Neurocomputing 86 (2012) 24–32

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Temporal Spectral Residual for fast salient motion detection

Xinyi Cui [a,*], Qingshan Liu [b], Shaoting Zhang [a], Fei Yang [a], Dimitris N. Metaxas [a]

[a] Department of Computer Science, Rutgers University, Piscataway, NJ, USA
[b] Nanjing University of Information Science and Technology, Nanjing, China

ARTICLE INFO

Article history:
Received 8 October 2011
Received in revised form
27 December 2011
Accepted 27 December 2011
Communicated by D. Tao
Available online 23 February 2012

Keywords:
Computer vision
Motion saliency detection
Spectral Residual

ABSTRACT

Motion saliency detection aims at finding the semantic regions in a video sequence. It is an important pre-processing step in many vision applications. In this paper, we propose a new algorithm, Temporal Spectral Residual, for fast motion saliency detection. Different from conventional motion saliency detection algorithms that use complex mathematical models, our goal is to find a good tradeoff between the computational efficiency and accuracy. The basic observation for salient motions is that on the cross section along the temporal axis of a video sequence, the regions of moving objects contain distinct signals while the background area contains redundant information. Thus our focus in this paper is to extract the salient information on the cross section, by utilizing the off-the-shelf method Spectral Residual, which is a 2D image saliency detection method. Majority voting strategy is also introduced to generate reliable results. Since the proposed method only involves Fourier spectrum analysis, it is computationally efficient. We validate our algorithm on two applications: background subtraction in outdoor video sequences under dynamic background and left ventricle endocardium segmentation in MR sequences. Compared with some state-of-art algorithms, our algorithm achieves both good accuracy and fast computation, which satisfies the need as a pre-processing method.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Saliency detection has attracted much attention in recent years. Different from conventional segmentation problem of separating the whole scene into discrete parts, saliency detection aims at finding semantic regions and filtering out the unimportant area. The idea of saliency detection comes from human visual system, where the first stage of human vision is a fast but simple pre-attentive process. Saliency detection is an important topic in computer vision, since it provides a fast pre-processing stage for many vision applications.

Saliency detection on both images and videos has been studied in recent years. For image-based saliency detection, we want to find salient regions that are different from the background, e.g. a deer in a forest. Many algorithms have been proposed. Itti and Koch have designed a model of simulating the human visual search process to detect saliency in static images [1–3]. It has also been extended to visual recognition tasks [4,5]. Recently Hou and Zhang [6] proposed a fast Fourier spectrum residual analysis for image saliency detection.

Video-based saliency detection is different from image saliency detection problem. It aims at finding salient motions from the background in a video sequence. The salient motion can be a running person on a beach or a beating heart in a MR sequence. Locating salient moving objects accurately and efficiently is a critical pre-processing step to many video understanding applications [7,8]. It can also be used in video qualify assessment [9]. But it is still a challenging problem, since videos or 3D volume data can have various background motions. To solve this problem, many algorithms have been proposed: Gaussian Mixture Model [10], Nonparametric Kernel Density Estimation [11], adaptive KDE combined with motion information [12], Bayesian Learning approach [13], Linear Dynamic Model [14], Robust Kalman Filter [15], etc.

Though these models have achieved good results, they all use sophisticated models or algorithms. They are not fast enough as a pre-processing method. In this paper, we propose a fast motion saliency detection method Temporal Spectral Residual. Different from the complex background modeling, the proposed method is computationally efficient. It does not need any initial labeling and is free of training. It satisfies the need of a pro-processing method. The main idea of our method comes from the observation: the moving trajectories of salient objects on the cross section along temporal slices have its salient region and redundant area. Thus our focus is to extract the salient information on the cross section, by utilizing the off-the-shelf algorithm Spectral Residual [6]. Spectral Residual is a method to find salient information on a 2D image. Majority voting strategy is also introduced to produce a robust result. Since our algorithm only involves Fourier Spectrum

* Corresponding author. Tel./fax: +1 732 445 2795.
E-mail address: xycui@cs.rutgers.edu (X. Cui).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2011.12.033

Analysis, it is computationally efficient. We validate our assumptions on two different applications: (1) background subtraction under dynamic background in video sequences; (2) cardiac motion localization in MR sequences. The experiments show that our method works well for diverse applications and can handle various dynamic background motions. It also shows that our algorithm is computationally efficient.

It is worth mentioning that our algorithm does not aim at finding the motion perfectly, our goal is to find a good tradeoff between the accuracy and efficiency. The experiments show that our algorithm can find salient objects in a good quality, and also run efficiently. It provides a good pre-processing method for other applications. Section 2 gives the details of our algorithms. The experiment Section 3 will talk about more information of the literature review and the details of the two applications. Experiments validate the effectiveness and efficiency of our algorithm.

2. Methodology

We propose a new and efficient method to find salient motion regions in video sequences. The main idea is to roughly remove the redundant part of a volume data (the static part of temporal slices) and keep the salient motion regions. This algorithm is able to provide reliable motion regions but does not need initial labeling or any training data. Our method uses Spectral Residual (SR) algorithm [6] as the building bricks. SR is a saliency detection algorithm on 2D images by doing statistics of Fourier Transformation. In this section, we will first give a brief introduction to SR algorithm, and then talk about our main method.

2.1. Spectral residual

Different from summarizing the properties of target objects, Hou and Zhang's Spectral Residual algorithm [6] focuses on exploring the properties of the background by exploiting the power of log spectrum. It is based on the observation that log spectra of different images share similar trends, though each containing statistical singularities. The similarities imply redundancies. If the similarity (trend of local linearity of natural images) is removed, the remaining singularity (spectral residual) should be the innovation of an image corresponding to its visual saliency. The main idea of Spectral Residual is to remove the redundant part of image's spectrum. It only needs Fourier Transformation of an image, so this algorithm is computationally efficient. The details are as follows:

Given an image I and its Fourier Spectrum f, its log spectrum representation is $L(f)=\log(A(f))$, where $A(f)$ is the magnitude of f. The spectral residual $R(f)$ is obtained by $R(f)=L(f)-\overline{L(f)}$, where $\overline{L(f)}$ denotes the general shape of log spectra. Spectral Residual algorithm proposed an approximate solution of $\overline{(f)}$, obtained by a local average filter $h_n(f)$. Transforming the spectral residual $R(f)$ back to spatial domain, the high value pixels correspond to the salient regions.

2.2. Temporal Spectral Residual

SR algorithm has been successfully applied to 2D natural scene images. But it cannot be directly used for motion saliency in video sequences or 3D volume, since the saliency information of motions is significantly different from pixel intensity distribution.

