Image Segmentation Based on GrabCut Framework Integrating Multi-scale Nonlinear Structure Tensor

Shoudong Han, Wenbing Tao, Desheng Wang Xue-cheng Tai, Xianglin Wu

ABSTRACT

In this paper, we propose an interactive color natural image segmentation method. The method integrates color feature with multi-scale nonlinear structure tensor texture (MSNST) feature and then uses GrabCut method [17] to obtain the segmentations. The MSNST feature is used to describe the texture feature of an image and integrated into GrabCut framework to overcome the problem of the scale difference of textured images in [28]. In addition, we extend the Gaussian Mixture Model (GMM) used in [17] to MSNST feature and GMM based on MSNST is constructed to describe the energy function so that the texture feature can be suitably integrated into GrabCut framework and fused with the color feature to achieve the more superior image segmentation performance than the original GrabCut method [17]. For easier implementation and more efficient computation, the symmetric KL divergence [30] is chosen to produce the estimates of the tensor statistics instead of the Riemannian structure of the space of tensor as in [28]. The Conjugate norm developed in [31] was employed using Locality Preserving Projections (LPP) technique as the distance measure in the color space for more discriminating power. An adaptive fusing strategy is presented to effectively adjust the mixing factor so that the color and MSNST texture features are efficiently integrated to achieve more robust segmentation performance. Lastly, an iteration convergence criterion is proposed to reduce the time of the iteration of GrabCut algorithm dramatically with satisfied segmentation accuracy. Experiments using synthesis texture images and real natural scene images demonstrate the superior performance of our proposed method.

Key word: Graph Cuts, Multi-scale Nonlinear Structure Tensor (MSNST), Interactive Image Segmentation, Adaptive Fusion.

I. INTRODUCTION

Extracting a foreground object in a complex environment is of great practical importance in computer vision. To extract objects from color images is even more challenging. Color images carry much more information than gray ones [1], and these information can be used to enhance the image analysis process and improve segmentation results. As a result, color image segmentation has been studied for decades, and recently received much attention

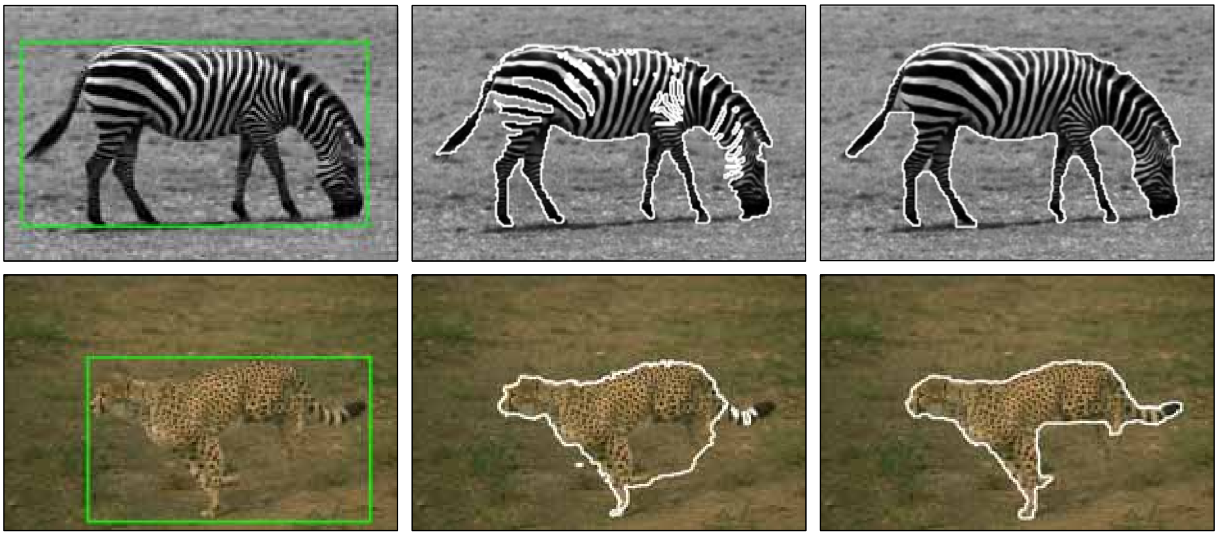


Fig. 1 A comparison between GrabCut [17] and our method. Left: the original image with an initial rectangle placed by user; Middle: the result by GrabCut [17]; Right: the result by our method.

for special effects in film, television, publication, photography and a number of desktop applications.

Due to the amount of information contained in images and their unpredictable complexity, manual segmentation is tedious and time consuming, lacking in precision and impractical when applied to long image sequences. A general purpose image segmentation technique should be able to accurately define the desired object boundaries or regions automatically or semi-automatically with minimal user input. Existing image segmentation algorithms can be generally classified into three major categories: feature-based, region-based and boundary-based [2]. As is typically the case, each strategy has its advantages and disadvantages and is better suited to segment certain types of images. Feature-based methods try to classify pixels based on their positions in feature space without explicitly considering their connectivity to similarly classified pixels. Common features include color intensity, gradient magnitude, texture, depth, motion, etc. Grayscale thresholding [3] and distance-based classification [4] all belong to this category. This strategy has some serious drawbacks as pixels from disconnected regions of an image may be grouped together if their feature spaces overlap. Region-based methods extend feature-based segmentation by specifically trying to maintain connectivity while grouping pixels with similar features, examples include blob coloring [3], region growing, region merging, region splitting and intelligent paint [5]. However, it may undesirably produce a very large number of small but quasi-homogenous regions. Boundary-based methods don’t have this disadvantage by attempting to define contours enclosing objects or subprojects and yield a minimum cost curve by optimizing the current edge criteria to approximate the real boundary, such as border tracing [6], dynamic programming [7], active contours [8] and intelligent scissors [9] etc. Due to the locality of edge information, if we want to obtain the desired global object, numerous interactions are necessary.

Recently, the graph-based approaches to object extraction have been shown to be efficient and accurate. One essential feature of the approach is that the segmentation energy function combines boundary regularization with regional properties. The common strategy underlying these approaches is the formation of a weighted graph, where each vertex corresponds to an image pixel or a region, and the weight of each edge connecting two vertices represents the similarity between them that they belong to the same segment. Additionally, another graph-based approach called Graph Cuts Method includes two extra terminals into the weighted graph, and the edge weight between vertex and terminals represents the possibility whether the vertex belongs to foreground or background. The weights are usually related to extracted features. The graph is then partitioned into multiple components that minimize some energy function. In the last few years, several graph-shaped methods have been developed for image unsupervised segmentations [10, 11, 12] or interactive segmentation [8, 13, 14, 15, 16, 17, 18].

Taking unsupervised approaches for example, Shi and Malik [12] proposed a general image segmentation approach based on normalized cut by solving an eigensystem, and Wang and Siskind [10] developed an image-partitioning approach by using a complicated graph reduction. Although they can robustly generate balanced clusters without any user interaction, high computation complexity is required and there is no way to alternate the final segmentation result in case some parts of the image are wrongly labeled. By contrast, Graph Cuts is a general purpose interactive segmentation technique. It uses binary Graph Cuts algorithms for object segmentation and is able to alleviate the problems inherent to fully automatic segmentation. The users have to hint on what they intend to segment first, and then the image is segmented automatically by computing a global optimum among all segmentations satisfying the provided hints. Such an approach enables the user to get some desired segmentation results with very intuitive interactions. This concept was first proposed and tested by Boykov and Jolly [13]. Since then, it has been widely studied in computer vision and graphics communities for image restoration, stereo and object segmentation etc. In the following we try to give a briefly overview on several Graph Cuts based methods: interactive Graph Cuts [13], Lazy Snapping [15], and GrabCut [17]. Interactive Graph Cuts [13] is a general purpose interactive segmentation method for monochrome N-dimensional images. The user needs to mark certain pixels as “object” or “background”. Afterwards, the histograms of grey values are used to describe image foreground and background grey-level distributions and Graph Cuts are used to find the globally optimal segmentation. However, it is impractical to construct adequate color space histograms for this method. Lazy Snapping [15] combines Graph Cuts with pre-computed over-segmentation and produces high quality cutouts for color images in near real-time, but for thin and branch structures it works very poor. Many user interactions are needed to achieve reasonable results. GrabCut [17] extends Graph Cuts to color images and incomplete trimaps. It replaces the monochrome image model based on histograms in [13, 14, 16] by Gaussian

Mixture Model and iteratively alternates between estimation and parameter learning to solve the min-cut problem until converges. Therefore, the user interaction can be relaxed to simply placing a rectangle or a lasso around the object, followed by a small amount of corrective editing. These developments make GrabCut more convenient to image editing such as foreground extraction.

In [13, 15, 16, 17], the process of separating an image into objects and background is mainly guided by regional statistics involving image values. However, directly computing the statistics on image values may not be enough to discriminate regions. In some cases, texture information is often a more appropriate discriminating feature. There are many different texture feature description approaches which have been employed in the literature for image segmentation applications, including those Markov Random Fileds [20, 21], multiple resolution techniques [22, 23], Gabor wavelet filters [24, 25] and so on. The structure tensor has been introduced for texture analysis as a fast local computation providing a measure of the presence of edges and their orientation. Various tensor segmentation methods have been proposed including active contours [26, 27, 30]. For example, the work of [26] uses a Gaussian approximation for the nonlinear structure tensor channels and a non-parametric histogram for the added image intensity channel, and then the segmentation proceeds to separate these distributions with active contours unsupervisedly. In addition, Graph Cuts technique has been recently demonstrated with diffusion tensor magnetic resonance imaging data in [29], which uses the symmetric KL divergence [30] as dissimilarity measure, and obtains the weights of terminal links by computing the average distance of each such tensor to each of the respective seed tensors. However, all these distances are weighted equally and the computations of terminal link weights increase dramatically with the sizes of image and seeds increase.

Malcolm et al. [28] generalized these tensor methods to segment images with multimodal object and background. This is done by taking into account the Riemannian geometry of the tensor space. Moreover, interactive Graph Cuts [13] technique is applied to segment multimodal tensor valued images. In order to improve the segmentation quality, intensity information was included with the image derivatives without losing its nice properties and a 3×3 extended structure tensor is constructed as the Graph Cuts' data input. However, the 3×3 tensor actually fixes the weights of color information and texture information both at approximate 0.5, which means that the mixing factor can not be adjusted adaptively to reduce the negative side effect of including too much useless information. Furthermore, the introduction of high-order tensor (5×5 when considering all the color channels) implies that the energy minimization has to be done in a higher dimensional space, which can be too difficult and result in multiple local minima [41]. Additionally, the tensor texture description used in [28] lacks of scale information. Therefore, the method will fail when two textures differ only in scale.

To deal with scale difference of the classical structure tensor [28] of texture images effectively, a multi-scale nonlinear structure tensor is proposed to describe the texture feature of images and integrated into GrabCut [17] framework instead of the interactive Graph Cuts method [13] used in [28] to achieve improved interactive image segmentation and simplify user interaction. To achieve this goal, we extend the color models of images based on GMMs used in [17] to multi-scale nonlinear structure tensors and construct texture feature models of images based on GMMs. To minimize the energy in a low dimensional space, we consider the rank-2 tensor and image color separately. The symmetric KL divergence [30] is chosen to produce the estimates of the tensor statistics instead of the Riemannian structure of the space of tensor as in [28] for easier implementation and more efficient computation. The symmetric KL divergence [30] is a low-dimensional approximating of the full distribution in Riemannian space, which has been proved to be robust and discriminative enough in [27, 29, 30]. The Conjugate norm developed in [31] was employed as the distance measure in color space using Locality Preserving Projections (LPP) technique for more discriminating power. To adaptively adjust the mixing factor so that the color and MSNST texture features are efficiently integrated into GrabCut framework to achieve more robust segmentation performance, we adopt an approximation of the KL divergence to compute the relative discriminative capabilities of the present global foreground and background color GMMs. Lastly, an iteration convergence criterion is proposed to reduce the time of the iteration of GrabCut algorithm dramatically with satisfied segmentation accuracy.

