Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Visual Saliency Map from Tensor Analysis

Bing Li
National Laboratory of Pattern Recognition
Institute of Automation
Chinese Academy of Sciences
Beijing, 100190, China
bli@nlpr.ia.ac.cn

Weihua Xiong
Omnivision Corporation
Sunnyvale
California, 95054, USA
wallace.xiong@gmail.com

Weiming Hu
National Laboratory of Pattern Recognition
Institute of Automation
Chinese Academy of Sciences
Beijing, 100190, China
wmhu@nlpr.ia.ac.cn

Abstract

Modeling visual saliency map of an image provides important information for image semantic understanding in many applications. Most existing computational visual saliency models follow a bottom-up framework that generates independent saliency map in each selected visual feature space and combines them in a proper way. Two big challenges to be addressed explicitly in these methods are (1) which features should be extracted for all pixels of the input image and (2) how to dynamically determine importance of the saliency map generated in each feature space. In order to address these problems, we present a novel saliency map computational model based on tensor decomposition and reconstruction. Tensor representation and analysis not only explicitly represent image's color values but also imply two important relationships inherent to color image. One is reflecting spatial correlations between pixels and the other one is representing interplay between color channels. Therefore, saliency map generator based on the proposed model can adaptively find the most suitable features and their combinational coefficients for each pixel. Experiments on a synthetic image set and a real image set show that our method is superior or comparable to other prevailing saliency map models.

Introduction

It is well known that primate visual system employs an attention mechanism that focuses on salient parts based on image itself or relevant visual tasks. Detecting and extracting these salient regions is a fundamental problem in computer vision, because it can help image semantic understanding in many applications, such as adaptive content delivery and region-based image retrieval(Itti, Koch, and Niebur 1998), etc.

Implicit issue in this problem is to compute saliency value for each pixel that represents the departure from its neighboring in terms of some kinds of low-level features. Therefore, two essential questions have to be addressed: (1) finding those features with good discriminating power; and (2) determining each feature's importance in combination (Meur et al. 2006)(Koch and Ullman 1985). Prior researches often consider several low level color or texture features, such as gray intensity, color channel and local shape orientation, separately. They firstly calculate the saliency values of each pixel in these different feature spaces; then combine them following a prefixed fusion model (Itti, Koch, and Niebur 1998)(Gopalakrishnan, Hu, and Rajan 2009)(Liu et al. 2007)(Valenti, Sebe, and Gevers 2009)(Harel, Koch, and Perona 2006) . These predefined features and combination strategies may obtain good performances for some images or certain parts of an image; but cannot always be useful for all images or pixels in some complex situations. In these cases, local-feature selection and adaptive-combination for each pixel can provide significant advantages for saliency map computation. In addition, just as what Hoang et al (Hoang, Geusebroek, and Smeulders 2005) and Shi et al (Shi and Funt 2007) have shown that, for applications on color images, using color and texture features in combination is better than using them separately.

According to the analysis above, we propose a new saliency map model based on tensor analysis. Tensor provides an efficient way to represent color and texture in combination. Its decomposition and reconstruction can not only explicitly represent image's color values into a unit, rather than 3 separate channels, but also imply the spatial interaction within each of the three color channels as well as the interaction between different channels. In the proposed model, the color image is organized as a tensor structure, and the first several bases from tensor decomposition of neighboring blocks of each pixel are viewed as the selected features for its saliency computation. These bases can reveal most significant information inherent in the surrounding environments, the projection of the central block on these bases is viewed as the combination weights of selected features, and the reconstruction residual error after recovering is set as the pixel's saliency value, since it implies whether the pixel includes the similar important features to its neighbors in terms of color and local texture.

Therefore, compared with other existing saliency map computations, our proposed algorithm has two major contributions: (1) The features used for each pixel's saliency computation are adaptively determined by tensor decomposition; (2) The combinational coefficients for all selected features are not predefined, but are gained from tensor reconstruction dynamically. Experiments on both synthetic image set and real-world image set show that our method is superior or

comparable to other prevailing saliency map computations.