For a motion saliency detection task, we have some basic observations in its general cases: (1) the region of foreground is usually smaller than that of background; (2) background motion is usually smaller than foreground object motion; (3) background has more regular patterns, even when dynamic background exists. Therefore, if we analyze the temporal slices of videos, the unexpected portion or distinct motion trajectories indicate the foreground moving objects. Fig. 1(b) shows an example, where the walking person forms a distinct trajectory from the background in a temporal slice on the XT plane.

To detect such distinct trajectories, we design an efficient algorithm Temporal Spectral Residual, by making use of the spectral residual in temporal domain. Different from conventional background modeling, the proposed method is computationally efficient. It does not need any training or labeling of initial frames. The general procedure is summarized in Fig. 1. We will talk about the details in the following section.

Denote the temporal axis of the video sequence or 3D volume as T, then the temporal slices are represented by XT and YT, where X and Y are the axes of each image frame. Samples of temporal slices are shown in Fig. 1(b) and (c). To detect distinct trajectories of moving objects, we apply Spectral Residual (SR) algorithm introduced in Section 2.1 on each temporal slice on XT plane and YT plane respectively to obtain the salient pixels. Without loss of generality, we take the temporal slices I_{XT} on XT plane for example, such as Fig. 1(b). Its spectral residual is calculated as

$$S_{XT_j}=\sigma(SR(I_{XT_j})) \tag{1}$$

where j is the index along Y-axis. $SR(\cdot)$ denotes the Spectral Residual algorithm applied on slice image I_{XT_j}. As described in the previous section, pixels with higher energy value are more

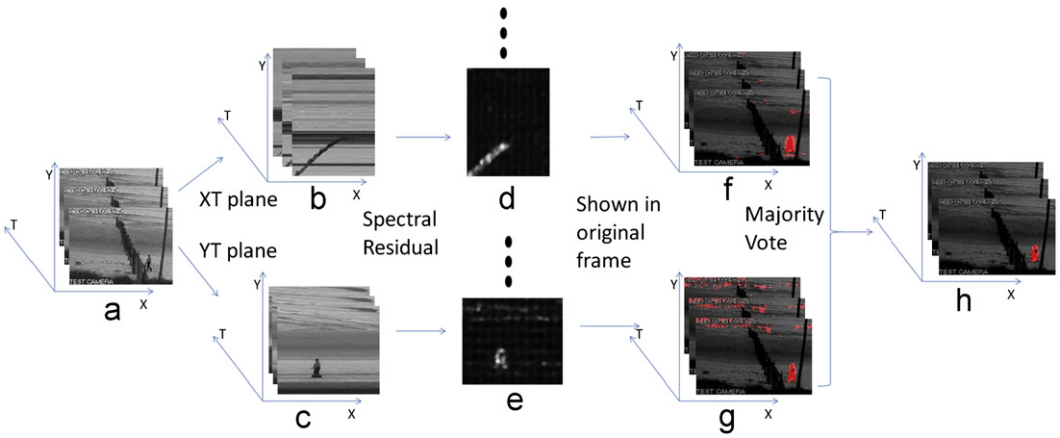


Fig. 1. Temporal Spectral Residual flow chart: (a) the original image sequences; (b) temporal slices on XT plane; (c) temporal slices on YT plane; (d) saliency map on XT plane; (e) saliency map on YT plane; (f) saliency map of XT projected back in original image sequences; (g) saliency map of YT projected back in original image sequences; (h) the final saliency map after majority vote.

salient than those with lower values. $\sigma(\cdot)$ is a threshold operator to remove low salient values:

$$\sigma(x) = \begin{cases} 1 & \text{if } x > t \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where t is the threshold to filter out pixels with low values and keep salient pixels. More details on how to define the threshold will be discussed in the Experiments section.

To show the reason why temporal spectral residual works for motion saliency detection, we performed experiments on a toy sample, shown in Fig. 2. The image has multiple lines. Each line has a constant intensity I_c plus a random noise I_{noise}. The image is obtained by $I = I_c + I_{noise}$. Both I_c and I_{noise} are generated randomly, where I_{noise} is a uniform distribution of [0,0.1]. These multiple lines simulate scene background on a temporal slice. It also has a white line, a black circle and a black skinny line on the left, simulating the trajectories of moving objects. Fig. 2(b) shows the spectral amplitude of (a); (d) shows the spectral residual amplitude; (b) is the result of performing Spectral Residual. The highlight region shows saliency, which almost matches the three salient shapes in the image.

We also validate our assumption on real outdoor video sequences. This video is a scenario of a person walking on the beach. The pedestrian's movement is the salient motion. Fig. 1(d) shows a saliency map S from a temporal slice. The major moving trajectory has been successfully detected.

Fig. 1(d) and (e) shows that salient motions appeared on both S_{XT} and S_{YT}. Moving objects generally have distinct trajectories on both temporal planes. But the motions from background noises are usually orderless and random, so they rarely have distinct patterns on both planes. Thus we can filter out the noisy pixels by collecting evidence on both planes. However when a salient pixel moves mainly along one direction, the majority voting is likely to discard it. The major salient regions still remain. In order to make the result more robust, we propose a majority voting strategy. Consider a video sequence as a 3D volume, we denote C as the 3D volume that contains all saliency maps along the temporal plane. Saliency majority voting is performed on the C_{XT} and C_{YT}, then we can obtain the final salient map by

$$C = C_{XT} \cdot C_{YT} \quad (3)$$

In Fig. 1(h), the major salient motion regions have been detected.