The remainder of the paper is organized as follows. In Section II, the multi-scale nonlinear structure tensor is described. How to integrate the MSNST into GrabCut framework is presented in Section III. Section IV presents a number of experimental results. Finally, some conclusions are drawn in Section V.

II. Multi-Scale Nonlinear Structure Tensor (MSNST)

It is not easy to estimate or even to represent the orientation information from a scalar valued image or a vector-valued image, which is indeed a major component of textures feature. The Gabor representation has been shown to be optimal in the sense of minimizing the joint two-dimensional uncertainty in space and frequency [32], but it unfortunately has the decisive drawback of inducing lots of redundancy. However, the structure tensor is widely accepted to compactly derive this feature by the use of image derivatives, which hold the whole orientation information. In other words, the components of the structure tensor are as powerful for the discrimination of textures as a whole set of Gabor filters of a fixed scale.

In this section, we will introduce the concept of MSNST based on the classical structure tensor, by making use of a redundant dyadic wavelet transform and the nonlinear diffusion [33]. With this concept, the orientation information of an image can be studied at different scales, just like the Gabor wavelets [34].

A. Gabor Wavelet

In this section, we review the basics of Gabor transforms. The Gabor wavelet has been studied by numerous researchers in the context of image representation, object recognition, texture classification and image retrieval due to its rich multiresolution representation and simplicity of implementation. It has the property that it can segment images with differences in spatial frequency, density of elements, orientation, phase, and energy. A 2-D Gabor filter [34] is an oriented complex sinusoidal grating modulated by a 2-D Gaussian function, which is given by

$$g(x,y)=\left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+2\pi jWx\right] \tag{1}$$

where σ_x and σ_y are the space constants of the Gaussian envelope along the *x*-axis and *y*-axis, and the frequency of the span-limited sinusoidal grating is given by *W*. Gabor functions form a complete but nonorthogonal basis set. Expending a signal using this basis provides a localized frequency description. Let *g*(*x*,*y*) be the mother Gabor wavelet, then we can generate a class of self-similar functions by means of appropriate translations, rotations and dilations:

$$\begin{aligned} &g_{mn}(x,y)=a^{-m}g(x^{'},y^{'}),\quad a>1\\ &x^{'}=a^{-m}(x\cos\theta+y\sin\theta)\\ &y^{'}=a^{-m}(-x\sin\theta+y\cos\theta)\end{aligned} \tag{2}$$

where $\theta=n\pi/K$, $n=0,1,\ldots,K-1$ and *K* is the total number of orientations, and $m=0,1,\ldots,M-1$ and *M* is the total number of scales in the multiresolution decomposition. More implementation details refer to [34].

The Gabor filter g_{mn} gives a complex-valued function, decomposed by $g_{mn}=g_{mn}^{R}+jg_{mn}^{I}$ into real and imaginary parts. Given the gray level distribution I_0 of the original textured image, its Gabor wavelet representation is then defined to be the convolution of the image with a bank of Gabor filters

$$(I_0)_{mn}=\sqrt{(I_0*g_{mn}^{R})^2+(I_0*g_{mn}^{I})^2}$$

A filter bank consisting of Gabor filters with different orientations and scales is usually used to extract the local image details. However, it contains many unavoidable drawbacks, for example, complexity both in memory and computational time involved in the convolution is high; low orientation texture discrimination when the number of the extracted orientation information is small; a rather large number of parameters need to fix manually.

B. Structure Tensor

The classical structure tensor [37] uses the tensor product of the smoothed image gradient to form the tensor, and all channels in RGB color space are taken into account by summing the tensor products along the particular channels [40]

$$\mathbf{T} = K_\rho * \sum_{n=1}^{N} \left(\nabla I_n \nabla I_n^T \right) = \begin{pmatrix} K_\rho * \sum_{n=1}^{N} (I_{n,x})^2 & K_\rho * \sum_{n=1}^{N} (I_{n,x} I_{n,y}) \\ K_\rho * \sum_{n=1}^{N} (I_{n,x} I_{n,y}) & K_\rho * \sum_{n=1}^{N} (I_{n,y})^2 \end{pmatrix} \tag{3}$$

In above, K_ρ is a Gaussian kernel with standard deviation ρ, the subscripts x and y denote partial derivatives of the *n*-th channel of the image I, and N is the total number of the channels. In the case of scalar images or vector-valued images, e.g., color images, we have $N = 1$ or $N > 1$ accordingly. Obviously the structure tensor yields only three feature channels for each scale without any parameters like $\theta \in 2\pi$ in Gabor filers. Comparing the representation of local orientation provided by the structure tensor with that obtained by Gabor filters reveals that the discreteness and the degree of freedom for the orientation known from Gabor filters is replaced by the smoothed compact versions of the image derivatives. Apart from this, structure tensor can be computed more easily with less memory consumption and information redundancy, and the calculation is acceptable even though all channels are taken into account. Moreover, tensor algebra is a solid mathematical body that supports further analysis in the tensor domain [41] and can provide more complete information to discriminate different feature regions.

The smoothing with a Gaussian kernel makes the classic structure tensor suffer from the dislocation of edges, leading to inaccurate segmentation results near region boundaries. For example, most of the final objects contours obtained in [28] seem larger than the desired boundaries. We also experiment in Fig. 2 to justify this problem of standard Gaussian smoothing. To address this problem, nonlinear diffusion, a technique used to reduce the smoothing in the presence of edges, has been proposed to replace the Gaussian smoothing. It was introduced by Perona and Malik [33], and extended to vector-valued data by Gerig et al. [42] using

$$\partial_t u_i = \operatorname{div}(g(\sum_{x=1}^{X} |\nabla u_x|^2) \nabla u_i) \quad \forall i \tag{4}$$

Above, $u_i(t = 0)$ is the *i*-th evolving vector channel of the structure tensor without Gaussian smoothing, and *X* is the total number of independent channels, *g* is a decreasing function. For applications, the following *g* is often used [43]

$$g(|\nabla u|) = \frac{1}{|\nabla u|^p + \varepsilon} \tag{5}$$

Above the small positive constant $\varepsilon = 10^{-3}$ is introduced to avoiding the division by zeros and the constant *p* is used to balance edge enhancement and smoothing. A larger *p* gives the better edge enhancement effect. However, a larger *p* requires a longer diffusion time to obtain an appropriate smoothing effect. A number of experimental results show that *p*=0.6 is a good compromise in practical applications. During the implementation of formula (4), explicit schemes are only stable for very small time steps, which lead to poor efficiency and limited practical use. Thus, we adopt the Additive Operator Splitting (AOS) scheme [44], which is stable for all time steps and can be computed at least ten times more efficient under typical accuracy requirements.

In segmentation methods such as active contours [26] and Graph Cuts technique [28], structure tensor has been widely used to represent the local orientation of the image. However, compared with the Gabor filters, the structure tensor used in [26, 28] reflects only the orientation information at a single scale and it fails to discriminate two textures that differ only in scale. In order to preserve the multi-scale property of the Gabor filters and the compact representing of orientation at a fixed scale, we shall introduce the MSNST which fusion the scale information to represent the texture feature of image.

C. Multi-Scale Nonlinear Structure Tensor

Multi-scale structure tensor was first defined by Scheunders [45] and named as multi-scale fundamental forms. It has been used for multispectral images fusion or merging, color images enhancing, and multivalued image noise filtering etc. To the best of our knowledge, this is the first time that MSNST has been applied to textured image segmentation.

An extension towards multi-scale structure tensor can be obtained by using the non-orthogonal (redundant) discrete wavelet frameworks introduced by Mallat [46, 47]. Let $\theta(x, y)$ be a 2-D differentiable smoothing function, for instance the Gaussian function given by formula (1). Define two wavelet function using partial derivatives such that $\psi^x(x,y) = \partial\theta(x,y)/\partial x$ and $\psi^y(x,y) = \partial\theta(x,y)/\partial y$. If we denote the dilated function θ_s, ψ_s^x and ψ_s^y at scale *s* in the manner of formula (2) with $\theta = 0$, then the wavelet transform of the image *I*(*x*, *y*) in RGB color space has two components defined by

$$\begin{pmatrix} D_s^x(x,y) \\ D_s^y(x,y) \end{pmatrix} = \begin{pmatrix} I * \psi_s^x(x,y) \\ I * \psi_s^y(x,y) \end{pmatrix} = a^s \nabla (I * \theta_s)(x,y) \qquad s = 0,1,\ldots,S-1 \tag{6}$$

where S is the total number of scales in the multiresolution decomposition, and *a* can be set as 2 to decrease the computation and storage cost. Therefore, the proposed multi-scale structure tensor can be constructed using the tensor product of the gradient of $(I * \theta_s)(x,y)$ at each scale similar to (3) but without the Gaussian smoothing as

$$\mathbf{T}_s = \sum_{n=1}^{N}\left(\nabla(I*\theta_s)_n \nabla(I*\theta_s)_n^T\right) = a^{-2s}\begin{pmatrix} \sum_{n=1}^{N}(D_{n,s}^x)^2 & \sum_{n=1}^{N}(D_{n,s}^x D_{n,s}^y) \\ \sum_{n=1}^{N}(D_{n,s}^x D_{n,s}^y) & \sum_{n=1}^{N}(D_{n,s}^y)^2 \end{pmatrix} \tag{7}$$

where the notations of *n* and *N* follow the definition in formula (3). For real applications, a fast algorithm called “algorithme à trous” [47] has been developed to approximate this transform through filters associated with a set of one dimensional filters iteratively.

Finally, the MSNST is computed by applying (4) with initial conditions $\mu_1 = a^{-2s}\sum_{n=1}^{N}(D_{n,s}^x)^2$, $\mu_2 = a^{-2s}\sum_{n=1}^{N}(D_{n,s}^y)^2$, $\mu_3 = a^{-2s}\sum_{n=1}^{N}(D_{n,s}^x D_{n,s}^y)$ for different scale *s*=0,1,…,*S*-1, which is just like the nonlinear structure tensor does, to replace the Gaussian smoothing. We should notice that each scale is nonlinearly diffused separately, and each scale has three different channels and these three channels share the same decreasing function *g* as described in formula (4). In addition, the initial conditions μ_1, μ_2 and μ_3 are three coefficient images computed using the method mentioned above corresponding to each scale. Now we can use MSNST together with the color information to segment textured images by making use of not only the color information but also the orientations and scales of the texture information.

III. INTEGRATING MSNST INTO GRABCUT FRAMEWORK

Since only color information sometimes isn’t enough to discriminate the interest regions, effectively fusing texture feature will greatly increase the performance of image segmentation methods. There are many excellent works about texture segmentation or effective natural image segmentation integrating color and texture features. In this Section, how to integrate MSNST into GrabCut framework will be presented. The GrabCut algorithm includes two parts: the hard segmentation and border matting. The hard segmentation part consists of two steps. The first step is the iterative segmentation by simply placing a rectangle or a lasso around the object. The second step is a user editing process similar to [13] which is to refine the result. The border matting process is independent of the iterative segmentation process. More detail about the GrabCut framework can be found in [17]. Compared with the other interactive image segmentation method based on Graph Cuts [13, 15, 16], the essential advantage of GrabCut [17] is the simpler interaction, i.e., just placing a rectangle or lasso around the object. In this work, we focus on the iterative segmentation process of GrabCut and try to improve the performance of the iterative segmentation by integrating the multi-scale structure tensor into GrabCut framework. This will result in even less user editing in the second step. Moreover, the integrated texture information is also helpful in the process of user editing operation.

A. Distance Measure

After the color feature and texture feature are extracted from image, we must choose suited distance measures so as to effectively discriminate these features, which is essential for the accuracy of image segmentation.