Related Work

Visual saliency map analysis can be dated back to the earlier work by Itti et al (Itti, Koch, and Niebur 1998), in which the authors give out a saliency map by applying the "Winner-talk-all" strategy on normalized center-surround difference of three important local features: colors, intensity and orientation. Then the prefixed-linear fusing strategy is used to combine values in these three feature spaces to obtain the final saliency map. Meur et al (Meur et al. 2006) build up a visual attention model based on a so-called coherent psychovisual (psychological-visual) space that combines the globally visual features (intensity, color, orientation, spatial frequencies, etc) of the image. Liu et al (Liu et al. 2007) feed Conditional Random Filed (CRF) technique with a set of multi-scale contrast, center-surround histogram and color spatial-distribution features to detect salient objects. Valenti et al (Valenti, Sebe, and Gevers 2009) combine color edge and curvature information to infer global information so that the salient region can be segmented from background. Hasel et al (Harel, Koch, and Perona 2006) apply graph theory and algorithm into saliency map computation by defining Markov chain over a variety of image maps extracted from different global feature vectors. The region-based visual attention model proposed by Aziz et al (Aziz and Mertsching 2008) combines five saliency maps on color contrast, relative size, symmetry, orientation and eccentricity features through a weighted average to obtain the final saliency map. More lately, Hae et al (Hae and Milanfar 2009) propose a bottom-up saliency detection method based on a local self-resemblance measure. Hou et al (Hou and Zhang 2007) introduce spectral residual and build up salient maps in spatial domain without requiring any prior information of the objects. Achanta et al(Achanta et al. 2009) point out that many existing visual saliency algorithms are essentially frequency bandpass filtering operations. They also propose a frequency-tuned approach (Achanta et al. 2009) in saliency map computation based on color and luminance features. Nearly all the aforementioned methods need to predefine features spaces and fusing strategies.

Tensor and Tensor Decomposition

Before introducing the concept of tensor, we define some notations used in this paper (Kolda 2006). Tensors of order three (cubic) or higher are represented by script letters,$\mathcal{X}$. Matrices (second-order tensors) are denoted by bold capital letters, $\mathbf{A}$. Vectors (first-order tensors) are denoted by bold lowercase letters,$\mathbf{b}$. Scalars (zero-order tensors) are represented by italic letters, i.

Tensor Products

Tensor, a multiple-dimensional array or N-mode matrix, is an element of the tensor product of N vector spaces, each of which has its own coordinate system. A tensor with order of N can be denoted as: $\mathcal{X} \in R^{I_1 \times I_2 \ldots \times I_N}$. There are several kinds of tensor products. A special case is the n-mode product of tensor $\mathcal{X}$ and a matrix $\mathbf{A}$, denoted as $\mathcal{X} \times_n \mathbf{A}$. Let $\mathcal{X}$ be of size $I_1 \times I_2 \ldots \times I_N$ and let $\mathbf{A}$ be of size $J_1 \times J_2$. The n-mode multiplication requires $I_n = J_2$. The result of $\mathcal{X} \times_n \mathbf{A}$ is a tensor with the same order as $\mathcal{X}$, but with the size I_n replaced by J_1. Suppose that $\mathbf{A}$ is of size $J \times I_n$, and $\mathcal{Y} = \mathcal{X} \times_n \mathbf{A}$, thus $\mathcal{Y}$ is of size $I_1 \times I_2 \times \ldots \times I_{n-1} \times J \times \ldots \times I_N$. The elements of $\mathcal{Y}$ are defined as follows:

$$\begin{aligned}(\mathcal{Y})_{i_1 \ldots i_2 j i_{n+1} \ldots i_N} &= (\mathcal{X} \times_n \mathbf{A})_{i_1 \ldots i_2 j i_{n+1} \ldots i_N} \\ &= \sum_{i_n=1}^{T_n} (\mathcal{X})_{i_1 \ldots i_N} \times (\mathbf{A})_{j i_n} \end{aligned} . \quad (1)$$

Given a tensor $\mathcal{X} \in R^{I_1 \times I_2 \ldots \times I_N}$ and the matrix $\mathbf{D} \in R^{J_n \times I_n}$, $\mathbf{E} \in R^{K_n \times I_n}$, and $\mathbf{G} \in R^{J_m \times I_m}$,$m \neq n$. The n-model product has the following properties:

$$(\mathcal{X} \times_n \mathbf{D}) \times_m \mathbf{G} = (\mathcal{X} \times_m \mathbf{G}) \times_n \mathbf{D} = \mathcal{X} \times_n \mathbf{D} \times_m \mathbf{G}. \quad (2)$$

$$(\mathcal{X} \times_n \mathbf{D}) \times_n \mathbf{E} = \mathcal{X} \times_n (\mathbf{E} \bullet \mathbf{D}). \quad (3)$$

Tensor Decomposition

Tensor decompositions are higher-order analogues of Singular Value Decomposition (SVD) of a matrix and have proven to be powerful tools for data analysis (Vasilescu and Terzopoulos 2002)(Savas and Elden 2007). The Higher-Order Singular Value Decomposition (HOSVD) (Kolda and Bader 2009) is a generalized form of the conventional matrix singular value decomposition (SVD). An N-order tensor $\mathcal{X}$ is an N-dimensional matrix composed of N vector spaces. HOSVD seeks for N orthogonal matrices $\mathbf{U}_1, \mathbf{U}_2, ..., \mathbf{U}_N$ which span these N spaces, respectively. Consequently, the tensor $\mathcal{X}$ can be decomposed as the following form:

$$\mathcal{X} = \mathcal{Z} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \ldots \times_N \mathbf{U}_N, \quad (4)$$

where $\mathcal{Z} = \mathcal{X} \times_1 \mathbf{U}_1^T \times_2 \mathbf{U}_2^T \ldots \times_N \mathbf{U}_N^T$, which denotes the core tensor controlling the interaction among the mode matrices $\mathbf{U}_1, \mathbf{U}_2, ..., \mathbf{U}_N$. Two popular solutions used in tensor decomposition are CANDECOMP/PARAFAC (Kolda and Bader 2009) and Tucker decompositions model (Kolda and Bader 2009).

