A LEARNING BASED HIERARCHICAL MODEL FOR VESSEL SEGMENTATION

Richard Socher*
Computer Science Department
Saarland University, Germany

Adrian Barbu
Statistics / Computational Science
Florida State University, USA

Dorin Comaniciu
Integrated Data Systems
Siemens Corporate Research, USA

ABSTRACT

In this paper we present a learning based method for vessel segmentation in angiographic videos. Vessel Segmentation is an important task in medical imaging and has been investigated extensively in the past. Traditional approaches often require pre-processing steps, standard conditions or manually set seed points. Our method is automatic, fast and robust towards noise often seen in low radiation X-ray images. Furthermore, it can be easily trained and used for any kind of tubular structure. We formulate the segmentation task as a hierarchical learning problem over 3 levels: border points, cross-segments and vessel pieces, corresponding to the vessel's position, width and length. Following the Marginal Space Learning paradigm the detection on each level is performed by a learned classifier. We use Probabilistic Boosting Trees with Haar and steerable features. First results of segmenting the vessel which surrounds a guide wire in 200 frames are presented and future additions are discussed.

***Index Terms*—** Blood vessels, Image segmentation, X-ray angiocardiography, learning systems

1. INTRODUCTION

Vessel Segmentation is an important task in medical imaging and has been investigated extensively in the past. In this paper, we develop an automatic segmentation method for vessels in coronary angiography.

Coronary angiography is a medical examination that uses X-Ray imaging to find stenoses in coronary arteries. To locate such an abnormal narrowing of a vessel a catheter is put into an artery in the groin or arm and guided to the heart. A contrast agent is injected several times to visualize the vessel and aid navigation of the catheter, guidewire, balloon and stent in the coronary tree. Segmentation is performed during the short period in which the vessel is visible in order to use this information later in the procedure and for future analysis.

There is a plethora of different segmentation methods for vessels. Some are specific to different kinds of vessels, such as retina vessels or different modalities such as CT or MRI. Only few papers handle the case of angiographic videos. [1] provides an extensive overview of different methods putting them in categories such as (i) pattern recognition, (ii) model based, (iii) tracking based and (iv) artificial intelligence.

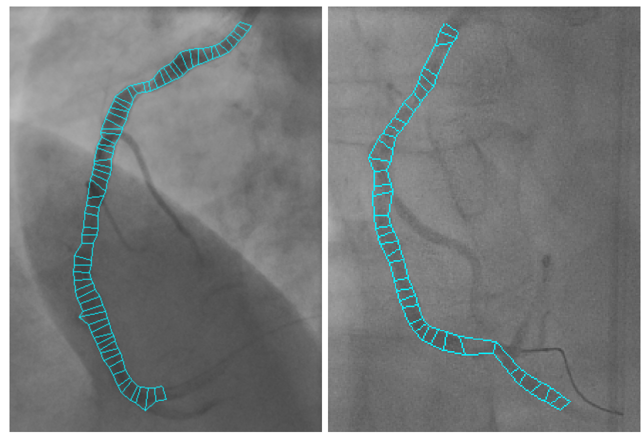

Fig. 1. Examples of main vessel segmentation in angiographic images.

Few papers exist that use machine learning techniques. [2] uses wavelet features and k-Nearest Neighbor to label pixels as inside or outside of a vessel. A similar approach that also uses k-Nearest Neighbor ([3]) is one of the few papers to present quantitative results. However, its results are impractical for our purpose, since the method needs about 15 minutes to segment one image of a retina vessel and is not robust against edges that are not vessels. In angiography, such edges frequently occur in form of background organs.

Many methods rely on standard conditions, heavy preprocessing steps such as morphological top hat filters or on manually set seed points. In contrast to these methods, our approach is purely learning based and may be used for segmenting other kinds of tubular structures such as streets. We do not require seed points, nor pre-processing of frames. Our algorithm is real-time and returns a probability as well as a width for each section of the vessel. We show first results on 200 frames.