Color features are extracted by representing each pixel with a three-dimensional color descriptor in a selected color space. Mostly, the Euclidean norm in RGB color space is employed [13, 15, 16, 17], but such a measure is notoriously unreliable for describing perceptually uniform and object boundaries [48]. Thus, in this work we choose to represent the color feature using L*a*b* color space which was shown to be approximately perceptually uniform by Wyszecki and Stiles [49]. However, it should be noticed that the Gabor filters are still computed using the grayscale values of image, and the classical structure tensor and our proposed MSNST are still extracted in RGB color space similar to [37, 40, 41].The L*a*b* color space is just used when the color features are computed. Inspired by the work of Grady et al. [48], the Conjugate norm developed in [31] was employed using the LPP technique as

$$dis_{\mathrm{C}}(I_m, I_n) = \left\| I_m - I_n \right\|_{Q^T Q} = (I_m - I_n)^T Q^T Q (I_m - I_n) \tag{8}$$

where the subscript C denotes that this formula is defined in the color domain and the other formulas follow this definition, and the subscripts *m* and *n* denote two different points in the L*a*b* color space, and *Q* is a matrix of size 3×3 . More implementation details refer to the LPP algorithm [31]. The advantages of LPP are linearity, generalization beyond the “training” points and robustness to outliers. These properties are helpful to distinguish object boundaries accurately.

To get good texture feature, each pixel is represented with the MSNST Γ, which is described in Section II-C as a set of matrixes of size 2×2 . The number of elements in Γ is the number of scales of the MSNST.

$$\boldsymbol{\Gamma} = \left\{ \mathbf{T}_0, \mathbf{T}_1, ..., \mathbf{T}_{S-1} \right\} \tag{9}$$

where $\mathbf{T}_s = a^{-2s} \begin{pmatrix} \sum_{n=1}^{N} \overset{\wedge}{(D_{n,s}^x)^2} & \sum_{n=1}^{N} \overset{\wedge}{(D_{n,s}^x D_{n,s}^y)} \\ \sum_{n=1}^{N} \overset{\wedge}{(D_{n,s}^x D_{n,s}^y)} & \sum_{n=1}^{N} \overset{\wedge}{(D_{n,s}^y)^2} \end{pmatrix}_{s=0,1,\ldots,S-1}$ and the “hat” denotes that the corresponding component has been nonlinearly diffused. One key factor in the tensor space analysis is a proper choice of tensor distance norm to measure similarity or dissimilarity between tensors and compute the tensor mean. The authors of

[26, 50] draw reduction of the full tensor to a single scalar, but lose much of the discriminating power. Notice that in [27, 51] the Frobenius norm was used, which relies on a very restrictive hypothesis. Taking into account the Riemannian structure of the tensor space, [52, 53] produce more accurate estimates of the tensor statistics. However, there is no closed form defined for the mean tensor which needs to be computed by using the gradient descent algorithm. Meanwhile, it is complicated to estimate the tensor statistics in Riemannian space when respecting the multi-scale structure and analyzing the Gaussian mixture distributions. In our proposed method, for easier implementation and more efficient computation we use the symmetric KL divergence [30] as a low-dimensional approximating of the full distribution in Riemannian space, which proved to be robust and discriminative enough in [27, 29, 30]. It naturally follows from the physical phenomena of diffusion and interprets the symmetric positive definite tensor as the covariance matrix of local Gaussian distribution, then defines the dissimilarity measure grounded in concepts from information theory. Although losing some discriminating power compared with the Riemannian measure, it still has the property of being affine invariant and offers the advantages of solving in closed form and computationally tractable. Additionally, our reliance on the MSNST feature and GMM statistics is quite robust and could be compensating although less discriminating. The experimental result (Fig. 7) verifies that our choice of a low-dimensional parametric representation is robust enough. The tensor distance for MSNST can be defined as the square root of the sum of the symmetric KL divergence for all scales, which have a very simple form [30] given by

$$dis_T(\mathbf{\Gamma}_m, \mathbf{\Gamma}_n) = \sqrt{\sum_{s=0}^{S-1}\left(\frac{1}{4}\left(tr(\mathbf{T}_{m,s}^{-1}\mathbf{T}_{n,s} + \mathbf{T}_{n,s}^{-1}\mathbf{T}_{m,s}) - 4\right)\right)} \tag{10}$$

Above $tr(\cdot)$ is the matrix trace operator, the other notations follow the definitions of (8) but in the MSNST space with texture domain.

B. Initial Clustering

In order to establish the GMMs, the initial foreground and background produced by placing a rectangle or a lasso around the object must first be classified into K_C clusters based on the color feature and K_T clusters based on the texture feature respectively. Then we will create a total of $2\times(K_C + K_T)$ Gaussian components. Note that K_C doesn't necessary equal to K_T and the clustering result in color space isn't necessary identical to that in texture space. The two clustering processes are independent of each other. How to choose the initial clustering algorithm and which algorithm to be employed isn't clarified in [17]. However, in our experiments, the initial clustering results are shown to be important to the performance and efficiency of the iterative segmentation in GrabCut. The tight and well-separated clusters will be helpful to the accuracy and efficiency of the iterative segmentation process in GrabCut. The better the initial clustering is, the more accurate the iterative segmentation is and the less the iterative time is.

There is a wide variety of excellent algorithms that could be used for the initial clustering, such as *k*-means and the improved version fuzzy *c*-means, and so on. Guided by Ruzon and Tomasi [54] and Chuang et al. [55], we use the binary tree quantization algorithm described by Orchard and Bouman [56] in color domain. In this work, color feature is extracted in L*a*b* color space as mentioned before. This algorithm starts with all pixels in a single cluster, and then calculates the mean value and covariance matrix of color over the cluster. This is then repeated to find in the resulting clusters whose covariance matrix has the largest eigenvalue and split it by using a function of the associated eigenvector as the split point until the desired number of clusters is achieved. For large clusters with Gaussian distributions it can be shown that this strategy is optimal [56]. However, in MSNST texture space, the extracted texture feature is regarded as matrix value but not vector value, and it is hard to define the matrix values' covariance matrices of the Gaussian approximation over regions due to the fact that we employ KL divergence but not the Riemannian manifold to measure the distances. Therefore, we must choose another clustering algorithm. Inspired by the work of Li et al. [15], we extend the *k*-means method by using formula (10) as a chosen distance measure and using formula (11) to calculate the cluster centre.

C. Energy Function

To define the Probability Density Function (PDF) of the extracted features, we follow the practice that is already used in [17] to construct the GMMs in color domain, the only difference is the color space applied: RGB is used in [17] and we use L*a*b*. For each GMM, there are K_C components and each component has three parameters, i.e. the vector-valued mean μ_C, the symmetric positive definite full-covariance matrix Σ_C and a real-valued mixture weighting coefficient π_C. More details refer to [17]. In MSNST texture space, we extend to use GMMs to model the statistics of MSNST feature analogously. Therefore, we can integrate the texture information appropriately into the GrabCut framework and ensure good expandability of the framework. Moreover, the color and MSNST features can be effectively fused into a unified energy function so that the good performance revealed by the traditional GrabCut method [17] can be completely inherited. Note that both the color GMMs and texture GMMs are created based on the initial clustering or learned from the previous segmentation in the iterative process.

To define the mean value of a MSNST field Γ over a region R using KL distance measure has a closed form [30] given by

$$\overline{\mathbf{M}}_T(\mathbf{\Gamma},R)=\left\{\overline{\mathbf{M}}_T(\mathbf{T}_0,R),\overline{\mathbf{M}}_T(\mathbf{T}_1,R),\ldots,\overline{\mathbf{M}}_T(\mathbf{T}_{S-1},R)\right\} \tag{11}$$

where $\left\{\overline{\mathbf{M}}_T(\mathbf{T}_s,R)=\sqrt{\mathbf{B}_s^{-1}}\left[\sqrt{\sqrt{\mathbf{B}_s}\,\mathbf{A}_s\sqrt{\mathbf{B}_s}}\right]\sqrt{\mathbf{B}_s^{-1}}\right\}_{s=0,1,\ldots,S-1}$, $\mathbf{A}_s=\int_R \mathbf{T}_s(x)dx$ and $\mathbf{B}_s=\int_R \mathbf{T}_s^{-1}(x)dx$. However, it is hard to define the covariance matrices as described in Section III-B. To address this problem, we follow the idea of [38] which gives the definition of variance in the Riemannian manifold case, and define the variance σ_T^2 of random variables over region Ω_T in the MSNST space as the expected value of the squared KL distance from the mean tensor, which is given by

$$\sigma_T^2=\frac{1}{\left|\Omega_T\right|}\int_{\Omega_T} dis_T^2(\mathbf{\Gamma}_x,\overline{\mathbf{M}}_T)dx \tag{12}$$

Therefore, each GMM in texture domain, one for the background and one for the foreground, is taken to be a general Gaussian density mixture analogous to the 1-D situation with K_T components in the MSNST texture space. For each component, there are also three parameters, *i.e.* the mean value $\overline{\mathbf{M}}_T$ (a set of matrixes of size 2×2), variance σ_T^2 (a real) and weight π_T (a real), where π_T can be easily obtained by computing the percentage between the number of points in this component and the number of all the points in the foreground or background.

The GMMs approximation is very robust for color channels as various experimental results have been shown for the original GrabCut [17]. When it is combined with the proposed approach for the MSNST field, the method can deal with a much larger range of images. In order to take into account the both extracted features, the general energy function is proposed as follows

$$E(\alpha)=\xi E_C(\alpha)+(1-\xi)E_T(\alpha) \tag{13}$$

where α denotes the assigned label, with 0 for background and 1 for foreground, and ξ is the mixing factor used to balance the relative weights of the color and texture based energy terms.

The color based energy term $E_C(\alpha)$ takes the following form

$$\begin{aligned}E_C(\alpha)=&\sum_{u\in U}-\log\sum_{k=1}^{K_C}\left\{\pi_C(\alpha_u,k)\frac{1}{\sqrt{(2\pi)^3\left|\Sigma_C(\alpha_u,k)\right|}}\times\exp\right.\\&\left.\left(-\frac{1}{2}\left[I_u-\mu_C(\alpha_u,k)\right]^{\mathrm{T}}\Sigma_C(\alpha_u,k)^{-1}\left[I_u-\mu_C(\alpha_u,k)\right]\right)\right\}\\&+\sum_{(m,n)\in\mathbf{N}}\left[\alpha_m\neq\alpha_n\right]\left\{\gamma_C dis^{-1}(m,n)\exp\left(-\beta_C dis_{\mathrm{C}}^2(I_m,I_n)\right)+\tau\right\}\end{aligned} \tag{14}$$

where (α_u,k) denotes the *k*-th component of foreground GMM when $\alpha_u=1$ or background GMM when

$\alpha_u = 0$ for each vertex $u \in U$, U represents the vertices located in the supplied rectangle, **N** is the set of pairs of neighboring pixels, τ is the denoising constant we newly include, other notations follow the original paper [17].

The texture based energy term $E_T(\alpha)$ is defined as

$$\begin{aligned} E_T(\alpha) = \sum_{u \in U} -\log \sum_{j=1}^{K_T} \left\{ \frac{\pi_T(\alpha_u, j)}{\sqrt{2\pi\sigma_T^2(\alpha_u, j)}} \exp\left(-\frac{dis_T^2(\Gamma_u, \overline{\mathbf{M}}_T(\alpha_u, j))}{2\sigma_T^2(\alpha_u, j)} \right) \right\} \\ + \sum_{(m,n)\in \mathbf{N}} [\alpha_m \neq \alpha_n] \left\{ \gamma_T dis^{-1}(m,n) \exp\left(-\beta_T dis_T^2(\Gamma_m, \Gamma_n) \right) + \tau \right\} \end{aligned} \tag{15}$$

We can adaptively set β_C and β_T during segmentation to be

$$\beta_C = \left(2 \frac{\sum_{(m,n)\in \mathrm{N}} dis_{\mathrm{C}}^2(I_m, I_n)}{|\mathrm{N}|} \right)^{-1} \quad and \quad \beta_T = \left(2 \frac{\sum_{(m,n)\in \mathrm{N}} dis_T^2(\Gamma_m, \Gamma_n)}{|\mathrm{N}|} \right)^{-1} \tag{16}$$

where $|\mathrm{N}|$ denotes the number of pairs in the set $\mathbf{N}$.

