RGBD Salient Object Detection via Deep Fusion

Liangqiong Qu, Shengfeng He*, Jiawei Zhang, Jiandong Tian, Yandong Tang, and Qingxiong Yang
http://www.cs.cityu.edu.hk/~jiawzhang8/saliency/TIP_saliency.htm

***Abstract*—Numerous efforts have been made to design various low-level saliency cues for RGBD saliency detection, such as color and depth contrast features as well as background and color compactness priors. However, how these low-level saliency cues interact with each other and how they can be effectively incorporated to generate a master saliency map remain challenging problems. In this paper, we design a new convolutional neural network (CNN) to automatically learn the interaction mechanism for RGBD salient object detection. In contrast to existing works, in which raw image pixels are fed directly to the CNN, the proposed method takes advantage of the knowledge obtained in traditional saliency detection by adopting various flexible and interpretable saliency feature vectors as inputs. This guides the CNN to learn a combination of existing features to predict saliency more effectively, which presents a less complex problem than operating on the pixels directly. We then integrate a superpixel-based Laplacian propagation framework with the trained CNN to extract a spatially consistent saliency map by exploiting the intrinsic structure of the input image. Extensive quantitative and qualitative experimental evaluations on three datasets demonstrate that the proposed method consistently outperforms state-of-the-art methods.**

***Index Terms*—RGBD saliency detection, Convolutional neural network, Laplacian propagation.**

I. Introduction

SALIENCY detection, which is the prediction of where a human being will look in an image, has attracted considerable research interest in recent years. It serves as an important pre-processing step for many tasks, such as image classification, image retargeting and object recognition [1], [2], [3], [4]. Unlike RGB saliency detection, which has received a great deal of research attention, there have been few explorations of the RGBD case. Recently emerging sensing technologies, such as Time-of-Flight sensors and the Microsoft Kinect, provide excellent capability and flexibility in capturing RGBD images [5], [6]. Detecting RGBD saliency has become essential for many applications, such as 3D content surveillance, retrieval, and image recognition [7], [8], [9]. In this paper, we focus on how to integrate the RGB information with the additional depth information for RGBD saliency detection [10], [11].

Depending on the definition of saliency used, saliency detection methods can be classified into two categories: top-down approaches and bottom-up approaches [17], [18]. Top-down saliency detection is a task-dependent process that incorporates high-level features to locate salient objects. By contrast, a bottom-up approach is task-free and utilizes low-level features that are biologically motivated to estimate salient regions. Most existing bottom-up saliency detection methods focus on the design of various low-level cues to represent salient objects. The saliency maps generated based on these low-level features are then fused into a master saliency map. Because human attention is preferentially attracted by high-contrast regions and their surroundings, contrast-based features (such as color, edge orientation or texture contrasts) play a crucial role in the extraction of salient objects. Background [19] and color compactness priors [20] consider salient objects from different perspectives. The former leverages the fact that most salient objects are located far from the image boundaries, whereas the latter utilizes the color compactness of salient objects. In addition to RGB information, depth has been shown to be a practical cue for extracting saliency [21], [22], [23], [24]. Most existing approaches for 3D saliency detection either use the depth information to weight the RGB saliency map [21], [24] or treat the depth cues as an independent image channel [22], [23].

Despite the demonstrated success of these features, no single feature is effective for all scenarios as they define saliency from different perspectives. The combination of different features might be a good solution. However, manually designing an interaction mechanism for integrating inherently different saliency features is a challenging problem. For example, linearly combining the saliency maps produced by these features cannot guarantee improved results (as shown in Figure 1h). Several other more complex combination algorithms have been proposed in [25], [26], [16], [27], [12], [14]. Qin *et al.* [16] proposed a Multi-layer Cellular Automata method (MCA, a Bayesian framework) to merge different saliency maps by exploiting the advantages of each saliency detection method. Recently, several heuristic algorithms have been designed for combining 2D-related saliency maps with depth-induced saliency maps [12], [14]. However, because they are restricted by the computed saliency values, these saliency map combination methods are not able to correct incorrectly estimated salient regions. For example, in Figure 1, heuristic-based algorithms (Figure 1d to 1g) cannot detect the salient object correctly. When these saliency maps are used for further fusion, neither simple linear fusion (Figure 1h) nor MCA integration (Figure 1i) is subsequently able to recover the salient object. We wonder whether a good integration method could be developed to address this problem by further adopting

* Corresponding author: Shengfeng He.