The background section describes marginal space learning which shapes our learning process, probabilistic boosting trees, which are used in all levels of learning and lastly steer-

*This work was performed while R. Socher and A. Barbu were with the Integrated Data Systems Department of Siemens Corporate Research.

able features. After defining the representation of the vessel we describe the training of our model. A section of experiments follows and a conclusion is given.

2. BACKGROUND

2.1. Marginal Space Learning

Many object detection applications face the problem of a high dimensional parameter space. Marginal Space Learning (MSL) first introduced by ([4]) uses the fact that the posterior distribution of the correct parameters given the data lies in a small region of the n-dimensional complete parameter space: $\mathfrak{R}_\mathrm{n} \subset \Omega_n$.

Let $P(\Omega_n|\mathfrak{D})$ be the true posterior given the data $\mathfrak{D}$. Instead of searching for $\mathfrak{R}_\mathrm{n}$ directly in Ω_n, MSL proposes to start in one of its low dimensional marginal spaces Ω_1 and sequentially increase the dimensionality of the search space:

$$\Omega_1 \subset \Omega_2 \ldots \subset \Omega_n \tag{1}$$

where $\mathtt{dim}(\Omega_k) - \mathtt{dim}(\Omega_{k-1})$ is usually small. Assume we learned the probability distribution over space Ω_k which results in the subspace Π_k with the most probable values. This allows us to restrict the learning and testing of the next higher dimensional marginal space to $\Pi_k \times X_{k+1}$. Hence, orders of magnitude fewer parameters have to be examined by restricting the final $\mathfrak{R}_\mathrm{n}$ early during the learning process. This is different from a normal cascade of strong classifiers in that it performs simple operations on a subset of Ω_k.

2.2. Probabilistic Boosting Tree

The Probabilistic Boosting Tree (PBT) introduced by ([5]) is used as a classifier in each marginal space. A PBT is similar to a decision tree but instead of using just one attribute at each node, a strong AdaBoost classifier is trained to find the probability of classes $y = \{+1, -1\}$ using several weighted weak classifiers $h(t)$: $H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$.

Based on $H(x)$ and the resulting probabilities $q(+1|x)$, $q(-1|x)$, each node recursively subdivides the samples into a left (S_{left}) and a right set (S_{right}). Assume, we have a sample (x_i, y_i):

$$
\begin{aligned}
&\texttt{if } q(+1|x_i) - 1/2 > \epsilon \\
&\quad \texttt{then } (x_i, y_i, 1) \rightarrow S_{right} \\
&\texttt{elseif } q(-1|x_i) - 1/2 > \epsilon \\
&\quad \texttt{then } (x_i, y_i, 1) \rightarrow S_{left} \\
&\texttt{else} \\
&\quad (x_i, y_i, q(+1|x_i)) \rightarrow S_{right} \texttt{ and } (x_i, y_i, q(-1|x_i)) \rightarrow S_{left}
\end{aligned} \tag{2}
$$

It then trains another strong classifier in both sets unless the empirical distribution $q(y) = \sum_i w_i \delta_i(y_i = y)$ directly defines the class or the maximum depth is reached. During testing, the complete posterior $\tilde{p}(y|x)$is recursively calculated from the entire tree by adding the probabilities $\tilde{p}_{left|right}(y|x)$ of its subtrees, weighted by the current classifier‘s posterior :

$$\tilde{p}(y|x) = q(+1|x_i)\tilde{p}_{right}(y|x) + q(-1|x_i)\tilde{p}_{left}(y|x) \tag{3}$$

2.3. Steerable Features

Steerable features is the name of a novel framework ([6]) which has been developed for 3d object segmentation with a given mean shape. It can capture the orientation, rotation and scale of an object while retaining a high degree of efficiency. The idea is to set up a small relative coordinate system for each given object with the center of the object being the origin. Then, few points are sampled from the area (or volume) around the origin by a certain pattern. Possible samples could be gray values, gradient values, probability maps that have been calculated before, a combination or transformation of those or even combined values of different sample points. Given those sampled features, a posterior probability can be computed for the parameters of a given object. Figure 2 shows an example coordinate system inside a vessel. For more details, see section 4.

