CGI'14
31st Computer Graphics International
10 - 13 June 2014
Sydney, Australia

Parallelized 4D Structure, Shape, and Motion Reconstruction of Vessels from Multiview X-Ray Angiograms

Xinglong Liu · Fei Hou* · Aimin Hao · Hong Qin

Abstract In this paper, we present a parallel 4D vessel reconstruction algorithm that simultaneously recovers 3D structure, shape, and motion based on multiple views of X-ray angiograms. The fundamental goal is to assist the analysis and diagnosis of interventional surgery in the most efficient way towards interactive and accurate performance. We start with a fully parallelized algorithm to extract vessels as well as their skeletons and topologies from dynamic image sequences. Then, instead of resorting to registration, we present an space voxelization algorithm to formulate the reconstruction problem as an energy minimization problem with color, coherence, and topology constraints to reconstruct the 3D vessel initially, which is robust to combat noise and incomplete information in images. Next, we incorporate temporal information into our energy optimization framework to track and reconstruct 4D kinematics of the dynamic vessels, which is also capable of recovering previous incomplete and misleading shapes acquired from static images otherwise. We demonstrate our system in coronary arteries reconstruction and movement tracking for percutaneous coronary intervention surgery to help medical practitioners learn about the 3D shapes and their motions of the coronary arteries of specific patient. We envision that our system would be of high assistance for diagnosis and therapy to treat vessel-related diseases in a clinical setting in the near future.

Keywords X-ray Angiograms · 3D Reconstruction · Motion Tracking · Belief Propagation

1 Introduction and Motivation

The morbidity of Cardiovascular Disease (CVD) is rapidly increasing over the past few decades. The golden standard for diagnosis of CVD is X-ray coronary angiography which is only offering 2D projection of relevant tissues/organs from 3D space. Yet, X-ray images not only lack a significant amount of 3D information of the coronary arteries, but also suffer from the viewing angle dependence, overlapping and blurring, etc. Accurate and rapid 3D reconstruction from limited views is necessary for medical practitioners towards earlier diagnosis and better treatment. This paper's originality hinges upon our novel and parallel solution to the simultaneous 4D structure, shape, and motion reconstruction from time-labeled image sequences.

Even though various work has been done to tackle the reconstruction problem in X-ray views, there are still some unsolved challenges existing in current methods. First, the noisy and blurry X-ray views may give rise to incorrect reconstruction. Second, accurate reconstructions need five or even more views of angiograms with exact angle requirements, which is hard to operate for clinical use. Third, current 3D reconstruction methods mostly rely on the registration between image pairs, which are less robust and much hard to incorporate with constraints such as consistency, continuity and coherence. Finally, since there are so many processing procedures involved during analysis and reconstruction, the overall computation is extremely time-consuming.

---

*Corresponding author, E-mail: houfei@vrlab.buaa.edu.cn

Xinglong Liu · Fei Hou · Aimin Hao
State Key Laboratory of Virtual Reality Technology and Systems, Beihang University.

Hong Qin
Department of Computer Science, Stony Brook University.

To overcome such shortcomings, we present an efficient vessel reconstruction and motion tracking system from multiple X-ray views. The pipeline is shown in Fig. 1 consisting of two stages: vessel extraction (Section 3) and 4D reconstruction (Section 4). In the first stage, we design a parallel algorithm to extract the vessels as well as their skeletons and topology in realtime. In the second stage, we devise a novel space voxelization algorithm to reconstruct 4D dynamic vessels robustly to resist noise and combat incomplete information in images. Moreover, the parallel CUDA implementation greatly enhances efficiency in our system. The main contributions of our work include:

- An efficient parallelized thinning and refining method for extracting vessel skeletons and key points.
- A novel space voxelization algorithm to reconstruct vessels based on energy optimization solved using belief propagation with *color*, *coherence* and *topology* constraints, which is robust to noisy and incomplete information from images.
- An improved energy formulation that unites temporal information with spatial information for better recovering 4D kinematics of vessels, whose advantages also include robustness to noisy and incomplete information in static images and ability of repairing misleading shapes.
- A vessel reconstruction and tracking system with the fully parallelized image processing algorithm and partially parallelized Belief Propagation algorithm.

2 Related Work

Our work relates to vessel extraction from images, vessel reconstruction, etc. We now briefly review them in the following categories.

Vessel Extraction. Hoover et al. [1] used a mathematical filter to entails a broad range of vessel enhancement and Li et al. [2] conducted this task using a non-linear filter. Frangi et al. [3] used the eigen values of Hessian matrix to extract the tube-like structures from X-ray images. Condurache et al. [4] used this method while adding a hysteresis thresholding method to purify the extracted data. But they are not robust to handle blurry images.

Skeleton Extraction. Centerline extraction can be divided into six categories of techniques: pattern recognition techniques, deformable model based techniques [5][6], tracking-based techniques [7][8][9], artificial intelligence based techniques, neural network-based techniques, and miscellaneous tube-like object detection techniques. Each one contains many sub-types such as multi-scale approaches, mathematical morphology approaches. Readers could refer to [10] for an overview of the centerline extraction technologies. Previously proposed techniques are mostly serial and usually hard to be parallelized.

3D Reconstruction. Definitely, 3D reconstruction from 2D projection images is feasible and reasonable. Wellnhofer et al. [11] and Messanger et al. [12] evaluated that 3D reconstructions of coronary arteries from 2D X-ray image sequences permit accurate results of the real data. The biplane X-ray system takes two (mostly) synchronized projection of the coronary arteries [11][12] while the mono-plane (single-plane) system [13] can just take one view at the same time, therefore selection of asynchronous images from multiple views is needed. Movassaghi et al. [14] used multiple projections for realistic vessel lumen simulation. Sprague et al. [15] utilized the benefits of three projections experimentally. Hansis et al. [16] had used multiple projections from a single rotational X-ray angiography to reconstruct the 3D centerline and the topology. Nguyen et al. [17] proposed a method based on motion and multiple views using a single-plane imaging system. Most of the methods above rely on hard registration which is rather unstable in the blurry X-ray views and the constrains used by registration are too simplex to take varied affection terms into account.Other routines such as knowledged-based or rule-based methods using the vascular network model [18] [19] are hard to be generalized.

