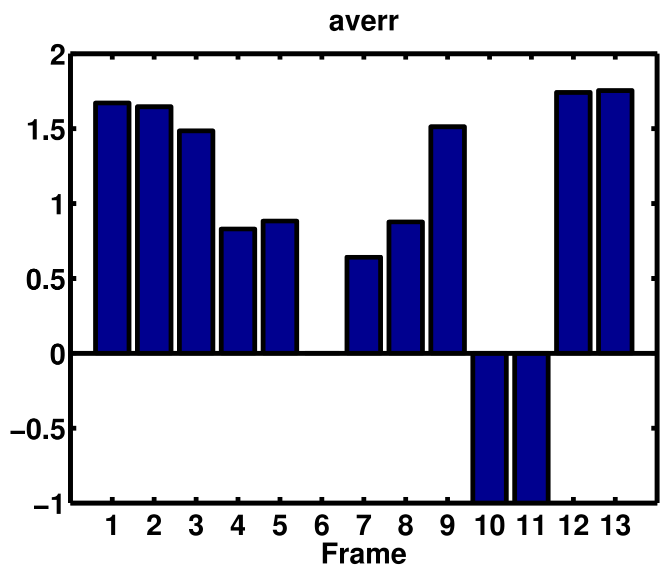
(m) Frame 13 (n) Frame 14

Figure 8: The Frames 1-14 from the bluespotted ribbontail ray video.

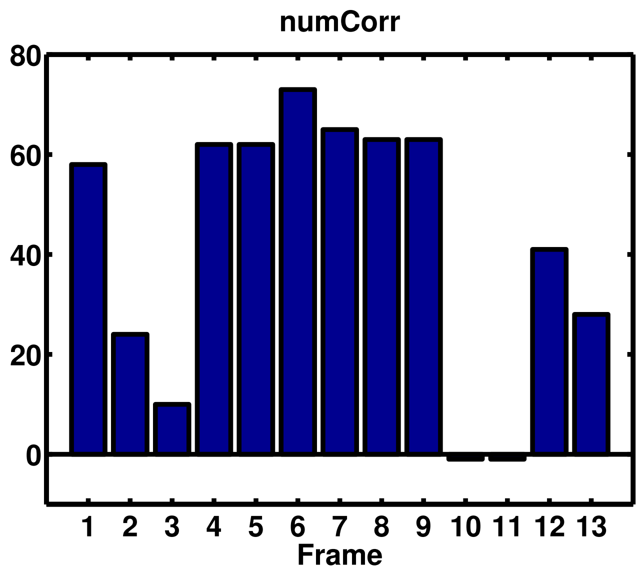
The GI-Algorithm can be applied to either track one individual ray in a video sequence or compare patterns of two different rays if the images can be registered by an Euclidean transform. The movement vector can be computed by determining the correspondences in two frames (Fig.10, Fig.11). For the registration of two frames (Fig.11) 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 $(\#\text{bin}_\alpha, \#\text{bin}_\beta, \#\text{bin}_\Delta) = (4, 4, 10)$, $\sigma = 10$ and $r = 20$.

For the comparison of two rays, one could compute the GI for the whole point pattern for each ray, thus obtaining one feature vector for each ray. The distance between these feature vectors can then be used to determine the similarity between the images of two rays, e.g. in order to identify the recorded ray.

In order to see the limitations of the method we have registered the sixth frame against all 14 frames. In Fig. we can see the average registration error in pixel distance and the number of found correspondences for each frame. If the registration failed the average error is negative. The registration failed for frame 10 and frame 11. For the data at hand (a video sequence of one ray) the GI-algorithm worked fine for most frames. However, in general the transformation of the ray in the image is better described by an affine transform.



(a) Average Error after registration with the GI-Algorithm.



(b) Number of Correspondencesafter registration with the GI-Algorithm.

Figure 9: The Frames 1-14 from the bluespotted ribbontail ray video are registered to Frame 6 with the GI-Algoirthm. The average error and the number of correspondences after registration are presented.

6.2 Registration of 3D image stacks

Three-dimensional zebrafish images are recorded from six different views using the Single Plane Illumination Microscopy (SPIM) ([5]). Fluorescent point markers (called beads) are embedded in a surrounding medium for the purpose of registration (see Fig. 12). The position of the beads is extracted by first smoothing the image, then extracting the local maxima and finally excluding points on the object with the help of morphological operators. The images are recorded with a red and a green channel, where points in the red channel mark the tissue of the zebrafish. The pixel size is $(1.14\mu m \times 1.14\mu m \times 3\mu m)$ per view resulting in a final output image with $959 \times 960 \times 250$px.

