Multi-Task Rank Learning for Visual Saliency Estimation

Jia Li, Yonghong Tian, *Senior Member, IEEE,* Tiejun Huang, *Member, IEEE,* and Wen Gao, *Fellow, IEEE*

***Abstract*—Visual saliency plays an important role in various video applications such as video retargeting and intelligent video advertising. However, existing visual saliency estimation approaches often construct a unified model for all scenes, thus leading to poor performance for the scenes with diversified contents. To solve this problem, we propose a multi-task rank learning approach which can be used to infer multiple saliency models that apply to different scene clusters. In our approach, the problem of visual saliency estimation is formulated in a pair-wise rank learning framework, in which the visual features can be effectively integrated to distinguish salient targets from distractors. A multi-task learning algorithm is then presented to infer multiple visual saliency models simultaneously. By an appropriate sharing of information across models, the generalization ability of each model can be greatly improved. Extensive experiments on a public eye-fixation dataset show that our multi-task rank learning approach outperforms 12 state-of-the-art methods remarkably in visual saliency estimation.**

***Index Terms*—Generalization ability, multi-task learning, pair-wise rank learning, visual saliency.**

I. Introduction

IN NATURAL scenes, the complexity of the input visual stimuli usually exceeds the processing capacity of human vision system. As a consequence, the important visual subsets will be selected and processed with higher priorities. In this selective mechanism, visual saliency often plays an essential role in determining which subset (e.g., pixel, block, region, or object) in a scene is important. Therefore, the central task in visual saliency estimation is to rank various visual subsets in a scene to indicate their importance and processing priorities.

In visual saliency estimation, each visual subset in a scene can be represented by a set of visual features. According to the feature integration theory [1], different visual features can be bound into consciously experienced wholes for visual saliency estimation. As such, visual saliency can be estimated by integrating the visual features that are able to effectively distinguish salient targets from distractors. Toward this end, visual saliency estimation should solve two problems: what features can distinguish targets from distractors in a scene and how to optimally integrate these features.

In existing work on visual saliency estimation, these two problems have been tentatively studied. Some stimuli-driven approaches (e.g., [2]–[8]) selected the preattentive visual features and integrated them in an ad-hoc manner. In contrast, some learning-based approaches (e.g., [9]–[11]) adopted the machine learning algorithms to learn the discriminant visual features and feature integration strategies. Generally speaking, these approaches can obtain impressive results in some cases but meanwhile may suffer poor performance in other cases since they often construct a unified model for all scenes. Actually, the features that can best distinguish targets from distractors may vary remarkably in different scenes. In surveillance video, for instance, the motion features can be used to efficiently pop-out a car or a walking person [as shown in Fig. 1(a)–(b)]; while to distinguish a red apple/flower from its surroundings, color contrasts should be used [as shown in Fig. 1(c)–(d)]. In most cases, it is infeasible to pop-out the targets and suppress the distractors by using a fixed set of visual features. Therefore, it is necessary to construct scene-specific models that adaptively adopt different solutions for different scene categories.

Toward this end, we propose a multi-task rank learning approach for visual saliency estimation. In this approach, visual saliency estimation is formulated as a pair-wise rank learning problem. Moreover, our approach constructs multiple visual saliency models, each for a scene cluster, by learning and integrating the features that best distinguish targets from distractors in that cluster. We also propose a multi-task learning algorithm to infer multiple saliency models simultaneously. Different from the traditional single-task learning approach, the multi-task learning approach can carry out multiple training tasks simultaneously with fewer training data per task [12], [13]. In this framework, the appropriate sharing of information across training tasks can be used to effectively improve the performance of each model. Extensive experiments on a public eye-fixation dataset [14] show that our approach outperforms several state-of-the-art bottom-up (e.g., [2]–[8]), top-down (e.g., [9]–[11]), and rank learning (e.g., [15], [16]) approaches in visual saliency estimation.

Manuscript received May 12, 2010; revised July 13, 2010 and September 23, 2010; accepted November 8, 2010. Date of publication March 17, 2011; date of current version May 4, 2011. This work was supported by grants from the Chinese National Natural Science Foundation, under Contracts 61035001, 60973055, and 90820003, by the National Basic Research Program of China, under Contract 2009CB320906, and by the Fok Ying Dong Education Foundation, under Contract 122008. This paper was recommended by Associate Editor F. Lavagetto.

J. Li is with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences (CAS), Beijing 100190, China, and also with the Graduate University of CAS, Beijing 100049, China.

Y. Tian, T. Huang, and W. Gao are with the National Engineering Laboratory for Video Technology, Key Laboratory of Machine Perception (MoE), School of Electronic Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: yhtian@pku.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2011.2129430

Fig. 1. Targets and distractors in different scenes can be best distinguished by different features. (a), (b) "Motion" feature. (c), (d) "Color" feature.

Compared with existing approaches, our main contributions are summarized as follows.

1) We formulate the problem of visual saliency estimation in a pair-wise rank learning framework. In this framework, the model can automatically select the visual features that best distinguish salient targets from distractors.
2) We present an approach to construct multiple visual saliency models that apply to various scene clusters. With these scene-specific models, different features can be optimally selected and integrated to distinguish targets from distractors in different scenes.
3) We propose a multi-task learning approach to infer multiple saliency models simultaneously. By an appropriate sharing of information across models, the generalization ability of each saliency model can be greatly improved.

The rest of this paper is organized as follows. Section II presents a brief review of related work and Section III formulates visual saliency estimation in a rank learning framework. Section IV describes the multi-task rank learning algorithm for visual saliency estimation. Experimental results are shown in Section V and this paper is concluded in Section VI.

II. Related Work

Typically, visual saliency can be influenced by two factors: the bottom-up one that is related to the input stimuli and the top-down one that involves the task, expectation and experience [17]. When watching a scene, the input stimuli will compete in the bottom-up manner to pop-out themselves, while the top-down factor can bias such competition in favor of a specific category of visual stimuli [18]. Accordingly, existing visual saliency models can be also grouped into two categories: the bottom-up one and the top-down one. The bottom-up approaches often estimate visual saliency by directly detecting and integrating the visual irregularities in various preattentive features. In contrast, the top-down approaches aim at learning the optimal irregularity detectors as well as the combination strategies.

As a representative bottom-up approach, Itti *et al.* [2] estimated visual saliency on image by integrating the "center-surround" contrasts in intensity, opponent-colors, and directions. Afterwards, the same framework was extended for video saliency estimation by incorporating the local contrasts in motion and flicker [3]. Similarly, Walther and Koch [19] extended this framework to detect salient proto-objects, which were defined as bottom-up units that could be bound into objects once attend to. Marat *et al.* [20] presented a biology-inspired model by simulating the low-level processes in retinal cells. A retina model was constructed to filter out the salient subsets in several spatial/temporal preattentive features. Particularly, some works focused on detecting irregular spatiotemporal variations for saliency estimation. For example, Itti and Baldi [4] detected "surprise" in video sequence by combining spatial contrast and temporal evolution. In [8], visual saliency was estimated by detecting irregular motions from inter-frame key-points matching. Moreover, Harel *et al.* [7] represented an image (or a video frame) as a weighted graph and adopted a random walker to select the less-visited nodes (pixels) as salient locations. Similarly, such irregularities can also be detected using spectrum analysis. For instance, Hou and Zhang [5] located the irregularities in the amplitude spectrum of intensity using Fourier Transform, while Guo *et al.* [6] detected such irregularities in the phase spectrum of intensity, opponent-colors, and motion using quaternion Fourier transform. Moreover, some approaches (e.g., [21], [22]) segmented images into regions and the irregular regions are selected as salient visual subsets. Other works, such as [23]–[25], incorporated the influences of various semantic clues (e.g., human face, speech and music, camera motion) in visual saliency estimation. In general, the bottom-up approaches can well pop-out the salient targets but may have difficulties in suppressing the distractors.

Since visual neurons exhibit tuning properties that can be optimized to respond to recurring features in the visual input [26], it can be safely assumed that the experience on past similar scenes can assist the estimation of visual saliency. Inspired by this idea, some top-down approaches tried to learn the correlations between various visual attributes and visual saliency. For example, Kienzle *et al.* [9] adopted a SVM classifier to learn the correlations between local visual attributes and saliency values. They also presented an approach to learn a set of irregularity filters to find the interesting locations in video [27]. Similarly, Pang *et al.* [28] presented a stochastic model for visual saliency estimation in video with a dynamic Bayesian network. In this network, a Kalman filter was used to estimate the stochastic saliency and a hidden Markov model was further adopted to predict the probable human-attended regions. Furthermore, Navalpakkam and Itti [10] proposed a learning algorithm to pop-out the targets and suppress the distractors through maximizing the signal-noise-ratio. Peters and Itti [11] estimated the projection matrix between global scene characteristics and eye density maps. On the regional saliency dataset, Liu *et al.* [29] adopted a conditional random field for salient object detection. After that, they extended the approach to detect salient object sequences in video [30]. Compared with the bottom-up models, these top-down approaches can provide impressive results since they can transfer the experience from the viewed scenes to new ones to guide the saliency estimation. However, one drawback of these top-down approaches is that they can only construct unified models which may be not robust to all scenes, particularly for the scenes with diversified contents.

Recently, the rank learning approaches, such as [15] and [16], were widely used in many applications and demonstrated impressive performance. Since the saliency values correspond to the processing priorities of different visual subsets in a scene, visual saliency estimation can be also formulated as a rank learning problem. Typically, a ranking model can assign an integer rank for each visual subset to indicate its processing

priority, which corresponds to visual saliency to some extent. In a tentative exploration [31], we found that the visual features, which best distinguish targets from distractors, can be effectively mined in a rank learning framework. Since such features can vary remarkably in different scenes, visual saliency can be estimated by inferring multiple ranking models that apply to various scene clusters. Furthermore, these models can be simultaneously trained in a multi-task learning framework to improve the generalization abilities and avoid overfitting.