L. Qu and J. Zhang are with the Department of Computer Science, City University of Hong Kong, Hong Kong. L. Qu is also with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, 110016, and the University of Chinese Academy of Sciences, Beijing, China, 100049. (E-mail: quliangqiong@sia.cn; jiawzhang8-c@my.cityu.edu.hk).

S. He is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China, 510006. (E-mail: hesfe@scut.edu.cn).

J. Tian and Y. Tang are with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, 110016 (E-mail: tianjd@sia.cn; ytang@sia.cn).

Q. Yang is with Didi research institute, Hangzhou, China.

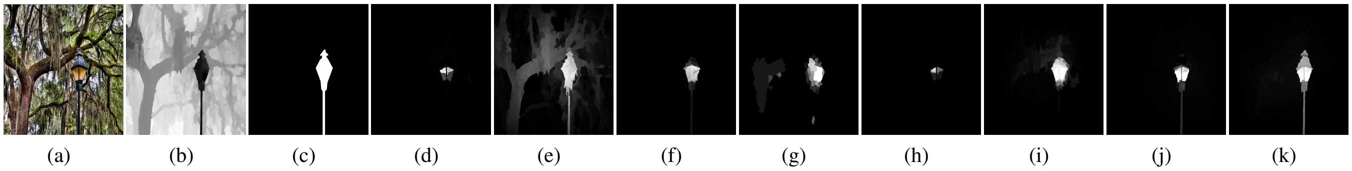


Fig. 1: Example illustrating the problems with various saliency map merging methods. (a) Original RGB image. (b) Original depth image. (c) Ground-truth saliency map. (d) Saliency map generated by LMH [12]. (e) Saliency map generated by ACSD [13]. (f) Saliency map generated by GP [14]. (g) Saliency map generated by LBE [15]. (h) to (j) show the saliency map integration results for (d), (e), (f), and (g). (h) Linear combination (i.e., averaging). (i) MCA integration [16]. (j) CNN-based fusion. (k) Saliency map generated via the proposed hyper-feature fusion method.

the Convolutional Neural Network technique to train a saliency map integration model. The resulting image shown in Figure 1j indicates that saliency map integration is strongly influenced by the quality of the input saliency maps. Based on these observations, we take a step back and consider rawer and more flexible saliency features.

In this paper, we propose a deep fusion framework to automatically learn the interaction mechanism between RGB and depth-induced saliency features for RGBD saliency detection. The proposed method takes advantage of the representation learning power of CNNs to extract hyper-features by fusing different hand-designed saliency features for the detection of salient objects (as shown in Figure 1k). We first compute several feature vectors from the original RGBD image, which include local and global contrasts, a background prior, and a color compactness prior. We then propose a CNN architecture to incorporate these regional feature vectors into more representative and unified features. Compared with the raw image pixels, these extracted saliency features are well designed and can more effectively guide the training of the CNN toward saliency optimization. Because the resulting saliency map may suffer from local inconsistencies and noisy false positives, we further integrate a superpixel-based Laplacian propagation framework with the proposed CNN. This approach propagates high-confidence saliency to other regions by considering color and depth consistency and the intrinsic structure of the input image [28]; thus, it is possible to remove noisy values and produce a smooth saliency map. The Laplacian propagation problem is solved with rapid convergence by means of the conjugate gradient method with a preconditioner. Experimental evaluations demonstrate that once our deep fusion framework is properly trained, it generalizes well to different datasets without any additional training and outperforms state-of-the-art approaches.

The main contributions of this paper are summarized as follows.

1. We propose a simple yet effective deep learning model to learn the interaction mechanism of RGB and depth-induced saliency features for RGBD saliency detection. In contrast to existing deep networks, which are fed with the raw image pixels, this deep model method takes various flexible and interpretable saliency feature vectors as inputs, which can more effectively guide the training of the CNN toward saliency optimization.

