Deep3DSaliency: Deep Stereoscopic Video Saliency Detection Model by 3D Convolutional Networks

Yuming Fang, *Senior Member, IEEE*, Guanqun Ding, Jia Li, *Senior Member, IEEE*, and Zhijun Fang, *Senior Member, IEEE*

***Abstract*— Stereoscopic saliency detection plays an important role in various stereoscopic video processing applications. However, conventional stereoscopic video saliency detection methods mainly use independent low-level features instead of extracting them automatically, and thus, they ignore the intrinsic relationship between the spatial and temporal information. In this paper, we propose a novel stereoscopic video saliency detection method based on 3D convolutional neural networks, namely, deep 3D video saliency (Deep3DSaliency). The proposed network consists of two sub-models: spatiotemporal saliency model (STSM) and stereoscopic saliency aware model (SSAM). STSM directly takes three consecutive video frames as the input to extract visual spatiotemporal features, while SSAM attempts to further infer the depth and semantic features from the left and right video frames by shared parameters from STSM. The visual spatiotemporal features from STSM and the depth and semantic features from SSAM are learned by an alternating optimization scheme. Finally, all these saliency-related features are combined together for the final stereoscopic saliency detection via 3D deconvolution. Experimental results show the superior performance of the proposed model over other existing ones in saliency estimation for 3D video sequences.**

***Index Terms*— Visual attention, stereoscopic video, spatiotemporal saliency, 3D convolutional neural networks.**

I. Introduction

VISUAL attention is an important characteristic in the Human Visual System (HVS) for visual information processing. It is a cognitive process of selecting the relevant regions while acquiring the most significant visual information from visual scenes. As an important and challenging problem in computer vision, saliency detection has attracted a lot of attention in the past decades, since it can be used in various multimedia processing applications such as object recognition [1], image retargeting [2], image compression [3], [4], object tracking [5], defect detection [6], abnormal event detection [7] and person re-identification [8].

Saliency detection methods generally can be categorized as either human eye fixation prediction [9]–[13] approaches and salient object detection [14]–[18] approaches. The first one aims to identify salient locations where human observers fixate during scene viewing, and we call it as eye fixation regions. The latter, salient object detection, focuses on predicting saliency values of pixels that determine whether the pixels belong to salient objects or not. In this paper, we focus on human eye fixation prediction task in stereoscopic video sequences.

Despite recent great progress in saliency detection for 2D images/videos, saliency detection for stereoscopic video sequences remains challenging. First, it is not easy to extract the accurate motion information in video sequences, and thus the small and fast moving objects in video sequences are usually difficult to be captured. For the early models of salient motion detection, they attempt to extract moving foreground objects as salient regions, but these methods cannot solve the occlusion problem due to the lost foreground objects. Furthermore, the depth properties of a visual scene typically have significant effect on visual fixations. Some existing 3D video saliency detection methods fuse the spatiotemporal and depth saliency maps with fixed weights for 3D video saliency prediction. This may ignore the intrinsic relationship between spatiotemporal and depth features. Thus, how to extract and combine the depth information and spatiotemporal features such as semantic/motion cues is important to design effective stereoscopic video saliency detection models.

Currently, there are several 3D video saliency detection models proposed for various multimedia processing applications [11], [19], [20]. For traditional 3D video saliency detection models, they first extract spatial, temporal and depth features to compute spatial, temporal and depth saliency maps; then the final saliency map for video sequences is predicted by combining the spatial, temporal and depth saliency maps with certain fusion method [19], [20]. Most of these methods manually extract low-level features such as color, luminance, and texture for spatial saliency estimation. However, they might lose some important high-level features such as semantic

Manuscript received May 2, 2018; revised October 19, 2018; accepted November 27, 2018. Date of publication December 5, 2018; date of current version January 30, 2019. This work was supported in part by the Natural Science Foundation of China under Grants 61822109, 61571212, 61772328, and 61461021, in part by the Natural Science Foundation of Jiangxi under Grant 20181BBH80002, in part by the Henry Fok Education Foundation under Grant 161061, and in part by the Beijing Nova Program under Grant Z181100006218063. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Damon M. Chandler. *(Corresponding author: Zhijun Fang.)*

Y. Fang and G. Ding are with the School of Information Management, Jiangxi University of Finance and Economics, Nanchang 330032, China (e-mail: fa0001ng@e.ntu.edu.sg).

J. Li is with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China, and also with the Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing 100191, China.

Z. Fang is with the School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China (e-mail: zjfang@sues.edu.cn).

information in 3D video sequences. Some existing methods attempt to use linear or nonlinear combination rules to fuse spatial and temporal saliency simply [11], [20], which may ignore the intrinsic relationship between spatial and temporal information due to the fixed weights used for the combination of spatial and temporal information.

Recently, deep learning has been successfully applied in object detection [21], semantic segmentation [22] and saliency detection [23]–[25]. Most of these existing models in the related fields of computer vision are designed for 2D image/video, which means they don't consider the depth information and they cannot be used for stereoscopic video processing. In this study, we adopt 3D convolutional and 3D deconvolutional neural networks to extract and fuse spatiotemporal and depth features simultaneously to build an effective stereoscopic video saliency detection model. One of the significant insights of this work is that, different from traditional stereoscopic video saliency detection methods utilizing computationally expensive optical flow for motion feature extraction, the proposed model learn spatial and temporal features from the raw video frames with 3D convolutional operation. Thus, it can reduce the computational time and decrease the required computer resources. The novelty of the proposed method is the idea of conversion from 2D features to 3D features for stereoscopic video saliency prediction. Admittedly, there have been a lot of studies investigating 2D features for 2D video saliency detection, while few works investigate the stereoscopic video saliency detection from 2D features to 3D features. Here, we investigate the conversion from 2D features to 3D features for stereoscopic video saliency detection by 3D convolutional networks and deconvlutional networks.

In sum, we propose a novel stereoscopic video saliency detection method based on 3D convolutional neural network (Deep3DSaliency). A spatiotemporal saliency model (STSM) is first designed to extract spatiotemporal features effectively. Considering the importance of intrinsic semantic and depth features in stereoscopic video saliency detection, we design a stereoscopic saliency aware model (SSAM) whose parameters are partially shared with STSM to effectively extract semantic and depth features for stereoscopic video sequences. Taking into account the fact that there is strong intrinsic relationship among spatiotemporal, depth and semantic features for stereoscopic video saliency detection, we design a novel joint training scheme for the proposed two sub-models (STSM and SSAM) to enhance the capability of feature learning and reduce the feature redundancy.

II. Related Works

In the past decades, many effective saliency detection methods have been proposed for images [9], [15], [16], [26]–[30]. Itti *et al.* [9] proposed an early saliency detection model by multi-scale center-surround contrast calculation on intensity, color and orientation features. Different from Itti's method [9] using low-level features, Liu *et al.* [16] incorporated both low-level and high-level features into saliency diffusion, and learn specific formulation and boundary condition of Linear Elliptic System with Dirichlet boundary (LESD) for images. Jia and Han [26] computed high-level saliency prior with the objectness concept to find potential object candidates, and then enforced the consistency among the salient regions using a Gaussian MRF with different weights to emphasize the influence of potential foreground pixels. Tong *et al.* [15] first calculated the bottom-up saliency map by considering global contrast information via low level features such as Histogram of Oriented Gradient (HOG) and Local Binary Pattern (LBP). Secondly, a top-down saliency map is formulated based on the reconstruction error by using a locality-constrained linear coding algorithm [15]. The final saliency map is predicted by combining the bottom-up and top-down saliency maps [15]. Ran *et al.* [27] utilized the patch distribution to compute pattern distinctness via Principal Component Analysis (PCA) to detect salient regions. Wang *et al.* [28] presented an unsupervised method that incorporates geodesic distance into saliency empowered video object segmentation. Tavakoli and Laaksonen [30] introduced a bottom-up unsupervised multi-scale hierarchical model with Independent Subspace Analysis (ISA) architecture for human eye fixation prediction.

Besides 2D image saliency detection algorithms, there have been also many effective saliency models proposed for 2D video sequences [31]–[39]. Liu *et al.* [31] constructed a superpixel-based graph with a virtual background node to represent the global motion, then the author design a spatiotemporal saliency propagation method in both forward and backward directions on inter-frame and intra-frame to obtain spatiotemporal saliency map for video saliency measurement. Similarly, Xi *et al.* [32] proposed a salient object detection method with bidirectional consistency propagation for video sequences based on spatiotemporal background priors. They first integrated multiple pairs of scale-invariant feature transform flows from multi-frames and then uses bidirectional consistency propagation method to generate spatiotemporal background priors. Finally, they adopted a dual-graph-based structure method with the background priors to calculate final saliency map. Different from the study [32] exploiting background priors, Aytekin *et al.* [35] calculated spectral foreground based on Quantum Cuts (QCUT) method to estimate spatiotemporal saliency by fusing local and global information including color and motion contrast, shape and background. In addition, the background prior is extracted for calculating the saliency map by a constructed spatially graph based on Manifold Ranking (MF) in several existing algorithms [33], [40].

Different from 2D image/video saliency detection, 3D image/video saliency detection is more challenging due to complicated depth and motion information existing in 3D video sequences [11], [12], [19], [20], [41]–[43]. Most of existing methods predict saliency maps of 3D images by fusing depth-related features and 2D visual features [11], [42]. Zhang *et al.* [42] proposed a bottom-up visual attention model for 3D video by combining the features including depth, motion, luminance, color and orientation. The authors claimed that pixels closer to viewers and in the front of visual scenes are more salient. Wang *et al.* [19] introduced a method to compute the depth saliency map and integrate it with 2D salient

features for the final stereoscopic saliency map calculation. Ferreira *et al.* [20] constructed a 3D video saliency detection model by fusing spatial, temporal and depth feature maps. Kim *et al.* [11] proposed a 3D video saliency model by using low-level features including luminance, chrominance, motion, and depth as well as high-level features of visual scenes. Fang *et al.* [12] released a large-scale eye fixation databases for stereoscopic video saliency detection and designed a stereoscopic video saliency detection model inspired by the laws of proximity, continuity and common fate in Gestalt theory.

Compared with traditional approaches, Deep Convolution Neural Network (DCNN) has made great success in the task of saliency detection [14], [23]–[25], [44], [45], since it can automatically learn rich features instead of hand-crafted features. Li *et al.* [14] built a multi-task learning framework to share the features between image semantic segmentation and saliency detection at the same time. Moreover, Liu *et al.* [44] proposed an eye fixation prediction model by adopting multi-resolution convolutional neural network (MR-CNN) to extract three types of saliency features including local contrast, global contrast and top-down visual factors. Banitalebi-Dehkordi *et al.* [45] designed a learning-based saliency detection model by incorporating low-level features such as brightness, color, texture, motion, and depth as well as high-level cues including face, person, vehicle, text, and horizon. Wang *et al.* proposed a saliency detection model by fusing local estimation and global search [25]. In that study, the local estimation utilizes the cues of local contrast, texture and shape to learn patch features, while the global search component is designed by global contrast information, geometric features, and object candidate cues [25]. Huang *et al.* [46] proposed a deep-based saliency detection architecture by fine-tuning the existing Deep Neural Networks (DNNs) including AlexNet [47], VGG16-net [48] and GoogLeNet [49]. Cornia *et al.* [63] designed a multi-level deep feature learning model for eye fixation prediction by a learned prior. Li and Yu [50] proposed a saliency detection model that incorporates multi-scale CNN features extracted from nested windows into a deep neural network with multiple fully connected layers.