The number of beads extracted in each view are presented in Tab. 3. They differ between 78 beads and 111 beads.

view	0°	60°	120°	180°	240 °	300°
# beads	78	92	87	82	111	99

Table 3: **The number of detected beads in each view of the SPIM recording.**

The SPIM obtains different views by rotating the probe around the y-axis (with respect to the recorded stack) 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. 13). However, as the field of view is manually adjusted, we need to calculate the translation between the images. There is also a slight rotation between the images, since the rotation aperture is never perfectly calibrated. Thus, the GI-algorithm needs to be applied in order to obtain a better registration of the images.

After extracting the positions of the beads we obtain two 3D point sets (P_1 and P_2) and apply the GI-algorithm for computing the multidimensional histogram for each point in P_i. The point descriptors from P_1 and P_2 are compared using the sum of absolute differences. Correspondences are assigned according to minimal distances d between the histograms. For the computation of the registration parameters, only correspondences with $d < \epsilon$ are considered. The Registration parameters R, t are computed using RANSAC [2] and absolute orientation [4].

The coarse registration is performed with the GI-Algorithm (Fig. 14(b)) with precision 1.49 px (min. 0.73 px max 2.02 px) on 200 correspondences. The parameters are $(\#\text{bin}_\alpha, \#\text{bin}_\beta, \#\text{bin}_\gamma, \#\text{bin}_\Delta) = (2, 2, 2, 10)$, $\sigma = 40$ and $r = 60$. For the registration of two SPIM images with 100 beads the GI-Algorithm requires 11 sec.

In Fig. 14 the registered six different SPIM views are presented. Due to the recording limitations, only the image portion enclosed by the smaller part of the white circle can be used for further processing (e.g. image fusion).

The results of the pre-registration are compared to the results obtained after applying the GI-Algorithm in Fig. 15. We can see the improvement especially on the right side of the image. There the beads which are contained in all views have a characteristic star shape, since for every view the orientation of the point spread function rotates when rotating the image. In the bottom left corner the registration is not very good. This might be due to the fact, that the gel surrounding the probe moves elastically when performing the registration. Better results could also be obtained by applying the GI-Algorithm in an iterative manner. In this example we have only registered each view separately to the zero view.

6.3 Registration of Interest Points

Two individual three day old zebrafish (e05 and e07) are recorded with a confocal microscope. In order to obtain a high cellular resolution, each zebrafish is recorded in 4 tiles, which are stitched together to form one image. The output image has the size $339 \times 339 \times 210$px with isotropic voxel size of $2\mu m$

After interest point detection using Gradient Vector Flow resulting in 5010 interest points for zebrafish e05 and 8050 interest points for zebrafish e07, the extracted interest points are registered using the GI-Algorithm. Since the zebrafish is symmetrical, the local features extracted at each point are not sufficient for determining correspondences. For example the features at the edge of the fish eye will be very similar at the left and the right eye. Hence, global information about the points is included by assigning them to one side of the axis of symmetry. This restricts the search for correspondences to only one side of the axis of symmetry.

The results of the registration for $(\#\text{bin}_\alpha, \#\text{bin}_\beta, \#\text{bin}_\gamma, \#\text{bin}_\Delta) = (2, 2, 2, 5)$, $r = 20$ and $\sigma = 10$ are presented in Fig. 16. The precision 1.47 px (min. 0.35 px max 2.0 px) on 391 correspondences is achieved.

The computation time is 250 seconds for 7000 interest points per image. For $r = 10$ the computation time is 36 seconds, but the result is not satisfying.

7 Conclusions

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, e.g. when the point density is high. The GI method is evaluated using different random distributions of points and is verified through three biological applications.

The computation time and the precision of the registration directly increase with the neighborhood size. However, it is shown that an upper bound for the neighborhood size exists, when the precision begins to decrease. Therefore it is possible to find an optimal neighborhood size providing shorter computation times while keeping the registration error small. An optimal size of the neighborhood r could be determined using point density statistics.

8 Acknowledgements

We would like to thank J. Huisken for providing the SPIM images, M. Keuper for providing the zebrafish interest points and D. Mai for providing the point labels for the bluespotted ribbontail ray. This study was supported by the Excellence Initiative of the German Federal Governments (EXC 294) and the SFB 592.

References

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

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

[3] A. W. Fitzgibbon. Robust registration of 2d and 3d point sets. In *In British Machine Vision Conference*, pages 411–420, 2001.

[4] B. K. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, 4:629–642, 4 1987.