2. We adopt a superpixel-based Laplacian propagation method to refine the resulting saliency map and solve it with fast convergence. Unlike the CRF model, our Laplacian propagation framework not only considers spatial consistency but also exploits the intrinsic structure of the input image [28]. Extensive experiments further demonstrate that the proposed Laplacian propagation technique is able to refine the saliency maps generated by existing approaches and thus can be widely adopted as a post-processing step.

3. We investigate the limitations of saliency map integration and demonstrate that superior performance can be achieved through the fusion of simple features.

II. Related work

In this section, we present a brief survey and review of RGB and RGBD saliency detection methods. Comprehensive literature reviews on these saliency detection methods can be found in [29], [12].

RGB saliency detection: As suggested by studies in the field of cognitive science [30], bottom-up saliency is driven by low-level stimulus features. This concept is also used in computer vision to model saliency. Contrast-based cues, especially color contrast, are the most widely adopted features in previous works. These contrast-based methods can be roughly classified into two categories: local and global approaches. Local methods calculate the color, edge orientation or texture contrast of a pixel/region with respect to a local window to measure its saliency [31], [32]. In [31], the authors propose an information-theoretic way to estimate the contrast between a center pixel and its surrounding distribution for salient object detection. Xie *et al.* [32] propose a Bayesian saliency model for estimating the saliency map based on the center-surroundings principle. However, based only on local contrast, these methods may overemphasize the boundaries of salient objects [20] and be sensitive to high-frequency content [33]. In contrast to local approaches, global approaches evaluate salient regions by estimating the contrast over the entire image. Achanta *et al.* [34] model saliency by computing color differences with respect to the mean image color. Cheng *et al.* [35] propose a histogram-based global contrast saliency method by considering spatially weighted coherence. Although these global methods achieve superior performance, they may be misled when the background shares a similar color with a salient object. Background and color compactness priors have been proposed as complements to contrast-based methods [19], [36], [20]. These methods are built on strong assumptions, which may be invalid in some scenarios.

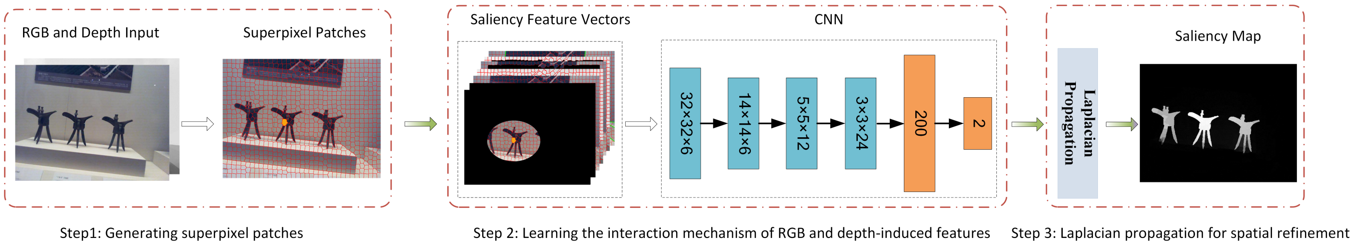

RGB and Depth Input
Superpixel Patches
Saliency Feature Vectors
CNN
32×32×6
14×14×6
5×5×12
3×3×24
200
2
Laplacian Propagation
Saliency Map
Step1: Generating superpixel patches
Step 2: Learning the interaction mechanism of RGB and depth-induced features
Step 3: Laplacian propagation for spatial refinement

Fig. 2: The pipeline of the proposed method. Our method consists of three modules. First, it generates different RGB and depth-based saliency feature vectors from the RGBD input image. These generated saliency feature vectors are then fed to the CNN. The CNN takes the saliency feature vectors of a superpixel as input (reshaped to dimensions of $32 \times 32 \times 6$) and outputs the saliency confidence value (the probability that this superpixel belongs to a salient region). Finally, Laplacian propagation is performed on the resulting probabilities to extract the final spatially consistent saliency map.

Because each type of feature has different strengths, some works focus on designing the integration mechanism for different saliency features [25], [36], [26], [37]. Liu *et al.* [25] use CRF to integrate three different features from both the local and global points of view. Yan *et al.* [26] propose a hierarchical framework for integrating saliency maps on different scales, which can handle small high-contrast regions well. Unlike these methods, which directly combine the saliency maps obtained from different saliency cues, the proposed method records low-level saliency features in vector form and jointly learns their interaction mechanism via a CNN to generate hyper-features.