Recently, the deconvolutional neural networks are effectively applied in visualization [51], semantic segmentation [22], [52], building extraction [53] and medical image processing [54]. Zeiler and Fergus [51] proposed a visualization method with deconvolutional network to investigate the function of intermediate feature layers and discover the performance contribution from each convolutional layers. In order to predict pixel-level semantic segmentation value, some studies [22], [52] utilize deconvolutional network to act as an feature combination and upsampling role for restoring the original image size. Huang *et al.* [53] proposed a end-to-end model based on deep deconvolutional networks for remote sensing images, where the final extraction result is fused by two saliency maps calculated from fully convolutional networks [53]. Fakhry *et al.* [54] used residual deconvolutional networks (RDN) to process brain electron microscopy images, and RDN consists of two information pathways from residual networks.

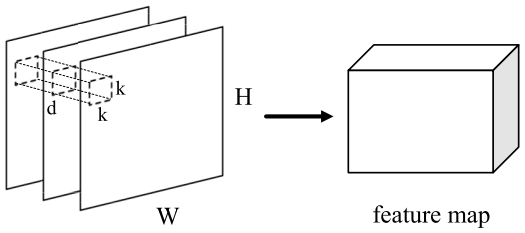


Fig. 1. Illustration of 3D convolutional operation. The kernel of 3D convolutional layer is cube with size $d \times k \times k$, where d represents the size of depth/temporal dimension and k stands for the spatial filter size. W and H denote width and height of feature maps, respectively.

As indicated above, the common yet key problems in 3D video saliency detection include how to effectively extract spatial, temporal, and depth features simultaneously, and combine them together when they are available. In this paper, we construct a novel stereoscopic video saliency detection model (namely Deep3DSaliency) by using 3D convolutional neural networks for effectively extracting and combining spatiotemporal and depth features in stereoscopic video sequences. Fig. 1 demonstrate the 3D convolutional operation. It can be used to efficiently learn spatiotemporal features such as motion cues for video sequences. Moreover, Tran *et al.* [55] demonstrated that 3D convolutional deep networks are useful and effective for learning spatiotemporal features by a set of empirically explored architectures, and the feature map generated from 3D convolutional layer can preserve temporal information of the input video sequences. Some studies [55], [56] have found that 3D convolutional neural networks can capture appearance and motion cues efficiently and obtain much better performance than 2D convolutional neural networks for video analysis tasks. Here, we adopt 3D convolutional and deconvolutional neural networks to construct the deep models for stereoscopic video saliency detection by extracting and combining the spatiotemproal, depth and semantic features. The proposed model is demonstrated in details in the next section.

III. Proposed Method

A. Architecture Overview

The proposed model is demonstrated in Fig. 2. As we can see from this framework, the proposed method includes two parts: STSM for spatiotemporal feature learning for video sequences, and SSAM for depth and semantic feature learning. Additionally, 3D DeconvNet of SSAM is used to learn saliency by fusing depth, semantic and spatiotemporal features. First, we feed three consecutive video frames (L_{t-1}, L_t, L_{t+1}) to pre-train STSM for learning spatiotemporal features. Then, we fine-tune SSAM with corresponding left and right video frames (L_t, R_t) as the input of 3D ConvNet, and feed the left frame (L_t) into 2D ConvNet. Besides, the ground truth map G_t of video frame (L_t) in the training set is used to calculate the loss of forward propagation.

For simplicity, we denote $d \times k \times k$ as the kernel/stride size for 3D convolutional layer, 3D pooling layer, 3D deconvolutional layer and 3D unpooling layer, where d represents the kernel/stride depth in temporal dimension and k stands

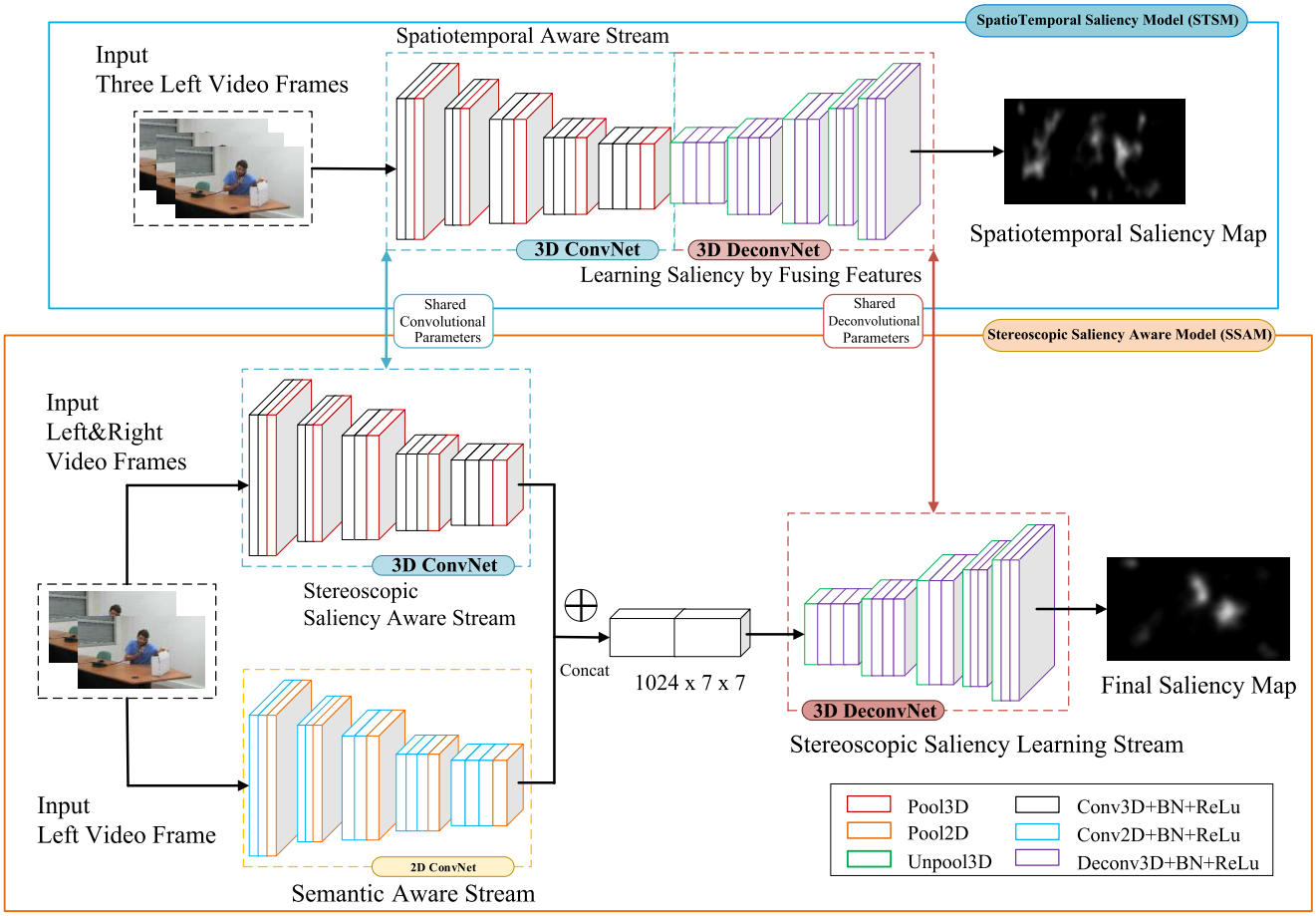


Fig. 2. Architecture of the proposed stereoscopic video saliency detection model. There are the following parts in the proposed framework: STSM with three consecutive video frames for spatiotemporal feature learning, 3D ConvNet of SSAM with corresponding left and right video frames as the input for depth feature learning, 2D ConvNet with left video frame as the input for semantic feature learning, and 3D DeconvNet for the final saliency learning.

for spatial filter/stride size. Besides, we intend to employ $n \times h \times w \times c$ to indicate the output shape of 3D convolution and deconvolution layers, where n represents the number of input video frames; h, w, and c are the parameters for height, width and channels of video frames or feature maps.

B. The Spatiotemporal Saliency Detection Model

In this subsection, we introduce STSM and explain how it could learn spatiotemporal features effectively. Compared with 2D convolutional neural networks, many studies [56] have shown that 3D convolutional neural network has the adequate and good capability to learn spatiotemporal features thanks to the operations of 3D convolution and 3D pooling.

As shown in Fig. 2, we construct a new deep neural network STSM including a 3D convolutional network and a 3D deconvolutional network. The 3D convolutional network consists of twelve 3D convolution layers and five 3D max-pooling layers. Each 3D convolutional layer includes a batch normalization [57] and a ReLu (Rectified Linear Unites) operation, and it is defined as follows:

$$f(x) = \begin{cases} x, & \mathrm{x} > 0 \\ 0, & \mathrm{x} \leq 0 \end{cases} \tag{1}$$

where x denotes the input feature.

In addition, due to the stride of convolutional and pooling operations, the output feature maps will be down-sampled and become sparse. This is the reason why we design a 3D deconvolutional network including 5 unpooling layers and twelve 3D deconvolutional layers to learn saliency by fusing spatiotemporal features for the proposed model. We explore a bunch of video frames as the input of the sub-model STSM, i.e., 2, 3, 5, 7, 9, 11, *etc.* We observe that noises increase and performance decreases with more frames since more redundant visual information will be introduced and 3D convolutional operation is sensitive to motion information. Thus, the sub-model STSM takes three consecutive left video frames (L_{t-1}, L_t, L_{t+1}) as the input of the constructed network to learn the coherence and motion information between video frames, which significantly contributes to 3D video saliency detection.