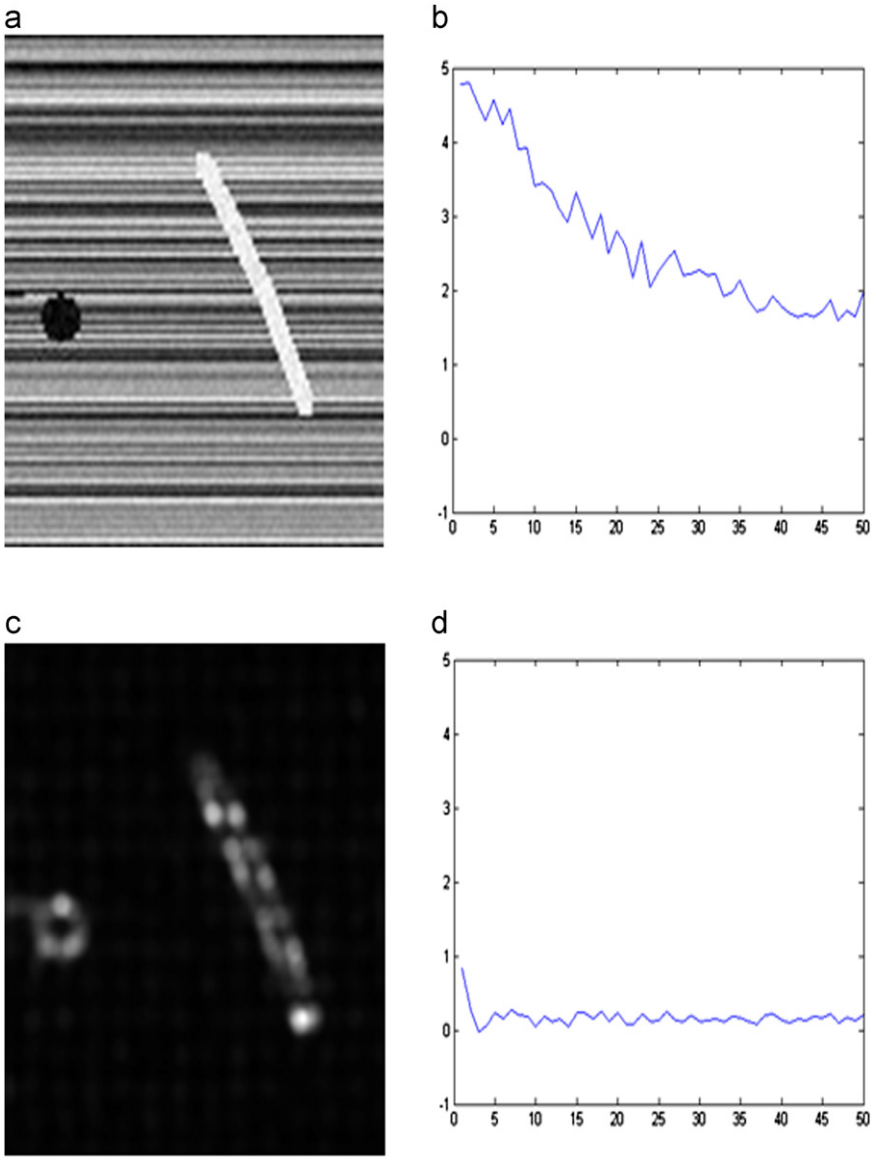


Fig. 2. Toy example: (a) original image, (b) log spectrum curve, (c) saliency image, (d) spectral residual curve.

2.2.1. Computational complexity analysis

As described above, our method is only based on Fourier spectrum analysis, thus it is computationally efficient. Fourier Transformation has its fast algorithm FFT, with the time complexity $O(n^2*\log(n))$, given an n by n matrix. A video clip with frame size M by N and frame number T only takes $2(M+N)*O(\mathrm{FFT})$. Then the running time to process one frame is

$$t = 2(M+N)*O(\mathrm{FFT})/T = 2(M+N)/T*O(n^2*\log(n)) \quad (4)$$

Note that in order to have the optimal performance of FFT, n needs to be an exponential term of 2. In our case, n is $\exp(2,\lceil\log_2(\min(M,N))\rceil)$. Our algorithm scales well. The computation time is linear to the frame number. We take a video block M by N by T as a unit for TSR each time. Thus we only need T_{all}/T times of TSR calculation, which is very efficient. T_{all} denotes the overall length of a video sequence. Section 3 will have more detailed discussion of running time analysis and comparison with other state-of-art algorithms.

3. Experiments

In this section, we apply our method in two applications: (1) background subtraction in video sequences with various types of backgrounds; (2) left ventricle segmentation from 2D MR sequences. We validate the robustness and computational efficiency of our algorithms on these two applications.

3.1. Background subtraction in video sequences

Background subtraction in video sequences detects moving foreground objects from background scenes. It is an important step in many vision problems. It separates objects from background clutter. The result of background subtraction can be used for tracking, object identification, action or activity recognition. And it is crucial to many applications including surveillance, human computer interaction, animation and video event analysis. One challenging problem for background subtraction is dynamic background, such as fountain, swaying tree leaves, ocean ripples, etc. The dynamic background makes it hard to separate the foreground movement from the background.

Many algorithms have been proposed in recent years. The naive approach to background subtraction is frame difference [16,17]. It thresholds the difference between two/three consecutive frames. Though this algorithm is fast, it does not handle dynamic background. In order to handle dynamic background, more complicated models have been studied, including Gaussian Mixture Model [10], Nonparametric Kernel Density Estimation [11], Adaptive KDE combined with motion information [12], Bayesian Learning approach [13], Linear Dynamic Model [14], Robust Kalman Filter [15]. These models have achieved good results, but they use complex mathematical model with high computational cost.

The dynamic background is a key issue for many background subtraction problems. Thus selecting a good threshold to remove the pixels on the dynamic background area is a key issue. In order to remove the dynamic background from video sequences, we propose adaptive threshold to remove the salient pixels from the Temporal Spectral Residual (TSR) algorithm.

After applying SR algorithm on each temporal slice, we obtain a saliency map for each slice. It is hard to tell where the background noise is from a single slice. But it is easy to recognize it if we can have information from the whole video sequence. Treating the video sequence as a cube, salient pixels on this cube belong to either the motion regions or dynamic background noises. Thus the adaptive threshold is designed to take all pixels from the whole video into account. Assuming values of salient energy from the video volume satisfy a Gaussian distribution $[\mu,\sigma]$, then an adaptive threshold is defined as $t_{adaptive} = \mu_{global} + 2\sigma_{global}$. Pixels with a lower energy than $t_{adaptive}$ are rejected. This procedure refines the motion area by filtering out the pixels with relatively higher value locally in temporal slices but still low value globally. Experiments below show that the adaptive threshold produces reliable results for various dynamic background.

To illustrate the effectiveness of our method, we tested the algorithm on four different types of background motions: static, moderately static, partially dynamic and fully dynamic.

Static background: The first experiment is conducted on relatively static background. The video clip is obtained from CAVIAR dataset [18]. It is captured by a fixed CCD camera in a shopping center. A person is walking across the lobby, while the background is static but cluttered. We compared our algorithm with one baseline method: Frame Difference (FD) and one widely used method Gaussian Mixture Model (GMM). FD is the fastest way to model background by simply subtracting the current frame from the previous frame. It is fast but the resulting quality is low, since it does not consider any long-term motion or background motion. GMM has been widely used in realistic scenes and can handle a range of realistic scenarios. It builds a mixture model of Gaussian distribution for each pixel along temporal axis. Parameters need to be updated using an online Expectation Maximization algorithm. It gives more reliable result than FD but it takes longer time. The result is shown in Fig. 3, where (d) is result from our TSR method. Comparing to these two methods, the result image quality is much better than FD, and comparative to GMM; while the computational efficiency is much higher than GMM and comparative to FD. The details of running time will be discussed later.