D. Adaptive Fusion

In previous works [26] and [28], the color and texture are approximate equally weighted as $\xi = 0.5$ by simply augmenting the feature vector, which make the mixing factor can not be adjusted efficiently and robustly to reduce the negative side effect. This will make the discriminating power of main feature decrease especially for the boundaries with low contrast. Although color is the main feature when dealing with natural images, in many important cases, texture information is often a more appropriate discriminating feature when the foreground and background differ more distinct in texture. Therefore, effectively fusing the two features will greatly improve the performance of the algorithm to segment natural images. To achieve this goal, we can set this parameter manually based on the experience of the user. However, to maximize robustness, an ideal system should adaptively adjust the mixing factor. A mixture fusion technique is proposed in [41], which adjusts this parameter depending on the relative discriminative power of texture and color terms. In order to reflect that of texture term, [41] assumes that the foreground and background satisfy the single Gaussian distribution with zero mean and mean tensor as covariance matrix, then computes the relative weight using the overlapping between both distributions. With respect to color term, Euclidean distance between the mean color values of foreground and background is employed.

In this paper, we can adaptively mix two models only according to the main feature of color and don’t need to measure the discriminative power of both feature terms so as to avoid adjust normalizing factor. In order to measure the discriminative power in color space, we follow the idea of considering the PDFs in both foreground and background regions and compute their KL distance as [19]. A large value for that distance means that the foreground and background features can be well separated. However, in our case the image is represented by a continuous probabilistic framework based on GMM but not single PDF, and there is no closed form expression for the KL divergence between two GMMs. In this paper, we adopt an approximation of the KL divergence between two GMMs models [39]

$$KL(GMM^F \| GMM^B) = \sum_{k=1}^{K^F} \pi_k^F \min_{i \in \{1,\ldots,K^B\}} (KL(N_k^F \| N_i^B) + \log \frac{\pi_k^F}{\pi_i^B}) \tag{17}$$

Here, the superscripts F and B denote the corresponding variables belong to foreground or background respectively, and N_k^F and N_i^B are the k-th component of foreground GMM and the i-th component of background GMM respectively. The KL divergence between N_k^F and N_i^B can be computed in L*a*b* color space as

$$\begin{gathered} KL_C(N_{C,k}^F \| N_{C,i}^B) = \frac{1}{2}(\log \frac{|\Sigma_{C,i}^B|}{|\Sigma_{C,k}^F|} + tr((\Sigma_{C,i}^B)^{-1}\Sigma_{C,k}^F) + \\ (\mu_{C,k}^F - \mu_{C,i}^B)^T (\Sigma_{C,i}^B)^{-1} (\mu_{C,k}^F - \mu_{C,i}^B)) \end{gathered} \tag{18}$$

where $tr(\cdot)$ is the matrix trace operator, see [39] for more details.

Consequently, our adaptive estimation of ξ can be designed to be

$$\xi = 1 - \exp\left(-KL_C(GMM_C^F, GMM_C^B) / \sigma_{KL_C}\right) \tag{19}$$

where σ_{KL_C} is a parameter to control the influence of KL_C, which can be used to balance the relative weights of texture and color terms. If the foreground and background color can be well separated, i.e., KL_C is large, then ξ is set to be large and the result will mainly rely on the color-based term. Otherwise, ξ is small and the texture-based term will make more contribution.

E. Iteration Convergence Criterion

The minimum of the general energy $E(\alpha)$ will yield a globally optimal segmentation for the current iteration of the iterative process or for the refine editing. In the process of iterative segmentation, when does the convergence happen? The straightforward criterion is to check whether the labels assigned to the pixels of image change or not after the iteration. However, it is often a waste of computing resources and prone to vibration. In this paper, an adaptive criterion can be defined as the KL distance between the current foreground and background in both color and texture fields, $\mathbf{KL} = \left(KL_C \;\; KL_T\right)^{\mathrm{T}}$. The iteration automatically terminates when the following formula holds

$$\left\|\mathrm{KL}_{\Lambda} - \mathrm{KL}_{\Lambda-1}\right\|^2 \le \sigma \left\|\mathrm{KL}_1 - \mathrm{KL}_0\right\|^2 \tag{20}$$

where KL_{Λ} indicates the vector composed of KL_C and KL_T at the Λ-th iteration, especially KL_0 indicates the KL distance between the initial foreground and background formed by the rectangle placed around the object by user, and σ is a decreasing coefficient which can be used to control the convergence speed and segmentation accuracy. Obviously, $\Lambda \ge 2$. The first Λ satisfying the constraint (20) gives the iterative numbers. The computation of KL_T is similar to KL_C by (17). However, the component of GMMs in MSNST texture space is defined as a general Gaussian density analogous to the 1-D situation, thus the KL divergence between N_k^F and N_i^B can not be computed using (18). Fortunately this situation is a special case of the generalized Gaussian density (GGD) [36], where the shape parameter is fixed and equals to 2. Then we can get the following closed form for the KL divergence between them in MSNST texture space

$$KL_T(N_{T,k}^F \left\| N_{T,i}^B\right.) = \frac{1}{2}(\log\frac{(\sigma_{T,i}^B)^2}{(\sigma_{T,k}^F)^2} + \frac{(\sigma_{T,k}^F)^2}{(\sigma_{T,i}^B)^2} - 1) \tag{21}$$

Since KL_C can be computed by (19) for dynamically adapting the relative weight, we only need to compute KL_T additionally, which is easy to compute using formula (21). A number of experiments show that this strategy can reduce the time of the iteration dramatically with satisfied segmentation accuracy.

IV. Experimental Results

In this section, a large set of color images with natural scenes have been used to test the performance of the proposed method. The compared experiments using the synthesized texture images and the real images based on GrabCut framework between MSNST and Gabor wavelet with different scales are carried out to reveal the powerful texture discriminating capability of MSNST. The comparison between different texture and color measures also demonstrates that the selected texture distance measure (KL measure) and color distance measure (Conjugate norm) can distinguish the regions more effectively than the usually used Euclidean measure. We also experimentally demonstrate that the KL measure is robust enough compared with the Riemannian measure for our reliance on the GMM statistics. Moreover, the effects of Section III-D and Section III-E are experimentally justified in this section. Finally, we conduct a number of compared experiments using the real images to demonstrate the superiority of the proposed method integrating the MSNST texture feature and color feature in GrabCut framework.

We employ the publicly available implementation of Gabor filters [34] and Maxflow Graph Cuts [14] in our experiments and build the whole system in c++. In order to simplify the user interaction and make the comparison more accurate, the initial "incomplete labelling" is supplied by a rectangle but not a lasso, and every group of the compared experiments share the same rectangle. Moreover, in all the experiments we only compute three orientations for each scale for Gabor filters as MSNST yields only three feature channels for each scale. Since the classical structure tensor is the special case of MSNST whose scale is one, in our experiment we denote the classical structure tensor as MSNST with one scale. It should be noticed that for most of the experiments expect for Fig. 11 only the iterative segmentation process of GrabCut framework has been used in this paper, which is the main advantage of the GrabCut method compared with the other interactive segmentation methods based on Graph Cuts and leads to simpler user interaction and better segmentation performance.

There are a number of parameters that must be appropriately determined for the implementation of the proposed method. Parts of the default values for these parameters have been given when we described the corresponding algorithms. For the point of clarity and integrity, we give the description of the parameters setting again. The implementation of the Gabor filters is the same as in [34], where the lower and upper center frequencies of interest are set 0.05 and 0.4 respectively and the radius is chosen as 60 to build the filter coefficients for the convolution. The parameters ε and p in formula (5) are fixed as 0.001 and 0.6, respectively. When implementing the nonlinear diffusion (4) using the AOS scheme, the time step and the number of steps are held as 5000 and 2 respectively in all the experiments. The parameters K_C and K_T in formula (14) and (15) denote the number of components in the GMMs for color and texture, respectively. The both parameters are always chosen as 5. The γ_C and γ_T in formula (14) and (15) are used to control the smoothness of color and texture, respectively. We fix them as 5 in all the experiments. The denoising constant τ in formula (14) and (15) is fixed as 2.5. The σ_{KL_C} in formula (19) controls the adaptive mixing factor and can be set as 10 by optimizing performance against ground truth over a training set of more than 20 images. The σ in formula (20) influences the speed of terminating and can be simply set as 0.01 for all the images.

The first experiment is that of a synthesized texture image with five different textures in Fig. 2, we display the obtained multi-scale structure tensor and justify the problem of standard Gaussian smoothing and the superiority of nonlinear diffusion. The original texture images are from the Brodatz texture database, and we set

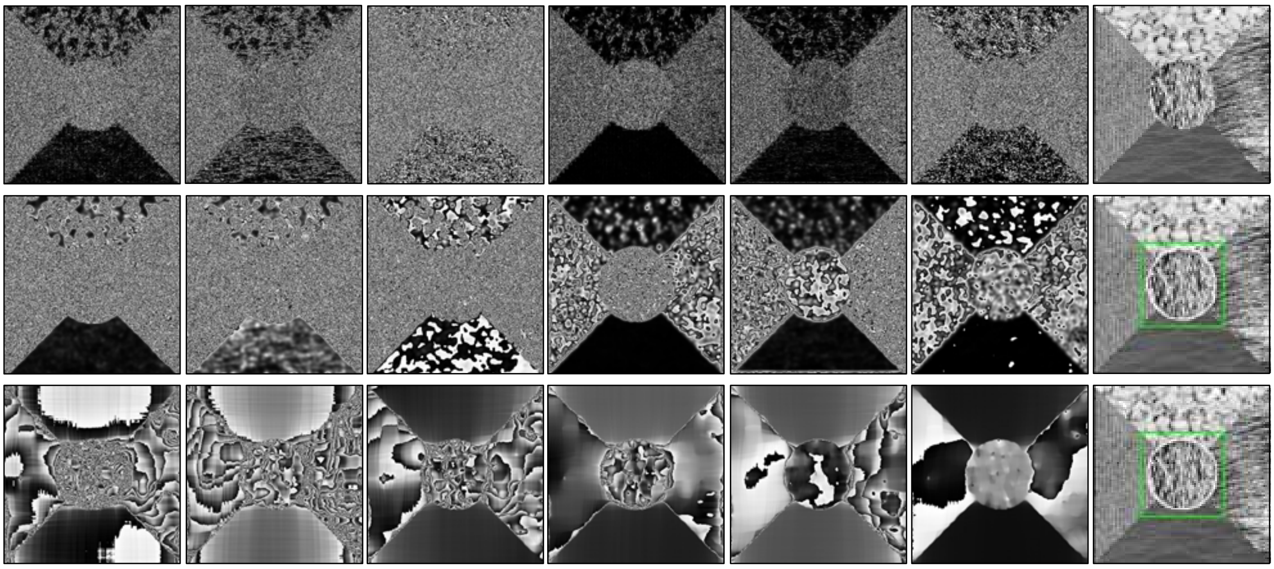


Fig. 2 The compared results between Gaussian smoothing and nonlinear diffusion for different channels (the first six columns from left to right given by $\mathbf{T}_0(1,1)$, $\mathbf{T}_0(2,2)$, $\mathbf{T}_0(1,2)$, $\mathbf{T}_1(1,1)$, $\mathbf{T}_1(2,2)$ and $\mathbf{T}_1(1,2)$) of multi-scale structure tensor and the final segmentations (the last column). Row 1 shows the obtained multi-scale structure tensor and the original image; Row 2 shows the corresponding results by Gaussian smoothing and the final segmentation; Row 3 shows the corresponding results by nonlinear diffusion and the final segmentation.

the standard deviation as 3 similar to [28] when applying the Gaussian smoothing. In this experiment, no color information is used so that we can obtain more convictive comparison. The results in the first six columns of Fig.2 show that the second scale of structure tensor has more powerful discriminating capability than the first scale, and the nonlinear diffusion with a more superior performance compared with the standard Gaussian smoothing in terms of denoising and preserving of edges. In addition, the final segmentations demonstrate that the standard Gaussian smoothing really suffers from the dislocation of edges while nonlinear diffusion performs better.

