Abstract

In this paper we address the problem of automatic selection of important vessel-depicting key frames within 2D angiography videos. Two different methods of frame selection are described, one based on Frangi filter, and the other based on detecting parallel curves formed from edges in angiography images. Results are shown by comparison to physician annotation of such key frames on 2D coronary angiograms.

1. Introduction

Coronary angiography is used routinely to assess coronary artery disease. It is used both pre-operatively to examine the arteries for evidence of stenosis as well as to monitor the progress of revascularization during angioplasty. The X-ray images can be either still images or videos captured at 15-30 frames per second to allow the cardiologist to evaluate the vascular anatomy and blood flow through vessels. During the analysis of the angiographic data, the cardiologist/radiographer often pauses at specific images to examine the vessels. Such images often depict the vessels with best clarity. In fact, much of their diagnosis is based on such ‘keyframes’.

To our knowledge, this is the first paper to address the key frame selection problem in angiography videos. While there is much work on shot detection and key frame selection for general videos [7, 11], they are not particularly applicable to angiography videos as the images have predominantly low intensities with roughly the same intensity histograms regardless of the content being depicted (background or vessels). Further, the subtle ‘waxing’ and ‘waning’ transition in the visibility of vessels is difficult to capture with existing shot boundary detection algorithms[7] that expect sharp changes (due to camera angle change).

2. Measuring vessel visibility

To accurately measure vessel visibility in each frame, we explore two methods, (a) the Frangi filter [1], a popular approach to vessel extraction, and (b) a new vessel extraction method from edge maps by detecting explicit tubular structures in curves. In both cases, we derive a single vessel visibility measure per image whose value will then be tracked over the angiography sequence. We then compare the impact of using the different vessel extraction methods in the quality of the key frames selected.

2.1 Vessel visibility using Frangi filter

The Frangi filter is a popular filter to extract vessel structures in coronary angiograms [1]. It uses the eigenvectors of the Hessian to compute the likelihood of an image region containing vessels or image ridges. Due
to the easy availability of its code, our first approach was to form a single vessel visibility number from an image obtained by applying the Frangi filter. The output of such filtered image is a grey scale image with brighter regions showing a higher probability of finding the ridge. Figure 3b shows the result of filtering the image of Figure 3a. As can be seen, the vessels have been delineated. However, in cases where there is no vessel depicted in an image, the filter can still find ridges although with not as much intensity. To avoid false positives, therefore, we apply a threshold to the Frangi-filtered image and retain the bright regions. Specifically, Otsu’s thresholding [5] algorithm is applied to separate the foreground and background by minimizing the combined spread (intra-class variance) using the shape of the histogram of pixel intensities. The resulting thresholded image is shown in Figure 3c. The number of non zero-valued pixels in this binary image is then taken as a measure of vessel visibility.

2.2. Visibility using edge-based vessel extraction

In addition to Frangi filter, we explored a new representation of angiography images by direct detection of tubular structures from edge images. For this, we first pre-process the image using a conventional edge detector (e.g. Canny edge detector). Since large intensity gradients are more likely to correspond to edges, a thresholding is applied at two levels to yield both weak (lower threshold) and strong (higher threshold) edges. We then extract curves from these edge maps using 8-connected neighbors and tracing curves using depth-first search between junctions. Let \( C^H = \{ c^H_1, c^H_2, \ldots, c^H_N \} \) be the curves formed from applying a low threshold. Similarly, let \( C^L = \{ c^L_1, c^L_2, \ldots, c^L_N \} \) be the curves formed using a high threshold. Then starting from the curves in \( C^L \), we retain all the curves from \( C^H \) that contain the curves in \( C^L \). That is, we retain a subset \( C^S \subset C^H = \{ c^H_1 \mid \exists (x,y) \in c^L_i \& (x,y) \in c^H_j \} \). This has the benefit of retaining long boundaries while still using edge strength to remove the background elements.

2.3. Extracting vessels using parallel curves

Next, to detect tube-like vessel structures, we look for evidence of closely-spaced almost parallel curves[9, 2] among the retained curves. Formally, a parallel of a curve is the envelope of a family of congruent circles of a fixed radius centered on the curve. For vessel detection though, this definition has to be relaxed to accommodate occlusion, spurious interjections of curve fragments and convergence as the vessel narrows towards the end. Thus only approximate test of parallelism can be done and partial matches of curve fragments have to be allowed.