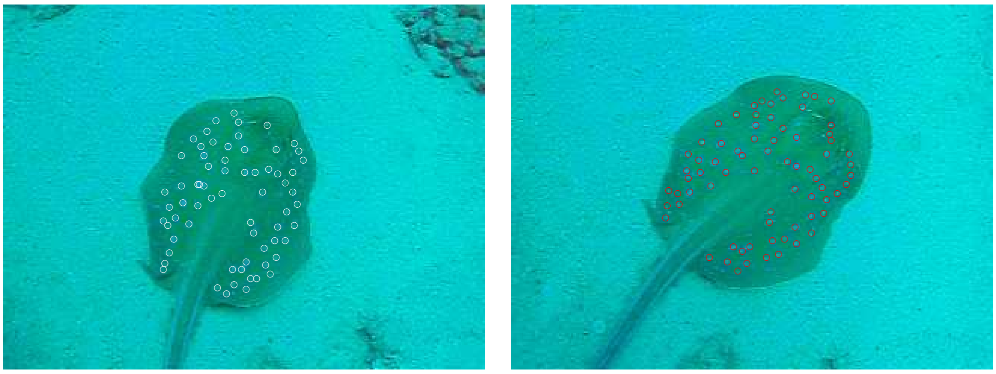
[5] J. Huisken, J. Swoger, F. D. Bene, J. Wittbrodt, and E. H. K. Stelzer. Optical sectioning deep inside live embryos by selective plane illumination microscopy. *SCIENCE*, 305:1007–1009, August 2004.

[6] A. Myronenko and X. Song. Point-set registration: Coherent point drift. Technical report, Oregon Health and Science University, May 2009.

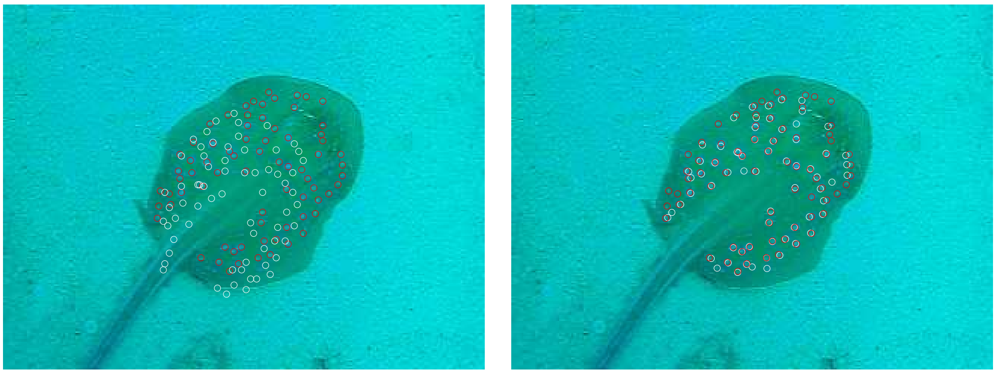
[7] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3d models with shape distribution. *Proceedings Shape Modeling International*, 2001.

[8] 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*, (72592S):1–10, 2009.

[9] M. Temerinac, M. Reisert, and H. Burkhardt. Invariant features for searching in protein fold databases. *International Journal on Computer Mathematics*, 84(5):635–651, May 2007.

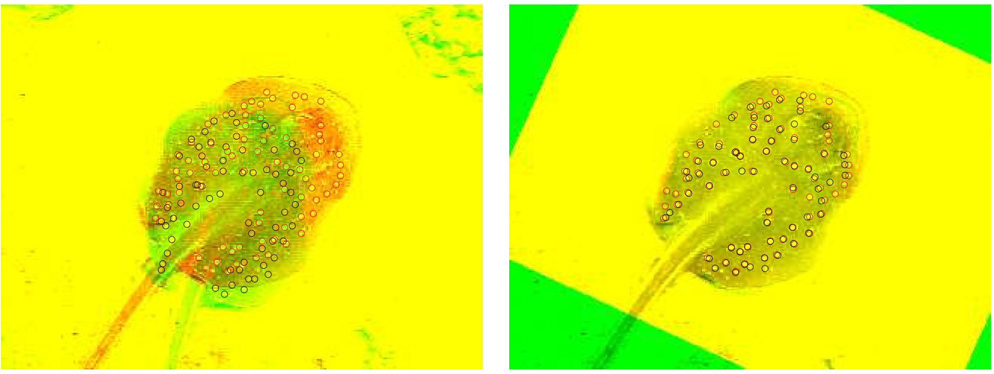


(a) Frame 1 with labeled Points (b) Frame 6 with labeled Points



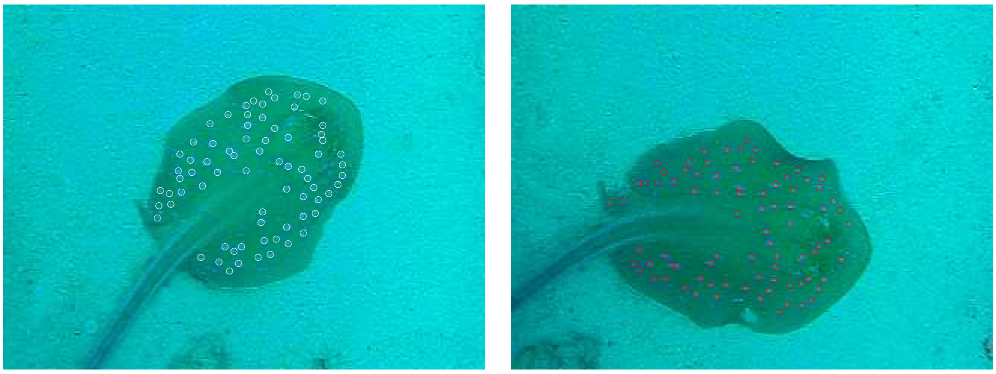
(c) Frame 6 with Points from Frame 1 before (left) and after (right) Registration

(d) Correspondences from Frame 1 and Frame 6 found with the GI-Algorithm are circled in the same color.