In Fig.3-5, we test the texture region discriminating capability of different texture features and their different distance measures based on GrabCut framework using the synthesized texture. In the three experiments, only texture feature is considered and color information isn’t used so that we can obtain more accurate comparison between different texture descriptions with changing scales and between different texture distance measures. All the textures used to synthesize the experimental images in Fig. 3-5 are of the same type respectively, and only differ in the orientation and scale. The five texture regions in the test image of Fig. 3 only differ in orientations and have the same scales, and the orientation differences between the five texture regions are at most $\pi/9$. The results in Fig. 3 show that the structure tensor has more powerful orientation discriminating capability and MSNST texture with only one scale can distinguish the texture region better than the Gabor filters with three scales. The main reason for this is that the Gabor filters selected by us include only three orientations. This means that they can only deal with the orientation difference which is bigger than $\pi/3$ when the scale is one. But in the synthesized texture images, the orientation differences between the five texture regions are at most $\pi/9$, which

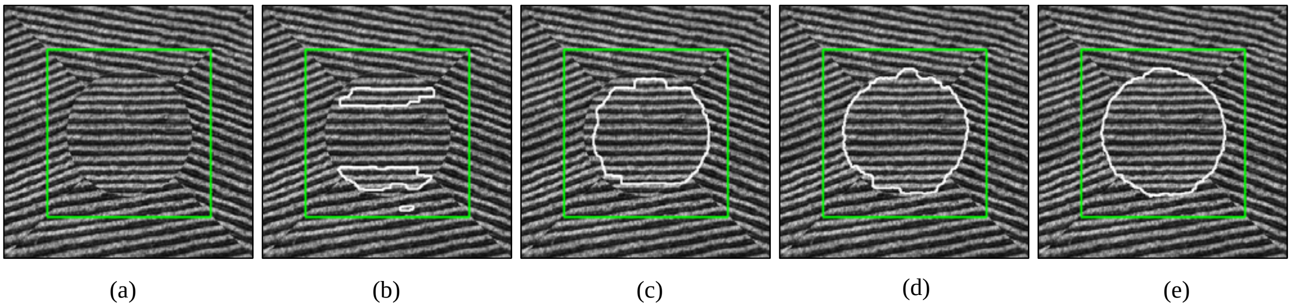


Fig. 3 (a) Gabor wavelet with one scale; (b) Gabor wavelet with two scales; (c) Gabor wavelet with three scales; (d) MSNST with one scale using Euclidean measure; (e) MSNST with one scale using KL measure;

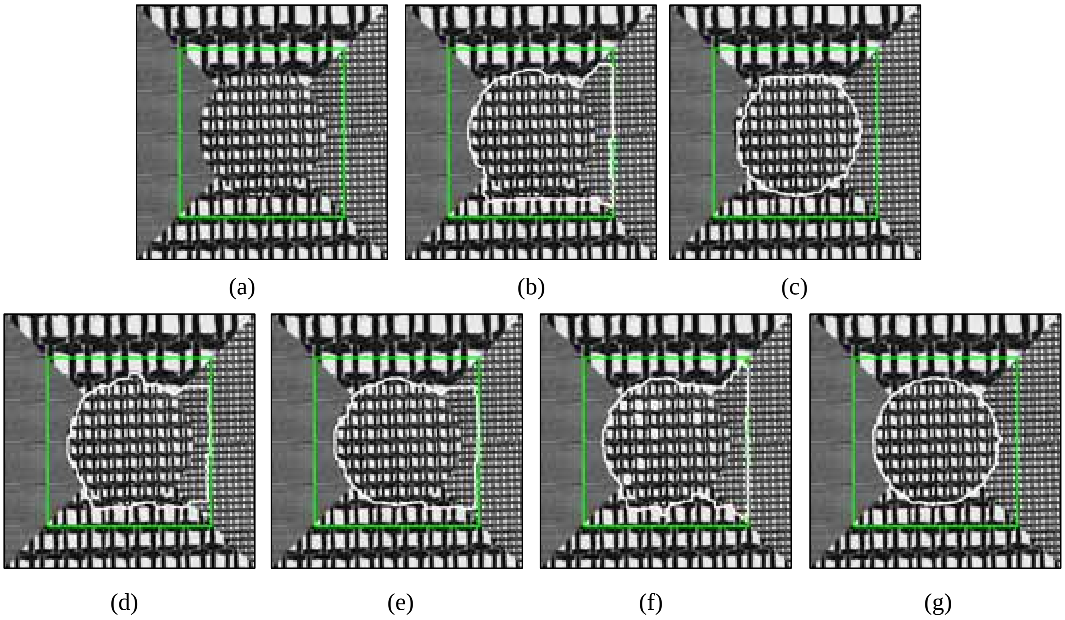


Fig. 4 (a) Gabor wavelet with one scale; (b) Gabor wavelet with two scales; (c) Gabor wavelet with three scales; (d) MSNST with one scale using Euclidean measure; (e) MSNST with one scale using KL measure; (f) MSNST with two scales using Euclidean measure; (g) MSNST with two scales using KL measure;

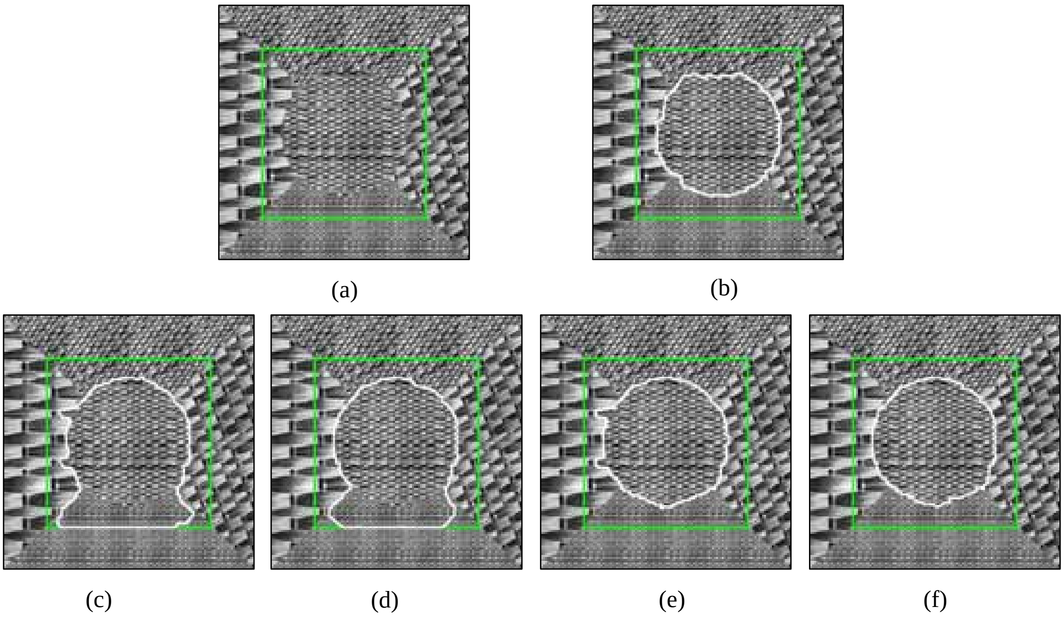
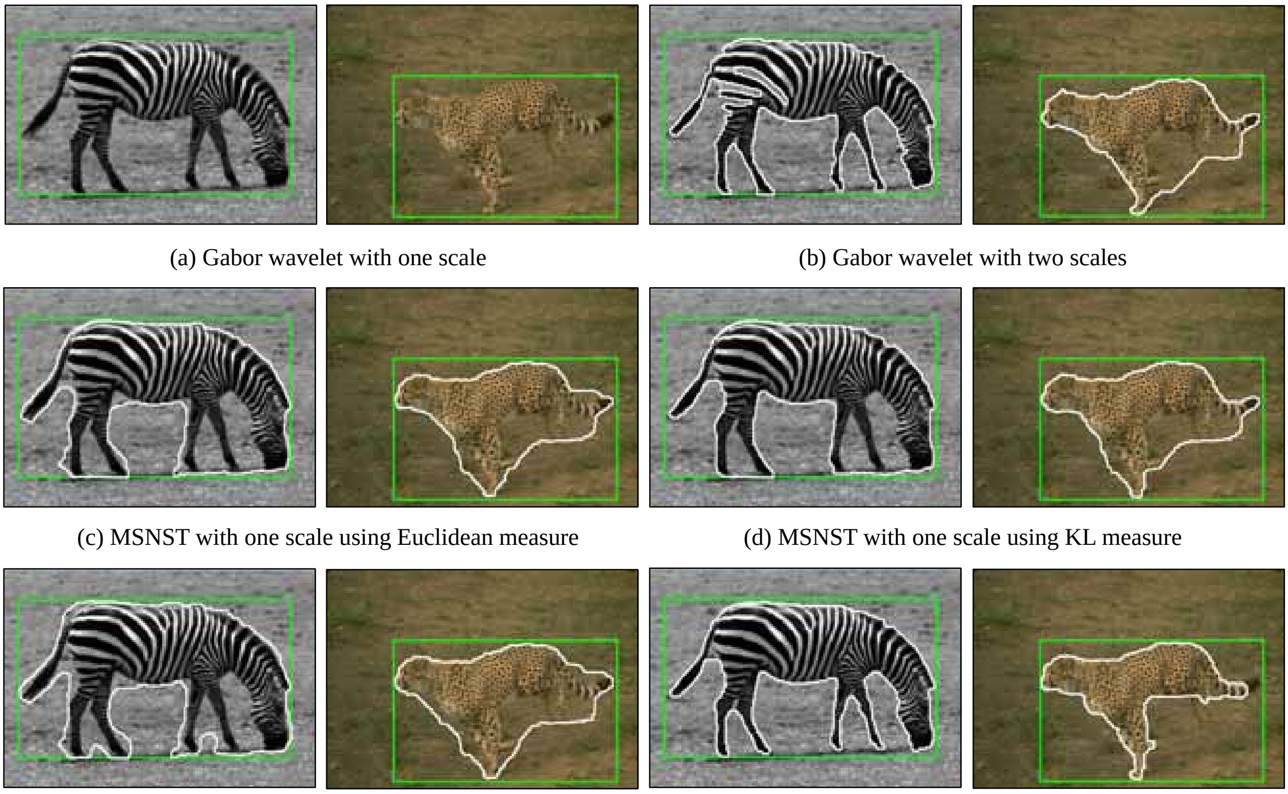


Fig. 5 (a) Gabor wavelet with one scale; (b) Gabor wavelet with two scales; (c) MSNST with one scale using Euclidean measure; (d) MSNST with one scale using KL measure; (e) MSNST with two scale using Euclidean measure; (f) MSNST with two scales using KL measure.