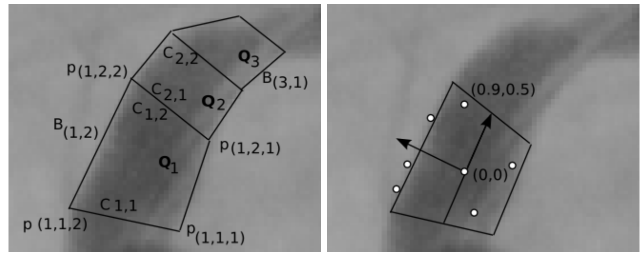

Fig. 2. Left: Example of a vessel represented as a list of quadrilaterals Q each consisting of two cross segments C. Each cross-segment consists of two endpoints p. Cross segments are connected through the boundaries B of the quadrilateral. Right: Example of a quadrilateral between two cross segments and the relative coordinate system used for sampling of steerable features at the white circles.

3. VESSEL REPRESENTATION IN MSL

In order to apply MSL effectively, we chose the following representation for a vessel. A vessel V is defined to be an ordered list of n quadrilaterals (each associated with a probability): $V = (Q_1, \ldots, Q_n)$. Each quadrilateral Q_i consists of a pair of cross segments C: $Q_i = (C_{i,1}, C_{i,2})$. Each cross segment C consists of its two endpoints (i.e. pixels in the image domain $\mathcal{I}$): $C_{i,j} = (p_{i,j,1}, p_{i,j,2})$. Neighboring quadrilaterals share the same cross segments, i.e.: $C_{i,2} = C_{i+1,1} \quad \forall i : 1 < i < n$. Vessel boundaries $B_{i,1}$ and $B_{i,2}$ are locally defined for each quadrilateral to be either a line with endpoints

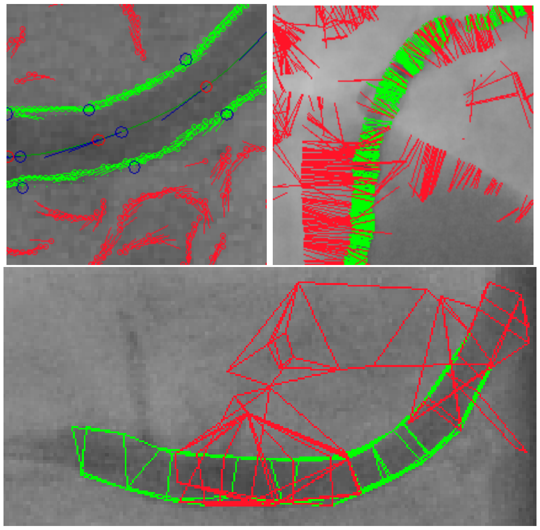

Fig. 3. Examples of positive (green) and negative(red) training samples for Ω_1 - edge detection (top left), Ω_2 - cross segments (top right) and Ω_3 - quadrilaterals (bottom)

$p_{i,1,1}, p_{i,2,1}$ or a smooth polynomial function whose derivatives are defined by the gradient at these endpoints. For the sake of simplicity, we will consider only the case of lines here. Figure 1 shows an example of such a vessel.

In this representation each level i (i.e. points, segments, quadrilaterals) corresponds to a learned classifier in space Ω_i in the segmentation algorithm.

4. TRAINING MSL-CLASSIFIERS

In this section, we describe the method to segment a vessel into the representation given by section 3.

4.1. Ω_1 - Learning Enhanced Edge Detection

The first level finds possible vessel edges by a simple gradient based method that is enhanced by a trained PBT classifier. For the classifier a sample is a pixel $p(x, y)$ with a direction $(g_y, -g_x)$, i.e. the gradient direction rotated counterclockwise by 90 degrees. This way, the corresponding Haar Features ([7]) of each sample are aligned and have a darker area (the vessel with contrast) on their right side. Through Haar Features more complex characteristics of vessel edges such as small side vessels may also be incorporated into the learning process. Samples are positive, if they have close proximity to the annotated vessel border and a large gradient. Figure 3 shows examples of training samples for all three levels.

