Guide-Wire Extraction through Perceptual Organization of Local Segments in Fluoroscopic Images*

Nicolas Honnorat[1,2,3], Régis Vaillant[3], and Nikos Paragios[1,2]

[1] Laboratoire MAS, Ecole Centrale Paris, Châtenay-Malabry, France
[2] Equipe GALEN, INRIA Saclay-Ile-de-France, Orsay, France
[3] General Electric Healthcare, Buc, France

Abstract. Segmentation of surgical devices in fluoroscopic images and in particular of guide-wires is a valuable element during surgery. In cardiac angioplasty, the problem is particularly challenging due to the following reasons: (i) low signal to noise ratio, (ii) the use of 2D images that accumulate information from the whole volume, and (iii) the similarity between the structure of interest and adjacent anatomical structures. In this paper we propose a novel approach to address these challenges, that combines efficiently low-level detection using machine learning techniques, local unsupervised clustering detections and finally high-level perceptual organization of these segments towards its complete reconstruction. The latter handles miss-detections and is based on a local search algorithm. Very promising results were obtained.

1 Introduction

The detection and the segmentation of the guidewires (GW) used in cardiac angioplasty is a challenging problem in biomedical image analysis. It is often addressed into two steps: first, interest points are detected and then linked (or grouped) together into a curvilinear structure corresponding to the GW. Both steps inherit severe technical challenges.

Detection is required due to the low signal to noise ratio. The aim is either to improve contrast between the guidewire and the background/other related anatomical structures or detect feature points corresponding to it. The first scenario is often addressed using low level operators, like dedicated filtering [1] and coherence enhancing diffusion [2]. State of the art for detection includes dedicated edge-detection methods [1], method based on the Hessian eigenvalues such in [2] and such as the Vesselness measure [3,4], steerable filters [5], and phase congruency [6]. The second class of methods aims to address detection directly using either a voting schema [7], or machine learning methods [8]. The central idea is to use patterns of appearance corresponding to vessels and learn a classifier that is able to separate them from the background. Pose and scale parameters are

* This work was supported by ANRT (grant 1062/2008) and GE Healthcare. The authors thank N. Komodakis for providing the clustering method.

T. Jiang et al. (Eds.): MICCAI 2010, Part III, LNCS 6363, pp. 440–448, 2010.

the most critical aspects to be handled in this process. Augmenting the training set towards encoding all possible variations of appearance is not feasible since it deteriorates the performance. However, compared to filtering methods this concept is quite promising.

Chaining of detections towards complete recovery of the guidewire is often considered afterwards [1,7,4]. The use of snake-splines on the feature images has also been investigated [2,6]. These methods are very sensitive to the presence of outliers as well as miss-detections. Such shortcomings can be addressed through the use of hierarchical grouping guided by a classification [8]. In all cases, sensitivity to the presence of outliers as well as miss-detections are challenging issues to be addressed.

In this paper we introduce a novel approach combining detection using boosting [9], with a mid-level grouping scheme based on clustering and a complete reconstruction through the minimization of a global criterion that encodes geometric and detection consistencies - through local search based on an inlier/outlier permutation model. Contrary to the other methods which can fail due to noise or low contrast [2,6], or do not recover undetected parts [1] our approach has been designed to be as robust as possible and to recover the GW as global optimum.

The most closely related work with our approach can be found in [8,10]. The first approach is purely learning based, is not invariant to the guide-wire pose parameters and cannot handle well miss-detections. The second approach shares the feature detection concept (the considered features are less efficient than the ones considered in this paper). However, [10] does not perform ordering, and is only able to provide a local grouping of segments into parts using linear programing. The method is unable to deal well enough with miss-detections as well as outliers because the final result is not necessarily perceptually meaningful.

The remainder of this paper is organized as follows: in section 2 we discuss the robust extraction of line segments that we perform and we present our approach for organizing them in section 3. The next section presents the experimental validation. Discussion concludes the paper.