III. Problem Formulation

As aforementioned, the central task in visual saliency estimation is to rank various visual subsets in a scene to indicate their importance and processing priorities. From the perspective of visual search, users tend to search the desired targets under the facilitation of experience derived from past similar scenes (i.e., the contextual cueing effect [26], [32]). Therefore, a saliency model can be represented as a ranking model which ranks all the visual subsets in a scene with respect to their relevance to the searching intention. These ranks, viewed as the priorities in searching (and processing) the desired targets, correspond to visual saliency to some extent. Without loss of generality, a subset denotes a macro-block in the remainder of this paper.

Given a scene S_k, we can represent its visual subsets $\{s_{kn} \in S_k\}_{n=1}^{N}$ with the local visual attributes $\{\mathbf{x}_{kn} \in \chi\}_{n=1}^{N}$. Thus, the goal of visual saliency estimation can be described as identifying the ranks (searching/processing properties) of $\{s_{kn}\}_{n=1}^{N}$ with respect to $\{\mathbf{x}_{kn}\}_{n=1}^{N}$. Toward this end, we infer a ranking function $\phi : \chi \rightarrow \mathbb{R}$ from past similar scenes (and user feedbacks) to assign real scores to $\{s_{kn}\}_{n=1}^{N}$. After that, these real scores can be used to rank the visual subsets. That is, $\phi(\mathbf{x}_{ku}) > \phi(\mathbf{x}_{kv})$ indicates that s_{ku} ranks higher than s_{kv} and maintains a higher saliency. Note that here the actual numerical value of $\phi(\mathbf{x})$ is immaterial and only the ordering is meaningful.

However, it is often difficult to obtain a unified ranking function that is robust to all scenes, particularly for those scenes with diversified contents. Therefore, we assume that scenes can be grouped into clusters $\{\mathbb{S}_m\}_{m=1}^{M}$ and different clusters correspond to different ranking functions $\{\phi_m\}_{m=1}^{M}$ (as shown in Fig. 2). Therefore, each ranking function ϕ_m should be optimized on cluster $\mathbb{S}_m$ to approximate the ground-truth (integer) ranks $\{y_{kn}\}_{n=1}^{N}$ with the estimated (integer) ranks $\{\pi_m(\mathbf{x}_{kn})\}_{n=1}^{N}$. Note that here y_{kn} can be obtained by ranking the ground-truth saliency values $\{g_{kn}\}_{n=1}^{N}$, while $\pi_m(\mathbf{x}_{kn})$ can be obtained by ranking the real scores $\{\phi_m(\mathbf{x}_{kn})\}_{n=1}^{N}$. For any two subsets, s_{ku} is more salient than s_{kv} if $g_{ku} > g_{kv}$.

Since the visual world is highly structured and such regularities can be consulted to guide visual processing [26], we can group K training scenes according to their global scene characteristics $\{\mathbf{v}_k\}_{k=1}^{K}$. Thus, the problem of visual saliency estimation can be formulated as inferring ranking functions $\Phi = \{\phi_m\}$ and scene-cluster labels $\alpha = \{\alpha_{km}\}$ from local visual attributes $\{\mathbf{x}_{kn}\}$, global scene characteristics $\{\mathbf{v}_k\}$ and ground-truth saliency values $\{g_{kn}\}$. Note that here $\alpha_{km} \in \{0, 1\}$ while $\alpha_{km} = 1$ indicates $S_k \in \mathbb{S}_m$.

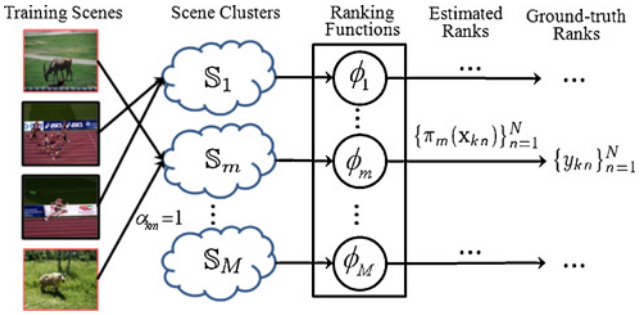

Fig. 2. Framework of our approach. In our approach, scenes with similar contents are grouped into the same cluster. For each cluster, a ranking function is optimized to give ranks for all subsets in a scene, while these estimated ranks are expected to approximate the ground-truth ranks.

IV. Multi-Task Rank Learning for Visual Saliency Estimation

In this section, we will describe the details of our multi-task rank learning approach for visual saliency estimation. First, we will present how to extract the local visual attributes and global scene characteristics. After that, we propose the multi-task rank learning approach, followed by the learning algorithm for optimizing visual saliency models and the computational complexity analysis.

A. Feature Extraction

First, we will introduce how to calculate the local visual attributes $\mathbf{x}_{kn}$ and the global scene characteristics $\mathbf{v}_k$. Often, the contrast-based visual irregularities in preattentive features can well recover the salient locations in a scene. By using the algorithm proposed in [3], we compute "center-surround" local contrasts for each visual subset in a scene. Typically, such local contrasts are robust to noise and quality degradation such as brightness changing, quality compression and blurring. In the computation, the local contrasts are generated from 12 preattentive visual channels in 6 scales, including intensity (6), red/green and blue/yellow opponencies (12), four orientations (24), temporal flickers (6) and motion energies in four directions (24). In total $L = 72$ local contrast features are obtained to form the local visual attributes $\mathbf{x}_{kn}$.

As proposed by [26], some properties of the visual environment, such as rough spatial layout information and predicable variations, do not change radically over time and can be encoded to guide the visual processing. Since these properties can work as the contextual priors for individuals to search and process the targets in similar environments, we can use them to form a global descriptor to characterize a scene. Typically, such global features can be obtained by summarizing the local visual attributes of a scene without encoding specific objects or regions [32]. Therefore, we can calculate the mean and standard deviation of the lth dimension of $\mathbf{x}_{kn}$ as follows:

$$\mu_{kl} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_{kn}(l) \quad \sigma_{kl} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} (\mathbf{x}_{kn}(l) - \mu_{kl})^2}. \tag{1}$$

The mean and standard deviation indicate whether visual subsets in the scene S_k are discriminative from each other in

the lth feature. After the calculation, $\{\mu_{kl}\}_{l=1}^{L}$ and $\{\sigma_{kl}\}_{l=1}^{L}$ can be used to form the global scene characteristics $\mathbf{v}_k$ (with $2L$ components). Since different visual features may span different ranges of values, we normalize each dimension of $\mathbf{x}_{kn}$ and $\mathbf{v}_k$ into [0, 1].

B. Multi-Task Rank Learning Approach

Given the local and global features, we can group K training scenes into M clusters and infer M ranking functions in a multi-task learning framework. Without loss of generality, we define $\phi_m(\mathbf{x}) = \omega_m^{\mathbf{T}}\mathbf{x}$ since various preattentive visual features are often integrated into experienced wholes with linear weights for saliency estimation (e.g., [2]–[4], [10]). Note that here ω_m is a column vector with L components. For the sake of simplicity, we let $\mathbf{W}$ be a $L \times M$ matrix with the mth column equals to ω_m. Therefore, the optimization objective can be defined as follows:

$$\begin{aligned} &\min_{\mathbf{W},\alpha} \mathcal{L}(\mathbf{W}, \alpha) + \Omega(\mathbf{W}, \alpha) \\ &\text{s.t.} \sum_{m=1}^{M} \alpha_{km} = 1 \quad \forall k \text{ and } \alpha_{km} \in \{0, 1\} \quad \forall m \end{aligned} \tag{2}$$

where $\mathcal{L}(\mathbf{W}, \alpha)$ is the empirical loss and $\Omega(\mathbf{W}, \alpha)$ is the penalty term that encodes the prior knowledge on the parameters. To focus on the features that can distinguish targets from distractors, we define the empirical loss $\mathcal{L}(\mathbf{W}, \alpha)$ in a pairwise manner

$$\begin{aligned} &\mathcal{L}(\mathbf{W}, \alpha) = \\ &\sum_{k=1}^{K} \sum_{m=1}^{M} \alpha_{km} \sum_{u \neq v}^{N} [g_{ku} < g_{kv}]_{\mathbf{I}} [\omega_m^{\mathbf{T}} \mathbf{x}_{ku} \geq \omega_m^{\mathbf{T}} \mathbf{x}_{kv}]_{\mathbf{I}} \end{aligned} \tag{3}$$

where $[x]_{\mathbf{I}} = 1$ if x holds, otherwise $[x]_{\mathbf{I}} = 0$. We can see that the empirical loss equals to the number of falsely ranked subset pairs on all training scenes.

Beyond the empirical loss, the prior knowledge on grouping scenes and training ranking functions should also be considered in the optimization process. That is, the optimization objective should comprise the penalty terms that encode the priors on scene clustering, model correlation and model complexity. These three penalty terms can be defined as follows.

1) *Scene Clustering:* To group scenes with similar contents into the same cluster, we set the penalty term Ω_s as follows:

$$\Omega_s = \frac{1}{K} \sum_{k_0 \neq k_1}^{K} \sum_{m=1}^{M} (\alpha_{k_0 m} - \alpha_{k_1 m})^2 \cos(\mathbf{v}_{k_0}, \mathbf{v}_{k_1}) \tag{4}$$

where $\cos\left(\mathbf{v}_{k_0}, \mathbf{v}_{k_1}\right)$ denotes the similarity of the k_0th and the k_1th scenes, which is computed as the cosine distance between two vectors $0 \preceq \mathbf{v}_{k_0}, \mathbf{v}_{k_1} \preceq 1$. We can see that the penalty term will be large when two similar scenes S_{k_0} and S_{k_1} are grouped into different clusters.