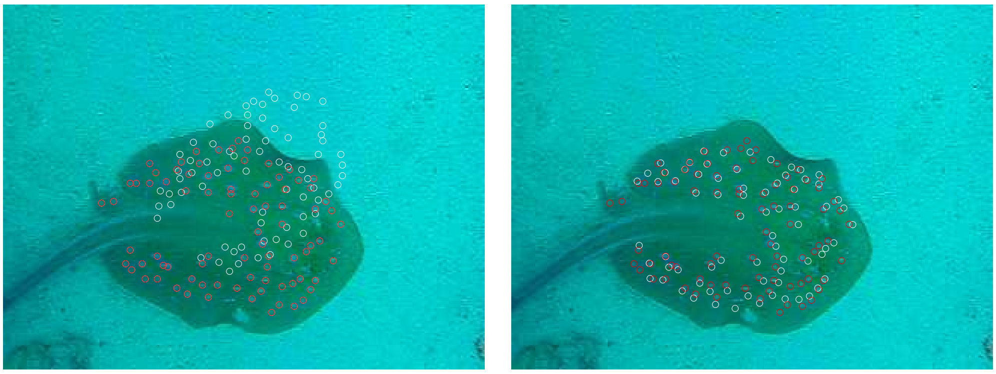


(e) Overlay from Frame 1 and Frame 6 before (left) and after (right) applying the GI-Algorithm.

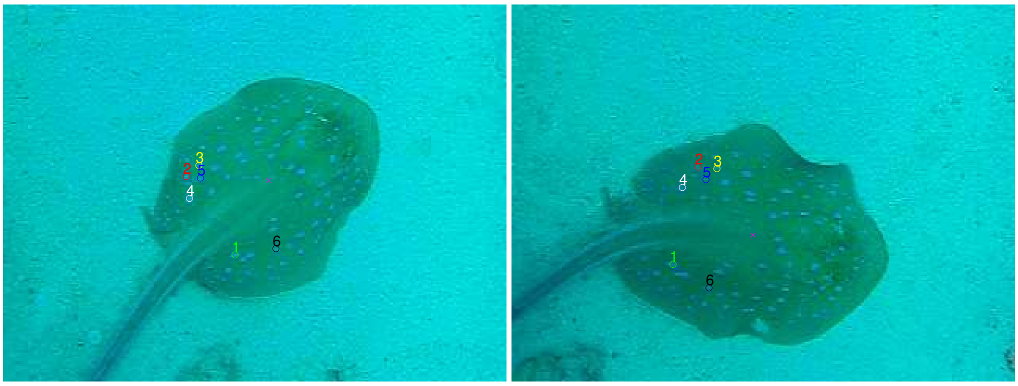
Figure 10: Registration of Frame 1 and Frame 6 of the bluespotted ribbontail rays with the GI-Algorithm.



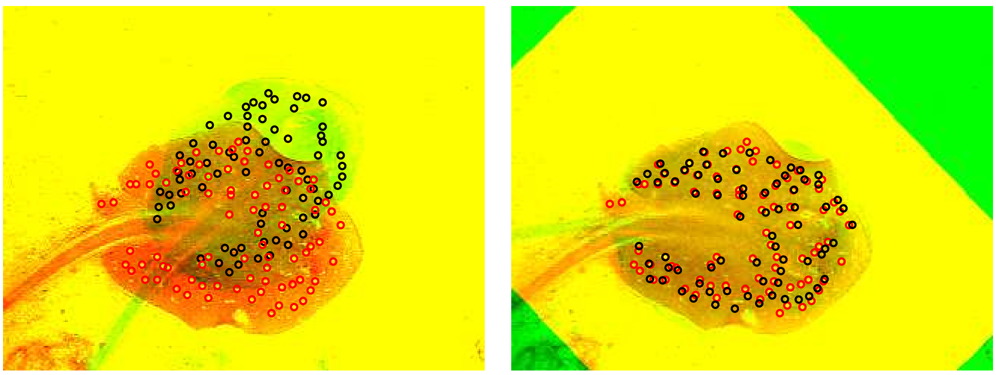
(a) Frame 6 with labeled Points (b) Frame 12 with labeled Points



(c) Frame 6 with Points from Frame 12 before (left) and after (right) Registration