As in the proposed method, CNNs have been adopted in several other works to extract hierarchical feature representations for detecting salient regions [38], [39], [40], [41], [42], [43]. Some of these approaches [39], [40], [42] mainly calculate hierarchical feature representations for saliency detection in a multi-scale fashion. By contrast, others [44], [45], [46] employ deep networks for salient object detection in a fully convolutional architecture (i.e., FCN [47]). These FCN methods treat the entire image as input and directly output the global saliency map. To better capture the object boundaries, various post-processing approaches are used to refine the outputs of the FCNs, such as CRF [48], [49], regularized nonlinear regression [46], and edge-aware erosion [45]. In contrast to most of these deep network methods, which take raw image pixels as input, the proposed method is aimed at designing a CNN framework to learn the interaction mechanism among different saliency cues.

RGBD saliency detection: Compared with RGB saliency detection, RGBD saliency has received less research attention [21], [24], [23], [22], [50]. The approach proposed by Maki *et al.* [21] is an early computational model of depth-based attention that measures disparity, flow and motion. Similar to color contrast approaches, Zhang *et al.* design a stereoscopic visual attention algorithm based on depth and motion contrast for 3D video [24]. Desingh *et al.* [23] estimate salient regions by fusing saliency maps independently produced based on appearance and depth cues. These methods either use the depth information to weight the RGB saliency map [21], [24] or consider the depth map as an independent image channel for saliency detection [22], [23]. By contrast, Peng *et al.* [12] propose a multi-stage RGBD model that combines both depth and appearance cues to detect saliency. Ren *et al.* [14] directly integrate a normalized depth prior and a surface orientation prior with RGB saliency cues for RGBD saliency detection. Instead of a depth prior, Feng *et al.* [15] introduce a novel local background enclosure feature to directly measure salient structures from depth information and then reweight this feature using depth and spatial priors. These methods combine a depth-induced saliency map with an RGB saliency map, either directly [13], [14] or in a hierarchical way, to calculate the final RGBD saliency map [12]. However, such saliency map level integration is not optimal as it is restricted by the determined saliency values. By contrast, we incorporate different saliency cues and fuse them via a CNN at the feature level.

III. Proposed method

As shown in Figure 2, the proposed deep fusion framework for RGBD salient object detection is composed of three modules. The first module generates various saliency feature vectors for each superpixel region. The second module extracts a hyper-feature representation from the obtained saliency feature vectors. The third module is the Laplacian propagation framework, which helps to generate a spatially consistent saliency map.

A. Extraction of saliency feature vectors

Given an image, our aim is to represent saliency by means of several demonstrated effective saliency features. Figure 3 shows an illustration of the proposed saliency feature extraction process. We first segment the image into N superpixels using the SLIC method [51]. Given an RGB image $\mathcal{I}$, we denote the N segmented regions by $\mathcal{P} = \{P_1, P_2, ..., P_i, ...P_N\}$. For each superpixel P_i, the vector of the calculated saliency features is denoted by Γ_{P_i}. In the following, we will take region P_i (the region marked in orange in Figure 3) as an example to show how we calculate different saliency feature vectors.

Unlike classical saliency detection methods, which directly calculate the saliency values for each superpixel, we record the saliency features for each superpixel and no further operation is performed on them to ensure that these saliency features as raw as possible. For region P_i, there are seven types of feature

