Averaging Centerlines: Mean Shift on Paths

Theo van Walsum[1], Michiel Schaap[1], Coert T. Metz[1], Alina G. van der Giessen[2], and Wiro J. Niessen[1]

[1] Biomedical Imaging Group Rotterdam
Departments of Radiology and Medical Informatics
[2] Hemodynamics Laboratory
Department of Biomedical Engineering
Erasmus MC – University Medical Center Rotterdam
t.vanwalsum,michiel.schaap,c.metz,
a.g.vandergiessen,w.niessen@erasmusmc.nl

Abstract. Generation of a reference standard from multiple manually annotated datasets is a non-trivial problem. This paper discusses the weighted averaging of 3D open curves, which we used to generate a reference standard for vessel tracking data. We show how weighted averaging can be implemented by applying the Mean Shift algorithm to paths, and discuss the details of our implementation. Our approach can handle cases where the observer centerlines take different branches in a natural way. The method has been evaluated on synthetic data, and has been used to generate reference centerlines for evaluation of vessel tracking algorithms.

1 Introduction

It is commonly understood that thorough evaluation of methodologies and algorithms is essential for progress in the field of medical imaging. Such evaluations require a set of test data, a reference standard (we prefer to use this phrase if there is no ground truth available), and a set of measures to quantify the results for evaluation.

The authors are involved in the coronary artery tracking challenge (`http://cat08.bigr.nl`), and therefore these issues were to be addressed in this context. As no ground truth central lumen lines are available for our clinical datasets, three observers manually annotated the central lumen lines: from a fixed starting position, the centerline of the main vessel should be annotated as distal as possible. However, the observers did not always take the same decision at bifurcations and also the handling of vessel pathology such as stenoses or regions with non-circular lumen cross sections was not always consistent. In our efforts to come to a reference standard from a set of three manually drawn centerlines we therefore soon came to the conclusion that plain averaging does not work. The most important requirement of the averaging process is that it should give reference

D. Metaxas et al. (Eds.): MICCAI 2008, Part I, LNCS 5241, pp. 900–907, 2008.

centerlines that are nearly always inside the vessel lumen, i.e. the reference centerline should follow the "majority votes" centerline. The main contribution of this paper is a method to "average" these centerlines into a reference standard.

Many publications have appeared that implement a way to compare open curves. If correspondence between centerlines is known, and bifurcations are not a problem, then approaches to estimate the mean of a set of curves can be applied [1]. To the best of our knowledge, no work has been published that addresses how to generate a reference centerline from several manually annotated lines, if there is no explicit correspondence between the lines.

Warfield et al. [2] address the related issue of generating a ground truth segmentation and propose a method called STAPLE. They use a Maximum Likelihood Expectation Maximization algorithm to estimate the performance parameters of the observers (and possibly software algorithms) given a set of segmentations, with the reference standard segmentation modelled as hidden data. This approach is appealing, as it not only determines a reference standard from multiple segmentations, but also addresses the issue of sensitivity and specificity of the observers and automated approaches.

More recently, Jomier et al. [3] showed how STAPLE could be used for evaluating centerlines in vessel segmentation. They propose to voxelize the open curves, dilate them and then apply STAPLE to the resulting segmentations, which yields a probability map of the ground truth segmentation. Their work is most related to ours.

The remainder of the paper is organized as follows. In Sect. 2 and 3, we describe our method and the implementation, in Sect. 4 our experiments are described, followed by Results and Discussion in Sect. 5 and 6 and Conclusions in Sect. 7.

2 Averaging Via Mean Shift

We propose to perform a weighted averaging to determine the reference centerline, where the weights depend on the distance from the reference. This suggests that averaging can be performed via the Mean Shift algorithm [4,5], which is an algorithm that iteratively shifts a data point along the gradient of a density that is determined by a set of data points and kernel functions, until the gradient vanishes, and the point has arrived at a mode (maximum) of the density.