Motion Tracking. Prior to the work of Ruan et al. [20], most analysis work focuses on static reconstruction using feature matching techniques. Similar method based on the same prediction-projection-optimization loop is proposed in [21]. In [22], a motion trajectory is computed for each point from the segmented artery tree independently, they used a set of vectors describing the general motion of each artery branch. Chen et al. [23] reconstructed the vessel tree and performed motion tracking on this tree with constraints. Based on the work of Mourgues et al. [24], Shechter et al. [25] presented a 3D method for tracking the coronary arteries through a temporal sequence of biplane X-ray angiograms. On the other side, Shechter et al. [26] proposed a parametric model to decompose the motion field into independent cardiac and respiratory components.

Meanwhile, Blondel et al. [27] presented a method to compute 4D tomographic representations of coronary arteries from a single view of rotational monoplane angiograms based on 4D B-spline solids to model motions. Bouattour et al. [28] formulated the tracking problem as a 3D-2D registration problem in which the 3D model deforms in space to best fit the given set of 2D angiogram. In [29], a projection-based motion compen-

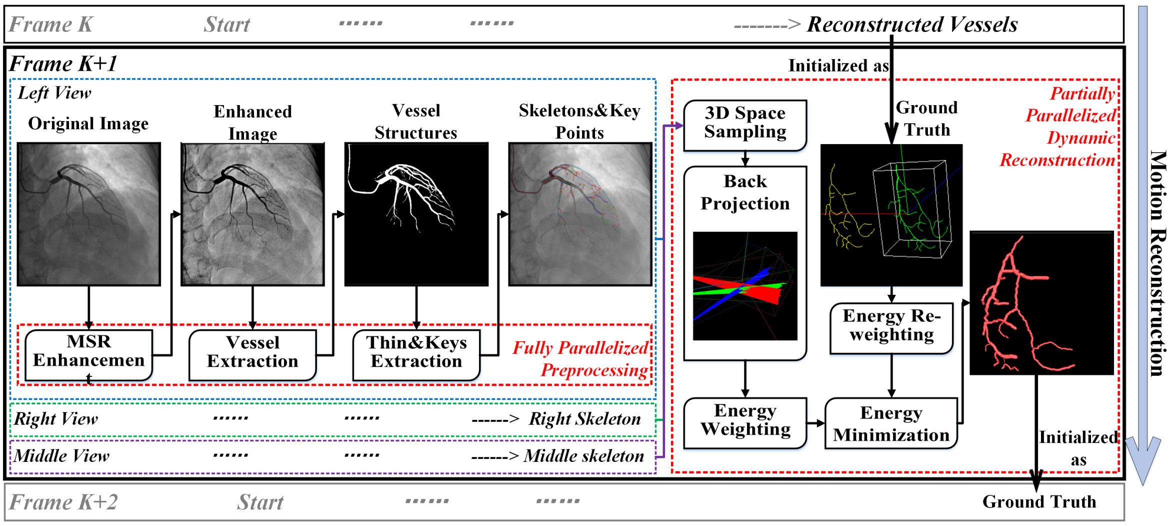


Fig. 1: Work pipeline. Left part: Image preprocessing. Right part: Dynamic reconstruction. From frame K to frame K+1: Reconstruction with time T.

sation and reconstruction method of coronary segments and cardiac implantable devices from rotational X-ray angiograms is developed.

Most of current tracking methods are either based on 3D to 2D projection or simple 2D registration using simple weighting term, leading to unsatisfactory results and less robustness. Methods based on mathematical analysis can simulate true motion of humans' heart, but they are specific and can not be generalized.

3 Vessel and Skeleton Extraction

Given X-ray angiograms, we design a realtime algorithm to extract the vessels and their skeletons as well as topologies robustly which are readily available for further reconstruction. We present a novel parallel method to extract vessels as well as their skeletons (Section 3.1) and finally the skeletons are segmented into segments split by bifurcated points to derive the topology structures (Section 3.2). All of these steps are run on GPU in realtime.

3.1 Vessel and Skeleton Extraction

Original angiograms acquired from the X-ray machine suffer from low contrast, low lumen, etc. To overcome these problems, we first apply the enhancement of radiography based on Multiscale Retinex (MSR). Then, we use the gain/offset method to fix the negative values. Finally we convolute the original images by four Gaussian filters with different scales and compute the weighted average of them, giving rise to satisfactory results.

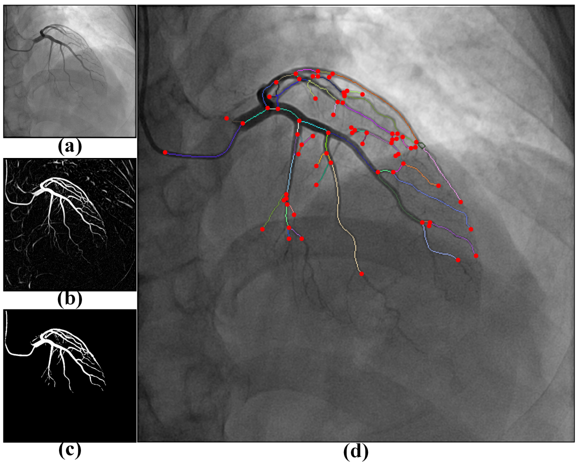


Fig. 2: Vessel extraction results. (a) Original image; (b) Possibility image; (c) Binary image; (d) Enlarged image with highlighted points and skeletons.

After the enhancement, we extract the vessel skeletons for further vessel reconstruction. We parallelize the algorithm proposed in [3] which relies on a multiscale Hessian matrix and apply it to MSR enhanced angiograms and extract tube-like structures. Extracted vessels could be found in Fig. 2 (b-c).