The second experiment is a video with moderately static background. A car is moving in the rain. Video is obtained from [11]. We compared our algorithm with three methods: FD, GMM and Kernel Density Estimation (KDE) as shown in Fig. 4. KDE is also an algorithm to build mathematical models for each pixel. It keeps samples of intensity values pixel and uses samples to estimate density function. In this video, the background our method obtained is cleaner than all three methods. The foreground qualify of our method is slightly lower than KDE, but it is good enough as an initialization for many vision applications.

Dynamic background: To further show the effectiveness of our algorithm, we conducted experiments on two more complicated videos with dynamic background. The third video "Beach" shown in Fig. 5 includes partially dynamic background. The upper part has dynamic motions from sea waves while the lower part of the

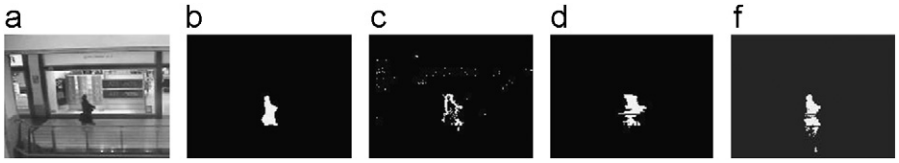


Fig. 3. Experimental results on shopping center video: (a) original frame, (b) ground truth, (c) frame difference, (d) Temporal Spectral Residual, (e) Gaussian Mixture Model.

Fig. 4. Experimental results on rain video: (a) original frame, (b) ground truth, (c) frame difference, (d) Temporal Spectral Residual, (e) Gaussian Mixture Model, (f) Kernel Density Estimation [11].

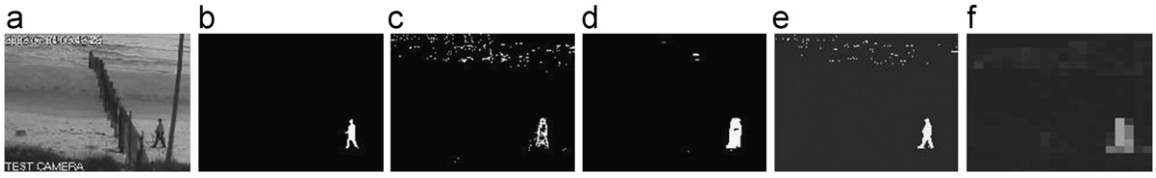


Fig. 5. Experimental results on beach video: (a) original frame, (b) ground truth, (c) frame difference, (d) Temporal Spectral Residual, (e) Gaussian Mixture Model, (f) Linear Dynamic Model [14].

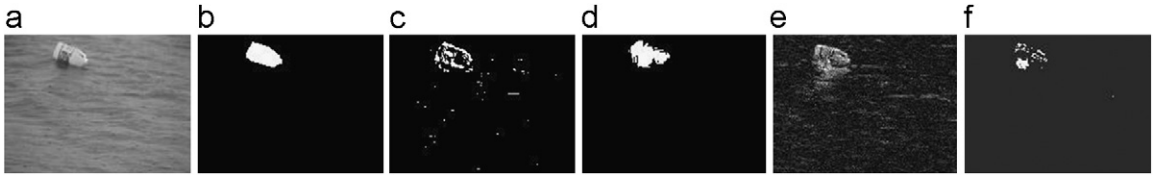


Fig. 6. Experimental results on water bottle video: (a) original frame, (b) ground truth, (c) frame difference, (d) Temporal Spectral Residual, (e) Gaussian Mixture Model, (f) Robust Kalman Filter [15].

scene is beach. Since it has dynamic background motion, it is generally a hard task for traditional background subtraction methods. We compared our approach with FD, GMM and Linear Dynamic Model (LDM). Our method produces almost the best result considering the both factors of accuracy and computational speed. Methods like FD and GMM fail to model the dynamic motion of the sea waves. LDM also has good results, but it takes longer time. Linear Dynamic Model builds a predictive model to capture the most important variation based on sub-space analysis of the signal, thus it is more computationally expensive.

It is worth mentioning that some pixels on the moving object in Fig. 5(d) are missing, due to the refinement procedure, but major salient regions still remain. The missing pixels can be easily compensated by simple post-processing with local appearance features. Additionally, the salient region of our result is slightly larger than the ground truth, due to the fact that Fourier Transformation is more sensitive to edges and boundaries, so the pixels around edges are captured. But there is merely no harm as a pre-processing stage aiming to provide salient candidates.

The fourth video ("Water Bottle" [15]) has a water bottle floating on water surface. The water surface is moving, and the whole scene is in dynamic texture. We compare our algorithm with FD, GMM and Robust Kalman Filter (RKF) [15]. See Fig. 6. FD and GMM algorithms fail to separate the water motion from the foreground. Our algorithm produces cleaner result than RKF. At the same time, the computational efficiency is higher. Note that all the results from the state-of-art algorithms are directly obtained from their published papers and research websites, in order to have a fair comparison. We selected the compared algorithms from the available results in the state-of-art papers. For our algorithm, we all use the same set of parameters.

Quantitative analysis: To further show the effectiveness of our method, we conducted quantitative analysis on the computational time and accuracy. We manually labeled the above four videos, each of which has a ground truth binary map of background and foreground. The ground truth images are shown in column (b) from Figs. 3–6. The way to measure the background subtraction is

$$P_{error} = \frac{\#\ mismatched\ pixels}{\#\ all\ pixels} \tag{5}$$

Table 1
Computational time analysis of background subtraction. It shows the time to process one frame.

Methods	Time (s)
Frame difference (FD)	0.0017 ± 0.0002
Temporal spectral residual (TSR)	0.0030 ± 0.0006
Gaussian mixture model (GMM)	60.12 ± 6.16

The computational time is calculated by the time to process one frame. We ran our code in Matlab on a 2.4GMH, 8 G Memory machine without code optimization with image size of 120 by 160. We implemented two algorithms Frame Difference (FD) and Gaussian Mixture Model (GMM). The running time analysis is shown in Table 1. Our method only takes 0.0030 s to process one frame, which is a real-time pre-processing method.