2 Robust Local Segments Extraction

Our algorithm consists of two steps: (i) a low-level detection of GW pixels, that are then grouped towards extracting line segments (ii) an ordering and removal of erroneous segments. The first task is addressed using boosting and unsupervised clustering, and the second through local permutation search.

2.1 Low-Level Detection

Boosting [9] refers to a powerful classification method that combines weak classifiers towards the creation of a strong one. This classification process is guided from the miss-detection/miss-classification error. Samples are weighted according to the classification error and these weights are continuously updated. Since in our data set, one expects important discrepancy between positive and negative samples (GW pixels represent less than 0.15% of whole image) as well as important presence of outliers, we adopt the variant called Gentle AdaBoost [11] and we replaced its re-weighting formula driving the choice of the next weak classifier with the following (where w_j is the weight of a data x_j of label $y_j \in \{-1, +1\}$ and h^m is the weak classifier chosen during the current AdaBoost step) in order to make it asymmetric like [12]:

$$w_j \leftarrow w_j e^{-y_j h^m(x_j)} e^{y_j (asymmetry)}$$

When training classifiers, we have considered around 500 000 samples, almost 87% of them being negative ones, and chosen an asymmetry of 0.1. The input to the classification process was features derived from the image after appropriate filtering and in particular with separable steerable filters [5] that are optimal in the sense of Canny criteria.

In addition to the steerable ridge detectors of fourth order of [5] for $\mu = 0.25$ and scales $\sigma = \{0.8, 1.2, 1.6, 2.0, 3.0\}$ and to the edge detector of third order of scale 5.0, we introduced three features specific to our problem: (i) Difference of Gaussian (DoG) computed with the raw, the log and the next images, (ii) variances of these features around the pixel considered (in a square of size 7), both for the couples of scales: $\{(0.8, 1.2)(1.4, 1.8)(2.0, 3.0)(3.0, 4.0)(4.0, 5.0)\}$; and (iii) variances computed on the 'subtraction image' (the current image minus the next one) in squares of size 5 and 9.

Log images have indeed a physical meaning and allow to better discriminate the structures from their background, but their level of noise has been locally modified, which produces numerous false detections. Considering both images allows to combine both advantages to avoid more false detections , and considering the next frame allows to better reject the confusing structures, because they are most of the time static (and variances provides a second order description).

Clustering these points towards extracting line segments both allows to reject remaining false detections due to noise (because most of them are isolated) and provides primitives we just need to link together (with a proper handling of outliers) to delineate the GW. The fact that clustering provides a global optimum guarantees far more robustness with respect to noise. The main challenge is that neither the number of segments/clusters nor the membership function for every detected point are known in advance.

2.2 Grouping

Let us consider the N candidate pixels $p_i = (x_i, y_i)$.

Clustering of these pixels can be formulated as follows: find a set of cluster centers $c_j = (cx_j, cy_j)$ (the unknown cardinality of this set is denoted as $|C|$) and an individual membership function $\mathcal{L}$ (labels in $\{1, \dots |C|\}$) such that all detected points p_i with the same label form a local line segment with minimal dispersion. We can express this problem within an optimization framework as follows:

$$min_{\{c_j, |C|, \mathcal{L}\}} \sum_{j=1}^{|C|} f(c_j) + \beta \sum_{i=1}^{N} \delta(\mathcal{L}(i) - j) g(c_j, p_i)$$

where f is the penalty for a cluster center, $\mathcal{L}(i)$ is the label indicating the cluster for the point i and $g(c_j, p_i)$ is the cost of attributing p_i to the cluster c_j. $\delta(\mathcal{L}(i) - j)$