To simplify the reconstruction, we use skeletons to present vessel structures. Our skeleton extraction method mainly consists of three steps. First, a typical two-step thinning method is used to extract the rough skeletons. Second, we refine the skeletons to ensure its one-pixel width and use a pattern based method to extract it-

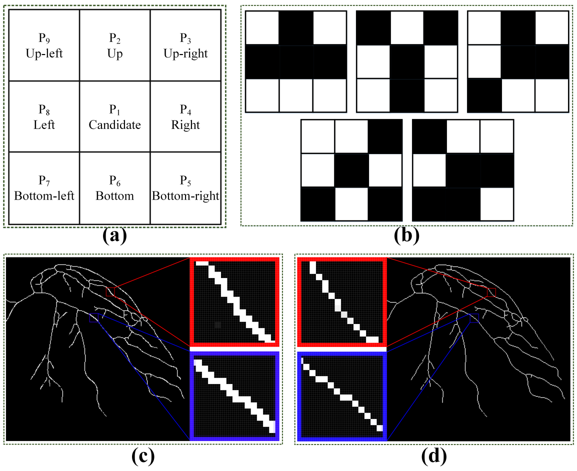


Fig. 3: Thinning refinement. (a) Designations of nine pixels in 3×3 window; (b) Bifurcation patterns; (c) Rough skeletons; (d) Refined skeletons.

s bifurcations and end points. At last, we collect the skeletons from bifurcations and derive the entire structure of the skeleton tree. All these three steps are done on GPU with the help of CUDA and achieve great efficiency.

In order to extract the skeleton, we use a two-step local-feature based thinning method [30] at beginning. Although the two-step thinning method provides a good initialization, the results are far from ideal enough for extracting the key points and structure of the skeletons because of the redundant points shown in Fig. 3 (c). We design a novel parallelized method to conduct the refinement while keeping efficiency.

First of all, we label points as *deleted* based on its neighbours. As shown in Fig. 3 (a), consider P_1 as the candidate, if $P_1 = 1$, P_1 is valid and vice versa. There are four ways that this point should be labeled as *deleted*: (1) $P_2 = P_8 = 1$ and $P_4 = P_5 = P_6 = 0$; (2) $P_2 = P_4 = 1$ and $P_6 = P_7 = P_8 = 0$; (3) $P_4 = P_6 = 1$ and $P_8 = P_9 = P_2 = 0$; (4) $P_6 = P_8 = 1$ and $P_2 = P_3 = P_4 = 0$.

Second, *deleted* information together with point neighbours are used to decide which points should really be deleted. Similarly, consider P_1 as one of the *deleted* labeled points according to Fig. 3 (a), there are two ways making this point really be deleted: (1) None of P_2, P_4, P_6, P_8 is labeled as *deleted*; (2) Only P_8 or P_6 are labeled.

Meanwhile, there are four ways deciding the removal of the neighbours of P_1: (1) If P_2 and P_4 are labeled as *deleted*, P_2 and P_4 should be removed; (2) If P_6 and P_8 are *deleted* while P_5 and P_9 are not *deleted*, P_6 and P_8 should be removed; (3) If P_4 and P_6 are labeled as *deleted*, P_4 and P_6 should be removed; (4) If P_2 and P_8 are *deleted* while P_3 and P_7 are not *deleted*, P_2 and P_8 should be removed.

Refined results are shown in Fig. 3 (d). Boxes with the same color are corresponding to the same areas between rough and refined skeletons. The refined skeletons would have just one-pixel width and easy to be processed by our following procedures.

3.2 Topology Identification

It is necessary to extract the key points to derive the topologies of the skeletons for further reconstruction. For bifurcations, we adopt a pattern-matching method, in which the five basic patterns are shown in Fig. 3 (b) and through changing these patterns by 90, 180, and 270 degree, all patterns could be found. On the other hand, for end points, we examine its eight neighbors and select those that just have one neighbor. Extracted key points could be found in Fig. 2 (d). Bifurcation and end points are labeled using filled circle marks.

Once the key points are extracted, skeleton segments are easy to be derived. Actually, skeleton segments start from bifurcation points and end at bifurcation or end points. In our approach, we start from the bifurcation points and examine the eight neighbors of current point.

Since one skeleton line consists of two bifurcations or one bifurcation with one end point, skeleton segments extracted using this method are redundant because each bifurcation point is computed twice. Therefore, we transverse all the skeletons and remove duplicate segments. The skeleton segments are drawn in Fig. 2 (d). Different colors indicate different skeleton branches.

4 4D Shape and Motion Reconstruction

Conventional image-based 3D reconstruction requires registration between different projection views. However, there are fewer features in the X-ray images than general camera images. Thus, it is hard to match the corresponding points between X-ray images reliably. The mistakes during registration may lead to much more huge errors during reconstruction. Meanwhile, it's also hard to add constraints such as the connectivity of the vessel neighbors, or prior knowledge for registration based methods , resulting in a significant waste of the various original information. Instead of relying on registration, we devise a space voxelization algorithm to recover the initial shape of the 3D vessels as an energy minimization problem (Section 4.1). Meanwhile, the continuously-moving frames provide strong clues for

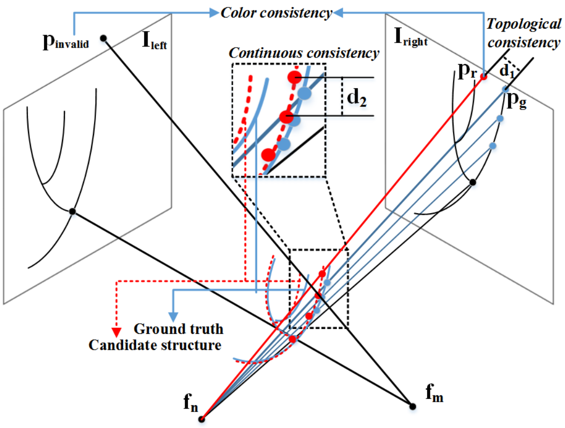


Fig. 4: Energy constraints determining the probability.

their subsequent frames. Based on the initial shapes, we generalize the above algorithm and design a robust motion tracking algorithm to reconstruct the dynamic vessels of the subsequent frames (Section 4.2).

4.1 Initial 3D Reconstruction

Instead of using conventional image reconstruction methods, we bypass the registration issue and discretize the space into voxels to formulate the 3D reconstruction problem as an energy minimization problem as described in Fig. 5. It is obvious that the unknown 3D skeleton should satisfy the following conditions: (1) The projection of every point of the 3D skeleton onto every 2D X-ray image is on the 2D skeleton; (2) The 3D skeleton is continuous with almost the same topology to the 2D skeleton except occlusion.