The relationship between processing time and accuracy is shown in Fig. 7. A good pre-processing algorithm is preferably to lie in the left bottom corner of the figure, which is an area of low computational time and low error percentage. The figure shows that simple method like FD stays in the right bottom area, which has a short running time (shown in blue circle), but the error percentage is high. Complex mathematical model such as GMM has a slightly higher accuracy than TSR in video (b,c), and similar performance in video (a). However, GMM is not always stable. GMM on video (d) produces a noisy result. This is because the GMM model does not handle fully dynamic background well. Furthermore, the computational time is significantly higher than TSR. For all four videos, the proposed method TSR (shown in Square) lies in the left bottom corner of the figure, which shows a good tradeoff between computational time and accuracy.

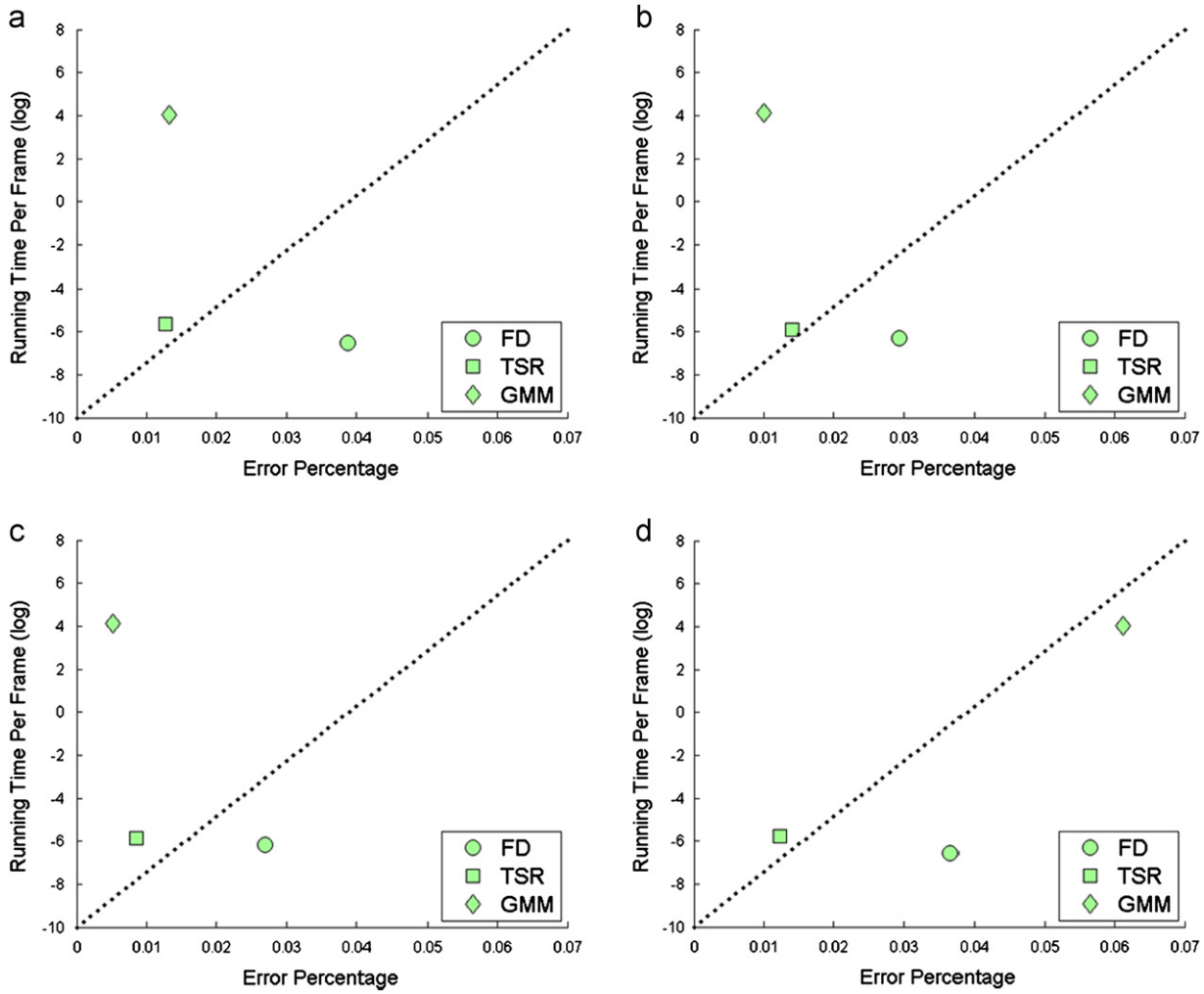


Fig. 7. Quantitative analysis on background subtraction quality and running time. It shows the algorithm performance on four videos individually. *X*-axis is the percentage of missing pixels over all pixels; *Y*-axis the running time to process one frame. Since the running time varies too much, we display it in natural log scale. Three algorithms are compared: FD, TSR and GMM: (a) shopping center, (b) rain, (c), beach, (d) water bottle.

3.2. Left ventricle segmentation

Image segmentation is a hot topic that has been studied in many years [19]. Automated object segmentation is a fundamental problem in medical image analysis. It is challenging to solve the problem robustly because of the common presence of cluttered objects, object texture, image noise, and various other artifacts in medical images. In recent decades, deformable model based segmentation methods have been extensively studied and has achieved considerable success, such as Snake [20], ASM [21], Level Set [22], GVF [23], Parametrically Deformable Model [24], because of their ability to integrate high-level knowledge with low-level image processing. Metamorphs method is one type of deformable models that take into account of "deforming disks or volumes". It is able to integrate both shape and appearance in a unified space, which is encoded as probability maps. Since the model has not only boundary shape but also interior appearance, it is more robust to ambiguous boundaries and complex internal textures. Metamorphs method shows its effectiveness on diverse tasks such as segmenting cardiac and liver from MRI, and prostate from ultrasound images. Several works have been proposed to extend and further improve the original Metamorphs method. Shen et al. proposed Active Volume Model (AVM) [25] to perform volume segmentation in 3D images. The AVM's shape is represented by a simplex mesh and its volumetric interior carries the various visual appearance feature statistics. Shape prior constraint based on Active Shape Model (ASM) [21] has also been incorporated into 3D Metamorphs. It constrains the intermediate shape by following the shape pattern from existing data, which makes it able to recover or preserve local shape details.

Metamorphs has been extended to 3D [26,27], and shape prior has also been incorporated [28,29]. However, one limitation of Metamorphs and its variations is that temporal information cannot be effectively incorporated. It is still not clear on how to find a nice representation of motion information for spatio-temporal segmentation, such as segmenting a cardiac cycle. Furthermore, since manual initialization is very time-consuming for temporal data, it is also desirable to automatically initialize the model instead of manually locating the organ as used in the standard Metamorphs.