Visual Saliency Map from Tensor Analysis

Overview of Proposed Method

In the proposed model, image is represented by tensors. We divide the image into blocks with $w \times w$ pixels and use 3-order tensor to represent color values in RGB channels of each block, as $\mathcal{B} \in R^{w \times w \times c}$, where w is the row and column size of each block, and c is the dimension of the color space. Since we always use RGB space in this paper, so $c = 3$. For any pixel with its location $\mathbf{p}$, the block centered on it is called 'Center Block' (CB) and the overlapped and directly adjacent blocks are named as 'Neighbor Blocks'(NB). An example is shown in Figure 1.

Here each block shown in Figure 1, CB or NB, is a 3-order tensor, and all neighbor blocks can be assembled into higher-order tensor. The basic idea to find the saliency value of pixel at location $\mathbf{p}$ is as follows: Decomposition of 4-order tensor packaged from 16 neighbor blocks (NBs) can be used to obtain most representative features embedded in

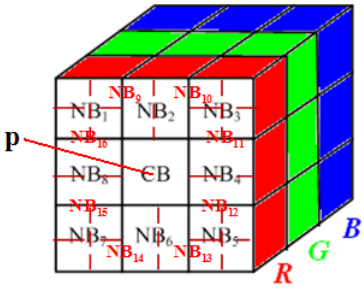

Figure 1: The center block (CB) of pixel $\mathbf{p}$ has 16 overlapped neighbor blocks with $w/2$ overlapping pixels: $NB_1, NB_2, ..., NB_{16}$. The size of each block is $w \times w$.

the surroundings. Then we project the central block (CB) on these bases and reconstruct the central block using these bases. The reconstruction residual error, which can indicate the difference between the center block and its neighbors in terms of color and texture,is set as its saliency output.

Saliency Map from Tensor Reconstruction

In this section, we detail the algorithm for calculating visual saliency value of each pixel from an image. The first stage is to extract the pixel's neighboring blocks and use a 4-order tensor $\mathcal{M} \in R^{b \times w \times w \times c}$ to represent their color and texture pattern, where $b = 16$ is the number of neighboring blocks here.

The second stage is to apply higher-order Tucker decomposition(Kolda and Bader 2009)(Kolda 2006) on the 4-order tensor and decompose it into different subspaces, as

$$\mathcal{M} = \mathcal{Z} \times_1 \mathbf{U}_{block} \times_2 \mathbf{U}_{row} \times_3 \mathbf{U}_{column} \times_4 \mathbf{U}_{color}, \quad (5)$$

where the core tensor $\mathcal{Z}$ reflects the interactions among 4 subspaces: $\mathbf{U}_{block}$ spans the subspace of block parameter, $\mathbf{U}_{row}$ spans the subspace of each block row's parameter and includes correlation between any two rows along all blocks, so each eigenvector represents different texture basis along y direction. Similarly, $\mathbf{U}_{column}$ spans the subspace of each block column's parameter and includes correlation between any two columns along all blocks, so each eigenvector represents different texture basis along x direction. $\mathbf{U}_{color}$ spans the subspace of color parameter and each eigenvector represents one kind of linear transformation of R,G,B color values.

Since $\mathbf{U}_{block}$ only represents the discrimination among all neighboring blocks, the decomposition output along this order will not be taken into account in the following analysis. So we keep its dimension to be 16×16. For the remaining three orders, we take first dr eigenvectors of $\mathbf{U}_{row}$ and $\mathbf{U}_{column}$ (respectively denoted as $\mathbf{U}^{dr}_{row}$ and $\mathbf{U}^{dr}_{column}$) that contain most important texture energy along y or x direction separately. We also take first dc most important linear transformations of the $\mathbf{U}_{color}$ eigenvectors (denoted as $\mathbf{U}^{dc}_{color}$) to emphasize color feature variations. Consequently, the dimension of tensor $\mathcal{M}$ is actually reduced to $b \times dr \times dr \times dc$. An example of this tensor decomposition is given in Figure 2.

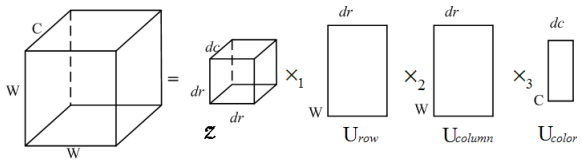

Figure 2: An example envision of 4-order Tucker decomposition viewed from 1st order: 'Block'.