2) *Model Correlation:* In the training process, each scene cluster may only contain limited number of scenes with similar contents. Thus the saliency models directly trained on these clusters may lack the generalization ability [as shown in Fig. 3(a), this corresponds to the typical single task learning]. To solve this problem, taking an appropriate sharing of information across training tasks can avoid overfitting and improve the performance of each model [12], [33]. Therefore, we set a penalty term Ω_d to incorporate the correlations between models

$$\begin{aligned} \Omega_d = &\frac{1}{M} \sum_{i \neq j}^{M} \sum_{k=1}^{K} \sum_{u \neq v}^{N} [g_{ku} < g_{kv}]_{\mathbf{I}} \times \\ &[\omega_i^{\mathbf{T}} \mathbf{x}_{ku} \geq \omega_i^{\mathbf{T}} \mathbf{x}_{kv}]_{\mathbf{I}} [\omega_j^{\mathbf{T}} \mathbf{x}_{ku} \geq \omega_j^{\mathbf{T}} \mathbf{x}_{kv}]_{\mathbf{I}}. \end{aligned} \tag{5}$$

The influence of this penalty is two-fold. First, a sample mistakenly predicted by most models will be emphasized in training ϕ_i (i.e., a large $\sum_{j \neq i} [g_{ku} < g_{kv}]_{\mathbf{I}} [\omega_j^{\mathbf{T}} \mathbf{x}_{ku} \geq \omega_j^{\mathbf{T}} \mathbf{x}_{kv}]_{\mathbf{I}}$). This ensures the diversity of training samples for ϕ_i, leading to improved generalization ability. Second, a sample successfully predicted by most models will be ignored in training ϕ_i (i.e., a small $\sum_{j \neq i} [g_{ku} < g_{kv}]_{\mathbf{I}} [\omega_j^{\mathbf{T}} \mathbf{x}_{ku} \geq \omega_j^{\mathbf{T}} \mathbf{x}_{kv}]_{\mathbf{I}}$). This guarantees the diversity of different models. With this penalty term, each model is actually related to all the training samples with different weights [as shown in Fig. 3(b)], leading to improved performance.

3) *Model Complexity:* To avoid optimizing complex models, we have to set a penalty term Ω_c as follows:

$$\Omega_c = \sum_{m=1}^{M} \omega_m^{\mathbf{T}} \omega_m. \tag{6}$$

Here, the penalty term on model complexity can be used to constrain the number of scene clusters. Thus over-complex models can be avoided.

With these penalty terms, the overall penalty $\Omega(\mathbf{W}, \alpha)$ can be written as the weighted linear combination of them

$$\Omega(\mathbf{W}, \alpha) = \epsilon_s \Omega_s + \epsilon_d \Omega_d + \epsilon_c \Omega_c \tag{7}$$

where ϵ_s, ϵ_d, and ϵ_c are three non-negative weights to combine these penalty terms. In this paper, these weights are empirically selected using the validation set.

C. The Learning Algorithm

By incorporating the empirical loss in (3) and penalty term in (7) into the optimization objective (2), we will encounter a non-convex optimization problem. Therefore, we use the EM algorithm [34] to iteratively solve the problem and ensure the convergence. First, scenes are simply grouped into M clusters according to their global scene characteristics using the K-means algorithm (M can be empirically selected through cross validation). After that, the scene-cluster labels α are initialized and the parameter matrix $\mathbf{W}$ can be initialized by minimizing (2). Here, we set $\epsilon_s = \epsilon_d = \epsilon_c = 0$ to only minimize the empirical loss without considering the inter-scene and

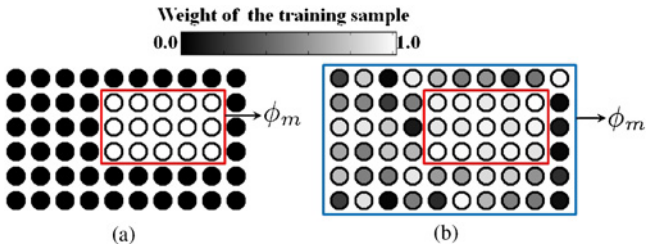

Fig. 3. Advantage of the multi-task learning approach. Each circle corresponds to a training sample and its intensity indicates its weight when training the ranking function ϕ_m. (a) In single task learning, each model is trained independently and the model correlations are not considered. In this case, ϕ_m is trained on limited samples (i.e., samples in the red box) and may lack the generalization ability. (b) In multi-task learning, an appropriate sharing of information across models is adopted by incorporating the penalty term Ω_d. In this case, ϕ_m is actually trained on the whole dataset (i.e., the samples in the blue box) by emphasizing different subsets of samples (i.e., the samples in the red box). Therefore, the generalization ability of ϕ_m can be improved.

inter-model correlations (this is also the baseline used in the experiments). After the initialization, we can iteratively update α and optimize $\mathbf{W}$ using the EM-algorithm. As in [15], [35], and [36], we replace the Boolean terms related to $\mathbf{W}$ in (2) with their convex upper bounds to facilitate the optimization

$$\begin{aligned}[\omega_m^{\mathbf{T}}\mathbf{x}_{ku} \geq \omega_m^{\mathbf{T}}\mathbf{x}_{kv}]_{\mathbf{I}} &\leq \exp\left(\omega_m^{\mathbf{T}}\mathbf{x}_{ku} - \omega_m^{\mathbf{T}}\mathbf{x}_{kv}\right) \\ &= \exp\left(trace\left(\mathbf{W}^{\mathbf{T}}\mathbf{X}_{kuv}^m\right)\right)\end{aligned} \tag{8}$$

where $\mathbf{X}_{kuv}^m$ is a $L \times M$ matrix with its mth column equals to $\mathbf{x}_{ku} - \mathbf{x}_{kv}$ and the other components equal to zero. Here we adopt the exponential upper bound since it is convex (although loose) and can facilitate the optimization. For the sake of simplicity, we let

$$\eta_{kuv}^m = \exp\left(trace\left(\mathbf{W}^{\mathbf{T}}\mathbf{X}_{kuv}^m\right)\right). \tag{9}$$

After the replacement, we can iteratively update α and optimize $\mathbf{W}$ as follows.

Step 1: For $k = 1, \ldots, K$, update $\alpha_k = [\alpha_{k1}, \ldots, \alpha_{kM}]$ by solving the problem which contains only the terms in (2) that are related to α_k

$$\begin{aligned}\min_{\alpha_k} &\sum_{m=1}^{M}\alpha_{km}\sum_{u\neq v}^{N}[g_{ku} < g_{kv}]_{\mathbf{I}}\eta_{kuv}^m \\ &+\frac{2\epsilon_s}{K}\sum_{k_0\neq k}^{K}\sum_{m=1}^{M}\left(\alpha_{km} - \alpha_{k_0 m}\right)^2\cos\left(\mathbf{v}_k, \mathbf{v}_{k_0}\right) \\ \text{s.t.} &\sum_{m=1}^{M}\alpha_{km} = 1 \text{ and } \alpha_{km} \in \{0, 1\} \quad \forall m.\end{aligned} \tag{10}$$

The optimization objective contains only quadratic terms with linear constraints and can be efficiently solved by 0–1 programming. In the optimization, the first term indicates that a scene will be grouped into the cluster with low prediction error. The second term indicates that such process is also influenced by the labels of similar scenes.

Step 2: To optimize $\mathbf{W}$, we have to solve the problem which contains only the terms in (2) that are related to $\mathbf{W}$

$$\begin{aligned}\min_{\mathbf{W}} &\sum_{k=1}^{K}\sum_{u\neq v}^{N}[g_{ku} < g_{kv}]_{\mathbf{I}}\sum_{m=1}^{M}\alpha_{km}\eta_{kuv}^m \\ &+\frac{\epsilon_d}{M}\sum_{k=1}^{K}\sum_{u\neq v}^{N}[g_{ku} < g_{kv}]_{\mathbf{I}}\sum_{i\neq j}^{M}\eta_{kuv}^i\eta_{kuv}^j \\ &+\epsilon_c trace\left(\mathbf{W}^{\mathbf{T}}\mathbf{W}\right).\end{aligned} \tag{11}$$

Since the exponential upper bound is convex, the objective function (11) turns out to be convex since it contains only quadratic and exponential terms of $\mathbf{W}$ with non-negative weights. Therefore, we can solve it with gradient descent method. Note that we have

$$\frac{\partial\eta_{kuv}^m}{\partial\mathbf{W}} = \frac{\partial\exp\left(trace\left(\mathbf{W}^{\mathbf{T}}\mathbf{X}_{kuv}^m\right)\right)}{\partial\mathbf{W}} = \mathbf{X}_{kuv}^m\eta_{kuv}^m. \tag{12}$$

Therefore, the gradient direction can be written as follows:

$$\begin{aligned}\triangle\mathbf{W} \propto 2\epsilon_c\mathbf{W} &+ \sum_{k=1}^{K}\sum_{u\neq v}^{N}[g_{ku} < g_{kv}]_{\mathbf{I}}\sum_{m=1}^{M}\mathbf{X}_{kuv}^m \\ &\times\left(\alpha_{km}\eta_{kuv}^m + \frac{2\epsilon_d}{M}\sum_{i\neq m}^{M}\eta_{kuv}^i\eta_{kuv}^m\right).\end{aligned} \tag{13}$$

From (13), we can see that each model is actually optimized on the whole training set by emphasizing different subset of training samples [as shown in Fig. 3(b)]. Actually, the optimization process of $\mathbf{W}$ mainly involves two steps. That is, estimating the prediction of each ranking function on each training sample to update η_{kuv}^m and re-weighting each training sample to calculate the gradient direction $\triangle\mathbf{W}$. By iteratively performing these two steps, the convex objective (11) can effectively reach its global minimum.

The detailed learning process is listed in Algorithm 1. By iteratively updating α and optimizing $\mathbf{W}$, we can obtain a decreasing overall loss. In the algorithm, we iteratively carry out these two steps until the algorithm converges or reaches a predefined number of iterations.