During detection, the result is a mask M of the image with probabilities entries:

$$M_{x,y} = P_{\texttt{PBT}}(\mathcal{I}_{(x,y)} = \texttt{edge}|\texttt{Haar}_{x,y}) \qquad (4)$$

4.2. Ω_2 - Cross Segments

Cross segments are defined in a similar way in ([8]), as a line that is perpendicular to the medial axis of the vessel, i.e. bisecting it. Based on M, the second level determines the width of the vessel by finding a suitable edge pixel in the opposite gradient direction for each candidate location a where M_a is high. A cross-segment sample is created if:

$$\exists b[||a - b|| < W_{\max} \wedge \exists t[\tilde{g}(t) = b]] \qquad (5)$$

where $\tilde{g}$ is the vector function in affine space, which starts at a and points in the opposite direction of the gradient. $W_{\max}$ is the maximum width a vessel could have.

For a cross segment to be a positive sample for the training of the PBT classifier, both of its endpoints have to be close to the annotated vessel edge and the induced line through them has to be close to perpendicular to the medial axis of the vessel. All other segments are negative samples.

During detection, the result is a set of cross segments that bisect the vessel and their probabilities: $\{(C, P_{\Omega_2}(C))\}_{1..\texttt{MaxC}}$, with maxC being the maximum number of cross segments per frame.

4.3. Ω_3 - Quadrilaterals

The goal of this level is to find pairs of cross segments that, if connected as a quadrilateral, capture an area of the vessel. It is important that the connecting lines are as close as possible to the real edge. See figure 2 for an example with annotation.

In order to give a probability for a quadrilateral, steerable features are sampled around its center, as seen in figure 2. Possible features include: image data such as gradient, gray value; or results of previous levels, such as the probability map of Ω_1 or the probabilities of the two cross segments from Ω_2, which form the quadrilateral. The PBT classifier will pick the best features.

Positive and negative samples are created as follows: A sample pool is created by finding the closest 20 cross segments on the right and left side for all the segments of the previously learned level. The remaining quadrilaterals are sorted into positive and negative samples for PBT training: A quadrilateral Q_i is a positive sample, if $\forall p \in B_{i,1} \bigcup B_{i,2}$

$$\exists d \in \{a_{(x,y)} | a_{(x,y)} \in \texttt{annotated boundary}\} : ||p - d||_2 < 2 \qquad (6)$$

The probability of a quadrilateral shows how likely two cross segments are connected for the final vessel.

4.4. Fourth Level: Dynamic Programming

Based on the outcomes of previous steps, we create a graph $G = (V, E)$, where the vertices are cross-segments. Cross-segments are connected through an edge, if they are likely to form a quadrilateral. The cost associated with each edge is calculated based on the quadrilateral's probability: $c =$

$\log(\frac{1-p}{p})$. Finding the main vessel then corresponds to finding the lowest cost path inside this graph. We use dynamic programming for solving this task.

5. EXPERIMENTS

Instead of segmenting the entire vessel tree, the goal of the following experiments is to segment only the vessel which surrounds the guide wire. This is useful during coronary angiography, when the surgeon has already identified the vessel which is blocked and needs to memorize its appearance, while the contrast agent is flowing through.

Training was performed with 134 frames from 7 sequences and testing with 64 frames from 5 different sequences. For evaluation purposes we compare the area under the quadrilaterals with that of the annotation, both counted in pixels. The detection rate in the test set is 90.1% and the false alarm rate is 23.5%. The high false alarm rate and the relatively low detection rate are both caused by dominant side vessels as shown in figure 4. Since the dynamic programming selects the path with the lowest cost, it follows the longer side vessels and hence misses the vessel that surrounds the guide wire. This problem will be solved in future versions by incorporating time coherency with guide wire detection of previous frames. The running time is around 2 seconds