In this section, we show that TSR can provide a nice representation of motion information for Metamorphs. TSR can automatically find motion information without tuning any parameters. The motion information can be effectively incorporated into the Metamorphs model, serving as both the initialization of the model and also the constraint in the running time.

We applied our TSR algorithm to spatio-temporal segmentation problem of left ventricle in MR sequences. TSR algorithm can have a good estimation of cardiac motions in MR sequences, which is a nice representation of motion features for Metamorphs. In addition, as

TSR algorithm is computationally efficient, the extra cost for Metamorphs is very low. Experiments show that compared to the standard Metamorphs, our method is fully automatic and efficient. It is able to effectively segment region-of-interest in the time series data.

TSR is an efficient method to find salient motion regions in video sequences. The main idea is to roughly remove the redundant part of a volume data (the static part of temporal slices) and keep the salient motion regions. Treating a sequence of 2D cardiac images as 3D volume, TSR is able to roughly locate the cardiac movement in temporal space, as shown in Fig. 8. A cardiac motion cycle contains cardiac motion (the motion saliency) and static regions (regions do not change much during the whole cardiac motion cycle).

We applied TSR algorithm on the 2D cardiac images, and Fig. 8 shows a result. The probability map detected by TSR is mostly around the boundary of left ventricle endocardium. It provides a nice representation of motion information for the 3D volume of cardiac movement. This can be used as the initialization and deformation constraint in Metamorphs. In addition, as this algorithm only needs Fourier transform, the computational cost is very low. It does not significantly increase the computational cost of Metamorphs.

The motion information obtained from TSR provides a roughly motion representation of the cardiac motion sequences. In order to effectively integrate the motion representation into the Metamorphs model, we represent the motion information (a probability map) as a new energy term, which can be seamlessly incorporated into the online deformation framework. The overall energy function is defined as

$$E = E_{int} + E_{ext} = E_{int} + \left(\sum_{i=1}^{n} E_R + k_T E_T \right) \tag{6}$$

where E_{int} is the internal (smoothness) energy, E_{ext} is the external (image) energy, E_R is the region term, n is the number of time frames, E_T is the temporal term (a probability map obtained from TSR algorithm), and k_T is the weight to balance the contributions of the two external energy terms. The balance between the internal and external energies is naturally controlled by the smoothness constraint of the shape model [25]. Different from standard Metamorphs, our object function is designed for the whole temporal data instead of each time frame. To initialize the model, we use TSR based saliency detection method to rapidly and automatically generate probability maps for all time frames, which denote the probabilities of being foreground or background. This energy drives the model to deform. After this initialization, an intermediate result is generated and energy from region term can be generated and iteratively updated. Although temporal energy continues to serve as a constraint during the evolution of the model, our algorithm adaptively decreases its importance since region term becomes more and more reliable when it is near by the boundary. We define the weight k_T as $k_T = 1 - e^{-|\nabla \mathcal{M}|}$, where $\mathcal{M}$ is the model. Thus $|\nabla \mathcal{M}|$ is the magnitude of deformation change in model shape. In the beginning, the shape deforms a lot so k_T is relatively large. It means that we trust more on the temporal energy, and put larger weights on it. After several iterations, the region term should be more important since the model is close to the boundary. At this time the shape deforms less, so k_T is smaller. The model converges when k_T approaches zero as the shape stops deforming.

We validate the spatio-temporal Metamorphs by segmenting left ventricle endocardium from MRI. Data were acquired with a 1.5 T Espree Siemens MRI scanner. The collected cine sets include short and long axis views on normal volunteers with a true fast imaging with steady-state precession sequence (TR/TE/a = 60.3 ms/1.4 ms/80°; slice thickness = 6 mm; acquisition matrix = 256 × 256). Segmentation experiments were performed on a set of 10 MRI cine sequences. Each sequence has 25 heart phases (frames) and total duration of 1 s (approximately one heart-beat). It is worth mentioning that cardiac segmentation has been extensively investigated [30–32]. Our focus, however, is to demonstrate TSR algorithm can provide a good motion presentation that improves the performance of standard Metamorphs on time series data.

This segmentation task is challenging because of two facts. The papillary muscle inside the left ventricle has high frequent movement. Such movement with high gradient information can adversely affect the accuracy of segmentation algorithms. Furthermore, the morphology of papillary muscle changes along time series. It is hard to obtain a consistent result by performing segmentation algorithm on a single frame. The bottom row in Fig. 9 shows a segmentation result of Metamorphs that fails to keep the consistency. The papillary muscle is captured in the first

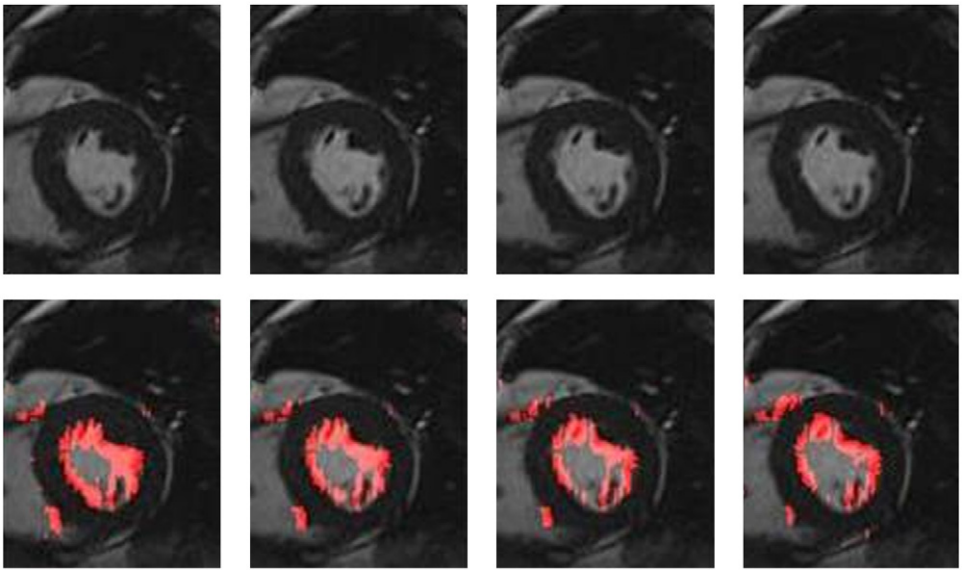


Fig. 8. Cardiac movement region detection from TSR Algorithm (marked in red color in the bottom row). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