Existing studies have shown that the convolutional filter with homogenous parameters of $3 \times 3 \times 3$ is effective for 3D convolutional networks [56], thus we set 3D convolutional kernel as $d \times 3 \times 3$ with stride $1 \times 1 \times 1$ in the proposed model. With direct extension of 2D max-pooling to the temporal dimension, many researches [56] have demonstrated that 3D max-pooling operation can work on multiple temporal samples. As can be seen from Fig. 2 and TABLE I, the stride sizes of five

TABLE I

THE DETAILED CONFIGURATION OF THE PROPOSED STSM AND SSAM SUB-MODELS. PLEASE NOTE THAT KERNAL AND STRIDE OF 3D OPERATIONS IS WITH DEPTH × HEIGHT × WIDTH, AND THE INPUT AND OUTPUT SHAPE ARE WITH [BATCH_SIZE, NUMBER_OF_FRAMES, HEIGHT, WIDTH, CHANNEL]. THE COLORED PARAMETERS ARE SHARED BETWEEN STSM AND SSAM

3D ConvNet of STSM				
Layer	Kernal@Chan.	Stride	Input	Output
Con3D1_1	3x3x3@64	1x1x1	[10,3,224,224,3]	[10,3,224,224,64]
Conv3D1_2	3x3x3@64	1x1x1	[10,3,224,224,64]	[10,3,224,224,64]
Pool3D1	~	2x2x2	[10,3,224,224,64]	[10,2,112,112,64]
Conv3D2_1	2x3x3@128	1x1x1	[10,2,112,112,64]	[10,2,112,112,128]
Conv3D1_2	2x3x3@128	1x1x1	[10,2,112,112,128]	[10,2,112,112,128]
Pool3D2	~	1x2x2	[10,2,112,112,128]	[10,2,56,56,128]
Conv3D3_1	2x3x3@256	1x1x1	[10,2,56,56,128]	[10,2,56,56,256]
Conv3D3_2	2x3x3@256	1x1x1	[10,2,56,56,256]	[10,2,56,56,256]
Pool3D3	~	1x2x2	[10,2,28,28,256]	[10,2,28,28,256]
Conv3D4_1	2x3x3@256	1x1x1	[10,2,28,28,256]	[10,2,28,28,256]
Conv3D4_2	2x3x3@256	1x1x1	[10,2,28,28,256]	[10,2,28,28,256]
Conv3D4_3	2x3x3@256	1x1x1	[10,2,28,28,256]	[10,2,28,28,256]
Pool3D4	~	1x2x2	[10,2,28,28,256]	[10,2,14,14,256]
Conv3D5_1	2x3x3@512	1x1x1	[10,2,14,14,256]	[10,2,14,14,512]
Conv3D5_2	2x3x3@512	1x1x1	[10,2,14,14,512]	[10,2,14,14,512]
Conv3D5_3	2x3x3@512	1x1x1	[10,2,14,14,512]	[10,2,14,14,512]
Pool3D5	~	2x2x2	[10,2,14,14,512]	[10,1,7,7,512]

3D ConvNet of SSAM				
Layer	Kernal@Chan.	Stride	Input	Output
Con3D1_1	2x3x3@64	1x1x1	[10,2,224,224,3]	[10,2,224,224,64]
Conv3D1_2	2x3x3@64	1x1x1	[10,2,224,224,64]	[10,2,224,224,64]
Pool3D1	~	1x2x2	[10,2,224,224,64]	[10,2,112,112,64]
Conv3D2_1	2x3x3@128	1x1x1	[10,2,112,112,64]	[10,2,112,112,128]
Conv3D1_2	2x3x3@128	1x1x1	[10,2,112,112,128]	[10,2,112,112,128]
Pool3D2	~	1x2x2	[10,2,112,112,128]	[10,2,56,56,128]
Conv3D3_1	2x3x3@256	1x1x1	[10,2,56,56,128]	[10,2,56,56,256]
Conv3D3_2	2x3x3@256	1x1x1	[10,2,56,56,256]	[10,2,56,56,256]
Pool3D3	~	1x2x2	[10,2,56,56,256]	[10,2,28,28,256]
Conv3D4_1	2x3x3@256	1x1x1	[10,2,28,28,256]	[10,2,28,28,256]
Conv3D4_2	2x3x3@256	1x1x1	[10,2,28,28,256]	[10,2,28,28,256]
Conv3D4_3	2x3x3@256	1x1x1	[10,2,28,28,256]	[10,2,28,28,256]
Pool3D4	~	1x2x2	[10,2,28,28,256]	[10,2,14,14,256]
Conv3D5_1	2x3x3@512	1x1x1	[10,2,14,14,256]	[10,2,14,14,512]
Conv3D5_2	2x3x3@512	1x1x1	[10,2,14,14,512]	[10,2,14,14,512]
Conv3D5_3	2x3x3@512	1x1x1	[10,2,14,14,512]	[10,2,14,14,512]
Pool3D5	~	2x2x2	[10,2,14,14,512]	[10,1,7,7,512]
Concat.	expend_pool2D5	~	[10,1,7,7,512]	[10,1,7,7,1024]
	Pool3D5	~	[10,1,7,7,512]	

2D ConvNet of SSAM				
Layer	Kernal@Chan.	Stride	Input	Output
Con2D1_1	3x3@64	1x1	[10,224,224,3]	[10,224,224,64]
Conv2D1_2	3x3@64	1x1	[10,224,224,64]	[10,224,224,64]
Pool2D1	~	2x2	[10,224,224,64]	[10,112,112,64]
Conv2D2_1	3x3@128	1x1	[10,112,112,64]	[10,112,112,128]
Conv2D1_2	3x3@128	1x1	[10,112,112,128]	[10,112,112,128]
Pool2D2	~	2x2	[10,112,112,128]	[10,56,56,128]
Conv2D3_1	3x3@256	1x1	[10,56,56,128]	[10,56,56,256]
Conv2D3_2	3x3@256	1x1	[10,56,56,256]	[10,56,56,256]
Pool2D3	~	2x2	[10,56,56,256]	[10,28,28,256]
Conv2D4_1	3x3@256	1x1	[10,28,28,256]	[10,28,28,256]
Conv2D4_2	3x3@256	1x1	[10,28,28,256]	[10,28,28,256]
Conv2D4_3	3x3@256	1x1	[10,2,28,28,256]	[10,28,28,256]
Pool2D4	~	2x2	[10,28,28,256]	[10,14,14,256]
Conv2D5_1	3x3@512	1x1	[10,14,14,512]	[10,14,14,512]
Conv2D5_2	3x3@512	1x1	[10,14,14,512]	[10,14,14,512]
Conv2D5_3	3x3@512	1x1	[10,14,14,512]	[10,14,14,512]
Pool2D5	~	2x2	[10,14,14,512]	[10,7,7,512]
expend_pool2D5	~	~	[10,7,7,512]	[10,1,7,7,512]

3D DeconvNet of STSM				
Unpool3D1	~	1x2x2	[10,1,7,7,512]	[10,1,14,14,512]
Deconv3D1_1	1x3x3@512	1x1x1	[10,1,14,14,512]	[10,1,14,14,512]
Deconv3D1_2	1x3x3@512	1x1x1	[10,1,14,14,512]	[10,1,14,14,512]
Deconv3D1_3	1x3x3@512	1x1x1	[10,1,14,14,512]	[10,1,14,14,512]
Unpool3D2	~	1x2x2	[10,1,14,14,512]	[10,1,28,28,512]
Deconv3D2_1	1x3x3@256	1x1x1	[10,1,28,28,512]	[10,1,28,28,256]
Deconv3D2_2	1x3x3@256	1x1x1	[10,1,28,28,256]	[10,1,28,28,256]
Deconv3D2_3	1x3x3@256	1x1x1	[10,1,28,28,256]	[10,1,28,28,256]
Unpool3D3	~	1x2x2	[10,1,28,28,256]	[10,1,56,56,256]
Deconv3D3_1	1x3x3@128	1x1x1	[10,1,56,56,256]	[10,1,56,56,128]
Deconv3D3_2	1x3x3@128	1x1x1	[10,1,56,56,128]	[10,1,56,56,128]
Unpool3D4	~	1x2x2	[10,1,56,56,128]	[10,1,112,112,128]
Deconv3D4_1	1x3x3@64	1x1x1	[10,1,112,112,128]	[10,1,112,112,64]
Deconv3D4_2	1x3x3@64	1x1x1	[10,1,112,112,64]	[10,1,112,112,64]
Unpool3D5	~	1x2x2	[10,1,112,112,64]	[10,1,224,224,64]
Deconv3D5_1	1x3x3@1	1x1x1	[10,1,224,224,64]	[10,1,224,224,1]
Deconv3D5_2	1x3x3@1	1x1x1	[10,1,224,224,1]	[10,1,224,224,1]

3D DeconvNet of SSAM				
Unpool3D1	~	1x2x2	[10,1,7,7,1024]	[10,1,14,14,1024]
Deconv3D1_1	1x3x3@512	1x1x1	[10,1,14,14,1024]	[10,1,14,14,512]
Deconv3D1_2	1x3x3@512	1x1x1	[10,1,14,14,512]	[10,1,14,14,512]
Deconv3D1_3	1x3x3@512	1x1x1	[10,1,14,14,512]	[10,1,14,14,512]
Unpool3D2	~	1x2x2	[10,1,14,14,512]	[10,1,28,28,512]
Deconv3D2_1	1x3x3@256	1x1x1	[10,1,28,28,512]	[10,1,28,28,256]
Deconv3D2_2	1x3x3@256	1x1x1	[10,1,28,28,256]	[10,1,28,28,256]
Deconv3D2_3	1x3x3@256	1x1x1	[10,1,28,28,256]	[10,1,28,28,256]
Unpool3D3	~	1x2x2	[10,1,28,28,256]	[10,1,56,56,256]
Deconv3D3_1	1x3x3@128	1x1x1	[10,1,56,56,256]	[10,1,56,56,128]
Deconv3D3_2	1x3x3@128	1x1x1	[10,1,56,56,128]	[10,1,56,56,128]
Unpool3D4	~	1x2x2	[10,1,56,56,128]	[10,1,112,112,128]
Deconv3D4_1	1x3x3@64	1x1x1	[10,1,112,112,128]	[10,1,112,112,64]
Deconv3D4_2	1x3x3@64	1x1x1	[10,1,112,112,64]	[10,1,112,112,64]
Unpool3D5	~	1x2x2	[10,1,112,112,64]	[10,1,224,224,64]
Deconv3D5_1	1x3x3@1	1x1x1	[10,1,224,224,64]	[10,1,224,224,1]
Deconv3D5_2	1x3x3@1	1x1x1	[10,1,224,224,1]	[10,1,224,224,1]

3D max-pooling layers of STSM sub-model are assigned as follows in the proposed method: $1 \times 2 \times 2$ for Pool3D2, Pool3D3, and Pool3D4 layers; $2 \times 2 \times 2$ for Pool3D1 and Pool3D5 layers. We set these parameters for all 3D max-pooling layers as above since we intend to learn more temporal features between video frames and do not expect to combine these temporal information at early stage. All strides of 3D unpooling layer are set as $1 \times 2 \times 2$ to upsample the spatial size of feature maps, while the temporal dimension is fixed to 1 since we aim to calculate the saliency map of unitary frame L_t.