Given a new scene, we have to identify a proper ranking function for estimating its saliency. Recall that the global descriptor can characterize a scene and guide the visual processes on it [26], we can select the ranking function for the new scene based on its global scene characteristics. That is, scenes with similar global characteristics are supposed to undergo the similar visual processes and the same ranking functions can be used for the saliency estimation. Therefore, we adopt a KNN classifier (1-NN in this paper) to find the scene in the training set with the most similar global scene characteristics [i.e., the cosine similarity as in (4)]. After that, the corresponding ranking function can be selected for visual saliency estimation.

In order to test the performance of the learned visual saliency model, we have to compare the estimated ranks with

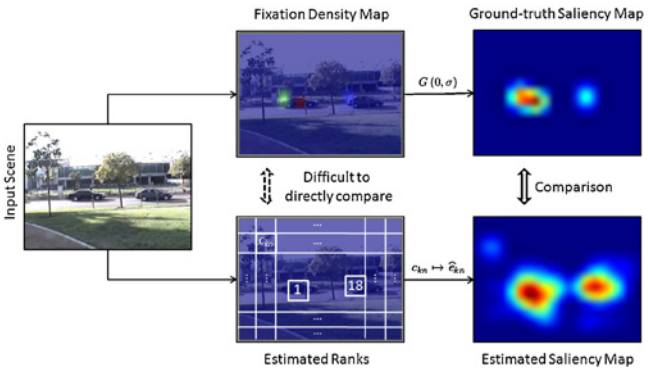

Fig. 4. Estimated ranks and eye fixations are compared by calculating the similarity between the estimated saliency maps and the ground-truth saliency maps.

eye fixation data. However, it is often difficult to directly perform such comparison. Therefore, we generate the estimated saliency map from these ranks and the ground-truth saliency map from eye fixations for further comparison (as shown in Fig. 4). Toward this end, we let $c_{kn} \in \{1, \ldots, N\}$ be the rank for the visual subset s_{kn} and empirically transform c_{kn} to the estimated saliency value $\widehat{e}_{kn}$

$$\widehat{e}_{kn} = G(0, \sigma) * \left(\frac{N - c_{kn}}{N} \right)^{\beta} \tag{14}$$

where $\beta > 0$ is a constant to pop-out the most salient targets and can be selected using the validation set. For larger β, the distractors can be suppressed more effectively. In our experiments, we set $\beta = 5$ and only limited locations can pop-out. In (14), a Gaussian kernel $G(0, \sigma)$ is adopted to generate the estimated saliency map from these locations by modeling the decrease in accuracy of the fovea with increasing eccentricity. Here we set $\sigma = 5$. For fair comparison, the same kernel is also used in the experiments to construct the ground-truth saliency maps from eye fixation data. For convenience, the estimated saliency values for all visual subsets in a scene are normalized into [0, 1].

D. Computational Complexity Analysis

In terms of computational complexity, the cost of the EM optimization, C_{EM}, can be written as follows:

$$C_{EM} = \sum_{i=1} C_{\alpha}^{i} + C_{\mathbf{W}}^{i} \tag{15}$$

where C_{α}^{i} ($C_{\mathbf{W}}^{i}$) denote the cost of updating α (optimizing $\mathbf{W}$) in the ith iteration. Thus, we can calculate these two costs separately to obtain the overall cost. For the sake of convenience, we let $N_a^k = \sum_{u \neq v}^{N} [g_{ku} < g_{kv}]_{\mathbf{I}}$ be the number of training samples in the kth scene and $N_a = \sum_{k=1}^{K} N_a^k$ be the total number of samples in all K scenes.

In updating α, the cosine distance of any two scenes can be obtained before the EM optimization and we suppose that $\{\eta_{kuv}^m\}$ have already been calculated (e.g., in optimizing $\mathbf{W}$). Therefore, α_k can be updated by taking the influences of the other $K - 1$ scenes into account. With respect to (10), the complexity of updating α_k is $O(N_a^k M + KM)$. Therefore, the overall complexity of updating $\alpha = \{\alpha_k\}$ is

$$C_{\alpha}^{i} = \sum_{k=1}^{K} O(N_a^k M + KM) = O(N_a M + K^2 M). \tag{16}$$

When optimizing $\mathbf{W}$, the cost depends on the convergence rate of the gradient-descent algorithm as well as the complexity for each gradient step. Let R_i be the number of gradient steps in optimizing $\mathbf{W}$ in the ith EM iteration and $C_{\triangle}^{i}$ be the computational complexity in each gradient step. Typically, R_i may vary remarkably in each EM iteration while the complexity $C_{\triangle}^{i}$ often remains constant in each gradient step. Therefore, the computational cost $C_{\mathbf{W}}^{i}$ can be written as follows:

$$C_{\mathbf{W}}^{i} = R_i \times C_{\triangle}^{i}. \tag{17}$$

From (13), we observe that $C_{\triangle}^{i}$ is related to two complexities.

1) The complexity in computing $\triangle \mathbf{W}$ using (13). Recall that only the mth column in $\mathbf{X}_{kuv}^{m}$ may have non-zero components (i.e., this column equals to $\mathbf{x}_{ku} - \mathbf{x}_{kv}$), the multiplication of $\mathbf{X}_{kuv}^{m}$ with a real value and the addition of $\mathbf{X}_{kuv}^{m}$ with another matrix have $O(L)$ complexity. Therefore, computing $\triangle \mathbf{W}$ using (13) has approximately $O(N_a M^2 L)$ complexity.
2) The complexity in computing $\{\eta_{kuv}^m\}$ using (9). From (8) and (9), we observe that $\eta_{kuv}^m = \exp(\omega_m^{\mathbf{T}} \mathbf{x}_{ku} - \omega_m^{\mathbf{T}} \mathbf{x}_{kv})$. Thus the complexity of updating $\{\eta_{kuv}^m\}$ is $O(N_a ML)$.

With these two complexities, $C_{\triangle}^{i}$ can be written as follows:

$$C_{\triangle}^{i} = O(N_a M^2 L) + O(N_a ML) \approx O(N_a M^2 L). \tag{18}$$

Algorithm 1: The multi-task rank learning algorithm

Input : Local visual attributes $\{\mathbf{x}_{kn}\}$, global scene characteristics $\{\mathbf{v}_k\}$, ground-truth saliency $\{g_{kn}\}$, iteration times T, threshold $\triangle e$.
Output: Model parameters $\mathbf{W}$, scene-cluster labels α.
begin
  Initialization:
  Group scenes into M clusters using K-means;
  Initialize α;
  Initialize $\mathbf{W}$ by minimizing (11) (set $\epsilon_s = \epsilon_d = \epsilon_c = 0$);
  $e^{(1)} \leftarrow \mathcal{L}(\mathbf{W}, \alpha) + \Omega(\mathbf{W}, \alpha)$;
  $t \leftarrow 1$;
  Optimization:
  repeat
    for $k \leftarrow 1$ **to** K **do** $\alpha_k^* \leftarrow$ minimize (10);
    $\mathbf{W} \leftarrow$ minimize (11);
    $e^{(t)} \leftarrow \mathcal{L}(\mathbf{W}, \alpha) + \Omega(\mathbf{W}, \alpha)$;
    $t \leftarrow t + 1$;
  until $t \geq T$ *or* $(e^{(t)} - e^{(t-1)}) < \triangle e$;
end

By incorporating (16)–(18) into (15), we observe that the overall computational complexity is tightly correlated with six parameters, including K (the number of training scenes), N_a (the number of training samples), M (the number of scene clusters), L (local feature dimensionality), $\{R_i\}$ (the numbers of gradient steps in optimizing $\mathbf{W}$), and the number of EM iterations. Among these six parameters, K is determined by the training set and different gradient-descent algorithms have different convergence rates, leading to different $\{R_i\}$ [37]. Moreover, we observe in the experiment that the EM optimization usually terminates in less than $T = 10$ iterations. For the other three parameters, there are three feasible ways to reduce the computational complexity.

1) Remove the redundant training samples to reduce N_a.
2) Reduce the cluster number M.
3) Reduce the features dimensionality L.

Often, the parameter L is predefined in different application (i.e., the number of candidate visual features in a specific application) and M should be optimized through cross validation. Therefore, we can reduce the computational complexity by removing the redundant training samples (e.g., by fusing the subsets in each scene with similar local visual attributes and ground-truth saliency values). In the experiment, we observe that when the scene number K, feature dimensionality L, and clusters number M are considered to be constants, the training time is linear with respect to the number of training samples. Compared with the typical multi-task learning approaches whose complexity may scale as the cube of the number of training samples (e.g., when using the regularization networks in [38] and [39]), the computational complexity of our approach is much less and is thus acceptable.

V. Experiments

In this section, we evaluate our approach on a public eye-fixation dataset [14]. The dataset, denoted as **ORIG**, consists of 46 489 frames in 50 video clips (25 min, 640×480). As shown in Table I, these video clips mainly contain genres such as "outdoors," "TV news," "sports," "commercials," "video games," and "talk shows." For these clips, the dataset also provides the eye traces of eight subjects recorded using a 240 Hz ISCAN RK-464 eye-tracker (four to six subjects per clip). Based on these eye traces, the fixation density in each 16×16 macro-block is calculated. After that, the ground-truth saliency map for each scene can be constructed by convolving the fixation density map with a Gaussian kernel ($\sigma = 5$) to model the decrease in accuracy of the fovea with increasing eccentricity.

On this dataset, the main objective of the experiment is to evaluate whether our approach can learn the features to distinguish targets from distractors and whether the learned features can be effectively transferred to new scenes for visual saliency estimation. Toward this end, we randomly select 1/10 scenes from the **ORIG** dataset to construct the training set, 1/10 scenes for validation and the rest scenes are used for testing. For the sake of convenience, the training/validation/testing sets are denoted as $\mathbb{D}_{\text{train}}$, $\mathbb{D}_{\text{validate}}$, and $\mathbb{D}_{\text{test}}$, respectively. On these training/validation/testing sets, four experiments are conducted. In the first experiment, the performance of our approach when using different parameters is tested. A set of optimal parameters are also selected for our approach, which will be used in the other three experiments. In the second and the third experiments, our approach is compared with the state-of-the-art saliency models and ranking models, respectively. These two experiments are designed to demonstrate the performance of our approach from different perspectives. Finally, we test the performance of all these models on different scene genres of the **ORIG** dataset in the last experiment.