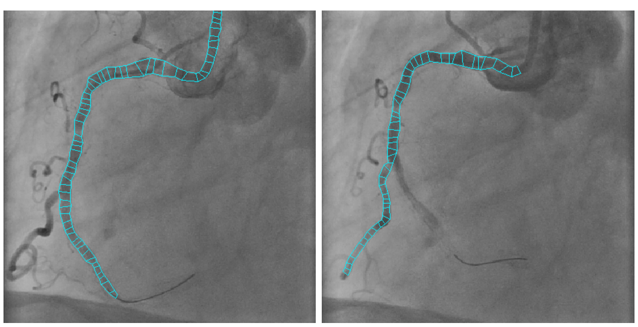

Fig. 4. Left: Correct segmentation of a vessel that surrounds the guide wire. Right: Instead of segmenting the vessel which surrounds the guide wire a dominant side vessel is segmented. The result is a large false alarm.

per frame. Figure 1 shows final results from two different sequences. Figure 5 demonstrates that our method generalizes well, since we have not used fluoroscopic images of such low quality during training, but the vessel is still correctly segmented.

6. CONCLUSION

In this paper, we presented a hierarchical learning based vessel segmentation method that is highly driven by data and generalizes well to lower quality X-ray images. We introduced a new representation of a vessel consisting of three marginal

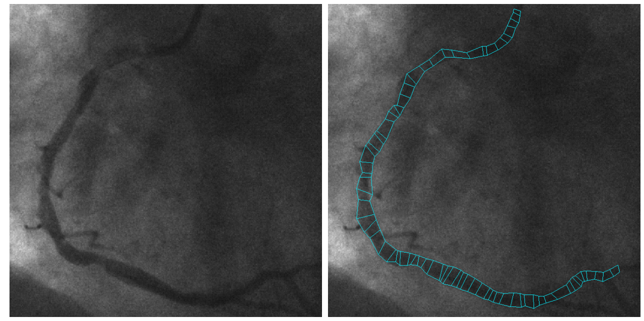

Fig. 5. Results in low-radiation X-ray images (test set). Left: Original image, Right: Segmentation of main vessel.

spaces: border points, vessel width and vessel pieces (quadrilaterals), each having an associated classifier. Our experimental results are preliminary but promising: they demonstrate that the vessel model is capable of segmenting vessels with high precision, even in low quality images. Further additions such as time coherency are needed to ensure that only the main vessel is segmented. Another extension would be to incorporate a bifurcation detector and use it to segment all branches of a vessel tree which start at a bifurcation and end at a tip.

7. REFERENCES

[1] Cemil Kirbas and Francis Quek, "A review of vessel extraction techniques and algorithms," *ACM Comput. Surv.*, vol. 36, no. 2, pp. 81–121, 2004.

[2] Jorge J. G. Leandro, Joao V. B. Soares, Roberto M. Cesar Jr., and Herbert F. Jelinek, "Blood vessels segmentation in non-mydriatic images using wavelets and statistical classi.ers," *sibgrapi*, vol. 00, pp. 262, 2003.

[3] J. Staal, M.D. Abramoff, M. Niemeijer, M.A. Viergever, and B. van Ginneken, "Ridge-based vessel segmentation in color images of the retina," vol. 23, no. 4, pp. 501–509, April 2004.

[4] A. Barbu, V. Athitsos, B. Georgescu, S. Boehm, P. Durlak, and D. Comaniciu, "Hierarchical learning of curves: Application to guidewire localization in fluoroscopy," CVPR, 2007.

[5] Zhuowen Tu, "Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering.," in *ICCV*, 2005, pp. 1589–1596.

[6] Y. Zheng, A. Barbu, B. Georgescu, M. Scheuering, and D. Comaniciu, "Fast automatic heart chamber segmentation from 3d ct data using marginal space learning and steerable features," in *IEEE Int'l Conf. Computer Vision (ICCV'07)*, Rio de Janeiro, Brazil, 2007.

[7] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *CVPR*, 2001.

[8] Ying Sun, "Automated identification of vessel contours in coronary arteriograms by an adaptive tracking algorithm," IEEE Transactions on Medical Imaging, Volume 8, Issue 1,pp. 78-88, Mar 1989.