Meanwhile, the parameters of 3D convolutional and deconvolutional layers are all randomly initialized by zero mean Gaussian distribution whose standard deviation is 0.01. All the weights of convolutional and deconvolutional layers are iteratively updated during the back propagation procedure. The l-th layer output feature map x^l of 2D/3D convolutional operation can be denoted as follows:

$$x^l = f(\sum W^l * x^{l-1} + b^l) \tag{2}$$

where $*$ is convolutional operation; $x^{(l-1)}$ denotes the *(l-1)*-th layer output feature map and W^l represents 2D/3D convolutional filter of the l-th layer. With a bias term b^l added to the convolutional results, an active function f is used to improve the hierarchical nonlinear mapping learning capability.

C. The Stereoscopic Saliency Aware Model

As shown in Fig. 2, SSAM includes three components: 3D convolutional network (3D ConvNet), 2D convolutional network (2D ConvNet) and 3D deconvolutional network (3D DeconvNet). We first restore the spatiotemporal saliency parameters of pre-trained STSM to 3D ConvNet and 3D DeconvNet of SSAM. Note that we skip the parameters of the first two convolutional layers and the first two deconvolutional layers of STSM, since we intend to re-train them with corresponding left and right video frames to learn stereoscopic saliency cues by randomly initialized weights, as shown in TABLE I. The detail training scheme is described in section IV in detail.

Additionally, we construct a 2D ConvNet to learn semantic saliency information. It is well known that VGG16-net [48] is a model pre-trained on the public ImageNet [58] dataset including rich semantic features. Here, we modify VGG16-net [48] for the saliency detection task by removing the last three fully-connected layers while preserving the layers before pool5 of VGG16-net [48]. We initialize the remain convolutional layers of VGG16-net [48] to 2D ConvNet for fine-tuning, which means we utilize the activation maps before the fully-connected layers. After concatenating the learned feature maps from 2D ConvNet and 3D ConvNet, we feed them into 3D DeconvNet to fuse semantic, depth and spatiotemporal features for 3D video sequences and up-sample the resolution of feature maps.

To train these two models, we propose a new robust loss function to update the parameters based on Gaussian function and Kullback-Leibler (KL) divergence. Huang *et al.* [46] demonstrated that KL divergence is effective for training saliency prediction deep network. The KL divergence between generated saliency map S_t and the human eye fixation map G_t

can be calculated as follows:

$$\mathcal{J}_{KL}(G_t, S_t) = G_t \log \frac{G_t}{S_t} \tag{3}$$

$$= \frac{1}{N}\sum_{i=1}^{N}[g_i * (\log(g_i + \epsilon) - \log(s_i + \epsilon))] \tag{4}$$

where g_i denotes the i-th element of the vector obtained by ground truth map G_t; s_i denotes the i-th element of the vector obtained by predicted saliency map S_t; ϵ is a small non-zero constant to avoid log-zero problem and we set $\epsilon = 1e-4$ during training. Here, we transfer G_t and S_t into one-dimension vector to calculate mean average J_{KL} with element-wise manner.

In addition, considering the Gaussian-like property in human eye fixation map, we measure the cost between S_t and G_t by the following function:

$$\mathcal{J}_G(G_t, S_t) = 1 - \exp^{(-\frac{(G_t - S_t)^2}{\sigma^2})} \tag{5}$$

$$= 1 - \frac{1}{N}\sum_{i=1}^{N}\exp^{-(\frac{(s_i - g_i)^2}{\delta})} \tag{6}$$

where g_i denotes the i-th element of the vector obtained by ground truth map G_t; s_i denotes the i-th element of the vector obtained by predicted saliency map S_t; δ is a small non-zero constant to avoid denominator to be zero, and we set $\delta = 1e-4$. Here, we transfer G_t and S_t into one-dimension vector to calculate mean average with element-wise manner.

In Eq. (3), the smaller KL divergence indicates higher accuracy in saliency prediction. Thus, the final objective loss function of the proposed deep model can be defined as follows:

$$\mathcal{J} = \mathcal{J}_G + \mathcal{J}_{KL} + \gamma \parallel W \parallel_2^2 \tag{7}$$

where $\parallel W \parallel_2^2$ denotes L_2-norm regularization term on parameters; γ is a hyper-parameter to balance the trade-off between the loss and the regularization.

Moreover, we adopt Adaptive Moment Estimation (Adam) [59] to optimize the proposed model. Adam is an optimization method using the first moment estimation and second moment estimation of gradient to update the learning rate adaptively.

IV. Experiment Results

In this section, in order to demonstrate the saliency prediction performance of the proposed model, we first describe the datasets used in the training and testing stages in detail. Then we introduce the implementation details of the proposed model and evaluation methods in this section. Besides, we report the performance evaluation results of three comparison experiments including: performance evaluation of sub-models in the proposed model, performance comparison by using other existing models, and cross dataset validation experiment.

A. Datasets and Evaluation Metrics

1) Datasets: In this study, we conduct the comparison experiments by using two public datasets of stereoscopic video saliency detection: DML-iTrack-3D [45] and FANG-Dataset [12]. Each dataset contains left-view and right-view 3D video sequences with the corresponding eye-fixations from human subjects. For convenient description, we provide the detailed information of these datasets in Table II and provide some samples in Fig. 3.

TABLE II

The Basic Information of Two Public Eye-Fixation Datasets for Stereoscopic Video Sequences: DML-iTrack-3D [45] and FANG-Dataset [12]

Datasets	Video Clips	Frames	Annotations
DML-iTrack-3D	27	5400	5400
FANG-Dataset	47	18715	18715

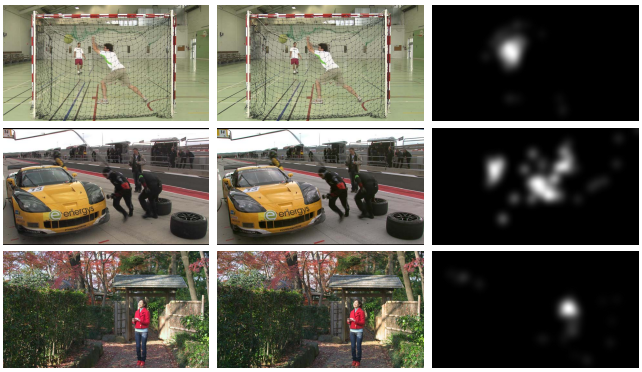


Fig. 3. Several visual examples from FANG-Dataset [12]. First column to final column: left-view video frames, right-view video frames, and human eye fixation maps.

Similar to the studies [23] and [25], we randomly choose 27 video sequences including about 10k frames from FANG-Dataset [12] as the training set, and the rest video sequences in this database are used as testing set. For DML-iTrack-3D [22], we use 15 video sequences including about 3k frames as training set, while the remaining video sequences are used as test set. For pre-training STSM sub-model, we use three consecutive left video sequences in FANG-Dataset [12] and DML-iTrack-3D [45] as the input to effectively learn spatiotemporal features. For re-training SSAM sub-model, we use the corresponding left and right video frames in FANG-Dataset [12] and DML-iTrack-3D [45] as the input to 3D ConvNet of SSAM for depth feature learning, and the left video frame is used as the input to 2D ConvNet of SSAM for semantic feature learning.

2) Implementation Details: During the training stage, all the parameters are learned by optimizing the loss function. In our experiments, the proposed deep network of 3D video saliency detection is implemented in Ubuntu operating system with the toolbox, Tensorflow library [60], an open source software for deep learning developed by Google. The experiments are conducted on a computer with Intel Core I7-6900K*16 CPU (3.20GHz), 64 GB RAM and Nvidia TITAN X (Pascal) GPU with 12 GB memory. The initial learning rate is set as $1e-5$ and divided by 10 after every 4 training epochs. These two sub-models stop training after 150K iterations in total.

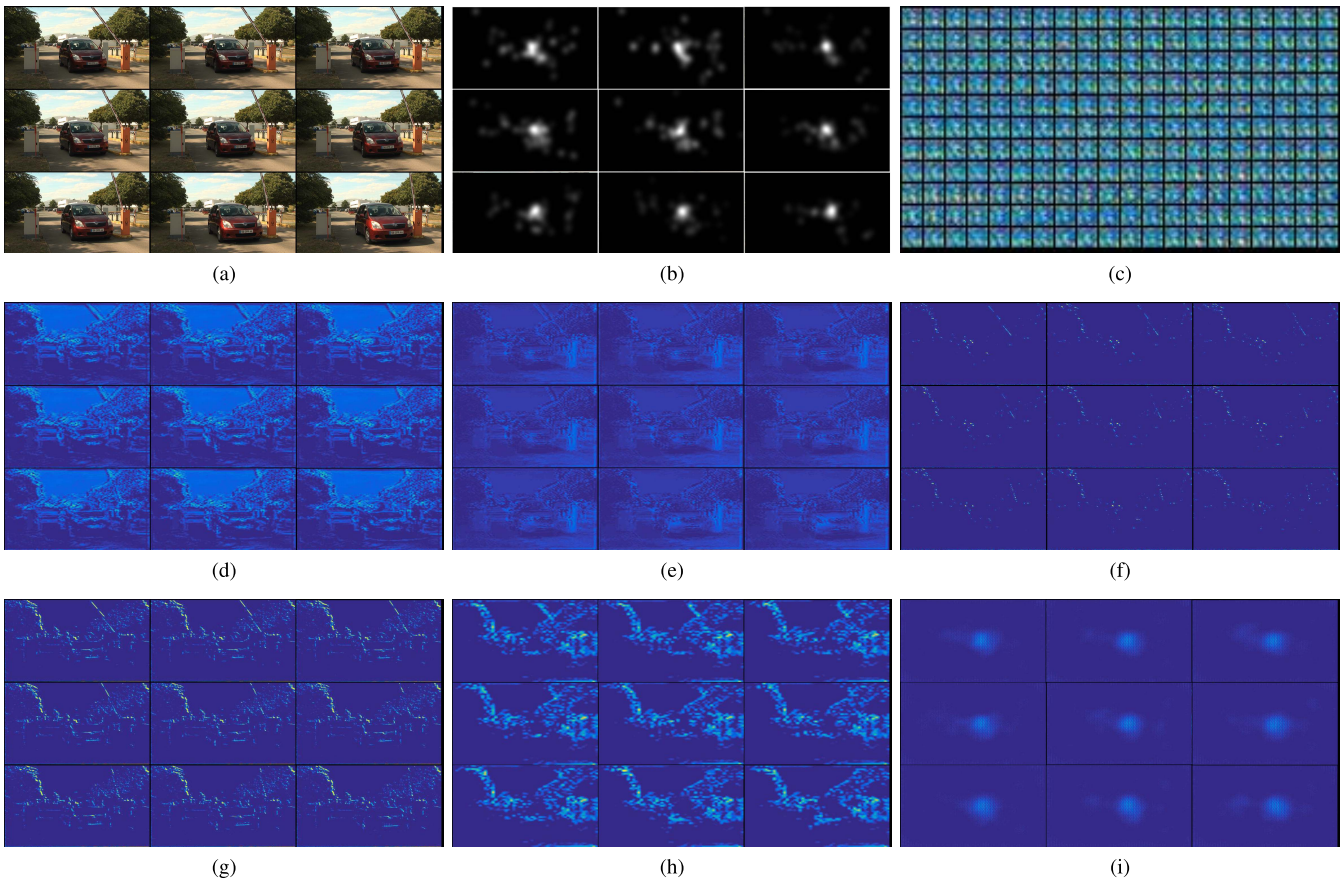


Fig. 4. Visualization of learned features. Left-top to right-down: (a) input video frames; (b) ground truth maps; (c) concatenated features of SSAM; (d) spatiotemporal features of STSM from Conv3D1_1; (e) stereoscopic features of SSAM from Conv3D1_1; (f) Deconv3D1_2 fused features of STSM; (g) Conv3D3_2 spatiotemporal features of STSM; (h) Conv3D3_2 fine-tuned features of SSAM; (i) Deconv3D5_2 fused features of SSAM.