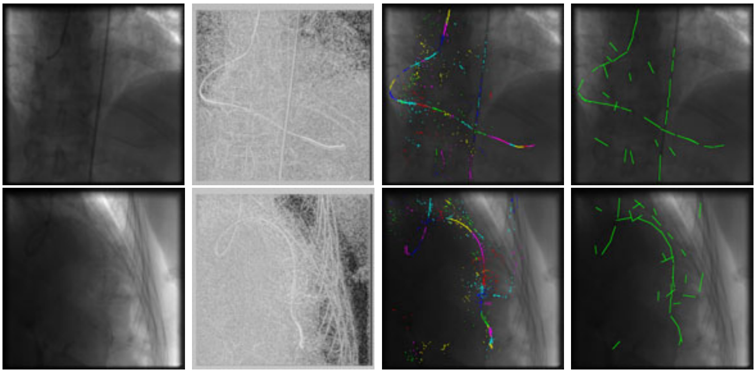

Fig. 1. 1) input 2) classification scores 3) detected points and clusters 4) detected segments (colors help to differenciate clusters in 3)

measures only the cluster cost with respect to the optimal attribution label. The definition of the functions f and g, as well as the optimization of the objective function are the main challenges to be addressed.

We consider f to be a constant (in our experiments, around 90000.0) and no-particular preference is given among the set of detected points. The definition of g is more challenging and for simplicity and clarity, let us introduce the following geometric elements: (i) (c_j, p_i) form a line segment l_{cp}, (ii) c_j and the axis of orientation given by the optimal steerable filter computed in c_j form a line l_c, and (iii) p_i and the optimal steerable filter axis computed in p_i also form a line l_p. In the ideal case, if point p_i is a member of c_j, then the three line segments will coincide.

In order to quantify this hypothesis, we consider two criteria: (i) The actual geometric distance between the two points, $d(c_i, p_j)$ since we would expect capturing local straight line segments of curvilinear structures. (ii) The distances $d_1(c_i, p_j) = d(c_i, p_j) \min(tan(\theta_1), t_{max})$ and $d_2(c_i, p_j) = d(c_i, p_j) \min(tan(\theta_2), t_{max})$ where θ_1 and θ_2 are the angles between the lines being formed from the two points and the ones given by steerable filters orientations, and t_{max} allows to avoid infinite distances when directions are orthogonal. These distances are combined to a single metric as follows (in our experiments, we took $\alpha = 0.3$):

$$g(c_i, p_j) = \alpha d^2(c_i, p_j) + (1 - \alpha)(d_1^2(c_i, p_j) + d_2^2(c_i, p_j))$$

The optimization of this objective function is done using [13], that finds the number of clusters and the pixels memberships simultaneously. Some clustering results are shown in: [Fig. (1)]. In order to extract the local line segments, we used a variant of RANSAC called MSAC [14] that is less sensitive to the geometric error introduced from the presence of outliers. There remains only to link these segments properly to delineate the GW - with handling outliers. Once again, we performed this task through a global optimization process, which guarantees robustness with respect to local minima created by noise.

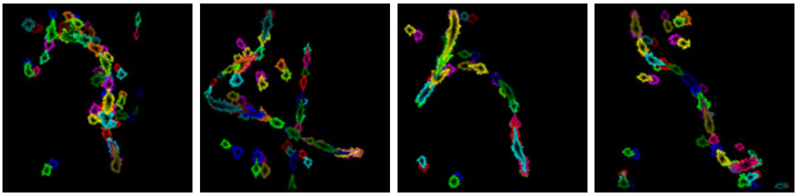

Fig. 2. Minimal geodesic costs determined using the fast marching algorithm. Colors help to differentiate and visualize costs (the brighter the color the higher the cost).

3 Perceptual Organization of Segments

Once we have obtained the line segments, we need to find which ones have to be linked and in which order for delineating the GW.