The next step is to represent the center block at location $\mathbf{p}$ as a 3-order tensor as $\mathcal{T} \in R^{w \times w \times c}$, then project it onto $\mathbf{U}^{dr}_{row}$, $\mathbf{U}^{dr}_{column}$ and $\mathbf{U}^{dc}_{color}$, the coefficient is represented as a 3-order tensor $\mathcal{Q} \in R^{dr \times dr \times dc}$; the reconstructed tensor $\mathcal{T}^R$ can be calculated as:

$$\begin{aligned}
\mathcal{T}^R &= \mathcal{Q} \times_1 \mathbf{U}^{dr}_{row} \times_2 \mathbf{U}^{dr}_{column} \times_3 \mathbf{U}^{dc}_{color} \\
\mathcal{T}^R &= \mathcal{T} \times_1 \left(\mathbf{U}^{dr}_{row}\right)^T \times_2 \left(\mathbf{U}^{dr}_{column}\right)^T \times_3 \left(\mathbf{U}^{dc}_{color}\right)^T \\
&\quad \times_1 \mathbf{U}^{dr}_{row} \times_2 \mathbf{U}^{dr}_{column} \times_3 \mathbf{U}^{dc}_{color} \\
\mathcal{T}^R &= \mathcal{T} \times_1 \left(\mathbf{U}^{dr}_{row} \left(\mathbf{U}^{dr}_{row}\right)^T\right) \times_2 \left(\mathbf{U}^{dr}_{column} \left(\mathbf{U}^{dr}_{column}\right)^T\right) \\
&\quad \times_3 \left(\mathbf{U}^{dc}_{color} \left(\mathbf{U}^{dc}_{color}\right)^T\right).
\end{aligned} \quad (6)$$

The final step is to calculate the reconstruction residual error $E(\mathbf{p})$ at pixel $\mathbf{p}$ as:

$$E(\mathbf{p}) = \sqrt{\sum_{i=1}^{w} \sum_{j=1}^{w} \sum_{k=1}^{3} \left(\mathcal{T}_{i,j,k} - \mathcal{T}^R_{i,j,k}\right)^2}. \quad (7)$$

The result $E(\mathbf{p})$ is used to be the saliency value of the processed pixel.

In this way, we approximate center block's color and texture pattern by the reconstruction using the learned patterns of neighbors. Obviously, if the central block has similar features with its neighbors in terms of color and local textures, the principal tensor components gained from neighbor blocks can represent major variance of center block so that the reconstruction error will be small, otherwise the reconstruction error will be higher and the pixel will have larger saliency value.

An example is shown in Figure 3. The center block has a different texture although its color is unchanged. When we process the center pixel inside the part and calculate its saliency value, we will firstly extract 16 neighboring blocks and get the 3 eigenvectors along rows, 3 eigenvectors along columns and 1 eigenvector along color dimension. All of these 7 eigenvectors are expressed as 3-order tensor and viewed as the selected features for the center pixel. Next, we project the central block's 3-order tensor and calculate the corresponding coefficients. The final one is to get the reconstruction value by back-projection. Now we can find that the recovery (Figure 3(C)) is far away from original features because texture bases derived from its neighboring are distinct from that inherent in center block, so the difference between them inevitably reflects a large saliency value. This example only shows the potential of our method; rigorous tests are presented in the following sections.

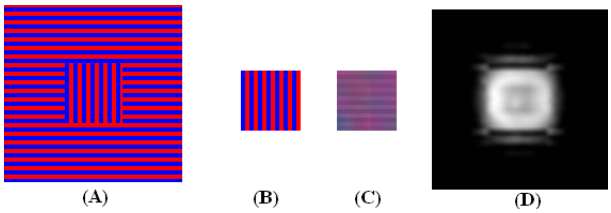

Figure 3: (A) Input Image (B) center block (C) the reconstruction result (D) saliency map output.

Pyramid Saliency Map Calculation

The pyramid architecture offers a framework for image saliency map calculation with increased solution quality. The image pyramid is a multiresolution representation of an image constructed by successive filtering and sub-sampling. It allows scale selection appropriate resolution for the task at hand.

In this paper, we use a pyramid with L different levels, denoted as $I_1, I_1, ..., I_L$, for the saliency map calculation ; where I_1 is the original image and I_L is the lowest resolution image. The pyramid level will be doubled at each step. The value of L is determined to be sure that the image's width and height of I_L cannot be less than 64 pixels. The normalized saliency map at each level is resized to the one with same size of original image. And the values of all saliency maps at different levels are averaged to gain the final saliency map, as:

$$SM(\mathbf{p}) = \frac{1}{L}\sum_{l=1}^{L}\hat{E}_l(\mathbf{p}), \tag{8}$$

where is $SM(\mathbf{p})$ the final saliency value of pixel $\mathbf{p}$; $\hat{E}_l(\mathbf{p})$ is the normalized saliency value of pixel $\mathbf{p}$ at the l^{th} level image.