In our implementation, the input video frame is with RGB format. The tensor shape of STSM sub-model is [10, 3, 224, 224, 3], where the vector means [batch size, number of frames, height, width, channels]. The tensor shape of 3D ConvNet of SSAM sub-model is [10, 2, 224, 224, 3]. In Fig. 1 of the manuscript, the kernel of 3D convolutional network is $d \times k \times k$, where d represents the size of depth or temporal dimension and k stands for the spatial filter size. This indicates that the perceptive field of convolutional layer is $d \times k \times k$. In the proposed STSM sub-model, we set the 3D convolutional kernel as $d \times 3 \times 3$, d means temporal dimension of the tensor shape.

In order to accelerate the saliency prediction speed and share the features between STSM and SSAM, we use an offline pre-trained STSM to fine-tune SSAM. Note that STSM is only used in the training stage and the learned spatiotemporal features can be preserved in SSAM well (as shown in Fig. 4 (g) and (h). In the testing stage, we use SSAM to compute the saliency map with the input 3D video frames (including corresponding left- and right- views). We train these two sub-models with different video frames (three video frames for STSM and two for SSAM). When initializing SSAM from STSM, we skip the first two 3D convolutional layers of SSAM to address the problem of the different numbers of input video frames to STSM and SSAM. The learned features of skipped layers are shown in (d) and (e) of Fig. 4. With the proposed framework, we first learn spatiotemporal features by STSM and then learn depth features by SSAM for stereoscopic video sequences, which implements the conversion from 2D features to 3D features for stereoscopic video saliency detection. The rich spatial and temporal saliency features can be transferred from STSM to SSAM, as shown in Fig. 4 (g) and (h). Finally, we use 3D deconvolutional networks to fuse the learned spatiotemporal, depth and semantic features, as shown in Fig. 4 (f) and (i).

Next, we describe the joint training scheme that shows how to train the proposed model and how to transfer the spatiotemporal and depth information between these two sub-models

TABLE III
THE TRAINING PROCESS OF THE PROPOSED MODEL

STEP	STSM	SSAM
Step 1	$\boldsymbol{\theta}^0_{STSM}$	-
Step 2	$\boldsymbol{\theta}^1_{STSM}$	-
Step 3	-	$\boldsymbol{\theta}^1_s$
Step 4	-	$\boldsymbol{\theta}^0_{SSAM}$
Step 5	-	$\boldsymbol{\theta}^2_s$ and $\boldsymbol{\theta}^1_{SSAM}$
Step 6	$\boldsymbol{\theta}^3_s$ and $\boldsymbol{\theta}^2_{STSM}$	-
Step 7	-	$\boldsymbol{\theta}^4_s$ and $\boldsymbol{\theta}^2_{SSAM}$
Step 8	Repeat steps (5)-(7)	-

with different input for feature learning. The training process is shown below. (We also provide the updating parameters for STSM and SSAM in each step in Table III.)

Step 1: The 3D convolutional/deconvolutional parameters $\boldsymbol{\theta}^0_{STSM}$ of STSM are initialized using random zero mean Gaussian distribution whose standard deviation is 0.01.

Step 2: We train STSM with three consecutive left-view video frames (L_{t-1}, L_t, L_{t+1}) to learn spatiotemporal features, and use Adam [59] to update the initial parameters $\boldsymbol{\theta}^0_{STSM}$ to obtain updated parameters $\boldsymbol{\theta}^1_{STSM}$.

Step 3: We initialize 3D ConvNet and DeconvNet parameters of SSAM with pre-trained $\boldsymbol{\theta}^1_{STSM}$ except the first two 3D convolutional and last 3D deconvolutional layers and denote these shared parameters as $\boldsymbol{\theta}^1_s$ (colored parts of TABLE I).

Step 4: We assign the remaining four layers of 3D ConvNet and DeconvNet in SSAM by random zero mean Gaussian distribution whose standard deviation is 0.01 and then initialize 2D ConvNet parameters of SSAM with pre-trained VGG16-net (remove the last three fully-connected layers). These parameters are denoted as $\boldsymbol{\theta}^0_{SSAM}$.

Step 5: The corresponding left-view and right-view video frames (L_t, R_t) are fed into 3D ConvNet of SSAM, and the left-view video frame (L_t) is used as input to 2D ConvNet of SSAM. Based on $\boldsymbol{\theta}^1_s$ (colored parts of TABLE I) and $\boldsymbol{\theta}^0_{SSAM}$, we utilize Adam [59] to optimize the loss function for updating the depth-aware parameters. Then we obtain $\boldsymbol{\theta}^2_s$ (colored parts of TABLE I) and $\boldsymbol{\theta}^1_{SSAM}$.

Step 6: We store the parameters $\boldsymbol{\theta}^1_{STSM}$ obtained in Step 2 as non-shared parameter set of STSM, and store the parameters $\boldsymbol{\theta}^2_s$ obtained in Step 5 as shared parameter set of STSM. With $\boldsymbol{\theta}^1_{STSM}$ and $\boldsymbol{\theta}^2_s$, we use Adam [59] to train STSM sub-model for collecting shared parameter set $\boldsymbol{\theta}^3_s$ (colored parts of TABLE I) and non-shared parameter set $\boldsymbol{\theta}^2_{STSM}$.

Step 7: We store the parameters $\boldsymbol{\theta}^1_{SSAM}$ obtained in Step 5 as non-shared parameter set of SSAM, and store the parameters $\boldsymbol{\theta}^3_s$ obtained in Step 6 as shared parameter set of SSAM. With $\boldsymbol{\theta}^1_{SSAM}$ and $\boldsymbol{\theta}^3_s$, we use Adam [59] to train SSAM sub-model for obtaining shared parameter set $\boldsymbol{\theta}^4_s$ (colored parts of TABLE I) and non-shared parameter set $\boldsymbol{\theta}^2_{SSAM}$.

Step 8: We repeat the above steps (5)-(7) to share parameters between STSM and SSAM. After three repeats, we obtain two robust sub-models and get the final parameters $\boldsymbol{\theta}_{STSM}$, $\boldsymbol{\theta}_{SSAM}$ and $\boldsymbol{\theta}_s$.

3) Evaluation Metrics: Similar with [12], [14], [41], and [61], we report the quantitative performance evaluation results with several popular metrics including: Pearson's Linear Correlation Coefficient (CC), Receiver Operating Characteristics (ROC) Curve, Area Under ROC Curve (AUC), Shuffled AUC (sAUC), Normalized Scanpath Saliency (NSS) and Kullback-Leibler Divergence (KLD). For these evaluation metrics, we downloaded the source code[1] for their implementations by MIT Saliency Benchmark.[2]

CC is used to quantify the correlation and dependence, demonstrating statistical relationship between the saliency maps and ground truth maps. CC is used to measure the degree of linear correlation between the saliency map and ground truth map, and it is commonly defined as follows:

$$CC(g,s) = \frac{cov(s,f)}{\sigma_s \sigma_g} \tag{8}$$

where $cov(s,g)$ denotes the covariance of saliency map s and ground truth map g; σ_s and σ_g stand for the standard deviation values of the saliency map s and ground truth map g, respectively. The range of CC values is [0,1]. Obviously, the lager CC value means the better performance of the saliency detection model. Specifically, it's a perfect correlation between predicted saliency map and human eye fixation map when the correlation score is close to 1.

As one of the most famous evaluation methods in the field of saliency detection, ROC curve and area under ROC curve (AUC) are also used for evaluating the performance of binary classifier with flexible thresholds. The pixel values of predicted saliency map above the threshold are classified as fixation points while the remain pixels are regarded as non-salient points. With the varied threshold, ROC curve can be plotted by false positive rate (FPR) and true positive rate (TPR), which are defined as follows:

$$FPR = \frac{M \cap \bar{G}}{\bar{G}} \tag{9}$$

$$TPR = \frac{M \cap G}{G} \tag{10}$$

where M represents the binary mask of the saliency map generated by a series of varying discrimination thresholds on the saliency map; G denotes the binary ground truth map while $\bar{G}$ stands for the reverse of G. The AUC and sAUC values are calculated by the area under ROC curve, which indicates the detection accuracy between predicted saliency maps and human eye fixations. In order to avoid the dramatic influence with AUC introduced by center bias effect, a shuffled AUC (sAUC) metric is widely used in standard evaluation of saliency detection. sAUC is proved to be more robust and credible than AUC [61]. Similar to CC, the lager AUC value also means the better performance of salient object detection model.

Furthermore, NSS attempts to collect the difference values between human fixation map and the saliency map with zero mean and unit standard deviation. It is also widely adopted to evaluate the performance of saliency detection methods. NSS can be defined as follows:

$$NSS(g,s) = \frac{1}{\sigma_s}(s(g_i,g_j) - \mu_s) \tag{11}$$

where s and g denote the saliency map and corresponding ground truth map; (g_i, g_j) is the pixel location of the ground truth map; μ_s and σ_s represent the mean value and the standard deviation of the saliency map, respectively. Typically, the higher NSS value means better performance of the saliency detection model.

KLD is also called relative entropy and it is used to measure the dissimilarity between the ground truth map and predicted saliency map. KLD can be calculated by the following for-

[1] https://github.com/cvzoya/saliency/tree/master/code_forMetrics
[2] http://saliency.mit.edu/

mula:

$$KLD(g,s) = \sum g(i,j) \log \frac{g(i,j)}{s(i,j)} \quad (12)$$

where s and g denote the saliency map and corresponding ground truth map; $g(i,j)$ and $s(i,j)$ denote the pixel value of the ground truth map and predicted saliency map at location (i,j), respectively. Generally, the performance of the saliency detection model is better with the smaller KLD value.