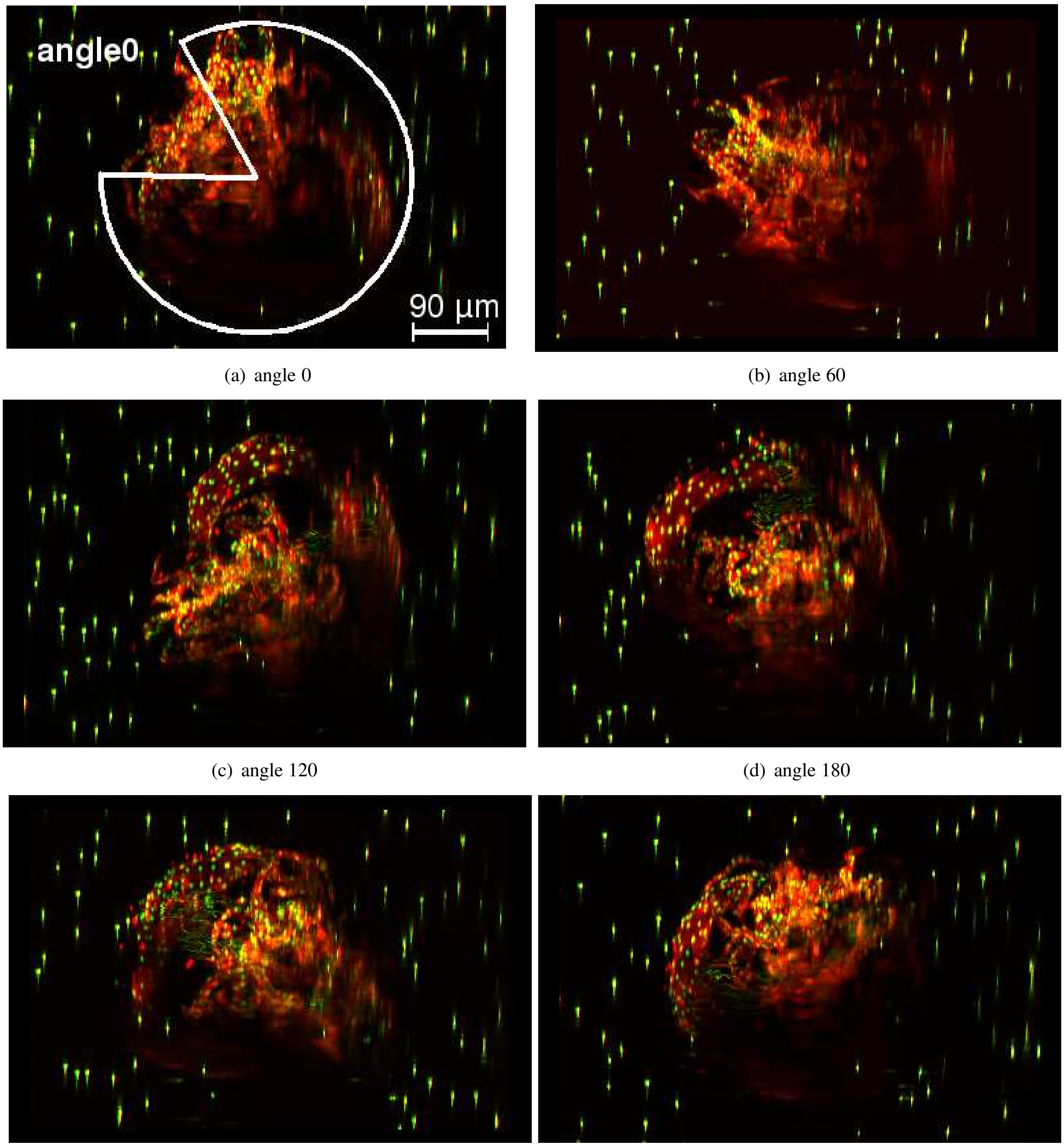


(d) Correspondences from Frame 6 and Frame 12 found with the GI-Algorithm are circled in the same color.



(e) Overlay from Frame 6 and Frame 12 before (left) and after (right) applying the GI-Algorithm.

Figure 11: Registration of Frame 6 and Frame 12 of the bluespotted ribbontail rays with the GI-Algorithm.