Experiments

We implement the proposed saliency map computation model in MATLAB 7.7 and compare it with other five prevailing algorithms, Itti's method (ITTI) (Itti, Koch, and Niebur 1998), Hou's method (HOU)[1] (Hou and Zhang 2007), Hae's method (HAE)[2] (Hae and Milanfar 2009), Graph-based visual saliency algorithm (GBVS)[3] (Harel, Koch, and Perona 2006) and Frequency-tuned Salient Region Detection algorithm (FS)[4] (Achanta et al. 2009), on both synthetic and real image data sets. The tensor tucker decomposition code used in the paper can be downloaded from (http://csmr.ca.sandia.gov/ tgkolda/TensorToolbox/).

Data Set and Error Measures

Firstly, we focus on saliency map computation from the perspective of texture analysis using a synthetic image set (referred to as Synthetic set). This dataset contains 100 synthetically or naturally textual images with manually salient patch. In order to construct this image set, we collect 100 images with different types of textures. For each image, we randomly extract out a small patch with size of nearly 40×40. Then we change the texture orientation or texture grain size in the image through rotation or zooming operation. The final stage is to paste the patch back in the original image at a random position. Now the patch is marked as ground truth region of saliency map. Some examples of synthetic images are shown in Figure 4(A)(B). The challenge of synthetic images is that all salient regions are caused only by texture change without color change.

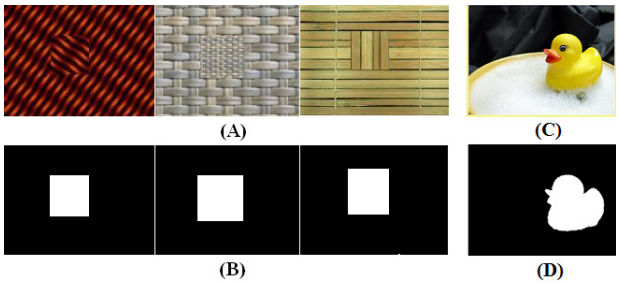

Figure 4: Example images and corresponding binary bounding box-based ground truth: (A)(B)in Synthetic set, (C)(D) in MS set.

The second one is from Microsoft Visual Salient image set (referred to as MS set) (Liu et al. 2007) that contains 5000 high quality images. Each image in MS set is labeled by 9 users requested to draw a bounding box around the most salient object (according to their understanding of saliency). For each image, all users' annotations are averaged to create a saliency map at location $\mathbf{p}$, $S = \{S(\mathbf{p})|S(\mathbf{p}) \in [0,1]\}$ as follows:

$$S(\mathbf{p}) = \frac{1}{M}\sum_{m=1}^{M} a_{\mathbf{p}}^{m}, \tag{9}$$

where M is the number of users and $a_{\mathbf{p}}^{m}$ are the pixels annotated by user m. However, Achanta et al (Achanta et al. 2009) point out that the bounding box-based ground truth is not accurate. They pick out 1000 images from the original MS set (referred to as 1000 MS subset) and create an object-contour based ground truth, the corresponding binary saliency maps are also given out. An example is shown in Figure 4(C)(D).

Given a ground truth saliency map $S(\mathbf{p})$ and the estimated saliency map $SM(\mathbf{p})$ of an image, the Precision (Pre), Recall (Rec), and F measure, which are formulated in Equation (10), are used to evaluate the performance of each algorithm. The same as previous work (Liu et al. 2007)(Valenti, Sebe, and Gevers 2009), α is set to be 0.5.

$$Pre = \frac{\sum_{\mathbf{p}} S(\mathbf{p})SM(\mathbf{p})}{\sum_{\mathbf{p}} SM(\mathbf{p})}, Rec = \frac{\sum_{\mathbf{p}} S(\mathbf{p})SM(\mathbf{p})}{\sum_{\mathbf{p}} S(\mathbf{p})}, \quad F_\alpha = \frac{(1+\alpha)\times Pre \times Rec}{(\alpha \times Pre + Rec)}, \tag{10}$$

Parameter Selection

The performance of tensor analysis based saliency map computation depends on the number of eigenvectors along

[1] http://www.its.caltech.edu/∼xhou/
[2] http://users.soe.ucsc.edu/∼rokaf/SaliencyDetection.html
[3] http://www.klab.caltech.edu/∼harel/share/gbvs.php
[4] http://ivrg.epfl.ch/supplementary_material/RK_CVPR09/index.html