B. Performance Evaluation of Sub-Models in the Proposed Model

We show the experimental results of the proposed two sub-models of STSM and SSAM to demonstrate the advantages of the proposed model. We first provide some visual comparison samples from these two sub-models in Fig. 6, which demonstrates that the pre-trained STSM model can be used to provide rich spatiotemporal features for stereoscopic video sequences. As can be seen from Fig. 6, even though STSM and SSAM model can obtain relatively accurate saliency results, there are some wrongly detected fixation locations in the spatiotemporal saliency maps. As shown by the saliency samples in Fig. 6, for SSAM, the moving objects such as the woman and bus in the fourth and fifth columns can't be detected as salient region since SSAM doesn't consider the motion factor between consecutive video frames. For STSM, the objects such as white box and pedestrians in the third and fifth columns cannot be detected completely, since this model doesn't consider the effect of semantic and depth cues for stereoscopic video. Compared with these two sub-models, the overall proposed model by combing SSAM and STSM can obtain much better saliency results, as demonstrated by the second row (ground truth maps) and the final row in Fig. 6.

Meanwhile, we show the quantitative experimental results of the proposed two sub-models in Table IV with AUC, CC and NSS values. From Table IV, we can observe that the proposed model can obtain better saliency prediction results than STSM and SSAM, as demonstrated by the higher AUC, CC and NSS values of the proposed model than those from STSM and SSAM in Table IV. Additionally, we provide ROC curves of these models in Fig. 5, which also demonstrate the performance improvement of different sub-models for the proposed model. These comparison results demonstrate that both the learned features from STSM and SSAM can contribute much to the final saliency prediction results for stereoscopic video sequences.

We further use some existing 2D video saliency detection models including SAG2DV [28], LGGR2DV [39], and FCN2DV [62] for performance evaluation of the proposed STSM. From Table IV, we can observe that STSM can obtain better performance of saliency prediction than other existing ones, which can be demonstrated by higher AUC, sAUC, CC and NSS values (lower KLD value) of STSM than these of other existing models.

C. Comparison Experiments by Using Other Existing Models

In this experiment, we compare the proposed model against several existing saliency detection methods including

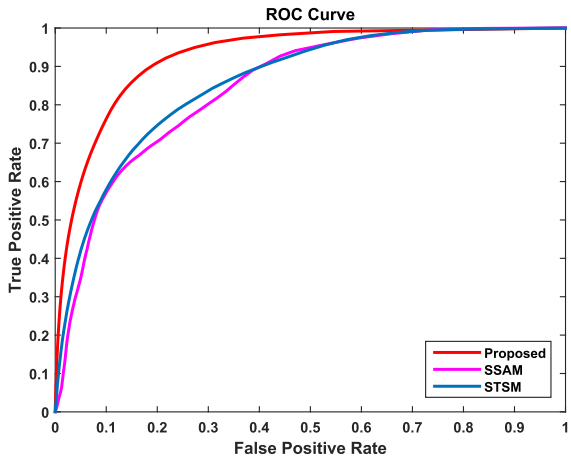


Fig. 5. ROC comparison of different models, including the SpatioTemporal Saliency Model (STSM), the Stereoscopic Saliency Aware Model (SSAM) and the proposed model.

TABLE IV

COMPARISON RESULTS OF DIFFERENT MODELS, INCLUDING STSM, SSAM, THE PROPOSED (THE PROPOSED MODEL), AS WELL AS THE 2D VIDEO SALIENCY MODELS (SAG2DV [28], LGGR2DV [39], FCN2DV [62])

Models	AUC	sAUC	CC	NSS	KLD
SSAM	0.8315	0.6885	0.6230	2.9304	0.8379
STSM	0.8859	0.6954	0.6531	3.0394	0.7967
The Proposed	**0.9355**	**0.8254**	**0.7222**	**3.5216**	**0.7835**
SAG2DV [28]	0.8465	0.6208	0.3438	1.4794	1.5843
LGGR2DV [39]	0.8587	0.6495	0.3785	1.6337	1.6427
FCN2DV [62]	0.8704	0.6697	0.5957	2.6436	1.4286

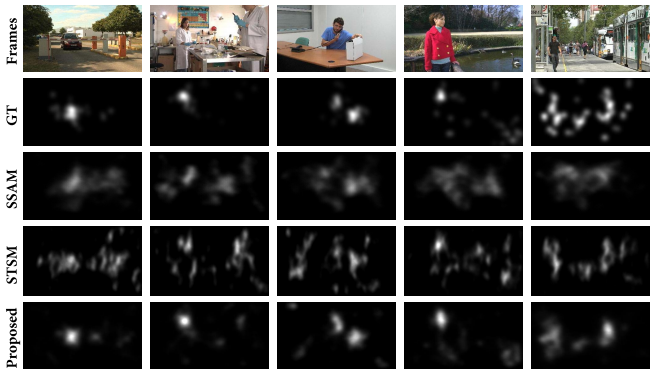


Fig. 6. Visual comparison samples from different models. First row to final row: original 3D video frames; the ground truth maps (GT); saliency maps from SSAM; saliency maps from STSM; final saliency maps from the proposed model.

Fang3DV [12], Ferreira3DV [20], Zhang3DV [42], SALICON [46], ML-NET [63], MDF [50], MultiTask [14], UHM [30]. Among these state-of-the-art approaches, Fang3DV [12], Ferreira3DV [20] and Zhang3DV [42] are designed for stereoscopic video saliency detection by hand-crafted low-level features; SALICON [46], ML-NET [63], MultiTask [14] are deep-based methods for image saliency detection; UHM [30] is a bottom-up eye fixation prediction method proposed for images; MDF [50] is a 2D video saliency detection model.

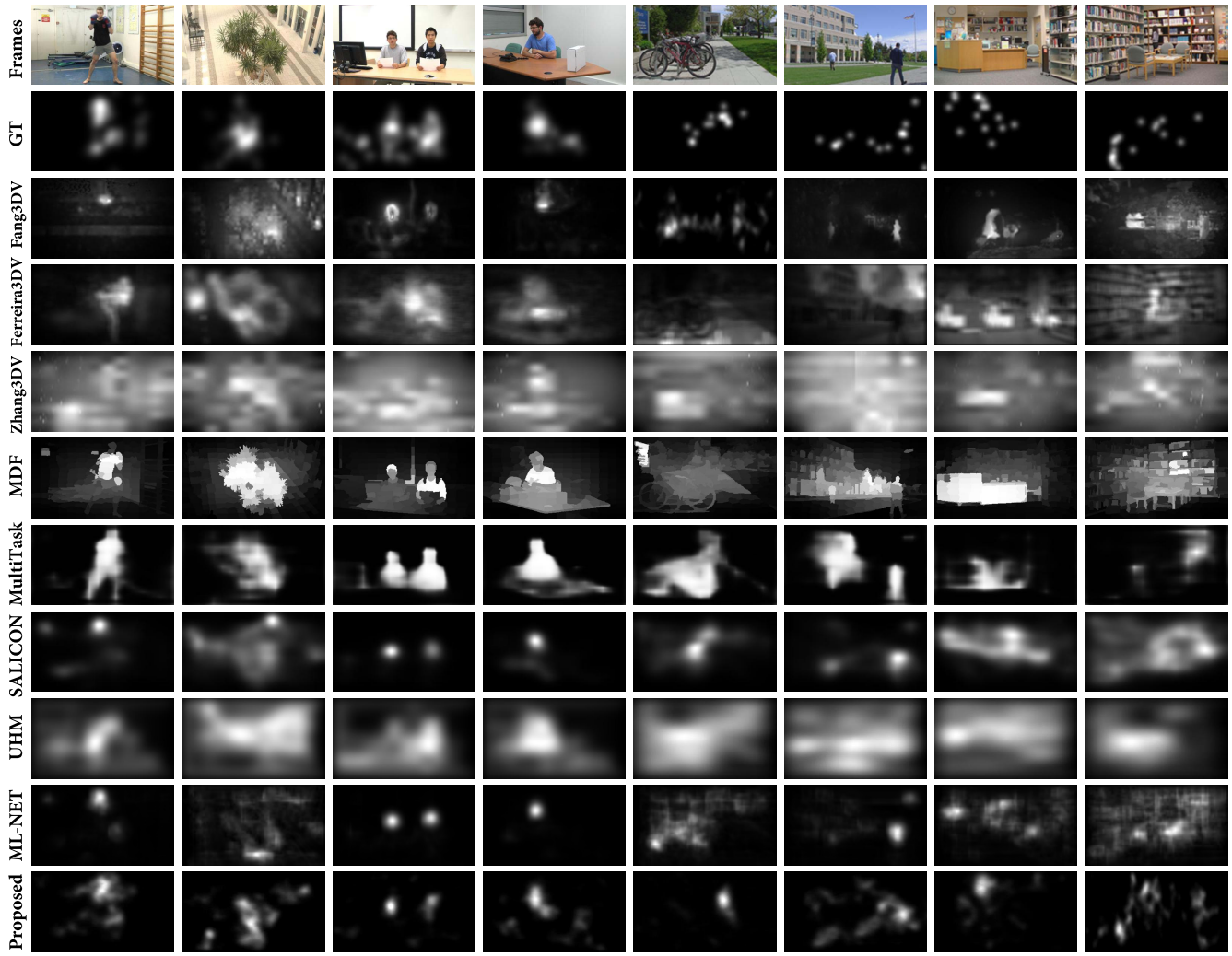


Fig. 7. Visual comparison samples from different saliency detection models on dataset FANG-Dataset [12] (first four columns) and DML-iTrack-3D [45] (final four columns). First row to final row: original 3D video frames, the ground truth maps (GT), saliency maps from Fang3DV [12], Ferreira3DV [20], Zhang3DV [42], MDF [50], MultiTask [14], SALICON [46], UHM [30], ML-NET [63], and the proposed model.

Currently, there is rarely deep neural network based video saliency detection models in the literature. And thus, we use several deep-based image saliency detection models (SALICON [46], ML-NET [63], MDF [50], MultiTask [14]) for comparison in this study. Note that we conduct this experiment on independent train and test set of FANG-Dataset [12] and DML-iTrack-3D [45].

Among these methods, Fang3DV [12] is a stereoscopic video saliency detection model published by ourselves, and we use the source code of our previous work to calculate saliency results. Since there is no public source code for Ferreira3DV [20] and Zhang3DV [42], we implement these two models based on their papers. For these implementations, we obtain the similar experiment results with those presented in their original papers [20], [42]. The authors of some existing saliency detection models including SALICON [46], ML-NET [63], MDF [50], MultiTask [14], UHM [30] publish their source code in the authors' Github or Homepages, and we download their source code to compute experimental results.