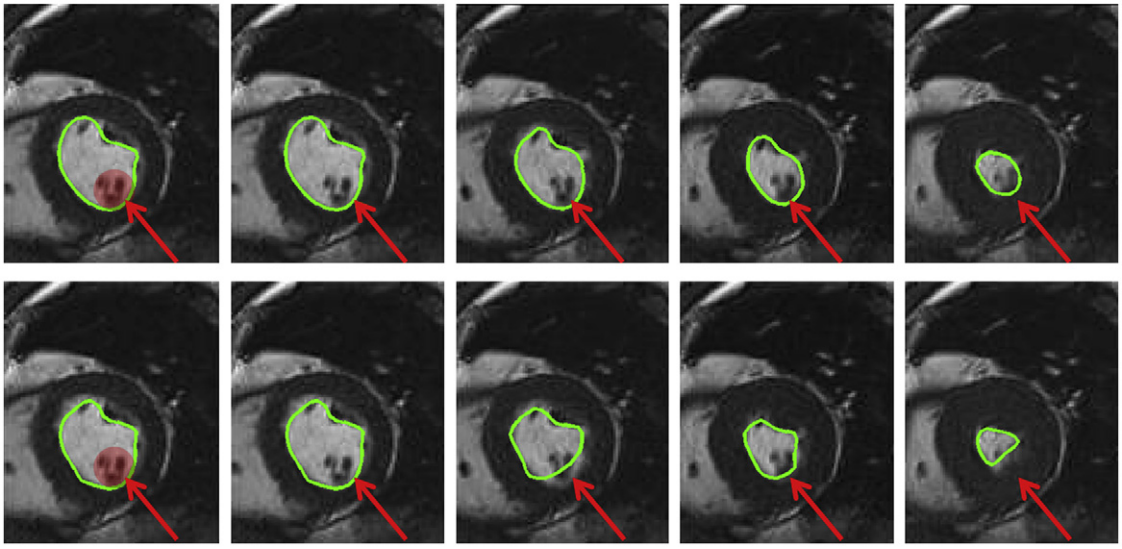


Fig. 9. Comparison between the proposed method (top row) and the standard Metamorphs (bottom row). The light red area in the first column is the papillary muscle inside the left ventricle. The high frequent movement makes it hard to capture the papillary muscle consistently.

Table 2
Quantitative comparisons of sensitivity, specificity and computational time. Mean values and standard deviations are reported.

Methods	Sensitivity	Specificity
Metamorphs	0.79 ± 0.14	0.94 ± 0.05
Our method	0.93 ± 0.05	0.96 ± 0.03

Table 3
Computational time analysis.

Methods	Time (s)
Standard metamorphs	30.1
TSR+stardard metamorphs	30.65
TSR+parallel metamorphs	4.8

frame, but then lost in the next. Our method uses the temporal constraint from TSR, thus it considers the papillary muscle movement over the whole MR sequences. It is able to keep the results consistent. The results of our method is shown in the top row in Fig. 9, where the papillary muscle area is successfully captured along all frames due to the constraint from temporal energy.

Table 2 quantitatively compares the algorithm performance. The proposed method achieves higher mean values of sensitivity and specificity with lower standard deviations. It further validates our assumption that our method is more accurate and stable.

We also conducted running time analysis in Table 3. We implemented this method using Matlab on a Quad CPU 2.4 GHz PC, with a volume data with 25 frames of size 256 by 256. Standard Metamorphs takes 30.1 s. Standard Metamorphs with TSR takes 30.65 s. TSR takes only 1.5% of running time, which is a small portion of the overall computation. Furthermore, the usage of the motion information from TSR can further boost the calculation speed by paralleling computing on a multi-core platform. The standard way is calculated sequentially, as the current frame is dependent on the previous frame as the initialization. But the TSR algorithm, each frame can be updated in parallel. So the computational time is further reduced to 4.8 s, which is only 15.95% of the standard Metamorphs. Thus the proposed method can greatly boost the calculation of the Metamorphs.

4. Conclusions

In this paper, we presented a fast motion saliency detection method Temporal Spectral Residual (TSR). Based on the observation that moving objects contain salient information along the temporal domain, we proposed to use Spectral Residual algorithm on temporal slices to detect the motion saliency. In order to produce reliable results, we also introduced majority voting strategy to further refine results. Since our algorithm only depends on Fourier Transform instead of complex models, it is computationally efficient. We conducted experiments on two applications: background subtraction in video sequences under dynamic background and left ventricle endocardium segmentation in MR sequences. Experiments show that our method accomplishes a good tradeoff between the accuracy and computational efficiency.

References

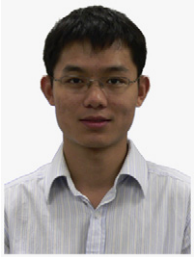
[1] L. Itti, C. Koch, A saliency-based search mechanism for overt and covert shifts of visual attention, Vision Res. 40 (10–12) (2000) 1489–1506.
[2] L. Itti, C. Koch, Computational modelling of visual attention, Nat. Rev. Neurosci. 2 (3) (2001) 194–203.
[3] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, IEEE Trans. Pattern Anal. Mach. Intell. 20 (11) (1998) 1254–1259.
[4] D. Walther, L. Itti, M. Riesenhuber, T. Poggio, C. Koch, Attentional selection for object recognition—a gentle way, in: the 2nd Workshop on Biologically Motivated Computer Vision, 2002, pp. 472–479.
[5] D. Gao, N. Vasconcelos, Discriminant saliency for visual recognition from cluttered scenes, in: Advances in Neural Information Processing Systems, vol. 1, 2004, pp. 481–488.
[6] X. Hou, L. Zhang, Saliency detection: a spectral residual approach, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[7] X. Cui, Q. Liu, M. Gao, D.N. Metaxas, Abnormal detection using interaction energy potentials, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011.
[8] R. Ji, H. Yao, X. Sun, Actor-independent action search using spatiotemporal vocabulary with appearance hashing, Pattern Recognition 44 (2011) 624–638.
[9] X. Gao, N. Liu, W. Lu, D. Tao, X. Li, Spatio-temporal salience based video quality assessment, in: IEEE International Conference on Systems, Man, and Cybernetics, 2010, pp. 1501–1505.
[10] C. Stauffer, W. Grimson, Adaptive background mixture models for real-time tracking, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 1999, p. 2, (xxiii+637+663).
[11] A. Elgammal, R. Duraiswami, D. Harwood, L.S. Davis, Background and foreground modeling using nonparametric kernel density for visual surveillance, Proc. IEEE 90 (7) (2002) 1151–1163.
[12] A. Mittal, N. Paragios, Motion-based background subtraction using adaptive kernel density estimation, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2004, pp. 302–309.