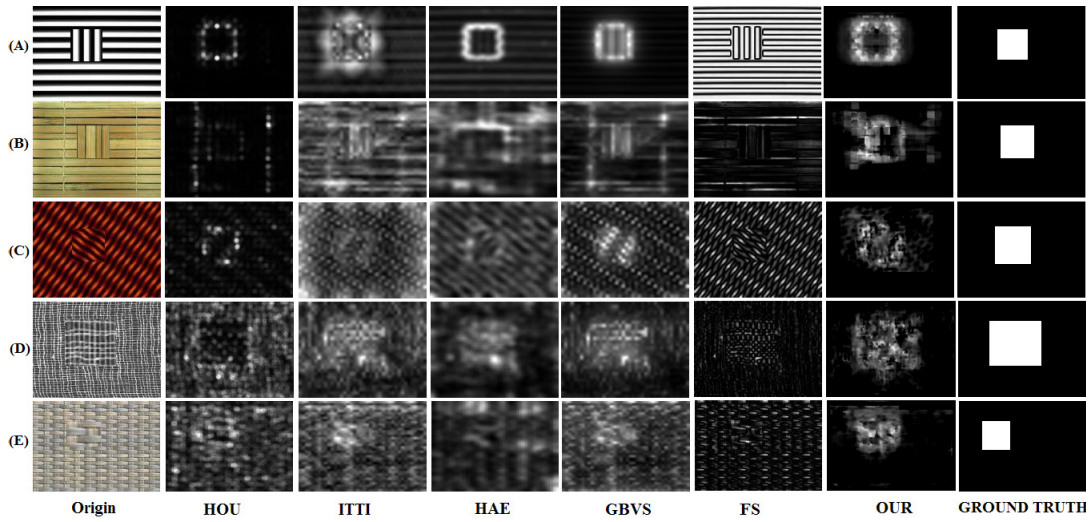

Figure 5: Saliency map examples from synthetic image set.

each order. Here we define the block size as $w = 7$, and each block has 16 overlapped neighbor blocks. We let dr be chosen from $\{1, 3, 5, 7\}$, and dc to be 1, 2 or 3. To ensure that the data set used for parameter selection and performance evaluation are truly independent, we use 4000 images from MS data set that has no intersection with 1000 MS subset to find the optimal basis number settings. Every possible values of dr and dc from candidate settings. For each candidate setting, we compute a saliency map for each image and the F measure is used to represent the performance for that candidate setting. The setting leading to the best performance is then chosen as final parameter setting. All of the following experiments are done based on the chosen parameter setting. We find experimentally that the best choice is $dr = 3$ and $dc = 1$, meaning that the method relies on the first three basis of texture characteristics, $\mathbf{U}^3_{row}$ and $\mathbf{U}^3_{column}$, and one special linear combination of color, $\mathbf{U}^1_{color}$.

Experiments on Synthetic Texture Data set

In this experiment, we work on the synthetic texture data set. We firstly compare our saliency computation method with others using original saliency maps without any further processing. For each saliency map generated by different algorithms, we normalize its values to be between [0, 1], represented as $SM(\mathbf{p})$, by min-max linear normalization method. The precision (Pre), recall (Rec) and F measure values from each method are calculated and compared in Figure 6(A). The results show that the proposed algorithm outperforms all other algorithms in this set.

We then compare all of these algorithms' outputs based on binary saliency map. For a given saliency map with saliency values in the range [0, 1], the simplest way to obtain a binary mask for the salient object is to threshold the saliency map at a threshold T within [0, 1]. The saliency value will be set as 1 if $SM(\mathbf{p}) \geq T$, will otherwise be set as 0. We follow a favorite method to decide the value of T adaptively (Hou and Zhang 2007)(Achanta et al. 2009):

$$T = \frac{2}{W \times H} \sum_{\mathbf{p}} SM(\mathbf{p}), \tag{11}$$

where W, H are the width and height of the image, respectively. The value of T actually is two times the mean saliency of the image. The precision (Pre), recall(Rec)and F measure values are evaluated in Figure 6(B). Moreover, a few saliency maps from different algorithms are given out in Figure 5.

All the results in Figure 5 and 6 show that our tensor based algorithm outperforms all other algorithms in textural salient region detection. It proves that the tensor decomposition can find rich textural information implicitly for detection task in despite of no obvious textural feature extraction. The FS algorithm has lowest performance due to the fact that it nearly takes no textual information into account. The results in Figure 5 show that other methods have some difficulties in getting correct saliency maps for these images, but our algorithm obtains good results. Especially for the salient region caused by textural grain change(Figure 5(E)), nearly no other algorithm can produce correct results, but saliency maps generated by our algorithm are very satisfying.

Experiments on 1000 MS Subset

The same as the previous experiment, we initially evaluate all algorithms' performances through comparing each method's output with original saliency map. The comparison results are shown in Figure 7(A). It tells us that the proposed algorithm is better than HOU, ITTI, HAE as well as GBVS and comparable to FS method. Its precision is 42.6%, recall is 38.1% and F measure is 41.0% respectively. Although HOU method has higher precision value, it has lower recall value. By comparison, our algorithm has both high precision and high recall. This result indicates that our algorithm not only promotes the salient region, but also restrains those unsalient regions.