Actually, the 3D skeletons could be regarded as a skeleton tree consisting of several skeleton segments or branches. Each segment could be seen as being made up of discretely sampled points. In order to solve the optimization problem, we quantize the 3D space into discrete voxels. Each 3D point p in the sampled space could be assigned with a probability of belonging to the 3D skeletons. The probability could be determined by the following terms which are described in Fig. 4:

- Number of views in which the projected point of p is valid, which is called *color consistency*.
- Distance between p and its neighbors in the same 3D skeleton to ensure continuity which is called *continuous consistency*.
- Distance between the projected 2D point and its nearest valid 2D skeleton point on the same view, which is called *topological consistency*.

Finally, the sampled 3D space between the optical center and the intensifier is discretized into voxels

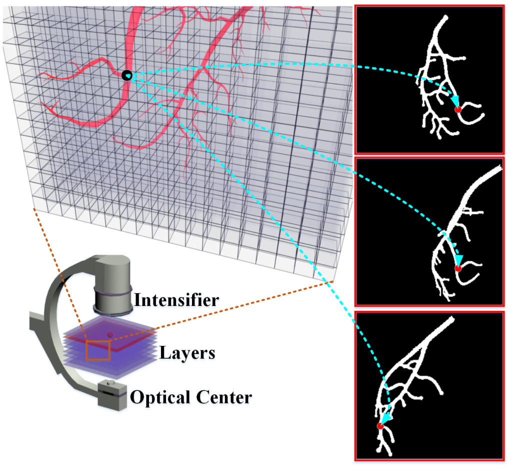


Fig. 5: 3D space sampling.

and could be regarded as a Markov Random Field such that the reconstruction problem could be formulated as an energy minimization problem with *color*, *continuous* and *topological* constraints. We use three views of angiograms in a cardiac cycle and choose the projection view I_1 as a reference view with least foreshortening and overlapping among all the views. The 3D space is divided into 3D slices which we call layer $L = (l_1, l_2 ... l_n)$ with a given depth interval. The 3D slicing and sampling is described in Fig. 5.

Each depth can be assigned a label l_i. Meanwhile, each skeleton point on reference view I_1 corresponds to one projected line from the source to the intensifier through all the layers. Therefore, for a given pixel p on I_1, the pair (p, l_i) uniquely determines a point in 3D space. So, the goal of 3D reconstruction is to optimally assign an label l_i to each p on the centerline of the reference view I_1. The energy function is defined as

$$E(f) = \sum_{p \in P} D_p(f_p) + \lambda \sum_{p,q \in N} V_{p,q}(f_p, f_q). \tag{1}$$

We define the $V_{p,q}(f_p, f_q)$ as the *Euclidean distance* to ensure the continuity between adjacent points p and q. And we define the $D_p(f_p)$ as the *color consistency* which is described as

$$D_p(f_p) = \frac{1}{(n-1)} \sum_{i=2}^{n} P_i(x, y), \tag{2}$$

where n is the number of back projected points, $P_i(x, y)$ is the projection value of point p on the i-th view, which is defined as

$$P_i(x, y) = \begin{cases} W_h, & p(x, y) \in I_i \\ W_l, & \mathcal{N}(p(x, y)) \notin I_i \\ \frac{1}{N} \sum_{i=1}^{N} V_i(x, y), & \text{otherwise} \end{cases}, \tag{3}$$

where $p(x, y) \in I_i$ denotes that $p(x, y)$ is on the skeleton of I_i. W_h and W_l are two constants that control the highest and lowest value (in our experiments they are set to 0.01 and 1.0, respectively). For a grey scale image, $\mathcal{N}(p(x, y))$ denotes point $p(x, y)$ and the 8 neighbors of $p(x, y)$. In our methods, if $p(x, y)$ can not be found in I_i, we compute its 8 neighbors and make the average value be the value of point $p(x, y)$. If none of its neighbors is valid, it is assigned to W_l.

We find the minimum of $E(f)$ using the Belief Propagation (BP) algorithm, which is comprised of two main steps, message propagation and energy minimization. In the message propagation step, the color of point $p(x, y) \in I_1$ is updated by $V_p = V_p + \alpha minD + (1 - \alpha)V_{p_{minD}}$, where α is a constant controlling the weight of the neighbors' *color consistency* and *dist consistency*. In our experiments, we set α to 0.85. $minD$ stands for the minimum distance from $p(x, y)$ to its neighbors, and $V_{p_{minD}}$ stands for the value of the minimum distance point. In our energy minimization, different from typical BP, the current energy of the i-th depth(l_i) is defined as $e_i(p_i) = min[\gamma D(p_i, q) + (1-\gamma)V(q) + e_{i-1}(q)]$, where q denotes the candidate 3D sample points corresponding to the projected point of $\mathcal{N}^o(p_i)$, and $\mathcal{N}^o(p_i)$ represents all neighbors of p_i except p_i itself.

At the end, we compute the minimum sum of all the grouped vessel skeletons' cost and obtain the optimal solution for the whole vessel skeleton tree. We obtain the initial 3D shape of the vessels.

4.2 Dynamic Reconstruction

Based on the initial reconstruction, we continue to generalize the above algorithm to track the vessel motions called *Dynamic Reconstruction*. Each image pair in the sequence could correspond to one special vessel skeleton in the motion sequence. Due to the characteristics of cardiac motion, these skeletons should be just changed slightly from its proceeding pose and changed slightly to its subsequent pose. Considering movement tracking of one frame to its next frame, reconstructed vessels of this frame can be regarded as prior knowledge to its subsequent pose. Deformed skeletons in the next frame should be close enough to the proceeding shape and the movement should be small. Since we treat the sample space as a discrete voxel space (a.k.a. Markov Random Field), this can be done by re-weighting the sampled valid points according to the distance to their corresponding prior points. Therefor, the energy term $E(f)$ from the standard energy is reformulated as

$$E^{'}(f) = E(f) + R_{p,k}(f_p, g_k), \tag{4}$$

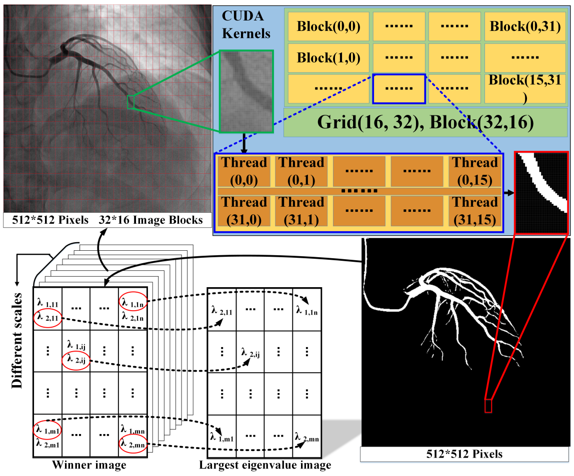


Fig. 6: GPU implementation for image preprocessing.