We define the tubular structures as pairs of curve fragments \((f(t_i), g(t_j))\) over parametric intervals \( t_i \in [T_1, T_2] \) and \( t_j \in [T_3, T_4] \) that have a large number of consecutive points in shape correspondence. To determine shape correspondence, we use the positional information as well as the local shape around the features. In order to allow local shape distortions due to stenosis as well as convergent lines, we allow for gaps and insertions while still finding the longest possible pairs.

To achieve this, we adapt the well-known technique of dynamic time warping used for time series matching [10] to find matching parallel curve fragments. The shape similarity needed for determining parallelism, as well as the constraints of maintaining a distance between the curves can be well-modeled in the distance function for alignment. Specifically, we consider a parametric representation of curves, and incorporate
shape constraints into the distance function by considering each point along a curve as a four-dimensional feature

\[ f(t) = (x_f(t), y_f(t), \theta_f(t), \varphi_f(t)) \]  

where \( x(t), y(t) \) are pixel coordinates of the points along the curve and \( \theta(t) \) is the included angle at the point by treating it as a corner with the adjacent pixels, and \( \varphi(t) \) is the orientation of the bisector of the include angle giving the local orientation at that pixel. Using the included angle ensures that narrow curvature points are not matched to wider curvature points. Similarly, orientation of the bisector ensures that the convexities do not match to shape concavities.

To find matching points on candidate curves that represent parallelism, we form a dynamic programming matrix \( H \) where the element \( H(i, j) \) is the cost of matching up to the \( i \)-th pixel along curve \( f(t) \) with the \( j \)-th pixel in the curve \( g(t) \) as

\[
H(i, j) = \begin{cases} 
H_{i-1, j-1} + d(f(t_i), g(t_j)), & \text{if } i, j > 0 \\
H_{i-1, j} + d(f(t_i), 0), & \text{if } j = 0 \text{ or } i = 0 \text{ and } j > 0 \\
H_{i, j-1} + d(0, g(t_j)), & \text{if } j = 0 \text{ or } i = 0 \text{ and } i > 0 \\
\infty & \text{otherwise}
\end{cases}
\]

with initialization as \( H_{0,0} = 0 \) and \( H_{0,j} = \infty \) and \( H_{i,0} = \infty \) for all \( 0 < i \leq K, \text{ and } 0 < j \leq M \), where \( M \) and \( K \) are the lengths of the candidate curves. The shape constraints for matching are incorporated in the term \( d(\cdot, \cdot) \). Here the first term \( d(f(t_i), g(t_j)) \) represents the distance of feature point \( f(t_i) \) to feature point \( g(t_j) \). The second term \( d(f(t_i), 0) \) represents the choice where no match is assigned to feature \( f(t_i) \). The number of matching pairs of points between the curves is then determined by a diagonal tracing of the dynamic programming matrix (similar to finding longest common subsequence of strings).

The distance \( d(f(t_i), g(t_j)) \) is then given as the Euclidean distance between the two fiducial points using the 4 parameters as

\[
d(f(t_i), g(t_j)) = 
\begin{cases} 
\sqrt{(x_f(t_i) - x_g(t_j))^2 + (y_f(t_i) - y_g(t_j))^2 + (\theta_f(t_i) - \theta_g(t_j))^2 + (\varphi_f(t_i) - \varphi_g(t_j))^2} & \text{if } |(x_f(t_i) - x_g(t_j))|^2 + (y_f(t_i) - y_g(t_j))^2 \leq \lambda_1 \\
\infty & \text{otherwise}
\end{cases}
\]

(3)

The thresholds \( (\lambda_1, \lambda_2, \lambda_3) \) are determined through a prior learning phase in which the expected variations per vessel class is noted.

We try all pairings of curves of length \( m \in C^R \) with a high resolution curves of length \( n \in C^H \) in \( C^H \) and retain those with high score of \( H(m, n) \) as parallel curves. Thus the final set of curves retained to generate the vesselness measure is given by \( C^R = C^H \cup \{c^H \mid c^H \in C^H \& \|c^H\| \in C^S \} \), where the symbol \( \| \) denotes the parallelism relation between the two curves. Using only curves that are within a distance of separation \( \delta \), the complexity of the parallel curve detection is kept linear in the number of curves \( C^S \). Typically, \( C^S \) and hence \( C^R \) is less than 200 while \( C^H \) can be over a 1000 curves, thus resulting in a great reduction in the number of curves that are retained. Finally, the vessel visibility measure \( V(f) \) for image frame \( f \) is simply the number of retained curves given by \( V(f) = |C^R| \).