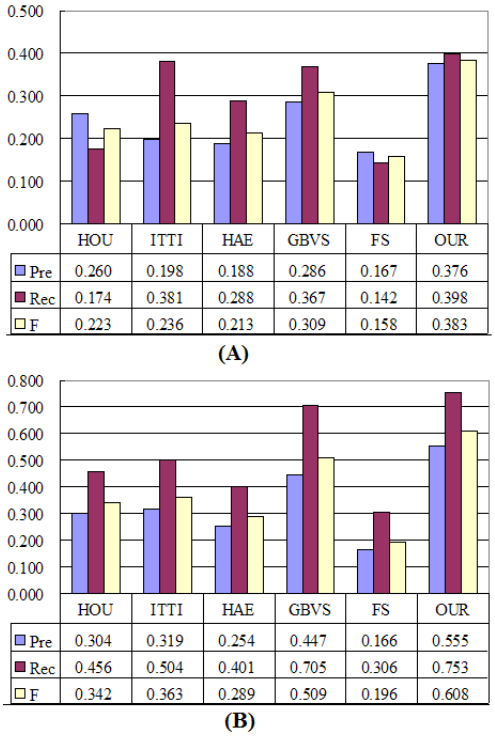

Figure 6: Comparison with existing visual saliency algorithms on synthetic set: (A)in terms of original saliency map (B)in terms of binary saliency map

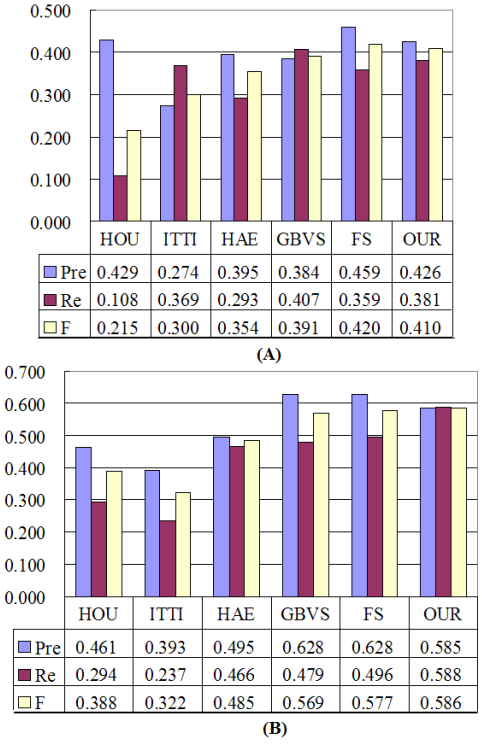

Figure 7: Comparison with existing visual saliency algorithms on 1000 MS subset: (A)in terms of original saliency map (B)in terms of binary saliency map

We also create the binary saliency maps and compare them with ground-truth. From the results in Figure 7(B), we can find that our method is comparable to other prevailing solutions. In order to intuitively compare saliency maps generated by different methods, we also give out some saliency map examples in Figure 8. They tell us that HOU method pays more attention on edges and fails to extract salient object's inner region. The FS algorithm is based on the difference between an image and its average image. It inevitably fails if salient object occupies major part of image with same color (Figure 8(C)) or salient object has similar color with background (the white cloth in Figure 8(C)). Obviously, although ITTI, HAE and GBVS can obtain good saliency maps on Figure 8(A), they give top background part of the image high saliency values incorrectly. By comparison, our method avoids this issue and assigns high saliency values only to those pixels in the salient region. Generally, the saliency maps from our algorithm, in contrast, can get high saliency values on both object's edges and inner regions.

Finally, the saliency map is also employed in salient object segmentation and extraction. The segmentation scheme used in this paper follows the one used in (Valenti, Sebe, and Gevers 2009). It firstly uses mean-shift algorithm to divide original images into many regions.Then an adaptive threshold T that is as two times the mean saliency (Equation 11), is also used to detect proto-objects. The Regions with average saliency values greater than T are viewed as salient, and their values in binary saliency object image are set as '1', while the other parts are set as '0'. The results in Figure 8 show that the extracted salient objects based on our saliency maps are pleasing. In particular, although it is very difficult to pick out the entire salient objects from Figure 8 (C) and (D), our algorithm can produce satisfied results.

Conclusion

Most existing computational visual saliency models follow a bottom-up framework that generates independent saliency map in each selected visual feature space and combines them in a predefined way. In this paper, the tensor representation and analysis of color image is introduced for saliency map computation. Compare to the existing bottom-up methods, two major advantages of our proposed algorithm can be obtained: (1) Considering and processing any image's color and local texture as a single entity; and (2) Using tensor decomposition to implicitly find the most important features for each pixel locally rather than explicitly select and define low level features used for all pixels. The power of the proposed method is demonstrated by experimental results in two challenge image sets.