where $R_{p,k}(f_p, g_k)$ is the new constraint to attract the shape not going far away from its prior shape g_k. We use the minimum distance between p and its corresponding prior point k to evaluate p and re-weight its energy. $R_{p,k}(f_p, g_k)$ is a piecewise linear function based on $D_{p,g(k)}$ which is the minimum distance from p to the prior point. In our implementation, we set five pieces: $[0, 0.5], (0.5, 1.0], (1.0, 2.0], (2.0, 5.0], (5.0, +\infty)$.

According to Eq.(4), points with distance close enough should be with small energy and have more chance to be selected, while points with large distance should be with low energy to penalize its score. Even worse, it might be considered to be deleted (ignored) from the candidate queue. In our implementation, we may have thousands of prior points and even millions of sampled candidate points. To compute the minimum distance efficiently, we design a three-dimensional kd-tree. It provides the ability of searching nearest neighbors in $O(log(n))$ time. During each tracking frame, for every sampled point achieved from CUDA kernel, we compute its distance to the prior point and add up the weighted energy. Then, we use belief propagation to compute the optimal solution. At the end, this optimized result is initialized as the prior shape for the next frame.

5 CUDA-based Parallel Algorithms

Our methods are evaluated carefully at the very start of attempting to design parallelized implementation for performance enhancement, which allows us to take the full advantage of CUDA in skeleton tracking and extraction and gain great efficiency in belief propagation messaging. Image preprocessing with CUDA is described in Fig. 6. The angiograms are divided into image blocks and each pixel corresponds to one CUDA thread.

Parallelized Vessel Extraction. Since the computation on every pixel is independent, the vessel extraction algorithm is very suitable to be parallelized and every pixel is mapped to a CUDA thread for parallelization. For every angiogram among the imaging sequence and for every specified σ, our parallelized extraction method consists of the following steps. First, we build the Gaussian kernel mask depending on σ on CPU side and transfer them into the GPU. Second, we convolve the entire image using this Gaussian kernel and each pixel point on the image corresponds to one CUDA kernel. Third, we extract the eigenvalues and eigenvectors and compute the coefficients for each point's Hessian matrix. This is also done per kernel on GPU. Finally, we use a double swap buffer on GPU to compute the possibility of a pixel being part of vessel structures (refer to Eq. 15 of [3] for details). In all the procedures, except initialization, data are on GPU side and stored for further processing.

Parallelized Thinning. After vessel extraction, extracted vessels are stored on GPU to be further processed for skeleton tracking. During tracking, each valid vessel point is mapped to one CUDA kernel. In each kernel, we compute the point's eight neighbors and assign zero to these not fulfilling our thinning condition as described in Section 3.1. Finally, any points not belonging to the skeleton are removed.

In the key point extraction step, every valid point is mapped to one CUDA thread. For every point, we iterate four different degrees and five patterns (Fig. 3(b)) to identify whether it is a bifurcation. Meanwhile, points with just one neighbor are recognized as end points.

Once we get the key points, we collect the skeletons in a parallelized way. Every bifurcation point is mapped to a CUDA thread and we perform nearest neighbor search in each CUDA kernel ending at either bifurcation or end point to extract skeleton segment.

Parallelized Message Propagation. Due to the serial characteristics of BP, we are unable to run it completely in parallel. Nonetheless, energy weighting based on distance computation among thousands of neighbor points during message propagation can be greatly accelerated by parallelization. In such situations, we implement a parallelized distance computation method with the help of CUDA which is described in Algorithm 1. The problem can be abstracted as follows: given two point vectors of dimension m and n, we intend to compute the minimum distances from every point of the first vector to the points of the second vector. We compute the $n \times m$ distance matrix using $n \times m$ CUDA threads and then search for the minimum values of ever columns parallelly to derive a m vector. Finally we normalize the vector and add them to the candidate energy values. Comparison for our parallelized message propagation and a classical BP is illustrated in Fig. 7. The horizontal axis indicates the number of processed points. The vertical axis shows the processing time. The blue line and red line respectively indicate processing method using CPU and GPU which have shown our method has gained great efficiency in message propagation.

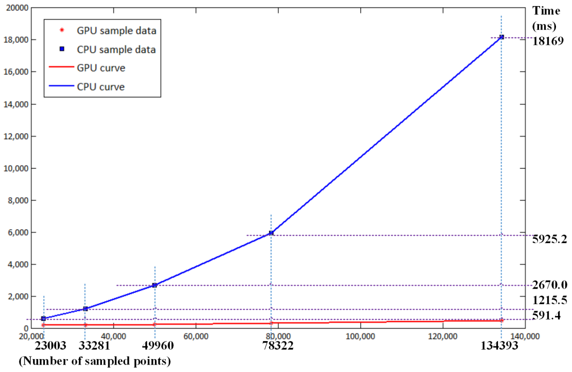


Fig. 7: Comparison of belief propagation in CPU and GPU. The time cost of CPU implementation increases fast while the sampled points increasing.

6 Experiments and Validation

6.1 Results

The reconstruction method is tested on both synthetic data and real clinical data which are definitely acquired at 5FPS and 15FPS speed for 512×512 images. Compared with the real data, the reconstruction of synthetic data is easy to be assessed because of the availability of the vessel ground truth. The final reconstruction results of synthetic data are shown in Fig. 8.

In the top row of Fig. 8, the yellow lines indicate the reconstructed skeletons using our method. The green lines indicate the ground truth of the synthetic data. The white box is the bounding box of the ground truth. Meanwhile, eight frames from synthetic data are shown on the top row of Fig. 12 and the reconstructed vessels are shown in the bottom row. It may be noted that we can reconstruct the vessels' motion as well.

Real clinical data contain more noise or even incomplete information. Some of the reconstructed dynamic vessels are shown in Fig. 9.