More formally, the Mean Shift algorithm with Gaussian kernels G_σ and for a set of m datapoints can be described by the following iterative procedure (see e.g. Carreira-Perpinan [5]):

$$\mathrm{x}^{\tau+1} = \sum_{m=1}^{M} \frac{G_\sigma(|\mathrm{x}^\tau - \mu_{\mathbf{m}}|)}{\sum_{m'=1}^{M} G_\sigma(|\mathrm{x}^\tau - \mu_{\mathbf{m}'}|)} \, \mu_{\mathbf{m}} \ , \tag{1}$$

where μ_{m} represents the m^{th} data point. After convergence of the algorithm, the mean shifted position x is the weighted average of the data points, where the weights are determined by the distance to the weighted average and the kernel function used.

This Mean Shift algorithm is a well-known algorithm, that has, among others, been used to find the modes of clusters in feature space analysis in image processing [4]. We adapt Eq. 1 for our application to paths (we will use path instead of centerline to underline that our method is applicable to any open 3D curve) in the following way.

Let $p_i(t)$ with $0 \leq i < m$ (m number of paths) and $t \in [0,1]$ be the path annotated by the i^{th} observer and parametrized by parameter t, and $r_i(t)$, $t \in [0,1]$ be the mean shifted path to be determined. Furthermore, let $s_i(t)$ be a bijection from $[0,1]$ to $[0,1]$ that defines the correspondence between r_i and p_i, i.e. $r_i(t)$ corresponds to $p_i \circ s_i\,(t)$. Thus s_i relates positions along r_i to positions along p_i that will be used in the Mean Shift algorithm. The mean shift r_i of path p_i is then given by the following equation:

$$r_i^{\tau+1} = \sum_j \frac{G_\sigma\left(|r_i^\tau - p_j \circ s_j|\right)}{\sum_k G_\sigma\left(|r_i^\tau - p_k \circ s_k|\right)}\, p_j \circ s_j \;\;, \tag{2}$$

where the corresondence function s_i may change after each iteration, and $r_i^0 = p_i$. This equations states that each point along r_i is shifted to the weighted average of the corresponding points on all paths p_j.

We apply this adapted Mean Shift algorithm to each of the manually drawn paths, using the manually annotated paths as data points. This results in a set of mean shifted paths, of which some will coincide for some part (following the same mode), and some may diverge at certain locations. To obtain the final reference path, the largest common part of the shifted paths is determined in a post processing step. Implementation details are discussed in the next section.

3 Implementation

3.1 Correspondence

Equation 2 contains correspondence functions s_i, which relate points from the path being shifted to the observer paths and also vice versa, as s_i is a bijection. Point correspondence must be known, as it determines which points of the paths p_i are involved in the mean shift of a position on r_i.

In our implementation, we use a discretized version S_i of the correspondence function s_i. Equidistant resampling of the input paths is performed before determining the correspondence. Next, the correspondence is determined by finding the minimum of the sum of the Euclidean lengths of all point-point connections that are connecting two paths over all valid correspondences.

Let O be a path, represented by an ordered set of n points $\{o_i\}$, $i \in [1,n]$, and let Q be a second path, represented of an ordered set of m points $\{q_j\}$, $j \in [1,m]$. We define a correspondence C for paths O and Q as the ordered set of connections $\{c_k\}$, $k \in [1, n+m-1]$ where c_k is a tuple $[i,j]$ that represents a connection from o_i to q_j, and we define a valid correspondence as a correspondence satisfying the following conditions:

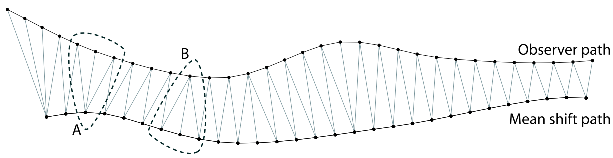

Fig. 1. Correspondence between two lines, as determined by the Dijkstra algorithm. A and B show regions where one point on one path corresponds to multiple points on the other path.

– The first connection c_1 connects the start points: $c_1 = [1, 1]$.
– The last connection c_{n+m-1} connects the end points: $c_{n+m-1} = [n, m]$.
– If connection c_k $(k < n + m - 1)$ equals $[i, j]$ then connection c_{k+1} equals either $[i + 1, j]$ or $[i, j + 1]$.