We provide some visual comparison samples from different saliency detection models in Fig. 7 on FANG-Dataset [12] (first five columns) and DML-iTrack-3D [45] (final five columns). It can be seen that the saliency maps obtained from other existing methods contain some noises, as shown by the fact that some non-salient regions are detected as the salient regions in some saliency maps generated from existing methods. For example, as shown in the forth and eighth columns in Fig. 7, we can see that the saliency maps from MDF [50] and Fang3DV [12] mainly detect the high-contrast edge and compact regions as salient (these regions belong to non-salient parts). The reason is that these saliency detection models of MDF [50], UHM [30] and Fang3DV [12] mainly consider the local/global contrast and compactness for saliency calculation. Although the saliency maps of Ferreira3DV [20] and MultiTask [14] can obtain relatively better results than Zhang3DV [42]. However, as shown in the fourth and seventh rows in Fig. 7, the saliency maps from Ferreira3DV [20] and MultiTask [14] include many wrongly detected salient

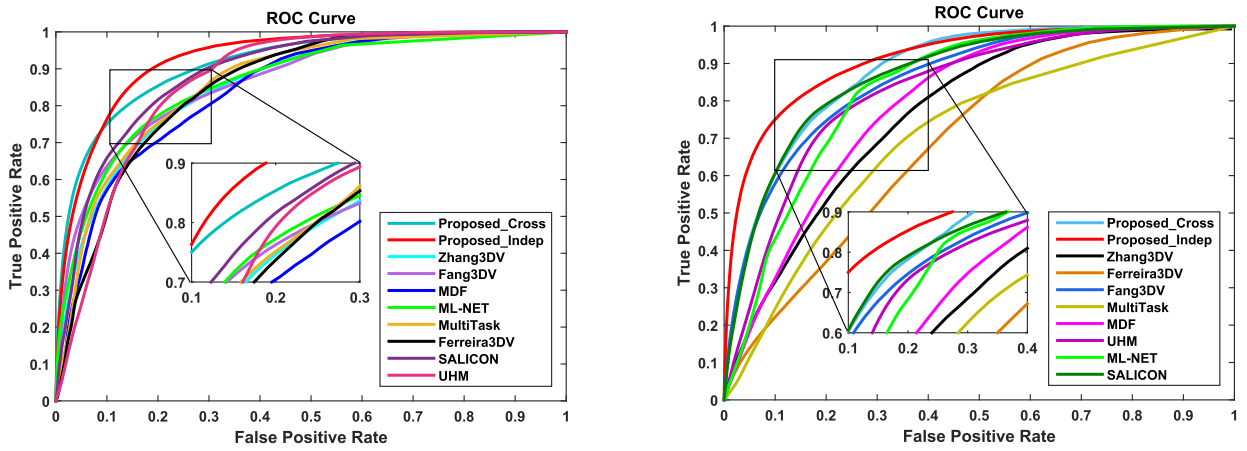


Fig. 8. ROC comparison of different saliency models on dataset FANG-Dataset [12] (left) and DML-iTrack-3D [45] (right). The results are collected from methods including Fang3DV [12], Ferreira3DV [20], Zhang3DV [42], SALICON [46], ML-NET [63], MDF [50], MultiTask [14], UHM [30], the proposed model with independent dataset validation (Proposed-Indep), and the proposed model with cross dataset validation (Proposed-Cross).

TABLE V

COMPARISON OF DIFFERENT SALIENCY DETECTION MODELS ON FANG-DATASET [12] AND DML-ITRACK-3D [45]. THE RESULTS ARE COLLECTED FROM METHODS INCLUDING FANG3DV [12], FERREIRA3DV [20], ZHANG3DV [42], SALICON [46], ML-NET [63], MDF [50], MULTITASK [14], UHM [30], THE PROPOSED MODEL WITH INDEPENDENT DATASET VALIDATION (PROPOSED-INDEP) AND THE PROPOSED MODEL WITH CROSS DATASET VALIDATION (PROPOSED-CROSS). **TIME** DENOTES THE TIME COST OF SALIENCY PREDICTION PER FRAME

Datasets	Models	AUC	sAUC	CC	NSS	KLD
FANG-Dataset [12]	Fang3DV [12]	0.9008	0.7434	0.2979	1.3274	1.4062
	Ferreira3DV [20]	0.8592	0.7003	0.3215	1.1949	1.6181
	Zhang3DV [42]	0.8490	0.6954	0.3229	1.8503	1.5925
	SALICON [46]	0.9061	0.7758	0.6188	3.0709	1.4310
	ML-NET [63]	0.8732	0.7042	0.4938	2.4234	1.6295
	MDF [50]	0.8398	0.6998	0.3367	1.5943	1.5473
	MultiTask [14]	0.8479	0.7103	0.3051	1.6141	1.7377
	UHM [30]	0.8630	0.6961	0.3902	1.5593	1.5238
	Proposed-Indep	**0.9355**	**0.8245**	**0.7222**	**3.5216**	**0.7835**
	Proposed-Cross	**0.9109**	**0.8022**	**0.7093**	**3.4392**	**0.7581**
DML-iTrack-3D [45]	Fang3DV [12]	0.7821	0.5595	0.2858	1.3812	1.3597
	Ferreira3DV [20]	0.6528	0.5418	0.2188	0.8242	1.3085
	Zhang3DV [42]	0.6905	0.5243	0.2442	1.0418	1.3681
	SALICON [46]	0.8347	0.6305	0.5964	2.2192	1.6731
	ML-NET [63]	0.7507	0.5374	0.4833	2.1019	1.5373
	MDF [50]	0.7212	0.5489	0.3520	1.1639	1.4986
	MultiTask [14]	0.6863	0.5261	0.2347	1.0418	1.5967
	UHM [30]	0.7476	0.5595	0.3895	1.2924	1.1090
	Proposed-Indep	**0.8575**	**0.6825**	**0.6440**	**2.8913**	**0.6954**
	Proposed-Cross	**0.8385**	**0.6339**	**0.5994**	**2.5199**	**0.6827**

regions. Compared with other existing methods, Fang3DV [12], SALICON [46] and ML-NET [63] can obtain relatively better saliency prediction results, as shown in the third, eighth and tenth rows of Fig. 7. However, there are still some background regions detected as salient in the saliency results by Fang3DV [12], SALICON [46] and ML-NET [63], as shown in the second and eighth columns in Fig. 7. On the contrary, the proposed saliency detection model can obtain more accurate saliency prediction results for stereoscopic video sequences thanks to the used STSM and SSAM for the semantic, spatiotemporal and depth aware feature extraction, as shown in the last column of Fig. 7.

Meanwhile, we show the quantitative experimental results in Table V on FANG-Dataset [12] and DML-iTrack-3D [45], where CC, AUC and NSS values are collected from the average of test set, including 20 video sequences in FANG-Dataset [12] dataset and 12 video sequences in DML-iTrack-3D [45] dataset, respectively. We denote the proposed model with independent dataset validation as Proposed-Indep in these tables. As shown in Table V, MDF [50] and UHM [30] obtain the lowest performance in 3D video saliency prediction, while Fang3DV and Ferreira3DV [20] can keep relatively high performance. This demonstrates that it is effective to consider the depth and spatiotemporal information for 3D video saliency detection. In Table V, we can observe that Fang3DV [12] can obtain better performance than other existing related methods, which further demonstrates that Gestalt based uncertainty weighting fusion method can obtained relatively good results for 3D video saliency detection. From Table V, we can observe that the proposed method can obtain better stereoscopic video saliency prediction performance than other related ones, as shown by the highest CC, AUC and NSS

TABLE VI

THE IMPLEMENTATION DETAILS AND TESTING CONDITIONS FOR THE COMPARED MODELS. **TIME** DENOTES THE TIME COST OF SALIENCY PREDICTION PER FRAME

Method	Code	OS	GPU/CPU	Time ($\times 10^{-2}$ sec.) FANG-Dataset [12]	Time ($\times 10^{-2}$ sec.) DML-iTrack-3D [45]
Fang3DV [12]	Matlab	Windows	CPU	86.36	45.74
Ferreira3DV [20]	Matlab	Windows	CPU	52.94	29.68
Zhang3DV [42]	Matlab	Windows	CPU	35.48	12.52
SALICON [46]	Python+Caffe	Ubuntu	GPU	30.45	8.32
ML-NET [63]	Python+Theano+Keras	Ubuntu	GPU	53.76	30.95
MDF [50]	Matlab+Caffe	Ubuntu	GPU	47.31	24.39
MultiTask [14]	Python+Caffe	Ubuntu	GPU	16.83	7.73
UHM [30]	Matlab	Windows	CPU	18.38	8.96
Proposed-Indep	Python+Tensorflow	Ubuntu	GPU	4.42	2.52
Proposed-Cross	Python+Tensorflow	Ubuntu	GPU	4.48	2.59

values among the compared models. We also provide the ROC curves of all these methods in Fig. 8 on FANG-Dataset [12] (left) and DML-iTrack-3D [45] (right), which also demonstrate the better results of the proposed model than other existing ones.

D. Cross Dataset Validation

To better demonstrate the performance of the proposed model, we conduct a cross dataset validation experiment on the training and testing sets of FANG-Dataset [12] and DML-iTrack-3D [45]. More specifically, we first use the training set of FANG-Dataset [12] to train the two sub-models, and then adopt the testing set of DML-iTrack-3D [45] to evaluate the performance of the proposed model. Similarly, another experiment is conducted by using the training set of DML-iTrack-3D [45] to train the proposed model, and predict the saliency results by using the testing set of FANG-Dataset.

We show the quantitative experimental results of cross dataset validation of the proposed model (denoted as Proposed-Cross) in Table V on FANG-Dataset [12] and DML-iTrack-3D [45]. We can observe that the proposed method can still obtain better performance than other existing ones, which demonstrates the robustness of the proposed model. We also provide the ROC curves of Proposed-Cross in Fig. 8 on FANG-Dataset [12] (left) and DML-iTrack-3D [45] (right), which also demonstrate better performance of the proposed model than other existing ones. Please note that we provide the independent dataset validation results for other existing learning-based models in Fig. 8 and Table V.

V. CONCLUSION

In this paper, a novel stereoscopic video saliency detection approach with 3D convolutional neural networks is proposed to effectively learn semantic, spatiotemporal and depth features for 3D video sequences. The proposed model mainly includes two sub-models: STSM and SSAM. We first pre-train STSM, which aims to provide plenty spatiotemporal features between consecutive video frames for SSAM. SSAM consists of three components: 3D convolutional network, 2D convolutional network and 3D deconvolutional network. After restoring the pre-trained STSM features for SSAM, we fine-tune SSAM with the input left and right video frames to extract rich depth cues for 3D video sequences. Experimental results have shown that there is great potential to build stereoscopic video saliency detection model with 3D convolutional operation for effectively learning semantic, depth and spatiotemporal features instead of time-consuming hand-crafted features.

REFERENCES

[1] U. Rutishauser, D. Walther, C. Koch, and P. Perona, "Is bottom-up attention useful for object recognition?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun./Jul. 2004, pp. 37–44.