(a) Gabor wavelet with one scale

(b) Gabor wavelet with two scales

(c) MSNST with one scale using Euclidean measure

(d) MSNST with one scale using KL measure

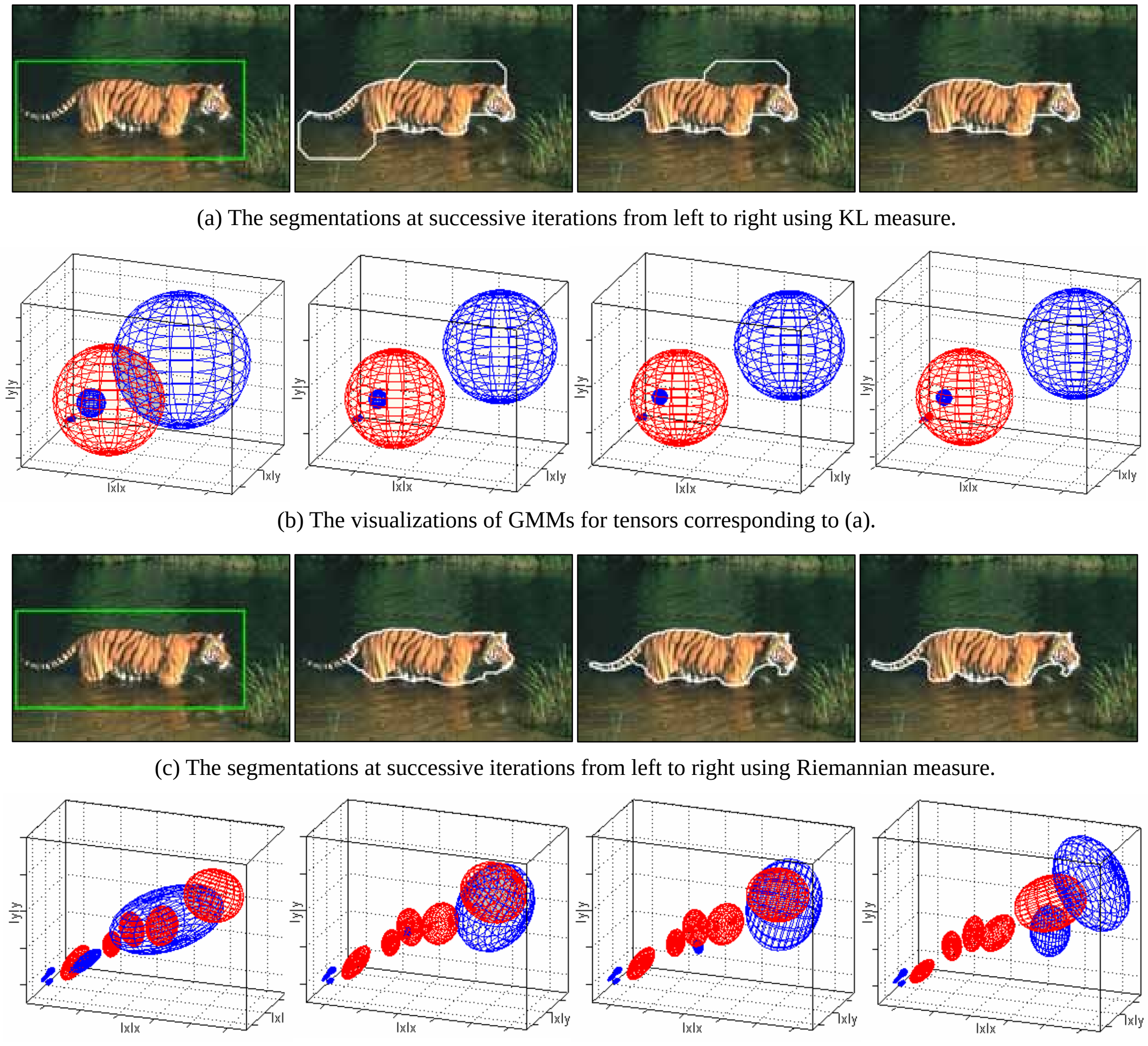
(e) MSNST with two scales using Euclidean measure

(f) MSNST with two scales using KL measure

Fig. 6 The compared experimental results based on different texture features, no color information is used.

makes the Gabor filters fail when the scale is one or two (see Fig. 3(a, b)). Until the scale increases to 3 (see Fig. 3(c)) the Gabor filters can roughly discriminate the texture object with inaccurate boundary. The results reveal that the structure tensor can describe the orientation information of textures more effectively than the Gabor filters, and actually can represent the whole orientation space of textures, as just like mentioned above. The compared results between Fig. 3(d) and 3(e) show that the KL divergence is more suited to discriminate MSNST texture feature than the Euclidean measure.

The five texture regions in the test image of Fig. 4 only differ in scales and have the same orientations. The Gabor filters can roughly discriminate the texture object with inaccurate boundary until the scale increases to 3 (see Fig.4(c)). The classical structure tensor [28], i.e. MSNST with one scale (in Fig.4(d, e)), has no way to separate the texture object, and the MSNST with two scales using the KL divergence works well, and can extract the object perfectly (in Fig.4(g)). In the synthesized texture image on Fig. 5, the five texture regions differ in both orientations and scales. As the difference between the five texture regions is more distinct than in Fig. 3 and 4, the Gabor filters with two scales can roughly discriminate the texture object (see Fig.5(b)), but MSNST with two scales using the KL divergence gives more prefect result (see Fig. 5(f)). Moreover, the Gabor filters with one scale (Fig. 5(a)) completely fail to distinguish anything, just like in Fig. 3(a) and Fig. 4(a), which is much worse than MSNST with one scale does. Therefore, using the MSNST texture can obtain better results with less feature channels than the Gabor filters. The compared results in Fig. 4 and 5 also reveal that the KL divergence is more suited to discriminate the MSNST texture feature than the Euclidean measure.



(a) The segmentations at successive iterations from left to right using KL measure.

(b) The visualizations of GMMs for tensors corresponding to (a).

(c) The segmentations at successive iterations from left to right using Riemannian measure.

(d) The visualizations of GMMs for tensors corresponding to (c).

Fig. 7 The compared results between KL measure and Riemannian measure with one scale MSNST for our reliance on the GMM statistics, no color information is used.

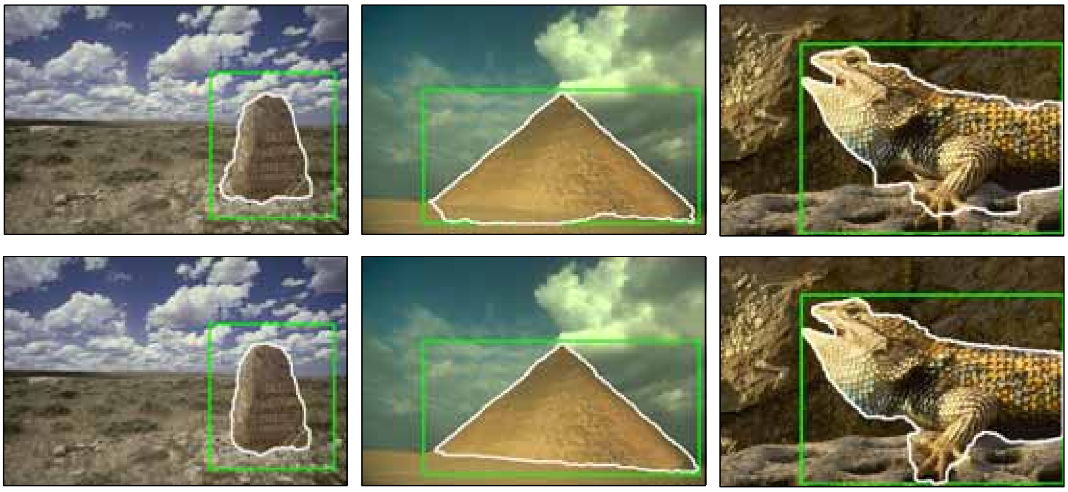
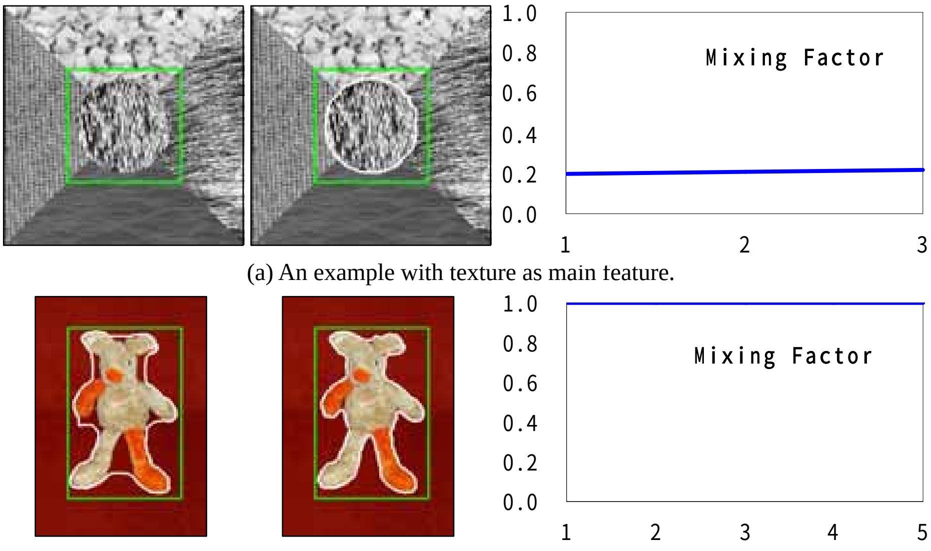


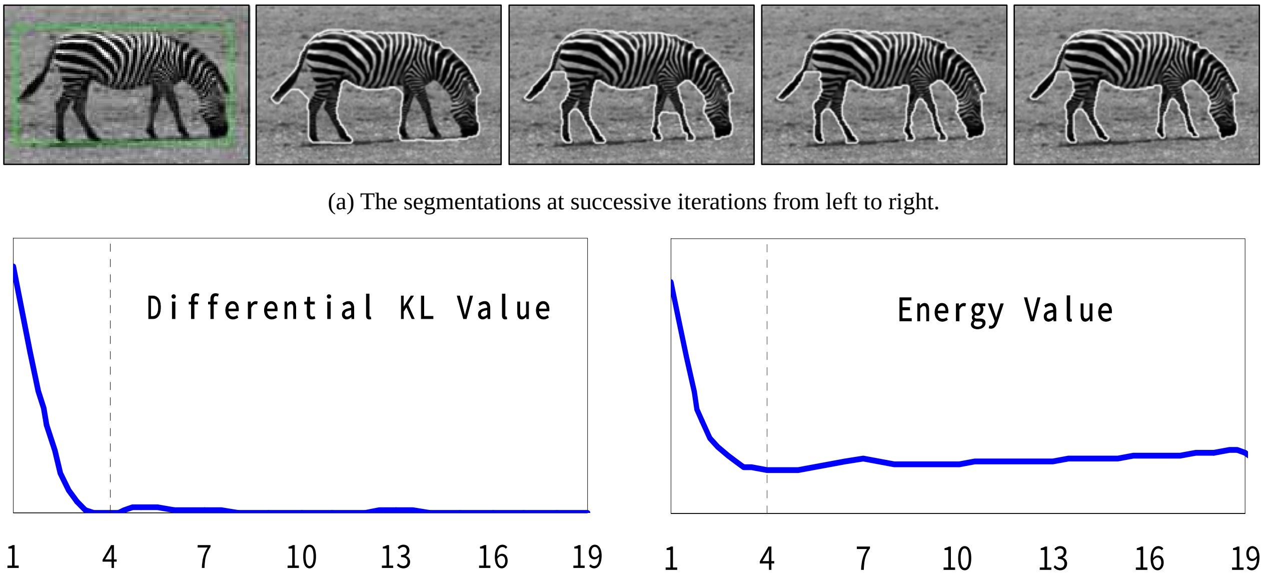
Fig. 8 The compared results between different color space and color distance measure. Just color information is used. Top: RGB + Euclidean norm; Bottom: L*a*b* + Conjugate norm.



(a) An example with texture as main feature.

(b) An example with color as main feature.

Fig. 9 The first column of (a) and (b) shows the results of fixing the mixing factor at 0.5; the middle column shows the results of our proposed adaptive fusion strategy; the last column shows the computed fixing factors at successive iterations respecting to the adaptive fusion.



(a) The segmentations at successive iterations from left to right.

(b) The differential KL value changes at successive iterations.

(c) The energy value changes at successive iterations.

Fig. 10 The displaying of segmentations, differential KL values and energy values at successive iterations.