These conditions guarantee that each point of O is connected to at least one point of Q and vice versa. Note that this is not a bijection, as multiple points on O may be connected to Q, and vice versa. A Dijkstra graph search algorithm [6] on a Cartesian grid with connection lengths as edge costs is used to determine the global minimum Euclidean length correspondence. See Fig. 1 for an example.

3.2 Mean Shift on Paths

Equation 2 shows how to determine the reference path given the positions and weighting factors, by taking the weighted average over all points involved. The correspondence as defined in Sect. 3.1 is not a real bijection, see Fig. 1: a point on the mean shifted path may be connected to several points on an observer path (case A), and vice versa (case B). To account for multiple observer points connected to one point on the mean shifted path (A), the weights are normalized with a factor of $\frac{1}{n_j}$, where n_j is the number of nodes on the observer path that is connected to a point of the mean shifted path. This means that the total weight of a path is not affected by the number of points connected to a point on the mean shifted path. In case one point on the observer path is connected to several points on the mean shifted path (B), that point is used in the mean shifts for each of the points of the mean shifted path it is connected to.

Equation 2 is thus implemented in our discretized Mean Shift algorithm as follows, with R_i, P_i and S_i the discretized versions of respectively the mean shifted path r_i, the observer path p_i and the correspondence s_i:

$$R_i^{\tau+1} = \sum_j \sum_l^{n_j} \frac{\frac{1}{n_j} G_\sigma \left(|R_i^\tau - P_j \circ S_{j,l}| \right)}{\sum_k \sum_{l'}^{n_k} \frac{1}{n_k} G_\sigma \left(|R_k^\tau - P_k \circ S_{k,l'}| \right)} P_i \circ S_{j,l} \ , \tag{3}$$

where n_j is the number of connections of path j to the position of R_i that is being evaluated, and $S_{j,l}$ corresponds to the l^{th} connection from the position on R_i to P_i and $\frac{1}{n_j}$ is the weighting factor for point multiplicity. This equation is evaluated for each of the points along R_i. The correspondence S_i is redetermined after each mean shift iteration, i.e. after a shift of all points on R_i.

3.3 Post Processing

The implementation of the Mean Shift algorithm on paths returns three shifted paths: each observer path is shifted to a mode of all input paths. In a subsequent post processing step, these paths are merged into one resulting path, representing the “major mode” of the paths. We perform this clustering task in a straightforward way. The local distance between each tuple of mean shifted paths is determined via correspondences established as described in Sect. 3.1. After thresholding this distance, the length of the common part that two paths share can be determined. The tuple with maximum common path length is subsequently chosen, and the average of the two paths of this tuple over the part of the paths that they share is determined.

4 Experiments

The method described in the previous section has been implemented and evaluated. Evaluation was done visually on synthetic test cases, to inspect the behaviour of the algorithms in case of paths ending at different positions, single bifurcations and double bifurcations. In these experiments, the bandwidth σ was set to 2 mm (the path length was around 80 mm, initial sampling distance as shown in the images was 1 mm). The effect of varying the bandwidth was also evaluated, with σ varying from 1 mm to 16 mm, where σ was doubled in each next experiment. These bandwidth experiments are shown together with a plain averaged path, i.e. a path that would be obtained if we would average without weighting factors. This averaging is performed by iteratively averaging tuples of observer paths over all connections that have been determined by the same Dijkstra algorithm, i.e. the plain average path is determined by iterating the following over all i: $p_i^{\tau+1} = \frac{1}{2}\left(p_i^\tau + p_{i+1}^\tau\right)$ where τ is the iteration number, and $i+1$ wraps at 3 in our case. This iterative approach converges to the average of the three paths.

The method was also used to generate the reference standards from manually annotated centerlines for evaluation of vessel tracking algorithms. Currently, four coronaries have been annotated in 24 coronary CTA datasets by three observers. The method has been tested on all 96 triplets of paths, and the reference paths generated in this way have been checked visually by displaying them in Curved Multiplanar Reformatted images and locally orthogonal views (oriented with the tangential of the centerline), together with interobserver variations. The observer centerlines contained centerlines that take different branches at bifurcations or that ended prematurely and also centerlines that have local deviations. For the mean shift of the centerlines of these coronary arteries we used a σ of 2 mm, which is approximately the average radius of the coronary vessels that we are tracking in this application. This implies that paths running apart much more than 2 mm will not converge to the same mode, which is exactly what we want. The mean shifted paths were determined at a sampling distance of 0.2 mm, and the input observer paths were resampled to a sampling distance of 0.03 mm.