6.2 Validation

In order to evaluate the precision of our algorithm, we project the reconstructed skeletons back onto the image

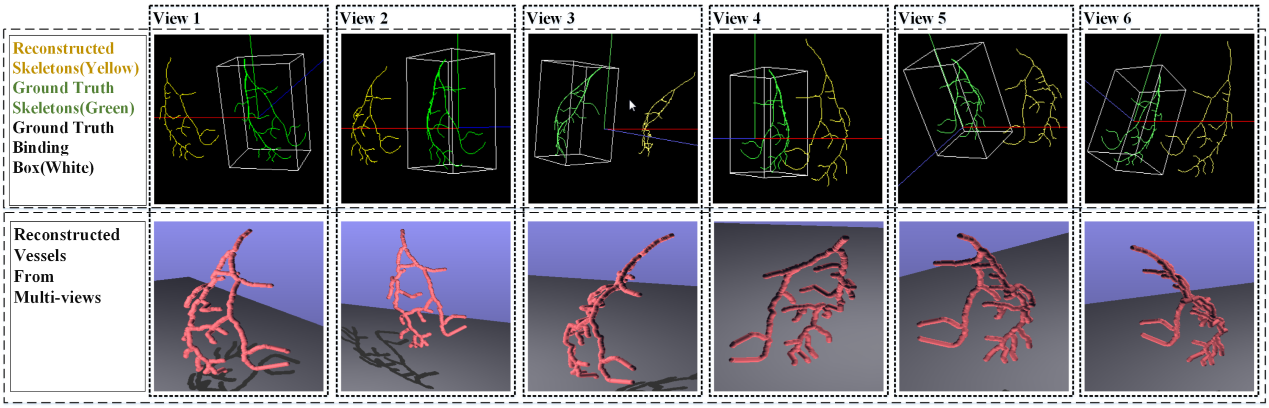


Fig. 8: Results of the synthetic data.

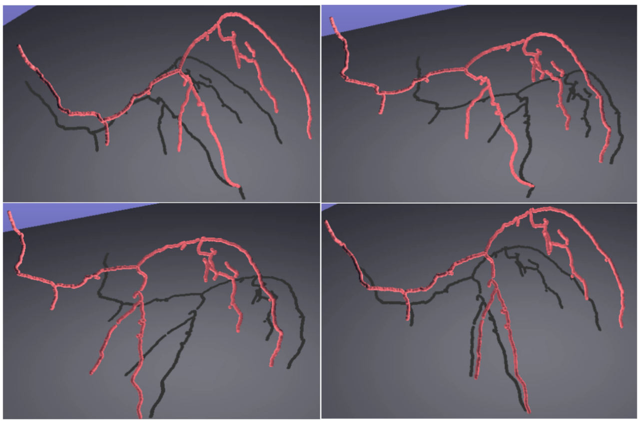
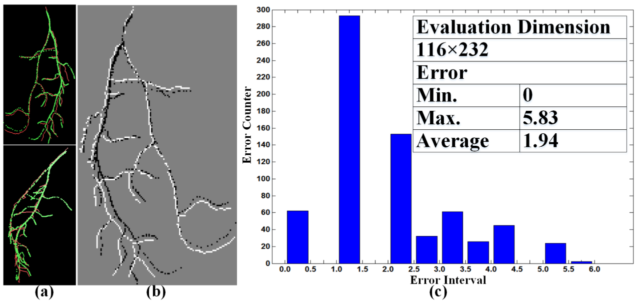
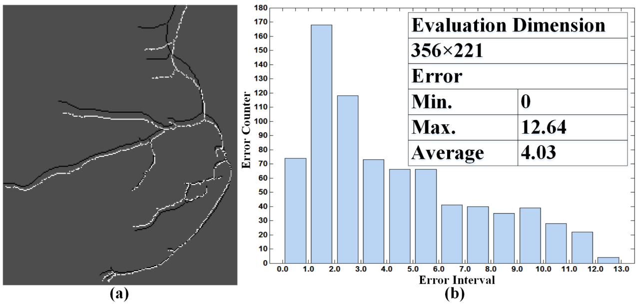


Fig. 9: Reconstruction from real data.



Evaluation Dimension	
116×232	
Error	
Min.	0
Max.	5.83
Average	1.94

Fig. 10: Re-projected error of the synthetic data and ground truth.



Evaluation Dimension	
356×221	
Error	
Min.	0
Max.	12.64
Average	4.03

Fig. 11: Re-projected error of the clinical data.

plane. Fig. 10 (a) shows the 3D ground truth and reconstructed skeletons. Fig. 10 (b) shows the projected real and reconstructed skeletons. Fig. 10 (c) shows the error statistics. Fig. 11 shows re-projected clinical data and the error statistics. For both Fig. 10 (c) and Fig. 11, the X axis represents the error interval, while the Y axis is the corresponding number. Errors are mostly distributed in the lower interval which demonstrates high accuracy of our method.

Because of heavily blurred and incomplete image data, some errors may occur in reconstruction during some frames as shown in the second frame of Fig. 12. The vessels marked in the white rectangle are missing in the second frame and they are recovered in the following frames, illustrating the robustness of our method that could combat the incomplete data.

6.3 Efficiency Analysis

Time analysis for our method is quantized and shown in Fig. 13. All these experiments are done on a workstation with one NVIDIA GTX-780 GPU, one Intel(R) Core(TM) i7-3770 CPU and 8GB RAM using a set of 37-frame sequence.