TABLE I
Main Scene Categories of the **ORIG** Dataset

Video Genre	Video Num.	Scene Num.
Outdoor	17	8357
Video game	9	15 809
Commercials	4	2618
TV news	7	8071
Sports	5	4851
Talk shows	4	4244
Others	4	2539

In these experiments, our multi-task rank learning (**MTRL**) approach is compared with 12 state-of-the-art saliency/ranking models as well as the baseline of our approach. In general, these models can be grouped into three categories.

1) Bottom-up models for saliency estimation:
   a) Itti98 [2] and Itti01 [3]: models based on local contrasts;
   b) Itti05 [4] and Zhai06 [8]: models that mainly focus on inter-frame variation;
   c) Harel07 [7]: a graph-based model by detecting spatiotemporal irregularities;
   d) Hou07 [5] and Guo08 [6]: models based on spectral analysis.
2) Top-down models for saliency estimation:
   a) Kienzle07 [9]: a model that learns the correlations between local features and visual saliency by using a SVM classifier;
   b) Navalpakkam07 [10]: a model that combines local contrasts in preattentive features by optimizing the signal-noise-ratio;
   c) Peter07 [11]: a model that learns the projection matrix between global scene characteristics and eye density maps.
3) Ranking models for saliency estimation:
   a) Freund03 [15]: a boosting algorithm for learning weak rankers as well as their combination strategies;
   b) Joachims06 [16]: a pair-wise rank learning algorithm using support vector machine;
   c) **Base-I**: the baseline of our approach which groups scenes into clusters and infers a model for each cluster without considering the inter-scene and inter-model correlations.

Note that here the ranking-based approaches such as Freund03 [15], Joachims06 [16] and **Base-I** can only give integer

ranks, we also turn them into real saliency values using the same way as **MTRL** for fair comparisons.

In the comparisons, the receiver operating characteristics (**ROC**) curve is used for performance evaluation.[1] In general, **ROC** curve is a useful tool to visualize the performance of binary classifiers [40]. Also, it is the most prevalent criteria for evaluating the performance of visual saliency models (e.g., in [7], [9], [19], [23], and [41]). In the evaluation, a set of thresholds $T_{\text{roc}} = \{0.00, 0.01, \ldots, 1.00\}$ are used to select the salient visual subsets from all the estimated saliency maps predicted by a specific saliency model. These salient subsets are then validated according to the ground-truth saliency maps. For the threshold T_{roc}, the true positives (TP), false negatives (FN), false positives (FP) and true negatives (TN) on all the *test data* can be calculated as follows:

$$\begin{aligned} TP &= \sum_k \sum_{n=1}^{N} [\widehat{e}_{kn} \geq T_{\text{roc}}]_{\mathbf{I}} \cdot g_{kn} \\ FN &= \sum_k \sum_{n=1}^{N} [\widehat{e}_{kn} < T_{\text{roc}}]_{\mathbf{I}} \cdot g_{kn} \\ FP &= \sum_k \sum_{n=1}^{N} [\widehat{e}_{kn} \geq T_{\text{roc}}]_{\mathbf{I}} \cdot [g_{kn} = 0]_{\mathbf{I}} \\ TN &= \sum_k \sum_{n=1}^{N} [\widehat{e}_{kn} < T_{\text{roc}}]_{\mathbf{I}} \cdot [g_{kn} = 0]_{\mathbf{I}} \end{aligned} \tag{19}$$

where $0 \leq \widehat{e}_{kn}, g_{kn} \leq 1$ are the estimated and ground-truth saliency values, respectively. After that, the false positive rate is calculated as $FP/(FP + TN)$ and the true positive rate is calculated as $TP/(TP + FN)$. Correspondingly, the **ROC** curve for the saliency model is plotted as the *false positive rate* versus *true positive rate*. Moreover, the area under the **ROC** curve (**AUC**) is also calculated to demonstrate the overall performance of a saliency model. We also compute the improvement of **MTRL** against other approaches on the **AUC** score, denoted as **IMP**.

A. Parameter Selection

This experiment is designed to evaluate the influence of different parameters to our approach. In this experiment, we demonstrate the performance of our approach when using various parameters and select a set of optimal parameters for the next three experiments. The computational complexities when using different parameters are also reported.

In our approach, there are many parameters involved in the processes of model optimization and saliency estimation. Among these parameters, K (the number of training scenes) and N (the number of visual subsets in each scene) are determined by the training set. Note that here a visual subset corresponds to a 16×16 macro-block, which is the same as the block used in calculating the ground-truth saliency maps. Other parameters, such as $\epsilon_s, \epsilon_d, \epsilon_c, M$, and β have to be optimized by cross validation. By using the training set $\mathbb{D}_{\text{train}}$ and validation set $\mathbb{D}_{\text{validate}}$, we test the influence of each parameter by fixing all the other parameters. The influence of various parameters are summarized as follows.

1) ϵ_s: In the EM optimization, some scene-cluster labels may vary radically with respect to the prediction errors when ϵ_s is too small. In contrast, the scene-cluster labels will rarely change when ϵ_s is too large.
2) ϵ_d: When ϵ_d is small, **MTRL** is slightly influenced by inter-model correlations and its performance will approximate that of **Base-I**. For large ϵ_d, the models trained on various scene clusters may lack the diversity, leading to decreased performance.
3) ϵ_c: This parameter is only used to avoid constructing over-complex models. A smaller ϵ_c indicates that a more complex model is acceptable.
4) M: The scene cluster number M is an important parameter in the optimization process. Therefore, we draw a curve to demonstrate the influence of M to the **AUC** score. As shown in Fig. 5(a), a larger M may lead to higher **AUC** score. However, the computational complexity will become extremely high when M is become too large. Therefore, the selection of M should simultaneously consider the algorithm performance and computational complexity. In our experiment, we start from $M = 1$ and select the optimal M as the smallest number of clusters on which the **AUC** score is larger than the scores on $M + 1$, $M + 2$ and $M + 3$. Here we test the **AUC** scores on four successive cluster numbers to avoid sudden perturbance
   Moreover, we also illustrate some typical scene clusters in Fig. 5(a). From these scene clusters, we can see that scenes with similar spatial layouts and predictable variations will be grouped together as M gets large (e.g., $M = 15$). However, an increasing M may not always guarantee an increasing **AUC** score [e.g., the **AUC** scores in Fig. 5(a) when $M = 22$ and $M = 15$]. Actually, **Base-I** may become over-fitting when M is too large, while **MTRL** can avoid such problem by utilizing the inter-scene and inter-model correlations.
5) β: This is an important parameter to turn the estimated integer ranks into real saliency values. Therefore, we also draw a curve to demonstrate the influence of β to the **AUC** score. As shown in Fig. 5(b), the **AUC** reaches its maximum around $\beta = 5$. For smaller β, distractors cannot be adequately suppressed, leading to noisy saliency maps. In contrast, a large β may also suppress some salient targets.

By iteratively carrying out the cross validation and varying these parameters, a set of optimal parameters are selected for **MTRL** (as shown in Table II). Note that the same cluster number M is also used for **Base-I** in the following experiments. Moreover, the parameters of all the other learning-based approaches, including [9]–[11], [15], and [16], have their parameters optimized in the same way to get fair comparisons in the following experiments.

In addition, we also conduct an experiment to test the computational complexity of **MTRL**. With the optimal parameters

[1]The source code for computing **ROC** curves is provided by Harel *et al.* [7] and is publicly available at http://www.klab.caltech.edu/ harel/share/gbvs.php.

TABLE II
SELECTED PARAMETERS FOR **MTRL**

Parameter	Value	Description
ϵ_s	0.10	Weight of penalty term Ω_s
ϵ_d	0.11	Weight of penalty term Ω_d
ϵ_c	0.14	Weight of penalty term Ω_c
M	15	Number of scene clusters
β	5	Parameter to turn ranks to saliency values

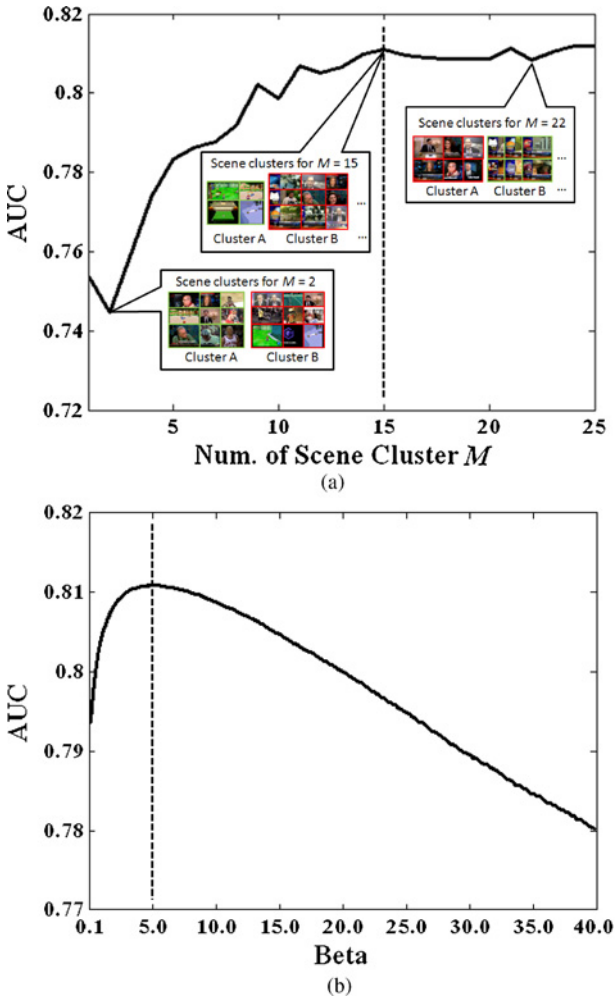

Fig. 5. **AUC** scores when using different parameters. (a) **AUC** scores when using different number of scene clusters ($\epsilon_s = 0.105, \epsilon_d = 0.105, \epsilon_c = 0.14, \beta = 5$). (b) **AUC** scores when using different β to turn integer ranks to real saliency values ($\epsilon_s = 0.105, \epsilon_d = 0.105, \epsilon_c = 0.14, M = 15$).

in Table II, we use different criteria to remove the redundant training samples (e.g., fusing the visual subsets in a scene with similar local visual attributes and ground-truth saliency values) and test the performance of **MTRL**. Note that here all the training processes are carried on a DELL Optiplex 960 computer with three threads.