Even though synthetic images allow us to precisely demonstrate the goodness of the segmentation method and enable a direct comparison with related approaches, the use of real-world images is more interesting and can also provide insight into the segmentation performance. The test results of the real-world images shown in Fig. 6 also demonstrate the superiority of the MSNST texture with the KL divergence. In addition, in order to verify that our choice of KL measure is robust enough compared with the Riemannian measure, we try mapping our MSNST using the procedure in [28] and constructing the GMM statistics of tensors following the ideas of [52, 53] respecting the Riemannian structure in Fig.7. The compared results in Fig. 7 demonstrate that with the iterations going on, our choice of KL measure can still robustly separate the foreground GMM (blue) from the background

GMM (red) in tensor space gradually like the Riemannian measure does, the foreground/background labelling becomes more and more accurate and nearly the same final segmentation results are obtained with little difference. However, the computation of Riemannian measure is at least 5 times slower than the using of a low-dimensional parametric representation such as KL divergence in our implementation. Therefore, for easier implementation and more efficient computation, the KL measure is chosen in our method since it is robust enough with our reliance on the GMM statistics for compensating. Notice that all these examples in Fig.6 and Fig.7 were performed using only the texture feature and no color information was used. For the convenience of visualizing the MSNST GMMs in the limited visualizing space, we only include one scale MSNST in Fig. 7 and the mean tensor manifolds are first mapped to a Euclidean tangent plane around the identity similar to [53].

The experiments presented in Fig. 8 give the comparison between different color feature spaces and color distance measures. We compare the color feature discriminating capability between RGB space + Euclidean norm used in the original GrabCut [17] and L*a*b* space + Conjugate norm used in our method. Note that these examples were performed using only the color information without any textured feature included. In the three test images in Fig. 8, all the objects surrounded by the rectangle given by user include the boundary with low contrast, which is difficult to completely determine by using the RGB color space and the Euclidean norm. However, by using the L*a*b* color space and the Conjugate norm, we can obtain more accurate results. We should notice that when the foreground and background have quite discriminative color distributions, there is little difference between these two strategies. But for the images with low contrast boundary like that in Fig. 8, our method is more robust and accurate than the traditional GrabCut when dealing with color natural images.

In Fig. 9, our proposed adaptive fusion strategy in Section III-D was tested against fixing the mixing factor at 0.5. Except for the mixing factor, the other parameters are set the same as described above. We include two extreme examples, one with texture as main feature and the other with color as main feature. The experimental results demonstrate the robustness of this strategy to estimate the main feature and obtain more accurate segmentations. When fixing the mixing factor at 0.5, Fig. 9(a) fails to segment the expected object because too much useless color information is included, while in Fig. 9(b) because too much useless texture information is included the segmentation accuracy isn’t very good. However, our fusion strategy can estimate the main feature and the approximate percentage, and reduce the negative side effect of including too much useless information.

In order to display the iterative process and justify the effect of our proposed new iteration convergence criterion in Section III-E, we show the iterative process of the zebra image in the first row of Fig.10. The results in Fig. 10(b) and Fig. 10(c) show that the proposed iteration convergence criterion is identical to the energy decreasing process, and can estimate the convergence robustly. If we use the changes of labelling as the criterion

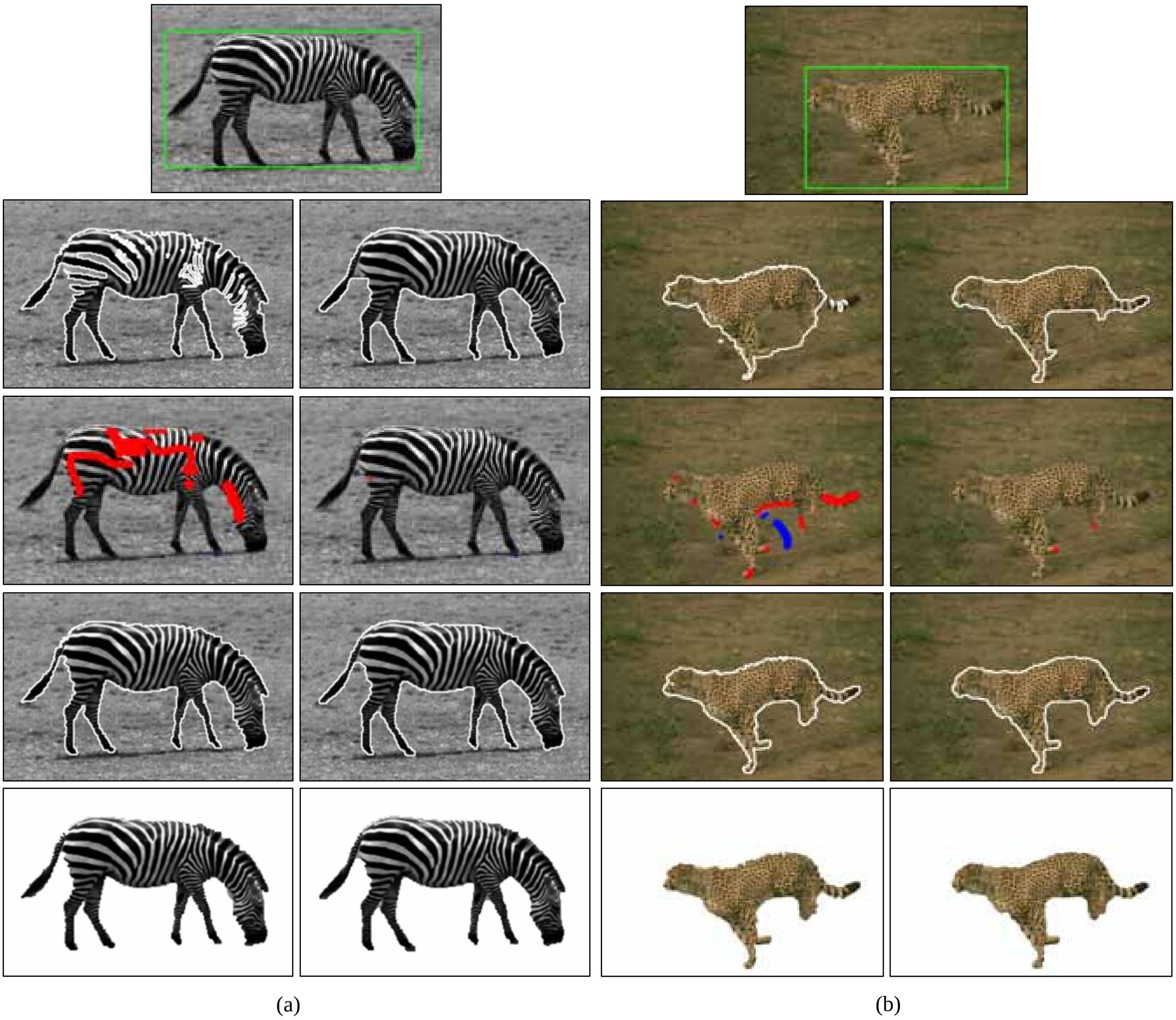


Fig. 11 The systematical compared experimental results between GrabCut [17] and our method for all the stages on two real-world images. (a) zebra. (b) leopard. The left of (a) and (b) shows the results of the GrabCut [17] and the right of (a) and (b) shows the results of the proposed method. Row 1 shows the original images with the rectangle around the object; Row 2 shows the results of iterative segmentation process; Row 3 shows the marks for the refining which are placed by user; Row 4 shows the result of the refining process; Row 5 shows the last object regions

like [17] does, the algorithm will converge after over 19 iterations. However, our proposed method terminates after 4 iterations with satisfied segmentation accuracy. Therefore, the computational efficiency is greatly enhanced.

The experiments in Fig. 11 compare the proposed method with the original GrabCut [17]. In the proposed method, we integrate the MSNST texture feature into the GrabCut framework. The color feature is represented by using the L*a*b* color space with the Conjugate norm and the texture feature is represented by using the MSNST feature with two scales and the KL divergence. The parameters are set as described above and the color feature and texture feature are adaptively fused by the scheme introduced in Section III-D. The comparison process includes the iterative segmentation process and the further refining by user editing. Notice that in the refining

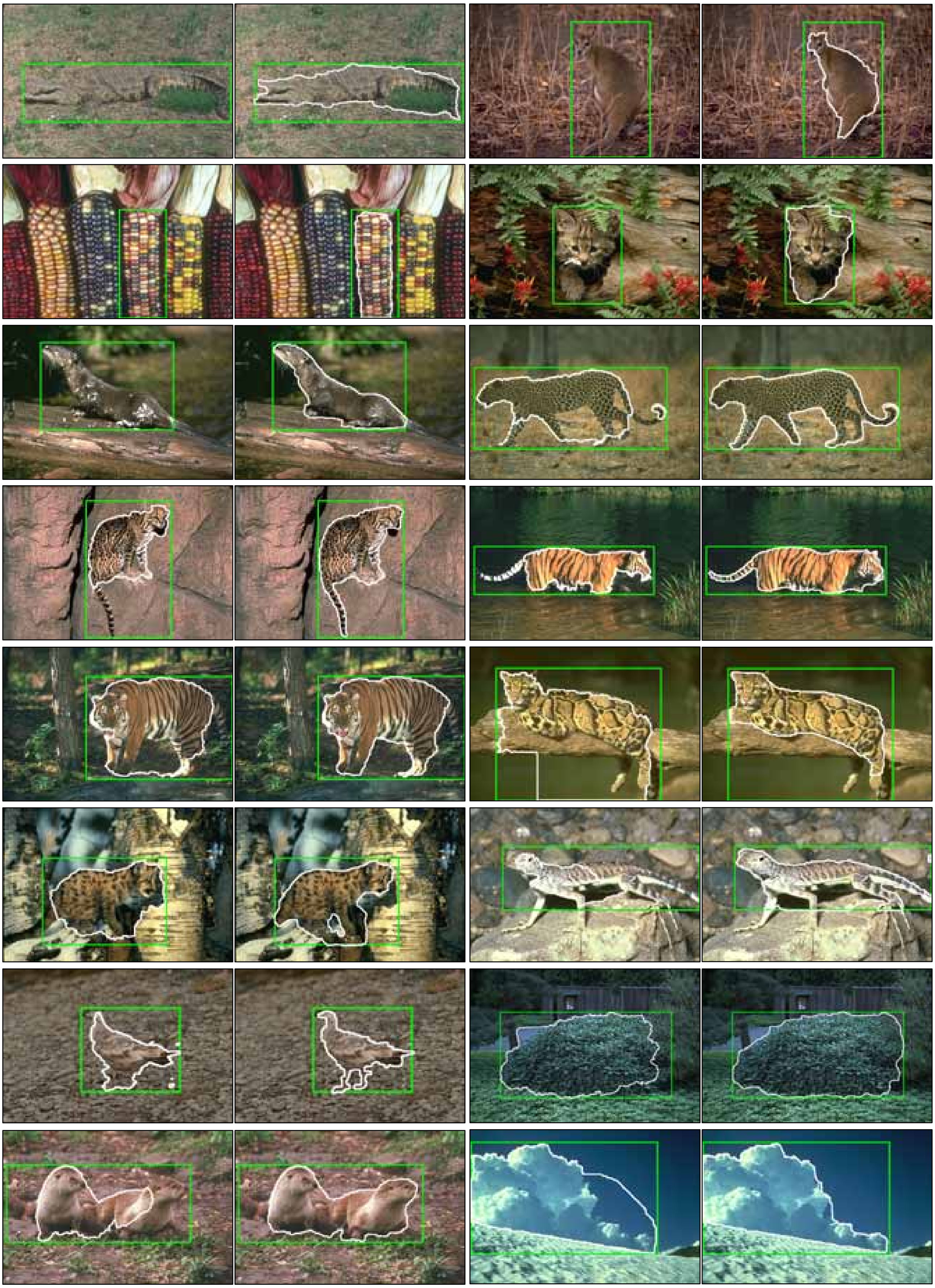


Fig. 12 The compared experimental results between the GrabCut [17] and our method. All the images come from the Berkeley Segmentation Dataset [35]. The colomn 1 and 3 are the results by GrabCut [17], and the colomn 2 and 4 are the results by our method.

process of the proposed method, we still use the integrated feature description just like the iterative process does. This makes the refining process of the proposed method also more effective than that of the original GrabCut [17]. In Fig. 11 two groups of the comparing experimental results are shown: the one is a zebra (Fig.11(a)) and the other is a leopard (Fig. 11(b)). From the results, we can find that the proposed method can get better iterative segmentation results. Thus, in the refining stage (in row 3 and 4), the proposed method can get the prefect results by adding less refining strokes. Moreover, from the final results (in row 5), the boundaries of the object regions are more accurate and smoother by the proposed method than by the original GrabCut [17]. This demonstrates that the MSNST texture feature is very essential to segment the images with rich texture information.