7 Conclusion and Discussion

We have developed a new 4D dynamic vessel reconstruction system from X-ray angiograms. The uniqueness of our system is its simultaneous handling on structure, shape, and motion during vessel reconstruction. At the technical core of our system are the parallel algorithms towards interactive performance. Specifically, at the vessel skeleton extraction stage, we developed

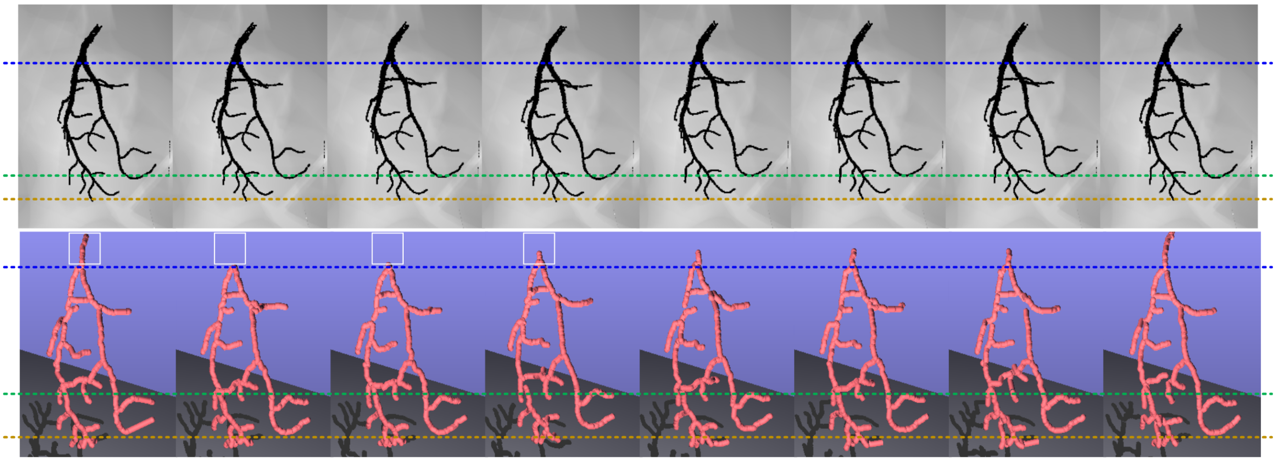
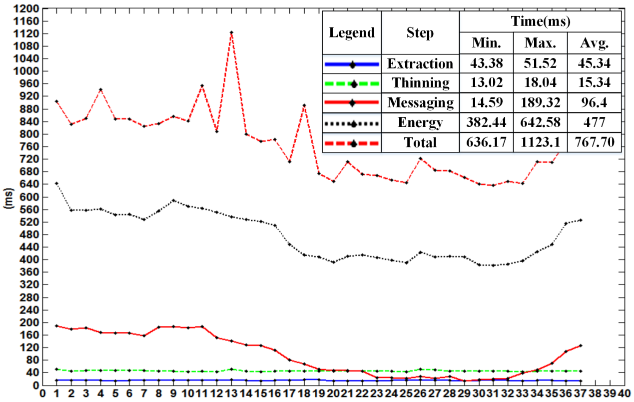


Fig. 12: Vessel motion tracking between sequential frames.



Legend	Step	Time(ms)		
		Min.	Max.	Avg.
—◆—	Extraction	43.38	51.52	45.34
- -◆- -	Thinning	13.02	18.04	15.34
—◆—	Messaging	14.59	189.32	96.4
···◆···	Energy	382.44	642.58	477
- -◆- -	Total	636.17	1123.1	767.70

Fig. 13: The overall time analysis together with the detailed statistics of each step based on 37 frames of synthetic data.

a realtime parallel method to extract vessels as well as their skeleton and topology from X-ray views. At the reconstruction stage, we have presented a space voxelization algorithm to formulate the dynamic reconstruction problem as an energy optimization problem solved by belief propagation without explicit registration. The experimental results from both synthetic and clinical data have shown that our method is robust for noise and even incomplete data because of the algorithms' global optimization nature. Our immediate goal for ongoing work is to continue to improve our system and its functionalities towards clinic trial in the near future.

Acknowledgements: This work is supported in part by National Natural Science Foundation of China (Grant No. 61190120, 61190121, 61190125, 61300068, 61300067), National Science Foundation of USA (Grant No. IIS-0949467, IIS-1047715, and IIS-1049448), the National High Technology Research and Development Program(863 Program) of China (Grant No. 012AA011503), Postdoctoral Science Foundation of China (Grant No. 2013M530512).

References

1. A. Hoover, V. Kouznetsova, M. Goldbaum, Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response, IEEE Transactions on Medical Imaging 19 (3) (2000) 203–210.
2. Q. Li, J. You, D. Zhang, Vessel segmentation and width estimation in retinal images using multiscale production of matched filter responses, Expert Systems with Applications 39 (9) (2012) 7600–7610.
3. A. F. Frangi, W. J. Niessen, K. L. Vincken, M. A. Viergever, Multiscale vessel enhancement filtering, in: MICCAI98, Springer, 1998, pp. 130–137.
4. A.-P. Condurache, T. Aach, Vessel segmentation in angiograms using hysteresis thresholding, in: IAPR Conference on Machine Vision Applications, 2005, pp. 269–272.
5. L. Sarry, J.-Y. Boire, Three-dimensional tracking of coronary arteries from biplane angiographic sequences using parametrically deformable models, IEEE Transactions on Medical Imaging 20 (12) (2001) 1341–1351.
6. C. Cañero, F. Vilariño, J. Mauri, P. Radeva, Predictive (un) distortion model and 3-d reconstruction by biplane snakes, IEEE Transactions on Medical Imaging 21 (9) (2002) 1188–1201.
7. A. C. Dumay, J. J. Gerbrands, J. H. Reiber, Automated extraction, labelling and analysis of the coronary vasculature from arteriograms, The International Journal of Cardiac Imaging 10 (3) (1994) 205–215.
8. S. R. Aylward, E. Bullitt, Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction, IEEE Transactions on Medical Imaging 21 (2) (2002) 61–75.
9. A. Sen, L. Lan, K. Doi, K. R. Hoffmann, Quantitative evaluation of vessel tracking techniques on coronary angiograms, Medical Physics 26 (5) (1999) 698–706.
10. C. Kirbas, F. Quek, A review of vessel extraction techniques and algorithms, ACM Computing Surveys (CSUR) 36 (2) (2004) 81–121.
11. E. Wellnhofer, A. Wahle, I. Mugaragu, J. Gross, H. Oswald, E. Fleck, Validation of an accurate method for