As shown in Fig. 6, the time costs before the convergence of the EM optimization are almost linearly correlated with the numbers of training samples. In contrast, the **AUC** scores will increase when **MTRL** is trained with more samples. However, we can see from Fig. 6 that reducing the training number to 30%–40% will not severely decrease the **AUC**

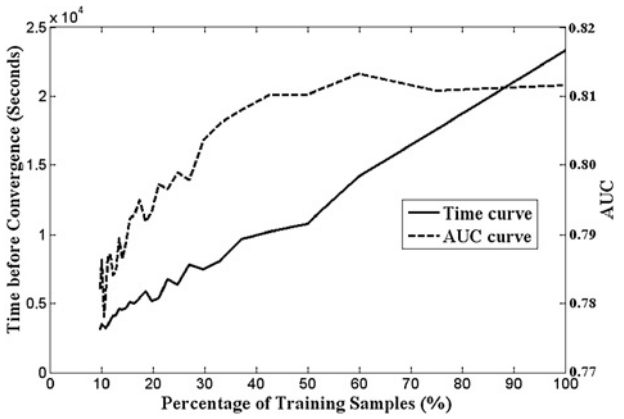

Fig. 6. Time costs in training **MTRL** and its **AUC** scores when using different numbers of training samples.

score. Therefore, reducing N_a is a feasible way to reduce the computational complexity while preserving the overall performance.

Moreover, we also test the efficiency of **MTRL** in estimating visual saliency. On a 2.66 GHz CPU, the time used in each major step (i.e., extracting local/global features, selecting ranking function through 1-NN, predicting ranks and transforming these ranks into real saliency values) is recorded. Note that here the I/O time is not taken into account. On average, it takes about 0.109 s to estimate the visual saliency for a new scene. From this result, we can see that our approach show high efficiency in visual saliency estimation, which is very important since the saliency estimation is usually the first step for many applications.

B. Comparisons with Saliency Models

In this experiment, we will test the performance of various visual saliency models and give explanations from the neurobiological perspective. The comparisons are mainly made between **MTRL** and ten bottom-up and top-down visual saliency models. In the experiment, the comparisons are conducted for ten times. In each comparison, the learning-based approaches are trained on $\mathbb{D}_{\text{train}}$ and tested on $\mathbb{D}_{\text{test}}$. After that, a **ROC** curve is generated for each model based on all the estimated saliency maps in all the ten comparisons. A unified **AUC** score is also reported to demonstrate the performance of each model. The **AUC** scores of various saliency models are shown in Table III. The **ROC** curves are illustrated in Fig. 7 and some representative results are given in Fig. 8.

From Table III and Fig. 7, we can see that **MTRL** outperforms all the other visual saliency models. As shown in Fig. 8(c) and (d), Itti98 [2], and Itti01 [3] only maintain the most salient subsets with the "winner-take-all" competition, leading to the low **AUC** scores. In contrast, Itti05 [4], Zhai06 [8] and Harel07 [7] have achieved a bit improvement by focusing on the spatiotemporal visual irregularities, while Hou07 [5] and Guo08 [6] perform even better by detecting such irregularities through spectrum analysis. As shown in Fig. 8(e)–(i), these five bottom-up approaches can well locate the salient subsets but may have difficulties in suppressing the distractors. In particular, we observe from Table III that

TABLE III
PERFORMANCE OF VARIOUS SALIENCY MODELS

Algorithm		AUC	IMP (%)
Bottom-Up	Itti98 [2]	0.557	45.5
	Itti01 [3]	0.554	46.4
	Itti05 [4]	0.622	30.4
	Zhai06 [8]	0.637	27.3
	Harel07 [7]	0.584	38.9
	Hou07 [5]	0.666	21.7
	Guo08 [6]	0.674	20.3
Top-Down	Kienzle07 [9]	0.539	50.3
	Navalpakkam07 [10]	0.697	16.3
	Peters07 [11]	0.693	17.1
	MTRL	**0.811**	

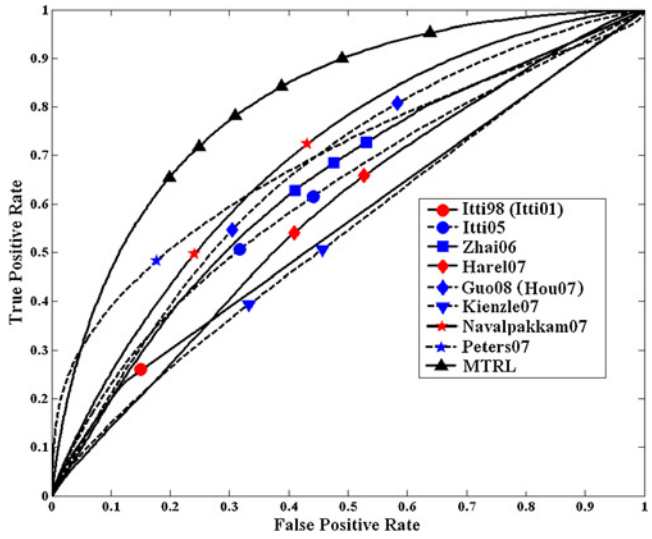

Fig. 7. **ROC** curves of **MTRL** and various saliency models.

the spatial saliency models [2], [5] may perform even better than some spatiotemporal saliency models (e.g., [3], [4], [7], [8]). This indicates that simply incorporating the temporal features (e.g., motion and flicker) may not always guarantee a better performance. Actually, human fixations can be driven by different spatial/temporal visual features in different scenes. Thus selecting the right features to distinguish targets from distractors is the most important issue for visual saliency estimation other than incorporating more candidate visual features into the model.

Often, it is believed that the experience from past similar scenes can assist the suppression of distractors. However, the top-down approach Kienzle07 [9] acts even worse than the bottom-up approaches, since it simply maps the local visual features to saliency values. However, the correlations between visual features and saliency values may not always hold in different scenes. In contrast, Peters07 [11] simply infers the relations between global scene characteristics and eye density maps. As shown in Fig. 8(l) and Table III, the advantage of Peters07 [11] is that it can well recover the most probable salient locations (e.g., the center of each scene), leading to a higher **AUC** score. However, they may also introduce a lot of noise into the estimated saliency maps. Particularly, Peters07 [11] have to infer an 832×300 projection matrix between global scene characteristics and eye density maps,

TABLE IV
PERFORMANCE OF VARIOUS RANKING MODELS

Algorithm	AUC	IMP (%)
Freund03 [15]	0.735	10.2
Joachims06 [16]	0.716	13.6
Base-I	0.739	9.72
MTRL	**0.811**	

while the training data is often insufficient to do so. In fact, the **AUC** score of Peters07 [11] can reach 0.745 if it is trained on 9/10 scenes of the **ORIG** dataset. This hampers the utilization of Peters07 [11] in some applications which may only provide sparse training data and user feedbacks.

As shown in Table III and Fig. 7, we can see that Navalpakkam07 [10] acts much better (**AUC** = 0.697) by considering both the influences of local visual attributes and target-distractor correlations. Moreover, **MTRL** have achieved the best performance (**AUC** = 0.811) by simultaneously taking the influences of local visual attributes and global scene characteristics into a pair-wise ranking framework. From the neurobiological perspective, the pair-wise ranking framework can assist finding features that can distinguish targets from distractors. With these features, the salient targets can be successfully pop-out while distractors can be effectively suppressed. Meanwhile, the experience of successfully popping-out targets and suppressing distractors in past similar scenes can be memorized and transferred to new scenes under the facilitation of global scene characteristics. As shown in Fig. 8(m), the saliency maps generated by **MTRL** contain less noise than other top-down approaches. This also explains the reason that **MTRL** outperforms Navalpakkam07 [10]. In **MTRL**, only the experience on past similar scenes are transferred to the new scenes, while Navalpakkam07 [10] infers a generalized solution for suppressing distractors. Often, such generalized experience may not apply to all scenes, particularly for those scenes with diversified contents. As shown in the last row of Fig. 8(k), the targets are suppressed while the distractors are mistakenly emphasized.

C. Comparisons with Ranking Models

In this experiment, we will test the performance of various learning-based ranking models on visual saliency estimation and give explanations from the perspective of machine learning. The comparisons are made between **MTRL**, **Base-I** and two rank learning approaches. This experiment adopts the same settings as the second experiment (e.g., the same training/testing sets, the same way to calculate the **ROC** curves). The **AUC** scores of various ranking models are shown in Table IV. The **ROC** curves are illustrated in Fig. 9 and some representative results are given in Fig. 10.