2.4. Keyframe selection

To select keyframes, we analyze the trajectory formed by computing the vessel visibility measure in each frame of a video sequence. We detect peaks in this trajectory as points \( \{(ij, V(ij) \} \) where \( V(ij) > V(ij-1) \) and \( V(ij) > V(ij+1) \). Then all peaks are ordered using the vessel visibility measure values \( V(ij) \). Based on the observation of a large number of angiographic sequences, a minimum separation of at least 5 frames is desirable for key frames. In our experiments, we retained peaks that were within 80% of the strongest peak as the selected key frames.

3. Results

We now present results of keyframe selection on a large database of angiogram sequences. The database was assembled from actual angiogram sequences recorded in a cardiac catheterization laboratory in our area. This data set depicts patients with various forms of coronary artery disease. Each angiogram study consists of several sequences called runs. Each run depicts the examination of a different vessel (e.g., left main, right coronary artery). In a typical study, there are 20-25 runs, with each run consisting of 15-120 images depending on the complexity of the arteries being shown and their pathology. In our database we have a collection of over 210 video sequences for a total of 5250 images.

We first illustrate the extraction of vessels using the new approach. Fig. 3a shows the original image. Fig. 3d shows the fine edges using a low threshold \( (C^H) \). Fig. 3e shows the result of parallel curve detection using dynamic time warping. The highlighted points are those found to be matching to the respective parallel curves. Finally, Fig. 3f shows the set of retained curves \( C^R \) that contain the tubular structures. These curves are retained for computing the vessel visibility. Comparing Fig. 3c with Fig. 3e, it can be seen that our measure not only highlights the vessels as well as the Frangi filter, but also results in accurate localization of vessel boundaries which can be helpful for stenosis characterization in the future.

**Figure 2.** Comparison of vessel visibility trajectories using Frangi filter versus edge-based tubular vessel detection.

Next, we illustrate the selected keyframes using the peaks in the vessel visibility trajectory. Fig. 1a and b show two angiography sequences, depicting the RCA (right coronary artery) and LAD (left anterior descending artery) respectively.
The vessel visibility trajectories using the two methods of vessel extraction are shown in Figure 2a and (b) and correspond to the runs shown in Figure 1a and b. The trajectory obtained by using the Frangi filter is shown in red while trajectory from tubular edges is shown in green in case of each run. Again, the set of frames determined to be keyframes using the vessel visibility trajectory from Frangi filtered images is shown by the frames highlighted in red. The set of keyframes determined using our edge-based vessel extraction algorithm are shown highlighted in green. As can be seen, our method detects fewer background images as keyframes, while still retaining frames with high vessel visibility.

3.1. Evaluation of key frame selection

To evaluate the accuracy of key frame selection using both approaches, we conducted the following experiments. We made storyboard pages of the frames of an angiography sequence HTML (Fig. 1). We asked three cardiologists to mark the frames in the sequence that they would use for examining the arteries. Next, we showed the cardiologists our selected frames via the highlighting. We evaluated the accuracy by (a) noting the fraction of runs for which our key frames annotated contained the consensus cardiologist-selected frames, and (b) the average distance between our identified frames and the cardiologists annotated frames in case of a mismatch. The results are tabulated in Table 1. As can be seen, the number of keyframes selected is substantially smaller than the total number of frames, although our vessel extraction method results in fewer frames (84.4%=100-(820/5250) compared to 70.8%=100-(1534/5250) using Frangi filter-based vessel extraction. The keyframes selected by our algorithm contain a physician-identified frame in 90% of the cases, and in case of misses, our identified frame was within 2 frames of the physician-identified frame.

The keyframes selected is substantially smaller than the total number of frames, although our vessel extraction method results in fewer frames (84.4%=100-(820/5250) compared to 70.8%=100-(1534/5250) using Frangi filter-based vessel extraction. The keyframes selected by our algorithm contain a physician-identified frame in 90% of the cases, and in case of misses, our identified frame was within 2 frames of the physician-identified frame. Thus we conclude that our vessel extraction based on parallel curve detection gives keyframes of both higher precision and higher recall while still achieving the goals of significant data reduction.

4. Summary

In this paper, we address the problem of key frame selection in angiogram videos using two different methods of vessel extraction. We show that the tubular curve method gives keyframes of higher precision and recall than those derived from the Frangi filter.

References