(a) angle 0

(b) angle 60

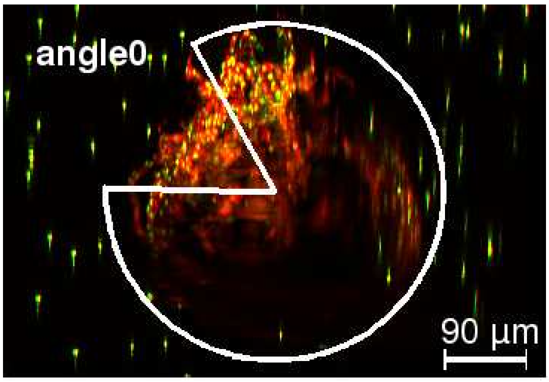
(c) angle 120

(d) angle 180

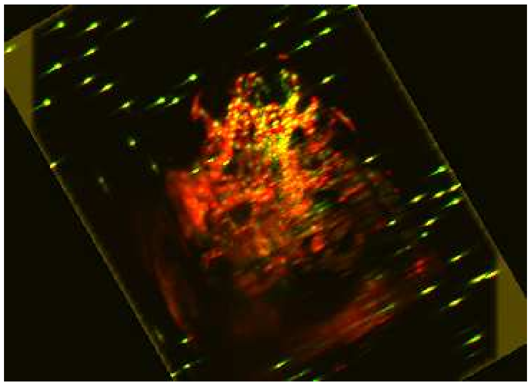
(e) angle 240

(f) angle 300

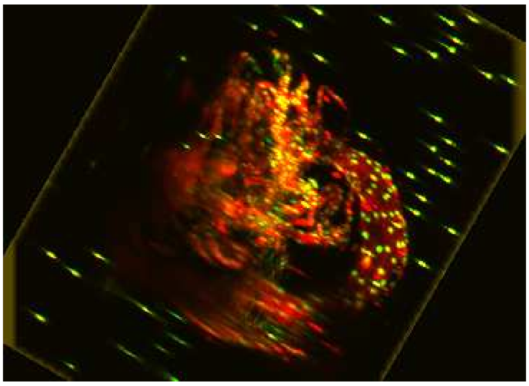
Figure 12: **Original data.** Six different views of the zebrafish recorded with SPIM are presented before registration. The maximum intensity projection of the stacks is displayed in the xz-view.