Let $|S|$ be the number of line segments and let us consider for every segment s_i a label $\{\theta_i, \omega_i, \phi_i\} \in \{1 \cdots |S|\} \times \{0, 1\} \times \{0, 1\}$. θ_i denotes the rank of the segment s_i, ω_i accounts for the orientation of this segment (the conventional one assumes that the first extremity corresponds to the small horizontal coordinate) with the 0 label corresponding to the conventional one and the 1 to the extremities being reversed. Introducing such a variable is a necessity since it modifies the cost of linking two successive segments. Last, let ϕ_i be the state, that is either part of the guide-wire (1) or an outlier (0). Given such notation, one can now proceed to the definition of the local cost $C(a, b)$ corresponding to the price to be paid towards linking two segments tips a, b.

3.1 Local Ordering Cost

In order to define the cost $C(a, b)$ we consider a linear combination of the scale invariant Elastica criterion defined in [15] and a linking cost equal to the minus log-likelihood of the most likely path linking a and b. The idea is that a and b should both be linked by a path of low curvature (i.e. low Elastica) and linked by a likely path. If we assume that pixels along the path are independent, then the last criterion can be computed as a sum over the pixels of the better path, and can therefore be computed by fast marching. Denoting with $h(x)$ the detection score at location x and using a sigmoid function to compute pixel likelihood, we finally used the following cost map $c(x)$ as fast marching input:

$$c(x) = -h(x) + log\left(e^{h(x)} + e^{-h(x)}\right)$$

Linking costs above an arbitrary threshold were not computed and approximated with the product between the Euclidean distance between the tips and the maximal linking cost normalized according to the euclidian distance between tips found by fast marching. [Fig. (2)] shows several cost maps computed with the algorithm presented in [16].

3.2 Global Ordering

Finding the optimal sequence of segments can be considered as an optimization problem. In simple words one would like to minimize:

$$\min_{\theta_i,\omega_i,\phi_i} \sum_{m=1}^{|S|}\sum_{n=1}^{|S|} \phi_m\phi_n\delta(\theta_n-\theta_m-1)C_{\omega_m,\omega_n}(m,n)+\beta\sum_{m=1}^{|S|}(1-\phi_m)$$

where product $\phi_m\phi_n$ guarantees that both s_m, s_n segments are retained and the term $\delta(\theta_n-\theta_m-1)$ guarantees that the segment s_n succeeds s_m. The second term penalizes the attribution of the outlier label to a segment. Last, the cost for connecting two (conventionally oriented) segments $s_m=(a_m,b_m)$ and $s_n=(a_n,b_n)$, denoted with $C_{\omega_m,\omega_n}(m,n)$, is derived from the one earlier presented:

$$C_{\omega_m,\omega_n}(m,n) = \begin{cases} C(b_m,a_n) & if \;\; \omega_m=0 \; and \; \omega_n=0 \\ C(b_m,b_n) & if \;\; \omega_m=0 \; and \; \omega_n=1 \\ C(a_m,a_n) & if \;\; \omega_m=1 \; and \; \omega_n=0 \\ C(a_m,b_n) & if \;\; \omega_m=1 \; and \; \omega_n=1 \end{cases}$$

Starting from a configuration where all the segments are considered to be inliers and have been linked greedily (one segment has been arbitrary chosen to be the first one, and for all n, the segment chosen to be at place $n+1$ is the nearest from the segment at place n that had not been chosen previously) our algorithm reduces the objective function iteratively until convergence. At each step, inliers and outliers are ordered, and all the possible permutations of a subsequence of inliers with a subsequence of outliers that reduce the objective function are applied.

The ordering algorithm we use for the first task tries to find by local search the order minimizing the sum of the links made between successive segments. In other words, it tries to find the 'Shortest Spanning Path' among the segments. The ordering of the outliers provides in this manner interesting subsequences for the permutation step without affecting the objective function, whereas the ordering of the inliers reduces the objective function. Given that all tasks reduce the objective function, convergence is guaranteed.