[13] O. Tuzel, F. Porikli, P. Meer, A Bayesian approach to background modeling, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition—Workshops, 2005. CVPR Workshops, vol. 1, 2005, p. 58.
[14] A. Monnet, A. Mittal, N. Paragios, V. Ramesh, Background modeling and subtraction of dynamic scenes, in: IEEE International Conference on Computer Vision, vol. 1, 2003, pp. 1556–1559.
[15] J. Zhong, S. Sclaroff, Segmenting foreground objects from a dynamic textured background via a robust Kalman filter, in: IEEE International Conference on Computer Vision, vol. 1, 2003, pp. 44–52.
[16] R. Jain, H. Nagel, On the analysis of accumulative difference pictures from image sequences of real world scenes, IEEE Trans. Pattern Anal. Mach. Intell. (1979) 206–214.
[17] Y. Kameda, M. Minoh, A human motion estimation method using 3-successive video frames, in: International Conference on Virtual Systems and Multimedia, 1996, pp. 135–140.
[18] 〈http://homepages.inf.ed.ac.uk/rbf/caviardata1/〉.
[19] X. Gao, B. Wang, D. Tao, X. Li, A relay level set method for automatic image segmentation, IEEE Trans. Syst. Man Cybern. Part B 41 (2) (2011) 518–525.
[20] M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models, Int. J. Comput. Vision 1 (1987) 321–331.
[21] T. Cootes, C. Taylor, D. Cooper, J. Graham, Active shape model—their training and application, Comput. Vision Image Understanding 61 (1995) 38–59.
[22] R. Malladi, J. Sethian, B. Vemuri, Shape modeling with front propagation: a level set approach, IEEE Trans. Pattern Anal. Mach. Intell. 17 (1995) 158–175.
[23] C. Xu, J. Prince, Snakes, shapes and gradient vector flow, Trans. Image Process. 7 (1998) 359–369.
[24] L. Staib, J. Duncan, Boundary finding with parametrically deformable models, IEEE Trans. Pattern Anal. Mach. Intell. 14 (11) (1992) 1061–1075.
[25] T. Shen, H. Li, X. Huang, Active volume models for medical image segmentation, IEEE Trans. Med. Imaging 30 (3) (2011) 774–791.
[26] T. Shen, S. Zhang, J. Huang, X. Huang, D. Metaxas, Integrating shape and texture in 3d deformable models: from metamorphs to active volume models, in: Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies, 2011.
[27] T. Shen, X. Huang, H. Li, E. Kim, S. Zhang, J. Huang, A 3D Laplacian-driven parametric deformable model, in: International Conference on Computer Vision, 2011.
[28] S. Zhang, J. Huang, M. Uzunbas, T. Shen, F. Delis, X. Huang, N. Volkow, P. Thanos, D.N. Metaxas, 3D segmentation of rodent brain structures using hierarchical shape priors and deformable models, in: International Conference on Medical Image Computing and Computer-assisted Intervention, Springer-Verlag, 2011, pp. 611–618.
[29] S. Zhang, Y. Zhan, M. Dewan, J. Huang, D.N. Metaxas, X.S. Zhou, Towards robust and effective shape modeling: sparse shape composition, Med. Image Anal. 16 (1) (2012) 265–277.
[30] Y. Zheng, A. Barbu, B. Georgescu, M. Scheuering, D. Comaniciu, Four-chamber heart modeling and automatic segmentation for 3-D cardiac CT volumes using marginal space learning and steerable features, IEEE Trans. Med. Imaging 27 (11) (2008) 1668–1681.
[31] X. Zhuang, K. Rhode, R. Razavi, D. Hawkes, S. Ourselin, A registration-based propagation framework for automatic whole heart segmentation of cardiac MRI, IEEE Trans. Med. Imaging 29 (9) (2010) 1612–1625.
[32] R. Chandrashekara, R. Mohiaddin, D. Rueckert, Cardiac motion tracking in tagged MR images using a 4D B-spline motion model and nonrigid image registration, in: International Symposium on Biomedical Imaging, 2004, pp. 468–471.

Xinyi Cui is a Ph.D. candidate in the Computer Science Department at Rutgers University. Her advisor is Dr. Dimitris N. Metaxas. Her major research interests are computer vision and machine learning. More specifically, she focuses on motion analysis for video sequences, human action/activity recognition, human behavior analysis, saliency detection, background modeling, object detection and recognition. she received her M.S. degree from the Computer Science and Engineering Department at Harbin Institute of Technology. She also received her B.E. from the same department.

Qingshan Liu is a professor in the School of Information and Control Engineering, Nanjing University of Information Science and Technology, China. He received his Ph.D. from the National Laboratory of Pattern Recognition, Chinese Academic of Science in 2003 and his M.S. from the Department of Auto Control in South-East University in 2000. Dr. Qingshan Liu was an assistant research professor in the Department of Computer Science, Computational Biomedicine Imaging & Modeling Center (CBIM), Rutgers, the State University of New Jersey from 2010 to 2011. Before he joined in Rutgers University, he worked as an associate professor at the National Laboratory of Pattern Recognition, Chinese Academic of Science, and he worked as an associate researcher at the Multimedia Laboratory in Chinese University of Hong Kong during June 2004 and April 2005. He received the president scholarship of Chinese Academy of Sciences in 2003. His research interests are Image and Vision Analysis including Face Image Analysis, Graph & Hyper-graph based Image and Video understanding, Medical Image Analysis, Event-based Video Analysis, etc.

Shaoting Zhang received his B.E. degree in Software Engineering from Zhejiang University, China, in 2005, M.S. degree in Computer Software and Theory from Shanghai Jiao Tong University, China, in 2007, and Ph.D. degree in Computer Science from Rutgers, the State University of New Jersey in 2011. His advisor is Dr. Dimitris N. Metaxas. His major research interests are focusing on deformable models, sparse learning methods, and their applications on medical image analysis, computer vision and computer graphics.



Fei Yang is a Ph.D. candidate in the Computer Science Department at Rutgers University. He received the B.E. degree from Tsinghua University in 2003, and the M.E. degree from the Chinese Academy of Sciences in 2006. He was a software engineer in Microsoft China from 2006 to 2007. His current research focuses on facial feature localization, face tracking and face animation.

Dimitris N. Metaxas is a professor in the Computer Science Department at Rutgers University. He is directing the Computational Biomedicine Imaging and Modeling Center (CBIM). He received the B.E. degree from the National Technical University of Athens, Greece, in 1986, M.S. degree from the University of Maryland in 1988, and Ph.D. from the University of Toronto in 1992. He has been conducting research toward the development of formal methods upon which computer vision, computer graphics, and medical imaging can advance synergistically.