From Tables III and IV, we can see that most ranking models outperforms the typical bottom-up and top-down visual saliency models. Compared with the learning-based top-down saliency models, the main advantage of ranking models is that they can better focus on the features that distinguishes targets from distractors. Particularly, we can see that the scene-specific models, including **Base-I** and **MTRL**, outperform the

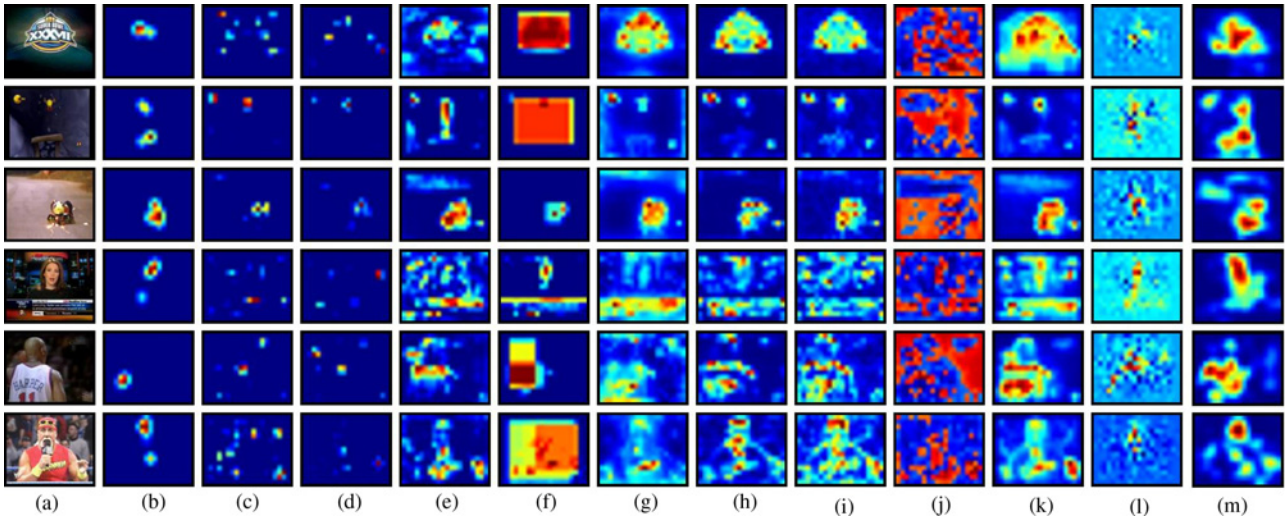

Fig. 8. Some representative results of visual saliency models. (a) Original scenes. (b) Ground-truth saliency maps. (c) Itti98 [2]. (d) Itti01 [3]. (e) Itti05 [4]. (f) Zhai06 [8]. (g) Harel07 [7]. (h) Hou07 [5]. (i) Guo08 [6]. (j) Kienzle07 [9]. (k) Navalpakkam07 [10]. (l) Peters07 [11]. (m) **MTRL**.

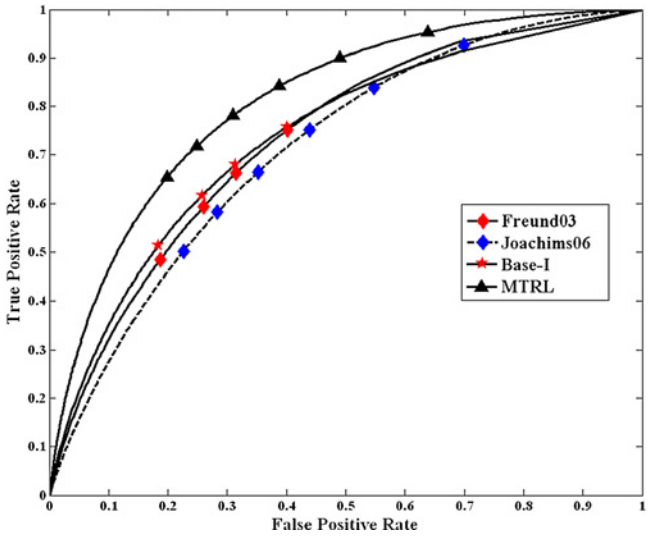

Fig. 9. **ROC** curves of **MTRL** and various ranking models.

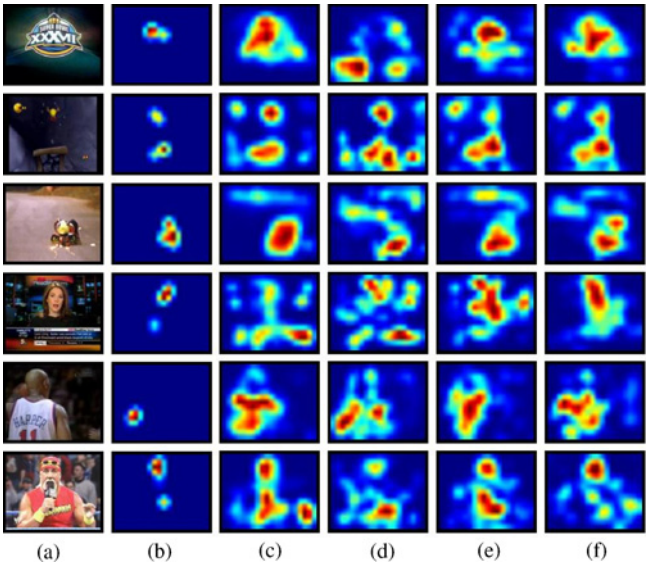

Fig. 10. Some representative results of ranking models. To facilitate the comparison between saliency models and ranking models, we show the results on the same frames. (a) Original scenes. (b) Ground-truth saliency maps. (c) Freund03 [15]. (d) Joachims06 [16]. (e) **Base-I**. (f) **MTRL**.

unified models Freund03 [15] and Joachims06 [16]. Generally speaking, a video is not simply a collection of randomized scenes. Successive video frames often have similar targets and distractors. By grouping similar training scenes into the same cluster, the model trained on this cluster can better apply to this kind of scenes. We can see that **Base-I** (**AUC** = 0.739) outperforms Joachims06 [16] (**AUC** = 0.716) by simply grouping scenes and constructing cluster-specific models. However, the model trained on each cluster (particular for the cluster with limited number of scenes) often lacks the generalization ability and may become over-fitting due to the low diversity of training samples. In some scenes, **Base-I** with $M = 15$ clusters may perform even worse than directly training **Base-I** (with $M = 1$) on all scenes.

To avoid the over-fitting problem, **MTRL** adopts a multi-task learning framework to train multiple saliency models simultaneously with considering the inter-model correlations. With a properly designed multi-task learning algorithm, each model in **MTRL** is actually trained on the whole training set by emphasizing different subset of the training data. Therefore, the saliency models in **MTRL** have improved generalization ability and avoid over-fitting. Meanwhile, the model diversity is also guaranteed. Therefore, **MTRL** can outperform **Base-I** remarkably (as shown in Table IV and Fig. 9).

D. Performance on Various Video Genres

In this experiment, we will test the performance of all the saliency and ranking models on different video contents. In the test, all the learning-based models are trained on $\mathbb{D}_{\text{train}}$ and tested on $\mathbb{D}_{\text{test}}$. Different from previous experiments, the performances are separately reported on the seven video genres of the **ORIG** dataset. The **AUC** scores of various approaches on different video genres are presented in Table V. Moreover,

TABLE V
AUC SCORES ON VARIOUS VIDEO GENRES

	Outdoor	Video Game	Commercials	TV News	Sports	Talk Shows	Others
Itti98 [2]	0.559	0.584	0.530	0.531	0.536	0.552	0.569
Itti01 [3]	0.558	0.565	0.538	0.534	0.537	0.580	0.553
Itti05 [4]	0.615	0.640	0.593	0.587	0.577	0.679	0.647
Zhai06 [8]	0.662	0.632	0.684	0.656	0.590	0.598	0.632
Harel07 [7]	0.646	0.540	0.648	0.545	0.538	0.668	0.664
Hou07 [5]	0.684	0.661	0.684	0.660	0.607	0.699	0.696
Guo08 [6]	0.691	0.689	0.690	0.629	0.632	0.687	0.705
Kienzle07 [9]	0.541	0.537	0.519	0.509	0.518	0.516	0.555
Navalpakkam07 [10]	0.713	0.683	0.681	0.706	0.668	0.751	0.697
Peters07 [11]	0.670	0.724	0.631	0.686	0.687	0.680	0.670
Freund03 [15]	0.734	0.736	0.719	0.724	0.714	0.783	0.751
Joachims06 [16]	0.690	0.713	0.709	0.625	0.644	0.680	0.640
Base-I	0.735	0.763	0.733	0.714	0.717	0.758	0.747
MTRL	**0.824**	**0.833**	**0.806**	**0.773**	**0.783**	**0.818**	**0.808**

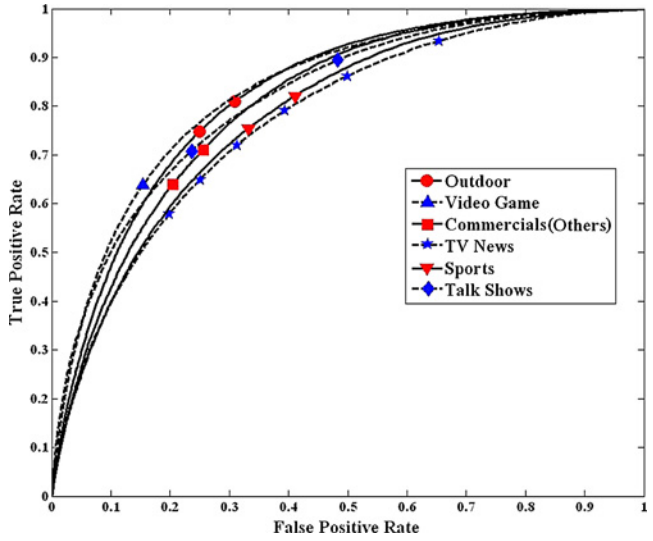

Fig. 11. **ROC** curves of **MTRL** on different video genres.

the **ROC** curves of **MTRL** on different video contents are illustrated in Fig. 11.

From Table V, we find that most approaches perform the best in the "video game" genre. Often, the scenes in "video game" have obvious salient targets which can be easily distinguished from distractors. Actually, when playing the game, users often adjust the scene (e.g., change the view angle and character position) to ensure that the targets (e.g., the game characters) can easily pop-out. Moreover, the scenes in "video game" are often relatively simpler than the scenes in other video genres (e.g., the "TV news"), which will also assist the saliency estimation.