Acknowledgment

This work is partly supported by the National Nature Science Foundation of China (No. 61005030, 60935002 and 60825204) and Chinese National Programs for High

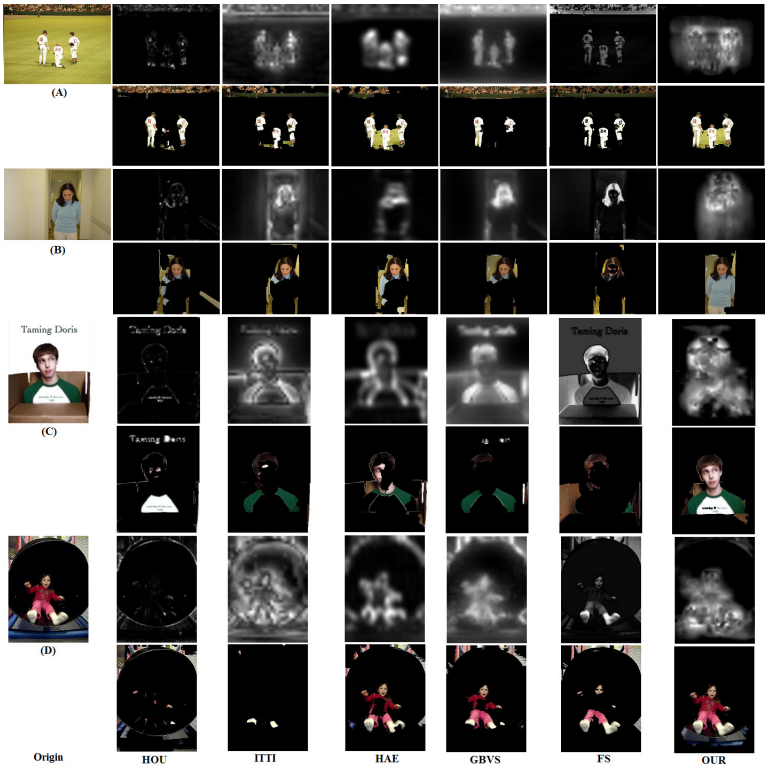

Figure 8: Examples of saliency maps, and extracted salient objects of different algorithms. For each given image, the first row includes saliency maps; the second row shows the extracted salient objects.

Technology Research and Development (863 Program) (No.2012AA012503 and No. 2012AA012504) as well as the Excellent SKL Project of NSFC (No.60723005).

References

Achanta, R.; Hemami, S.; Estrada, F.; and Ssstrunk, S. 2009. Frequency-tuned salient region detection. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 1597–1604.

Aziz, M. Z., and Mertsching, B. 2008. Fast and robust generation of feature maps for region-based visual attention. *IEEE Trans. on Image Processing* 17(5):633–644.

Gopalakrishnan, V.; Hu, Y.; and Rajan, D. 2009. Salient region detection by modeling distributions of color and orientation. *IEEE Trans. on Multimedia* 11(5):892–905.

Hae, J. S., and Milanfar, P. 2009. Static and space-time visual saliency detection by self-resemblance. *The Journal of Vision* 9(12):1–27.

Harel, J.; Koch, C.; and Perona, P. 2006. Graph-based visual saliency. In *Proc. of Annual Conf. on Neural Information Processing Systems*, 545–552.

Hoang, M. A.; Geusebroek, J. M.; and Smeulders, A. W. M. 2005. Color texture measurement and segmentation. *Signal Processing* 85(2):265–275.

Hou, X., and Zhang, L. 2007. Saliency detection: A spectral residual approach. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 1–8.

Itti, L.; Koch, C.; and Niebur, E. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 20(11):1254–1259.

Koch, C., and Ullman, S. 1985. Shifts in selection in visual attention: Toward the underlying neural circuitry. *Human Neurobiology* 4(4):219–227.

Kolda, T. G., and Bader, B. W. 2009. Tensor decompositions and applications. *SIAM Review* 51(3):455–500.

Kolda, T. 2006. Multilinear operators for higher-order decompositions. In *Technical Report*, SAND2006–2081.

Liu, T.; Sun, J.; Zheng, N.; and Tang, X. 2007. Learning to detect a salient object. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 1–8.

Meur, O. L.; Callet, P. L.; Barba, D.; and Thoreau, D. 2006. A coherent computational approach to model bottom-up visual attention. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 28(5):802–817.

Savas, B., and Elden, P. 2007. Handwritten digital classification using higher order singular value decomposition. *Pattern Recognition* 40(3):993–1003.

Shi, L., and Funt, B. 2007. Quaternion color texture segmentation. *Computer Vision and Image Understanding* 107(1):88–96.

Valenti, R.; Sebe, N.; and Gevers, T. 2009. Image saliency by isocentric curvedness and color. In *Proc. of Int. Conf. on Computer Vision*, 2185–2192.

Vasilescu, M. A. O., and Terzopoulos, D. 2002. Multilinear analysis of image ensembles: Tensor faces. In *Proc. of European Conf. on Computer Vision*, 447–460.