We remind that local search is a heuristic that minimizes an energy depending on a configuration by passing from the current configuration to a neighbouring one while it allows reducing the energy. Particular attention is to be paid when chosing the neighborhood definition. We chose the neighborhood defined by all the configurations reached (from the current one) when applying one of the following operations: (i) reversion of a subsequence of segments (ii) shift of a subsequence (iii) reversion followed by a shift of a subsequence.

Because local search is a meta heuristic, we cannot claim getting the global optimum, but the great size of neighborhood for both the orderings ($O\left(n^3\right)$ if n is the number of inliers) and the subsequence exchanges ($O\left(n^2m^2\right)$ if m is the number of outliers) substantially increases the probability of finding the global optimum. In practice this was very frequently observed.

The last step of the method consists of fitting a B-spline approximation to the retained ordered segmentations towards completing the missing content.

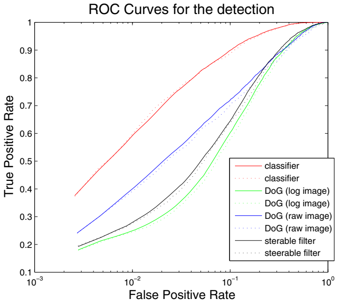

Fig. 3. Detection performances (for 5 million points from the 45 test images; dashed lines: for 1 million)

stages	MD	FD
detection	15.4%	53.2%
segments extraction	19.3%	53.0%
segments linking	23.8%	14.3%
splines fitting	25.8%	13.2%

Fig. 4. missed detections(MD) and false detections(FD) after each step. The state of the art [8] reports 22 ± 3% MD and 10 ± 3% FD.

4 Experimental Validation

Experiments were carried out on a database of 15 sequences of 10 images 1000x1000 acquired during interventions on 13 patients with a frame rate of 15 images per second, where clinical experts have manually marked GW. We performed a cross-validation: we built 15 classifiers using 75 images from 14 sequences only and tested each classifier using 3 images from the remaining sequence, using the same features and parameters for training and testing (a scale of 3 for MSAC and $\beta = 20.0$ for the ordering, in addition to constants given in preceding sections).

[Fig. (3)] compares the ROC curves of one of the classifiers with the ROC curves for three of the features that it uses: DoG of scales $(1.4, 1.8)$ computed on raw or on log images and the ridge detector for $\mu = 0.25, \sigma = 1.6$ of [5].

Given that even features designed for curvilinear structure detection perform poorly, machine learning methods like ours are necessary. Our method produces for example 4 times less false detections than the better single feature when detector is thresholded to recover 60% of GW.

We validated our algorithm with the same metrics as in [8]: we measured the proportion of pixels of GW at more than 5 pixels (1.0 mm) of the detected structures (*missed detections*,MD) and the proportion of pixels of the detected structures at more than 5 pixels of the GW (*false detections*,FD). Because GW parts lying in catheters are difficult to detect and not always of interest, but often help delineating the other parts, we decided not to take these parts into account for the computation of *missed detection* ratio.

[Fig. (4)] presents our results. Most of the MD are due to long parts of GW hardly visible producing prohibitive linking costs. There finally remains few FD, either due to rib borders or due to linking of line segments after the GW tip. A GW tip detector like the one used in [8] could therefore helps removing even more FD. This point is however out of the scope of this article.

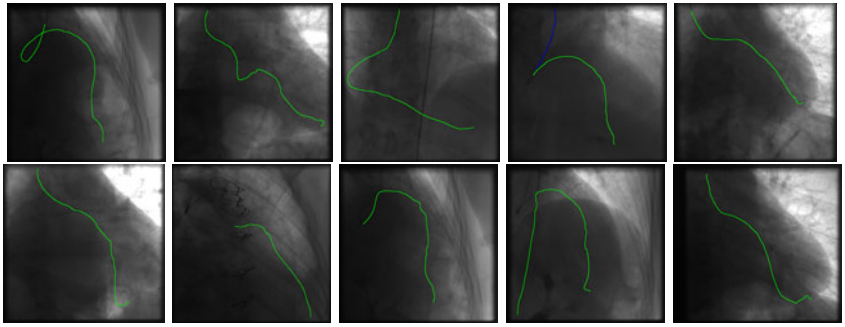

Fig. 5. Several final results (from 10 different sequences)