In contrast to the "video game" genre, most algorithms achieve the worst performance on the "TV news" and "sports" genres. The main reason is that the scenes in "TV news" and "sports" are more complex than the scenes in other video genres. For example, a scene of "news" can have anchor, caption, scrolling text, logo and other contents, each of which can be a probable salient target [as shown in the fourth row of Fig. 8(a)]. In such scenes with rich contents, it is difficult to distinguish targets from distractors only using the visual features. In some cases, the targets and distractors can be effectively separated only by using semantic clues. Therefore, incorporating the influences of various semantic clues (e.g., human face [23], [25] and camera motion [24]) into visual saliency estimation could be a challenging research direction.

VI. CONCLUSION

In this paper, we proposed a novel approach for visual saliency estimation. Compared with existing methods, our approach has three main advantages. First, the pair-wise rank learning framework can effectively learn the visual features that best distinguish targets from distractors. Second, our approach can effectively pop-out the targets and suppress the distractors in various scenes by using scene-specific models. Third and the most importantly, a multi-task learning framework is adopted to infer multiple visual saliency models simultaneously. By an appropriate sharing of information across models, the performance of each model is greatly improved. From the results obtained so far, our approach outperforms several state-of-the-art bottom-up, top-down and rank learning approaches remarkably. In the future work, we will extend this approach to object-based visual saliency estimation. Moreover, we will also explore the way to reduce the computational complexity (e.g., finding the analytical solutions).

REFERENCES

[1] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive Psychol.*, vol. 12, no. 1, pp. 97–136, 1980.
[2] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.
[3] L. Itti and C. Koch, "Computational modeling of visual attention," *Nature Rev. Neurosci.*, vol. 2, no. 3, pp. 194–203, 2001.
[4] L. Itti and P. Baldi, "A principled approach to detecting surprising events in video," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2005, pp. 631–637.
[5] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2007, pp. 1–8.
[6] C. Guo, Q. Ma, and L. Zhang, "Spatio-temporal saliency detection using phase spectrum of quaternion Fourier transform," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2008, pp. 1–8.

[7] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Advances in Neural Information Processing Systems*, vol. 19. Cambridge, MA: MIT Press, 2007, pp. 545–552.
[8] Y. Zhai and M. Shah, "Visual attention detection in video sequences using spatiotemporal cues," in *Proc. ACM Int. Conf. Multimedia*, 2006, pp. 815–824.
[9] W. Kienzle, F. A. Wichmann, B. Scholkopf, and M. O. Franz, "A nonparametric approach to bottom-up visual saliency," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2007, pp. 689–696.
[10] V. Navalpakkam and L. Itti, "Search goal tunes visual features optimally," *Neuron*, vol. 53, no. 4, pp. 605–617, Feb. 2007.
[11] R. J. Peters and L. Itti, "Beyond bottom-up: Incorporating task-dependent influences into a computational model of spatial attention," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2007, pp. 1–8.
[12] L. Jacob, F. Bach, and J.-P. Vert, "Clustered multi-task learning: A convex formulation," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2008, pp. 745–752.
[13] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2007, pp. 41–48.
[14] L. Itti, "Crcns data sharing: Eye movements during free-viewing of natural videos," in *Proc. Collaborative Res. Comput. Neurosci. Annu. Meeting*, Jun. 2008, pp. 1–4.
[15] Y. Freund, R. Iyer, R. E. Schapire, Y. Singer, and G. Dietterich, "An efficient boosting algorithm for combining preferences," *J. Mach. Learning Res.*, vol. 4, pp. 170–178, Dec. 2004.
[16] T. Joachims, "Training linear SVMS in linear time," in *Proc. ACM Conf. Knowl. Discovery Data Mining*, 2006, pp. 217–226.
[17] L. Itti, G. Rees, and J. Tsotsos, *Neurobiology of Attention*. San Diego, CA: Elsevier, 2005.
[18] C. Frith, "The top in top-down attention," in *Neurobiological Attention*, L. Itti, G. Rees, and J. K. Tsotsos, Eds., 1st ed. Amsterdam, The Netherlands: Elsevier Press, 2005, pp. 105–108.
[19] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Netw.*, vol. 19, no. 9, pp. 1395–1407, 2006.
[20] S. Marat, T. H. Phuoc, L. Granjon, N. Guyader, D. Pellerin, and A. Guerin-Dugue, "Modeling spatio-temporal saliency to predict gaze direction for short videos," *Int. J. Comput. Vision*, vol. 82, no. 3, pp. 231–243, 2009.
[21] Y. Hu, D. Rajan, and L.-T. Chia, "Robust subspace analysis for detecting visual attention regions in images," in *Proc. ACM Int. Conf. Multimedia*, 2005, pp. 716–724.
[22] H. Liu, S. Jiang, Q. Huang, C. Xu, and W. Gao, "Region-based visual attention analysis with its application in image browsing on small displays," in *Proc. ACM Int. Conf. Multimedia*, 2007, pp. 305–308.
[23] M. Cerf, J. Harel, W. Einhauser, and C. Koch, "Predicting human gaze using low-level saliency combined with face detection," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2008, pp. 241–248.
[24] G. Abdollahian, Z. Pizlo, and E. J. Delp, "A study on the effect of camera motion on human visual attention," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2008, pp. 693–696.
[25] Y.-F. Ma, X.-S. Hua, L. Lu, and H.-J. Zhang, "A generic framework of user attention model and its application in video summarization," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 907–919, Oct. 2005.
[26] M. M. Chun, "Contextual guidance of visual attention," in *Neurobiological Attention*, L. Itti, G. Rees, and J. K. Tsotsos, Eds., 1st ed. Amsterdam, The Netherlands: Elsevier Press, pp. 246–250, 2005.
[27] W. Kienzle, B. Scholkopf, F. A. Wichmann, and M. O. Franz, "How to find interesting locations in video: A spatiotemporal interest point detector learned from human eye movements," in *Proc. 29th DAGM Symp.*, 2007, pp. 405–414.
[28] D. Pang, A. Kimura, T. Takeuchi, J. Yamato, and K. Kashino, "A stochastic model of selective visual attention with a dynamic Bayesian network," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Jun. 2008, pp. 1073–1076.
[29] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2007, pp. 1–8.
[30] T. Liu, N. Zheng, W. Ding, and Z. Yua, "Video attention: Learning to detect a salient object sequence," in *Proc. IEEE Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.
[31] J. Li, Y. Tian, T. Huang, and W. Gao, "Cost-sensitive rank learning from positive and unlabeled data for visual saliency estimation," *IEEE Signal Process. Lett.*, vol. 17, no. 6, pp. 591–594, Jun. 2010.
[32] A. Torralba, "Contextual influences on saliency," in *Neurobiological Attention*, L. Itti, G. Rees, and J. K. Tsotsos, Eds., 1st ed. Amsterdam, The Netherlands: Elsevier Press, pp. 586–592, 2005.
[33] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *J. Mach. Learning Res.*, vol. 6, pp. 615–637, Apr. 2005.
[34] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Royal Statist. Soc. Series B*, vol. 39, no. 1, pp. 1–38, 1977.
[35] M.-R. Amini, T.-V. Truong, and C. Goutte, "A boosting algorithm for learning bipartite ranking functions with partially labeled data," in *Proc. SIGIR*, 2008, pp. 99–106.
[36] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge University Press, 2008.
[37] R. L. Watrous, "Learning algorithms for connectionist networks: Applied gradient methods of nonlinear optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Jun. 1987, pp. 619–627.
[38] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *Mach. Learning Res.*, vol. 6, pp. 615–637, Apr. 2005.
[39] G. Pillonetto, G. D. Nicolao, M. Chierici, and C. Cobelli, "Fast algorithms for nonparametric population modeling of large data sets," *Automatica*, vol. 45, no. 1, pp. 173–179, Jan. 2009.
[40] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.
[41] N. D. Bruce and J. K. Tsotsos, "Saliency based on information maximization," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2006, pp. 155–162.

Jia Li received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 2005. He is currently working toward the Ph.D. degree in computer science from the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing.

His current research interests include visual attention/saliency modeling, multimedia analysis, and online video advertising.

Yonghong Tian (M'05–SM'10) received the Ph.D. degree in computer applications from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China.

He is currently an Associate Professor with the National Engineering Laboratory for Video Technology, School of Electronic Engineering and Computer Science, Peking University, Beijing. His current research interests include machine learning and multimedia content analysis, retrieval, and copyright management.

Dr. Tian is a member of ACM.

Tiejun Huang (M'01) received the Ph.D. degree in pattern recognition and intelligent systems from the Huazhong University of Science and Technology, Wuhan, China.

He is currently a Professor with the School of Electrical Engineering and Computer Science and the Deputy Director of the National Engineering Laboratory for Video Technology, Peking University, Beijing, China. His current research interests include image understanding, video coding, digital libraries, and digital copyright management.

Dr. Huang is a member of ACM.

Wen Gao (M'92–SM'05–F'09) received the M.S. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1985, and the Ph.D. degree in electronics engineering from the University of Tokyo, Tokyo, Japan, in 1991.

He was a Professor of computer science with the Harbin Institute of Technology from 1991 to 1995. He was a Professor of computer science with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, from 1996 to 2005. He is currently a Professor with the School of Electronic Engineering and Computer Science and the Director of the National Engineering Laboratory for Video Technology, Peking University, Beijing. He published four books and over 500 technical articles in refereed journals and proceedings in the areas of signal processing, image and video communication, computer vision, multimodal interface, pattern recognition, and bioinformatics. His current research interests include all fields of digital media technology.

Dr. Gao received many awards, including five national awards for research achievements and activities. He has served the academic community in many positions, including as the General Co-Chair of the IEEE International Conference on Multimedia and Expo in 2007, and the Head of Chinese Delegation to the Moving Picture Expert Group of the International Standard Organization since 1997. He is the Chairman of the Working Group responsible for setting a National Audio Video Coding Standard for China.