[2] Y. Fang, J. Wang, Y. Yuan, J. Lei, W. Lin, and P. L. Callet, "Saliency-based stereoscopic image retargeting," *Inf. Sci.*, vol. 372, pp. 347–358, Dec. 2016.

[3] H. Hadizadeh and I. V. Bajić, "Saliency-aware video compression," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 19–33, Jan. 2014.

[4] C. Guo and L. Zhang, "A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 185–198, 2009.

[5] P. Zhang, T. Zhuo, W. Huang, K. Chen, and M. Kankanhalli, "Online object tracking based on CNN with spatial-temporal saliency guided sampling," *Neurocomputing*, vol. 257, pp. 115–127, Sep. 2017.

[6] X. Bai, Y. Fang, W. Lin, L. Wang, and B.-F. Ju, "Saliency-based defect detection in industrial images by using phase spectrum," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2135–2145, Nov. 2014.

[7] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 18–32, Jan. 2014.

[8] R. Zhao, W. Ouyang, and X. Wang, "Unsupervised salience learning for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3586–3593.

[9] 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.

[10] J. Zhang, M. Wang, S. Zhang, X. Li, and X. Wu, "Spatiochromatic context modeling for color saliency analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1177–1189, Jun. 2016.

[11] H. Kim, S. Lee, and A. C. Bovik, "Saliency prediction on stereoscopic videos," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1476–1490, Apr. 2014.

[12] Y. Fang, C. Zhang, J. Li, J. Lei, M. P. Da Silva, and P. Le Callet, "Visual attention modeling for stereoscopic video: A benchmark and computational model," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4684–4696, Oct. 2017.

[13] J. Zhang and S. Sclaroff, "Exploiting surroundedness for saliency detection: A Boolean map approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 5, pp. 889–902, May 2016.

[14] X. Li *et al.*, "DeepSaliency: Multi-task deep neural network model for salient object detection," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3919–3930, Aug. 2016. [Online]. Available: https://github.com/zlmzju/DeepSaliency

[15] N. Tong, H. Lu, Y. Zhang, and X. Ruan, "Salient object detection via global and local cues," *Pattern Recognit.*, vol. 48, no. 10, pp. 3258–3267, Oct. 2015.

[16] R. Liu, J. Cao, Z. Lin, and S. Shan, "Adaptive partial differential equation learning for visual saliency detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3866–3873.

[17] C. Chen, S. Li, Y. Wang, H. Qin, and A. Hao, "Video saliency detection via spatial-temporal fusion and low-rank coherency diffusion," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3156–3170, Jul. 2017.

[18] S. Wang, M. Wang, S. Yang, and K. Zhang, "Salient region detection via discriminative dictionary learning and joint Bayesian inference," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 5, pp. 1116–1129, May 2017.

[19] J. Wang, M. P. da Silva, P. Le Callet, and V. Ricordel, "A computational model of stereoscopic 3D visual saliency," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2151–2165, Jun. 2013.

[20] L. Ferreira, L. A. da Silva Cruz, and P. Assuncao, "A method to compute saliency regions in 3D video based on fusion of feature maps," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jun./Jul. 2015, pp. 1–6.

[21] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.

[22] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[23] G. Li and Y. Yu, "Deep contrast learning for salient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2016, pp. 478–487.

[24] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1265–1274.

[25] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3183–3192.

[26] Y. Jia and M. Han, "Category-independent object-level saliency detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Dec. 2013, pp. 1761–1768.

[27] R. Margolin, A. Tal, and L. Zelnik-Manor, "What makes a patch distinct?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1139–1146.

[28] W. Wang, J. Shen, and F. Porikli, "Saliency-aware geodesic video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3395–3402. [Online]. Available: https://github.com/shenjianbing/Saliency-Aware-Video-Object-Segmentation

[29] J. Lei *et al.*, "A universal framework for salient object detection," *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1783–1795, Sep. 2016.

[30] H. R. Tavakoli and J. Laaksonen, "Bottom-up fixation prediction using unsupervised hierarchical models," in *Proc. Asian Conf. Comput. Vis.*, 2016, pp. 287–302. [Online]. Available: https://github.com/hrtavakoli/UHM

[31] Z. Liu, J. Li, L. Ye, G. Sun, and L. Shen, "Saliency detection for unconstrained videos using superpixel-level graph and spatiotemporal propagation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 12, pp. 2527–2542, Dec. 2017.

[32] T. Xi, W. Zhao, H. Wang, and W. Lin, "Salient object detection with spatiotemporal background priors for video," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3425–3436, Jul. 2017.

[33] D. Tao, J. Cheng, M. Song, and X. Lin, "Manifold ranking-based matrix factorization for saliency detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1122–1134, Jun. 2016.

[34] K. Fu, I. Y.-H. Gu, and J. Yang, "Saliency detection by fully learning a continuous conditional random field," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1531–1544, Jul. 2017.

[35] Ç. Aytekin, H. Possegger, T. Mauthner, S. Kiranyaz, H. Bischof, and M. Gabbouj, "Spatiotemporal saliency estimation by spectral foreground detection," *IEEE Trans. Multimedia*, vol. 20, no. 1, pp. 82–95, Jan. 2017.

[36] Q. Wang, W. Zheng, and R. Piramuthu, "GraB: Visual saliency via novel graph model and background priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 535–543.

[37] Y. Fang, Z. Wang, W. Lin, and Z. Fang, "Video saliency incorporating spatiotemporal cues and uncertainty weighting," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 3910–3921, Sep. 2014.

[38] J. Li, Y. Tian, T. Huang, and W. Gao, "Probabilistic multi-task learning for visual saliency estimation in video," *Int. J. Comput. Vis.*, vol. 90, no. 2, pp. 150–165, 2010.

[39] W. Wang, J. Shen, and L. Shao, "Consistent video saliency using local gradient flow optimization and global refinement," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4185–4196, Nov. 2015. [Online]. Available: https://github.com/shenjianbing/Consistent-video-saliency

[40] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3166–3173.

[41] Y. Fang, J. Wang, M. Narwaria, P. Le Callet, and W. Lin, "Saliency detection for stereoscopic images," *IEEE Trans. Image Process.*, vol. 23, no. 6, pp. 2625–2636, Jun. 2014.

[42] Y. Zhang, G. Jiang, M. Yu, and K. Chen, "Stereoscopic visual attention model for 3D video," in *Proc. Int. Conf. Adv. Multimedia Modeling*, 2010, pp. 314–324.

[43] Y. Park, B. Lee, W.-S. Cheong, and N. Hur, "Stereoscopic 3D visual attention model considering comfortable viewing," in *Proc. Image Process.*, Jul. 2012, pp. 1–5.

[44] N. Liu, J. Han, T. Liu, and X. Li, "Learning to predict eye fixations via multiresolution convolutional neural networks," *IEEE Trans. Image Process.*, vol. 29, no. 2, pp. 392–404, Feb. 2018.

[45] A. Banitalebi-Dehkordi, M. T. Pourazad, and P. Nasiopoulos, "A learning-based visual saliency prediction model for stereoscopic 3D video (LBVS-3D)," *Multimedia Tools Appl.*, vol. 76, no. 22, pp. 23859–23890, Nov. 2017.

[46] X. Huang, C. Shen, X. Boix, and Q. Zhao, "SALICON: Reducing the semantic gap in saliency prediction by adapting deep neural networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 262–270.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[48] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 1, 2014, pp. 568–576.

[49] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[50] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5455–5463. [Online]. Available: https://sites.google.com/site/ligb86/mdfsaliency/

[51] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[52] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2016, pp. 1520–1528.

[53] Z. Huang, G. Cheng, H. Wang, H. Li, L. Shi, and C. Pan, "Building extraction from multi-source remote sensing images via deep deconvolution neural networks," in *Proc. Geosci. Remote Sens. Symp.*, Jul. 2016, pp. 1835–1838.

[54] A. Fakhry, T. Zeng, and S. Ji, "Residual deconvolutional networks for brain electron microscopy image segmentation," *IEEE Trans. Med. Imag.*, vol. 36, no. 2, pp. 447–456, Feb. 2017.

[55] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2014, pp. 4489–4497.

[56] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1–10.

[57] S. Ioffe and C. Szegedy. (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift." [Online]. Available: https://arxiv.org/abs/1502.03167

[58] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014.

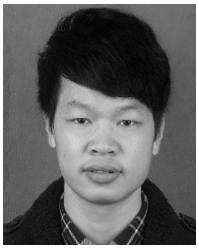
[60] M. Abadi *et al.* (2015). "TensorFlow: Large-scale machine learning on heterogeneous distributed systems." [Online]. Available: https://arxiv.org/abs/1603.04467

[61] A. Borji, D. N. Sihite, and L. Itti, "Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 55–69, Jan. 2013.

[62] W. Wang, J. Shen, and L. Shao, "Video salient object detection via fully convolutional networks," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 38–49, Jan. 2018. [Online]. Available: https://github.com/wenguanwang/ViSalientObject

[63] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "A deep multi-level network for saliency prediction," in *Proc. Int. Conf. Pattern Recognit.*, Dec. 2016, pp. 3488–3493. [Online]. Available: https://github.com/marcellacornia/mlnet

Yuming Fang (M'13–SM'17) received the B.E. degree from Sichuan University, Chengdu, China, the M.S. degree from the Beijing University of Technology, Beijing, China, and the Ph.D. degree from Nanyang Technological University, Singapore. He is currently a Professor with the School of Information Management, Jiangxi University of Finance and Economics, Nanchang, China. His research interests include visual attention modeling, visual quality assessment, image retargeting, computer vision, and 3D image/video processing. He serves as an Associate Editor for IEEE ACCESS. He is on the editorial board of *Signal Processing: Image Communication*.



Guanqun Ding is currently a graduate student with the School of Information Management, Jiangxi University of Finance and Economics, Nanchang, China. His research interests include saliency detection, object detection, and computer vision.

Jia Li (M'12–SM'15) received the B.E. degree from Tsinghua University in 2005 and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, in 2011. He was with Nanyang Technological University, with Peking University, and also with Shanda Innovations. He is currently an Associate Professor with the School of Computer Science and Engineering, Beihang University, Beijing, China. He has authored or co-authored over 50 technical articles in refereed journals and conferences such as TPAMI, TIP, IJCV, ICCV, and CVPR. His research interests include computer vision and multimedia big data, especially the cognitive vision toward evolvable algorithms and models.

Zhijun Fang (SM'13) received the Ph.D. degree from Shanghai Jiaotong University, Shanghai, China. He is currently a Professor and Dean with the School of Electronic and Electrical Engineering, Shanghai University of Engineering Science. His current research interests include image processing, video coding, and pattern recognition. He was the General Chair of the Joint Conference on Harmonious Human Machine Environment 2013 and the International Symposium on Information Technology Convergence 2017.