5 Discussion

In this paper, we have presented a global approach for GW segmentation designed to achieve high robustness. We used classifiers built by a variant of Boosting [9] and combining rotation invariant features such as improved steerable filters [5] to detect GW pixels and clustering to group them towards extracting line segments. We finally performed the delineation of the GW from the set of segments by solving an ordering problem with outlier rejection. Future work will focus on tracking the GW. Knowing the GW movement is indeed of great interest for many applications, such as cardiac motion monitoring.

References

1. Palti-Wasserman, D., Bruckstein, A.M., Beyar, R.P.: Identifying and tracking a guide wire in the coronary arteries during angioplasty from x-ray images. IEEE Tr. on Bio. Eng. 44(2), 152–164 (1997)
2. Baert, S.A.M., Viergever, M.A., Niessen, W.J.: Guide-wire tracking during endovascular interventions. IEEE Tr. on Medical Imaging 22(8), 965–972 (2003)
3. Frangi, A.F., Niessen, W.J., Vincken, K.L., Viergever, M.A.: Muliscale vessel enhancement filtering. In: Wells, W.M., Colchester, A.C.F., Delp, S.L. (eds.) MICCAI 1998. LNCS, vol. 1496, pp. 130–137. Springer, Heidelberg (1998)
4. Hauke Heibel, T., Glocker, B., Groher, M., Paragios, N., Komodakis, N., Navab, N.: Discrete tracking of parametrized curves. In: CVPR, pp. 1754–1761 (2009)
5. Jacob, M., Unser, M.: Design of steerable filters for feature detection using canny-like criteria. IEEE PAMI 26(8), 1007–1019 (2004)
6. Slabaugh, G., Kong, K., Unal, G., Fang, T.: Variational guidewire tracking using phase congruency. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) MICCAI 2007, Part II. LNCS, vol. 4792, pp. 612–619. Springer, Heidelberg (2007)
7. Franken, E., Rongen, P., van Almsick, M., ter Haar Romeny, B.M.: Detection of electrophysiology catheters in noisy fluoroscopy images. In: Larsen, R., Nielsen, M., Sporring, J. (eds.) MICCAI 2006. LNCS, vol. 4191, pp. 25–32. Springer, Heidelberg (2006)

8. Barbu, A., Athitsos, V., Georgescu, B., Boehm, S., Durlak, P., Comaniciu, D.: Hierarchical learning of curves application to guidewire localization in fluoroscopy. In: CVPR, pp. 1–8 (2007)
9. Freund, Y., Shapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. In: ICML (1996)
10. Honnorat, N., Vaillant, R., Paragios, N.: Robust guidewire segmentation through boosting, clustering and linear programming. In: ISBI, pp. 924–927 (2010)
11. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. The annals of statistics 28(2), 337–407 (1998)
12. Viola, P., Jones, M.: Fast and robust classification using asymmetric adaboost and a detector cascade. In: NIPS (2001)
13. Komodakis, N., Paragios, N., Tziritas, G.: Clustering via lp-based stabilities. In: NIPS (2008)
14. Torr, P., Zisserman, A.: Robust computation and parametrization of multiple view relations. In: ICCV, pp. 727–732 (1998)
15. Sharon, E., Brandt, A., Basri, R.: Completion energies and scale. IEEE PAMI 22(10), 1117–1131 (2000)
16. Yatziv, L., Bartesaghi, A., Sapiro, G.: O(n) implementation of the fast marching algorithm. Journal of Computational Physics 212, 393–399 (2006)