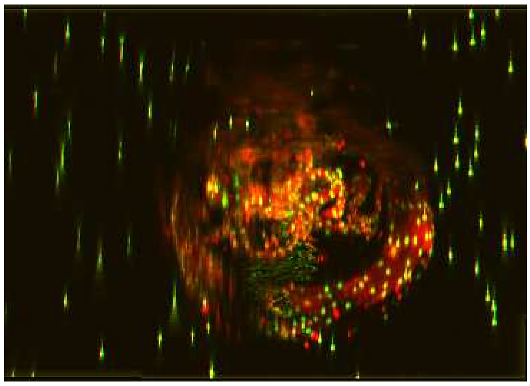
(a) angle 0



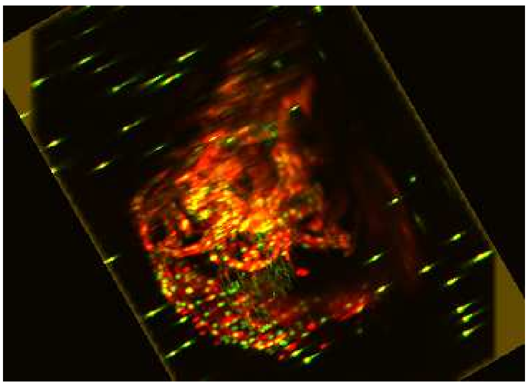
(b) angle 60



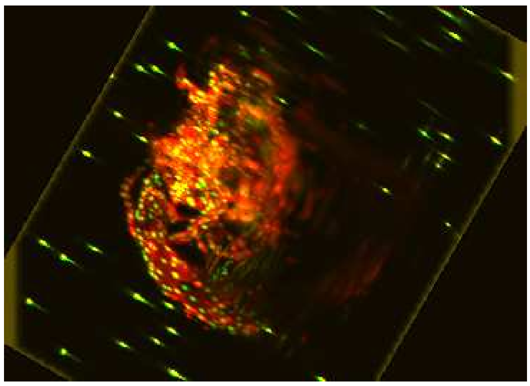
(c) angle 120



(d) angle 180

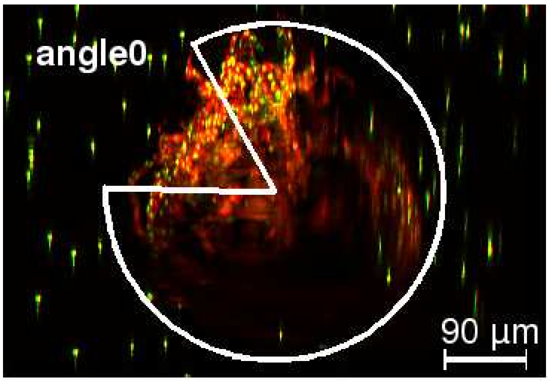


(e) angle 240

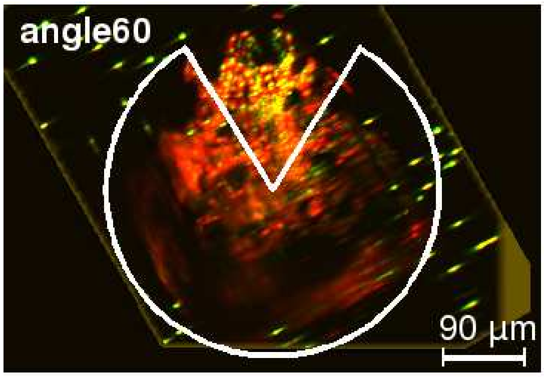


(f) angle 300

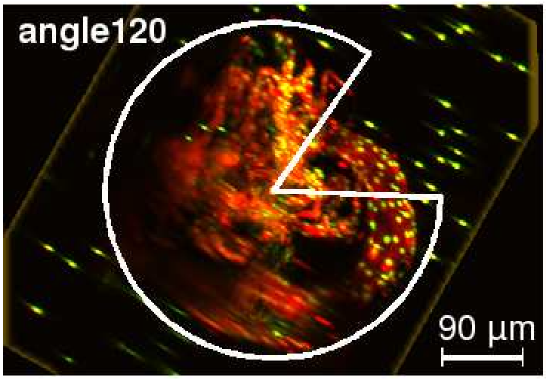
Figure 13: **Pre-registration step.** Six different views of the zebrafish recorded with SPIM are presented after rotation around the angle provided with the image around the y-axis. The maximum intensity projection of the stacks is displayed in the xz-view.