A number of comparing experimental results are shown in Fig. 12 using the images from the well known Berkeley Segmentation Dataset (BSDS) [35]. The selected feature spaces and distance measures of color and texture information and the parameters setting are the same as the experiments in Fig.11. The comparison between the proposed method and the original GrabCut method [17] includes only the iterative segmentation process with the same initialized rectangle placed by user. The results in Fig. 12 demonstrate the superiority of the proposed method.

V. Conclusion

An interactive color image segmentation method integrating MSNST texture feature based on the GrabCut framework is proposed to achieve an improved segmentation performance. We propose to exploit the MSNST to describe the texture feature of images in such a way that not only the orientation texture difference can be dealt with effectively, but also the scale problem is overcome perfectly. Applications of our method to synthetic texture images show that it is more powerful to discriminate the texture objects than the Gabor filters and the classical structure tensor [28]. The energy function for the iterative process of the GrabCut method is constructed based on the adaptively integrated L*a*b color and MSNST texture features by extending the color GMM used in [17] to MSNST texture with more discriminating distance measures, such as the Conjugate norm for color and the symmetric KL divergence for MSNST. The performance comparison between the proposed method and the original GrabCut method [17] is conducted using a large set of color images with natural scenes. The comparison demonstrates that the proposed method can achieve more superior interactive image segmentation results by simply placing a rectangle around the objects of interest so that the further user refining is less or even not needed. As future work, we would like to improve the ideas of [52, 53] and solve the GMM statistics respecting the Riemannian structure in closed form or speed it up, which will let us design a more discriminative and more computationally practicable segmentation process.

REFERENCES

[1] H. D. Cheng, X. H. Jiang, Y. Sun and J. Wang, "Color image segmentation: advances and prospects", *Pattern Recognition*, vol.34, pp. 2259-2281, 2001.

[2] E. N. Mortensen, "Simultaneous Multi-Frame Sub-pixel Boundary Definition using Toboggan-Based Intelligent Scissors for Image and Movie Editing", Ph.D. Thesis dissertation, Department of Computer Science, Brigham Young University, Provo, UT., 2000.

[3] D. H. Ballard and C. M. Brown, *Computer Vision*, Prentice Hall Professional Technical Reference, 1982.

[4] J. T. Tou and R. C. Gonzalez, *Pattern recognition principles*, Addison-Wesley Reading, Mass, 1974.

[5] L. J. Reese, "Intelligent Paint: Region-Based Interactive Image Segmentation", *Master's thesis, Brigham Yound University*, 1999.

[6] M. Sonka, V. Hlavac and R. Boyle, *Image processing, analysis, and machine vision. 2nd*, vol.770. Pacific Grove, CA: PWS Publishing, 1999.

[7] R. E. Bellman and S. E. Dreyfus, *Applied dynamic programming*, Princeton: NJ: Princeton University Press, 1962.

[8] N. Xu, N. Ahuja and R. Bansal, "Object segmentation using graph cuts based active contours", *Computer Vision and Image Understanding*, vol.107, pp. 210-224, 2007.

[9] E. N. Mortensen and W. A. Barrett, "Intelligent scissors for image composition", *Proc. ACM Siggraph*, pp. 191-198, 1995.

[10] S. Wang and J. M. Siskind, "Image segmentation with ratio cut", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.25, pp. 675-690, 2003.

[11] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: theory and its application to image segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.15, pp. 1101-1113, 1993.

[12] J. Shi and J. Malik, "Normalized cuts and image segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.22, pp. 888-905, 2000.

[13] Y. Boykov and M. P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images", *Proc. International Conference on Computer Vision*, pp. 105-112, 2001.

[14] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.26, pp. 1124-1137, 2004.

[15] Y. Li, J. Sun, C. K. Tang and H. Y. Shum, "Lazy snapping", *Proc. SIGGRAPH Conference*, pp. 303-308, 2004.

[16] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient N-D image segmentation", *International Journal of Computer Vision*, vol.70, pp. 109-131, 2006.

[17] C. Rother, V. Kolmogorov and A. Blake, "GrabCut: interactive foreground extraction using iterated graph cuts", *ACM Transactions on Graphics (TOG)*, vol.23, pp. 309-314, 2004.

[18] H. Lombaert, Y. Sun, L. Grady and C. Xu, "A multilevel banded graph cuts method for fast image segmentation", *Proc. International Conference on Computer Vision*, 2005.

[19] J. Cardelino, G. Randall and M. Bertalmio, "An Active Regions Approach for the Segmentation of 3D Biological Tissue", *Proc. IEEE International Conference on Image Processing*, pp. 277-280, 2005.

[20] G. R. Cross and A. K. Jain, "Markov Random Field Texture Models", *IEEE Transactions on Pattern Analysis andMachine Intelligence*, vol.5, pp. 25 – 39, 1983.

[21] B. S. Manjunath and R. Chellappa, "Unsupervised texture segmentation using Markov random field models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.13, pp. 478-482, 1991.

[22] C. Bouman and B. Liu, "Multiple resolution segmentation of textured images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.13, pp. 99-113, 1991.

[23] B. G. Kim, J. I. Shim and D. J. Park, "Fast image segmentation based on multi-resolution analysis and wavelets", *Pattern Recognition Letters*, vol.24, pp. 2995-3006, 2003.

[24] B. Sandberg, T. Chan and L. Vese, "A level-set and Gabor-based active contour algorithm for segmenting textured images", Mathematics Department, UCLA, Los Angeles, USA, Technical Report 39, 2002.

[25] C. Sagiv, N. A. Sochen and Y. Y. Zeevi, "Texture segmentation via a diffusion-segmentation scheme in the gabor feature space", *Proc. Texture 2002, 2nd International Workshop on Texture Analysis and Synthesis,* pp. 123-128, 2002.

[26] M. Rousson, T. Brox, R. Deriche, O. I. Projet and F. Sophia-Antipolis, "Active unsupervised texture segmentation on a diffusion based feature space", *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[27] Z. Wang and B. Vemuri, "Tensor field segmentation using region based active contour model", *Proc. European Conference on Computer Vision*, pp. 304-315, 2004.

[28] J. Malcolm, Y. Rathi and A. Tannenbaum, "A graph cut approach to image segmentation in tensor space", *Proc. Workshop on Component Analysis Methods (CVPR),* pp. 18-25, 2007.

[29] Y. T. Weldeselassie and G. Hamarneh, "DT-MRI segmentation using graph cuts", *Proc. SPIE*, pp. 1-9, 2007.

[30] Z. Wang and B. C. Vemuri, "An affine invariant tensor dissimilarity measure and its applications to tensor-valued image segmentation", *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 228-233, 2004.

[31] X. He and P. Niyogi, "Locality Preserving Projections", *Proc. Advances in Neural Information Processing Systems 16(NIPS 2003),* 2003.

[32] J. G. Daugman, "Complete discrete 2-D Gabor transforms by neural networks for imageanalysis and compression", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol.36, pp. 1169-1179, 1988.

[33] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.12, pp. 629-639, 1990.

[34] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.18, pp. 837-842, 1996.

[35] D. Martin, C. Fowlkes, D. Tal and J. Malik, "A Database of Human Segmented Natural Images and Its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics", *Proc. Eighth Int'l Conf. Computer Vision*, pp. 416-423, 2001.

[36] M. N. Do and M. Vetterli, "Wavelet-based texture retrieval using generalized Gaussian densityand Kullback-Leibler distance", *IEEE Transactions on Image Processing*, vol.11, pp. 146-158, 2002.

[37] J. Bigun, G. H. Granlund and J. Wiklund, "Multidimensional orientation estimation with applications to texture analysis and optical flow", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.13, pp. 775-790, 1991.

[38] P. T. Fletcher and S. Joshi, "Principal Geodesic Analysis on Symmetric Spaces: Statistics of Diffusion Tensors", *Proc. ECCV Workshops CVAMIA and MMBIA*, pp. 87-98, 2004.

[39] J. Goldberger, S. Gordon and H. Greenspan, "An efficient image similarity measure based on approximations of KL-divergence between two gaussian mixtures", *Proc. CVPR*, pp. 487-494, 2004.

[40] Z. S. Di, "A note on the gradient of a multi-image", *Computer Vision, Graphics, and Image Processing,* vol.33, pp. 116-125, 1986.

[41] L. G. de, R. Deriche and C. Alberola-L, "Texture and color segmentation based on the combined use of the structure tensor and the image components", *Signal Processing*, vol.88, pp. 776-795, 2008.

[42] G. Gerig, O. Kubler, R. Kikinis and F. A. Jolesz, "Nonlinear anisotropic filtering of MRI data", *IEEE Transactions on Medical Imaging,* vol.11, pp. 221-232, 1992.

[43] T. Brox, M. Rousson, R. Deriche and J. Weickert, "Unsupervised segmentation incorporating colour, texture, and motion", *Proc. 10th International Computer Analysis of Images and Patterns,* pp. 353-360, 2003.

[44] J. Weickert, B. Romeny and M. A. Viergever, "Efficient and reliable schemes for nonlinear diffusion filtering", *IEEE Transactions on Image Processing*, vol.7, pp. 398-410, 1998.

[45] P. Scheunders, "A multivalued image wavelet representation based on multiscale fundamental forms", *IEEE Transactions on Image Processing*, vol.11, pp. 568-575, 2002.

[46] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.14, pp. 710-732, 1992.

[47] S. Mallat, *A Wavelet Tour of Signal Processing, 2nd*, New York: Academic Press, 1999.

[48] L. Grady, T. Schiwietz, S. Aharon and R. Westermann, "Random walks for interactive alpha-matting", *Proc. Visualization Imaging and Image Processing(VIIP)*, pp. 423 – 429, 2005.

[49] G. Wyszecki and W. S. Stiles, *Color science: concepts and methods, quantitative data and formulae*, Wiley, New York, 1982.

[50] L. Zhukov, K. Museth, D. Breen, R. Whitaker and A. Barr, "Level set modeling and segmentation of DT-MRI brain data", *Journal of Electronic Imaging*, vol.12, pp. 125-133, 2003.

[51] M. R. Wiegell, D. S. Tuch, H. B. Larsson and V. J. Wedeen, "Automatic segmentation of thalamic nuclei from diffusion tensor magnetic resonance imaging", *Neuroimage*, vol.19, pp. 391-401, 2003.

[52] C. Lenglet, M. Rousson, R. Deriche and O. Faugeras, "Statistics on the Manifold of Multivariate Normal Distributions: Theory and Application to Diffusion Tensor MRI Processing", *Journal of Mathematical Imaging and Vision*, vol.25, pp. 423-444, 2006.

[53] X. Pennec, P. Fillard, and N. Ayache, "A Riemannian Framework for Tensor Computing", *International Journal of Computer Vision*, vol. 66, pp. 41-66, 2006.

[54] M. A. Ruzon and C. Tomasi, "Alpha estimation in natural images", *Proc. IEEE Conf. Comp. Vision and Pattern Recog.*, pp. 18-25, 2000.

[55] Y. Y. Chuang, B. Curless, D. H. Salesin and R. Szeliski, "A Bayesian approach to digital matting", *Proc. CVPR*, pp. 264-271, 2001.

[56] M. T. Orchard and C. A. Bouman, "Color quantization of images", *IEEE Transactions on Signal Processing*, vol.39, pp. 2677-2690, 1991.