Algorithm 1 Message Propagation

```
Input:
    $pCandi$,sampled candidate points.
    $pIMG$,2D image.$szData$,sizes of skeletons.
    $szLn$,line counts.$pNum$,3D valid point counts
Output:
    propagated candidate points.

    function bpMessage($pNum, pCandi, pIMG, szData, szLn$)
        $pCollect \leftarrow$collectValidSize($pNum$)
        $nMAXVSZ \leftarrow max(pCollect)$
        $pCache \leftarrow allocCache(nMAXVSZ)$
        for $i = 0 \rightarrow ITERATIONS$ do
            bpmessage_oneStep($pCandi, pCache$)
        end for
    end function

    function bpmessage_oneStep($pCandi, pCache$)
        $pValid1 \leftarrow pCandi[i], pValid2 \leftarrow pCandi[j]$
        $fMinD \leftarrow$cudaGetMinD($pValid1, pValid2, pCache$)
        for $k = 0 \rightarrow szValid1$ do
            $pValid1[k].fEnergy+ = fMinD$
        end for
    end function

    function cudaGetMinD($pValid1, pValid2, pCache$)
        $threadId \leftarrow getGlobalIdx2D()$
        $nDataIdx \leftarrow getGlobalIdx1D()$
        $pCache[threadId] \leftarrow$dist($pValid1[x], pValid2[y]$)
        $syncthreads()$
        $fMinD \leftarrow pCache[nDataIdx]$
        for $i = 0 \rightarrow nSZ0$ do
            if $pData[nDataIdx + i * nSZ1] < fMinD$ then
                $fMinD \leftarrow pData[nDataIdx + i * nSZ1]$
            end if
        end for
        return $fMinD$
    end function

    function collectValidSize($pValidPtsNum$)
        $threadId \leftarrow getGlobalIdx2D()$
        if $threadId > IMAGE\_DIM$ then return ;
        end if
        if $pValidPtsNum[threadId] > 0$ then
            $n \leftarrow atomicAdd(gValidCounter, 1)$
            $pDEVValidSZ[n] \leftarrow pValidPtsNum[threadId]$
        end if
    end function
```

three-dimensional reconstruction and quantitative assessment of volumes, lengths and diameters of coronary vascular branches and segments from biplane angiographic projections, The International Journal of Cardiac Imaging 15 (5) (1999) 339–353.

12. J. C. Messenger, S. J. Chen, J. D. Carroll, J. Burchenal, K. Kioussopoulos, B. M. Groves, 3d coronary reconstruction from routine single-plane coronary angiograms: clinical validation and quantitative analysis of the right coronary artery in 100 patients, The International Journal of Cardiac Imaging 16 (6) (2000) 413–427.
13. R. R. Gollapudi, R. Valencia, S. S. Lee, G. B. Wong, P. S. Teirstein, M. J. Price, Utility of three-dimensional reconstruction of coronary angiography to guide percutaneous coronary intervention, Catheterization and Cardiovascular Interventions 69 (4) (2007) 479–482.
14. B. Movassaghi, V. Rasche, M. Grass, M. A. Viergever, W. J. Niessen, A quantitative analysis of 3-d coronary modeling from two or more projection images, IEEE Transactions on Medical Imaging 23 (12) (2004) 1517–1531.
15. K. Sprague, M. Drangova, G. Lehmann, P. Slomka, D. Levin, B. Chow, et al., Coronary x-ray angiographic reconstruction and image orientation, Medical physics 33 (2006) 707–718.
16. E. Hansis, D. Schäfer, O. Dössel, M. Grass, Projection-based motion compensation for gated coronary artery reconstruction from rotational x-ray angiograms, Physics in medicine and biology 53 (14) (2008) 3807–3820.
17. T. V. Nguyen, J. Sklansky, Reconstructing the 3-d medial axes of coronary arteries in single-view cineangiograms, IEEE Transactions on Medical Imaging 13 (1) (1994) 61–73.
18. J. A. Fessler, A. Macovski, Object-based 3-d reconstruction of arterial trees from magnetic resonance angiograms, IEEE Transactions on Medical Imaging 10 (1991) 25–39.
19. I. Liu, Y. Sun, Fully automated reconstruction of three-dimensional vascular tree structures from two orthogonal views using computational algorithms and production-rules, Optical Engineering 31 (10) (1992) 2197–2207.
20. S. Ruan, A. Bruno, J.-L. Coatrieux, Three-dimensional motion and reconstruction of coronary arteries from biplane cineangiography, Image and Vision Computing 12 (10) (1994) 683–689.
21. J. Puentes, C. Roux, M. Garreau, J.-L. Coatrieux, Dynamic feature extraction of coronary artery motion using dsa image sequences, IEEE Transactions on Medical Imaging 17 (6) (1998) 857–871.
22. C. Ingrassia, P. Windyga, M. Shah, Segmentation and tracking of coronary arteries, in: BMES/EMBS Conference, 1999, Vol. 1, IEEE, 1999, pp. 203–203.
23. S.-Y. Chen, J. D. Carroll, Kinematic and deformation analysis of 4-d coronary arterial trees reconstructed from cine angiograms, IEEE Transactions on Medical Imaging 22 (6) (2003) 710–721.
24. B. Naegel, N. Passat, C. Ronse, Grey-level hit-or-miss transforms-Part I: Unified theory, Pattern Recognition 40 (2) (2007) 635–647.
25. G. Shechter, F. Devernay, E. Coste-Manière, A. Quyyumi, E. R. McVeigh, Three-dimensional motion tracking of coronary arteries in biplane cineangiograms, IEEE Transactions on Medical Imaging 22 (4) (2003) 493–503.
26. G. Shechter, J. R. Resar, E. R. McVeigh, Displacement and velocity of the coronary arteries: cardiac and respiratory motion, IEEE Transactions on Medical Imaging 25 (3) (2006) 369–375.
27. C. Blondel, G. Malandain, R. Vaillant, N. Ayache, Reconstruction of coronary arteries from a single rotational x-ray projection sequence, IEEE Transactions on Medical Imaging 25 (5) (2006) 653–663.
28. S. Bouattour, R. Arndt, D. Paulus, 4d reconstruction of coronary arteries from monoplane angiograms, in: Computer Analysis of Images and Patterns, Springer, 2005, pp. 724–731.
29. G. Schoonenberg, R. Florent, P. Lelong, O. Wink, D. Ruijters, J. Carroll, B. ter Haar Romeny, Projection-based motion compensation and reconstruction of coronary segments and cardiac implantable devices using rotational x-ray angiography, Medical Image Analysis 13 (5) (2009) 785–792.
30. T. Y. Zhang, C. Y. Suen, A fast parallel algorithm for thinning digital patterns, Commun. ACM 27 (3) (1984) 236–239.