5 Results

Results of the method for the experiments with synthetic data are shown in Fig. 2a–h. In all cases, the reference path is not much affected by a path that chooses another vessel, although a small bump is sometimes noticeable, the size of which depends on the bandwidth of the algorithm. Also paths that end less distal than others do not affect the reference path.

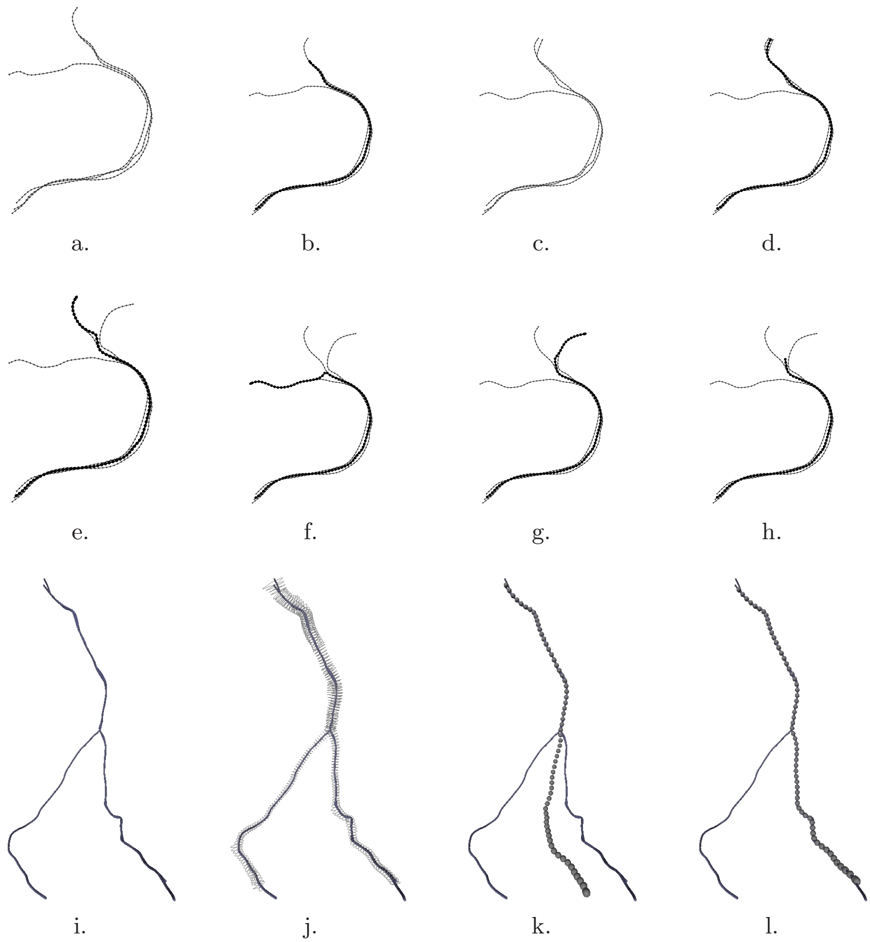

Fig. 2. Examples of the Mean Shift algorithm: a–h to 2D synthetic data, i–l to the 3D paths data. a) three paths, where not all paths run to the distal end; b) mean shift result of a; c) bifurcation in observer paths; d) mean shift result of c; e), f) and g) mean shift result for each of the three observer paths separately; h) final result after clustering the paths; i) centerlines; j) vessels; k) average path; l) mean shift path.

The effect of varying σ on the final reference path is small if changes in σ are small. When σ is large enough, all vessels will be averaged, even if they follow different branches, and the resulting path will be outside all vessels. A slight bump can be observed at the bifurcations with low values for σ, as the bifurcating path initially is near enough to also contribute to the major mode.

All the CTA reference paths have been visually inspected, and all could be used directly for the evaluation of tracking algorithms, none of them needed additional manual editing. Examples are shown in the bottom row of Fig. 2.