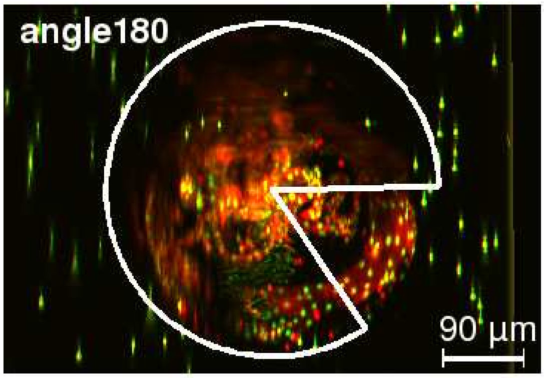
(a) angle 0



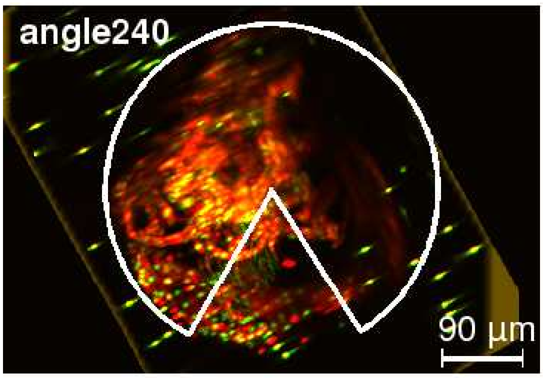
(b) angle 60



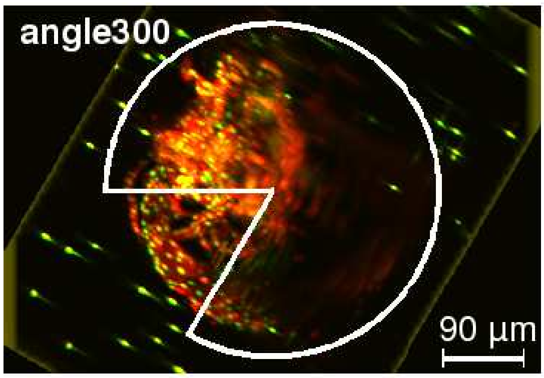
(c) angle 120



(d) angle 180

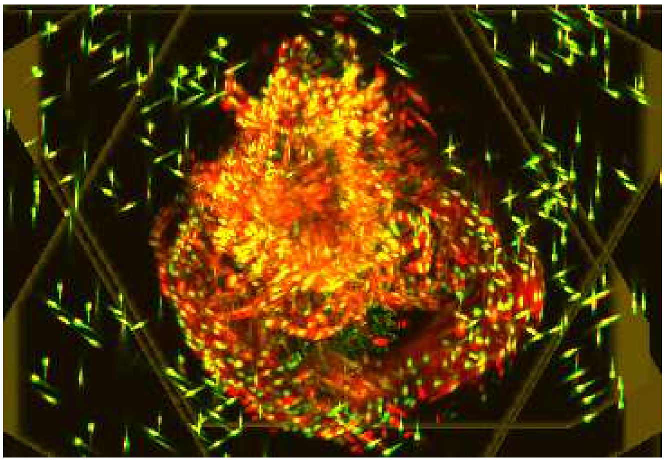


(e) angle 240

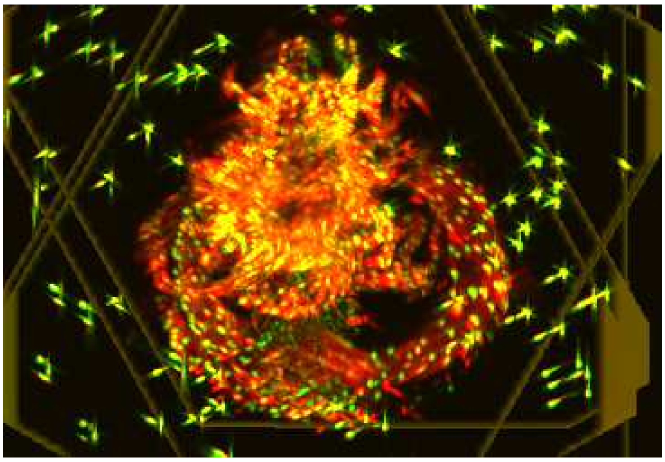


(f) angle 300

Figure 14: **Fine registration with the GI-Algorithm.** Six different views of the zebrafish recorded with SPIM are registered using the position of the beads with the GI-Algorithm. The maximum intensity projection of the stacks is displayed in the xz-view.

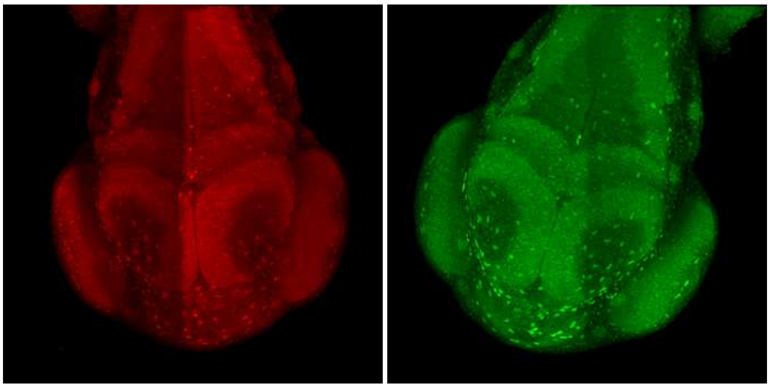


(a) Result of selecting the maximal value of all six pre-registered views (only rotation)



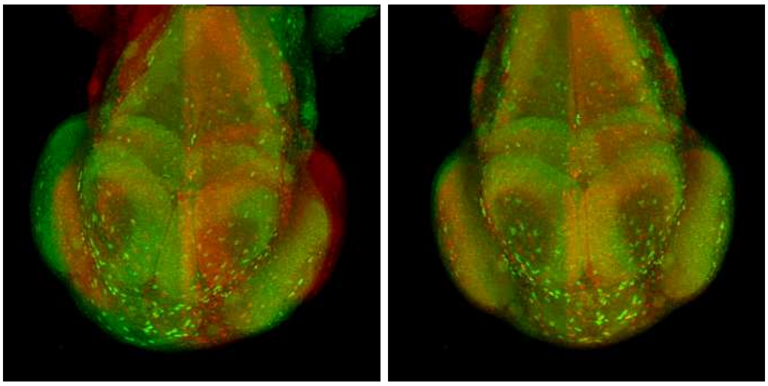
(b) Result of selecting the maximal value of all six registered views (GI-Algorithm on pre-registered)

Figure 15: Overlay of the six angles after pre-registration (a) and after applying the GI-Algorithm once to the pre-registered images (b).

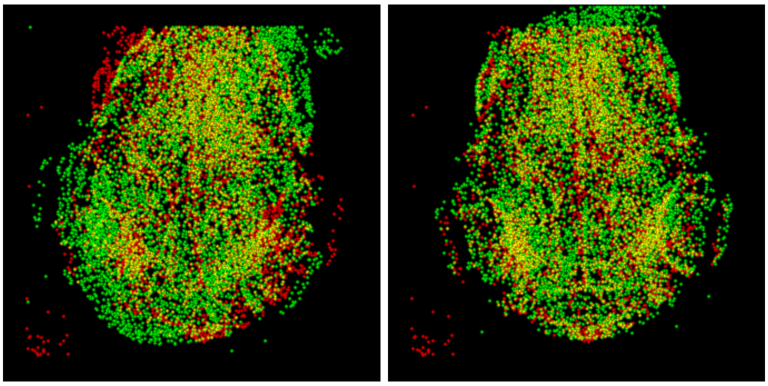


(a) Zebrafish e05, 5010 interest points detected

(b) Zebrafish e07, 8050 interest points detected



(c) Overlay Zebrafish e05 and e07 before (left) and after (right) registration



(d) Overlay of the interest points from e05 and e07 before (left) and after (right) registration

Figure 16: On the top the original images before (left) and after registration (right) are displayed as an overlay. On the bottom the detected interest points are overlayed. The three dimensional images are displayed as a maximum intensity projections.