6 Discussion

We have shown how the Mean Shift algorithm can be used to determine a weighted average of three manually annotated open curves. Correspondence between two curves was determined by the set of connections with minimal total Euclidean length. Our application involves averaging of three 3D (spatial) curves, and the algorithms discussed have been developed for this specific case. Both the spatial dimensionality and the number of input curves, however, are not essential to the algorithm: our approach can easily be applied to curves in higher- or lower-dimensional space, and also on sets of more than three curves.

The choice of a kernel and bandwidth determines the final result of the algorithm. We chose our bandwidth according to the expected variance of the data: all paths should be in the same vessel, and thus the bandwidth was similar to the vessel radius. We choose the Gaussian kernels because of their assumed better behaviour. The infinite support of the Gaussian kernel, which makes them computationally expensive in other cases [4], is not an issue in our case.

The resulting paths in our implementation may have slight bumps or kinks near paths that diverge. Depending on the value of the bandwidth σ, the diverging path will to some extent drag the major mode away from the course of the other paths. In the future, we may try to resolve this by following an iterative approach: based on the results of an initial application of our method, we can detect locations where paths are diverging, and reduce the influence of single diverging paths with a suitable chosen weighting factor, or vary the bandwidth along paths.

We do not yet fully exploit all the possibilities of our approach. Instead of picking the largest overlap between the shifted paths, a more advanced analysis of the convergence modes could be performed. One could imagine that, in case of a substantial number of observers, several important modes can be detected, even consisting of varying sets of observer paths, all of which should be part of the final reference. In the future, we intend to investigate this issue.

The work most closely related to ours is the STAPLE approach of Jomier et al. [3]. Our approach differs from theirs in several aspects. First, the STAPLE approach requires a voxelization of the input centerlines, whereas our approach is subvoxel accurate. Second, the STAPLE approach yields an explicit probability map that can be used for evaluation, whereas our approach gives a reference centerline, that can be used to derive quantitative measures on the position of other centerlines. The probability map could be used to derive a reference path, but this is not trivial, and it is not discussed in the work of Jomier et al. Third, the STAPLE approach gives a sensitivity and specificity value for the complete curve, whereas we use a weighting factor that varies along the curve. The advantage of the latter approach is that in proximal regions, where all curves track the same vessel, all curves are incorporated in calculating the average with approximately the same weights, even if one of the curves would track another part incorrect, e.g. by tracking the wrong branch at a distal bifurcation, and that weights are decreased locally in case of local errors.

In the future, we want to extend the approach to sets of more than three vessels. Also, we want to investigate how the analysis of the resulting modes can contribute to a better definition of the reference standard. Furthermore, a similar approach could be pursued to determine reference standards for other types of data.

7 Conclusion

We have shown how the Mean Shift algorithm can be used to determine the average centerline from a set of centerlines annotated by observers. The method can handle bifurcating centerlines and centerlines that stop at different distances along the vessel in a natural way. The technique has been applied successfully to 96 manually annotated coronary artery centerlines, and will be used in the evaluation of vessel tracking algorithms.

Acknowledgment

This work was sponsored by the Dutch Foundation for Scientific Research, NWO, STW.

References

1. Rice, J.A., Silverman, B.W.: Estimating the Mean and Covariance Structure Nonparametrically when the Data are Curves. Journal of the Royal Statistical Society. Series B (Methodological) 53(1), 233–243 (1991)
2. Warfield, S.K., Zou, K.H., Wells, W.M.: Simultaneous truth and performance level estimation (STAPLE): an algorithm for the validation of image segmentation. IEEE Transactions on Medical Imaging 23(7), 903–921 (2004)
3. Jomier, J., LeDigarcher, V., Aylward, S.R.: Comparison of vessel segmentations using staple. In: Duncan, J.S., Gerig, G. (eds.) MICCAI 2005. LNCS, vol. 3749, pp. 523–530. Springer, Heidelberg (2005)
4. Comaniciu, D., Meer, P.: Mean Shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5), 603–619 (2002)
5. Carreira-Perpiñán, M.: Gaussian Mean-Shift is an EM algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(5), 767–776 (2007)
6. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik 1, 269–271 (1959)