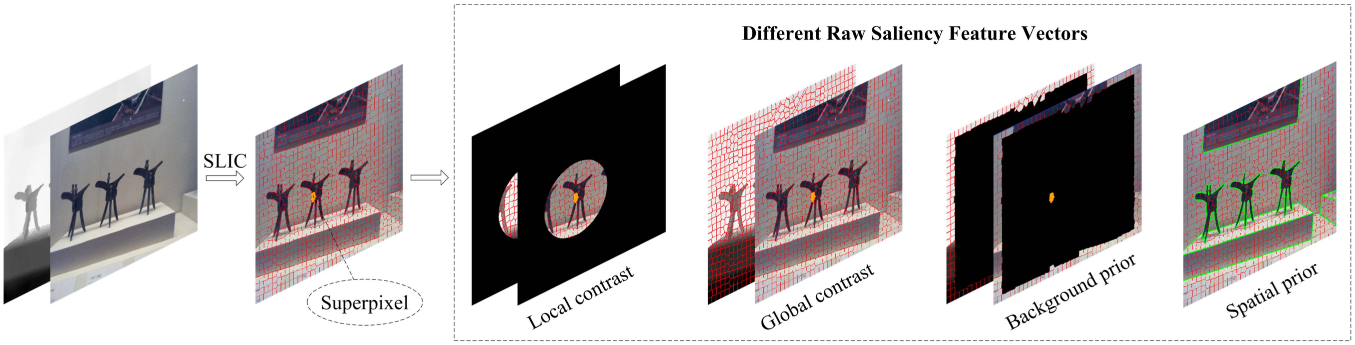


Fig. 3: Saliency feature extraction.

vectors: $\Gamma_{P_i} = \{\Theta_i^{CL}, \Theta_i^{CG}, \Theta_i^{DL}, \Theta_i^{DG}, \Theta_i^{CB}, \Theta_i^{DB}, \Theta_i^{CS}\}$, where C and D represent color and depth information, respectively; L indicates that saliency is determined in the local scope, whereas G indicates the global scope; and B and S represent the background and color compactness priors, respectively. More specifically, the color-based feature vectors are recorded in the following form:

$$\left\{\begin{array}{l} \Theta_i^{CL} = \{P_{i,1}^{CL}, ..., P_{i,j}^{CL}, ..., P_{i,N}^{CL}\} \\ \Theta_i^{CG} = \{P_{i,1}^{CG}, ..., P_{i,j}^{CG}, ..., P_{i,N}^{CG}\} \\ \Theta_i^{CB} = \{P_{i,1}^{CB}, ..., P_{i,j}^{CB}, ..., P_{i,N_b}^{CB}\} \\ \Theta_i^{CS} = \{P_{i,1}^{CS}, ..., P_{i,j}^{CS}, ..., P_{i,N}^{CS}\} \end{array}\right. , \quad (1)$$

and the depth-based feature vectors are defined similarly. We compute the color-based features in the Lab color space.

The local color contrast $P_{i,j}^{CL}$ is calculated as

$$P_{i,j}^{CL} = t(j)\phi_L(i,j) \left\| \boldsymbol{c}_i - \boldsymbol{c}_j \right\|_2 , \quad (2)$$

where $t(j)$ is the total number of pixels in region P_j and a larger superpixel contributes more to the saliency; $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$ are the mean color values of the regions P_i and P_j, respectively; $\phi_L(i,j)$ is used to control the spatial influence distance, where this weight is defined as $\exp(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2}{2\sigma_{Lr}^2})$; and $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are the centers of the corresponding regions. In our experiment, the parameter σ_{Lr} is set to 0.15 to make the neighbors have a higher influence on the calculated contrast values, whereas the influence of other regions is negligible. Similar to the local color contrast vector, the global color contrast vector is defined as

$$P_{i,j}^{CG} = t(j)\phi_G(i,j) \left\| \boldsymbol{c}_i - \boldsymbol{c}_j \right\|_2 . \quad (3)$$

The difference between the global contrast and the local contrast lies in the spatial weight $\phi_G(i,j)$; for the global contrast, the parameter σ_{Gr} is set to 0.45 to cover the entire image.

Likewise, the depth contrast between region P_j and region P_i can be calculated as in Eq. 4 and Eq. 5:

$$P_{i,j}^{DL} = t(j)\phi_L(i,j) \left| d_i - d_j \right| , \quad (4)$$

$$P_{i,j}^{DG} = t(j)\phi_G(i,j) \left| d_i - d_j \right| , \quad (5)$$

where d_i and d_j are the mean depth values of the regions P_i and P_j, respectively.

Generally speaking, the colors of an object are compactly clustered together, whereas the colors belong to the background are widely distributed throughout the entire image. The elements $P_{i,j}^{CS}$ of the color-compactness-based feature vector are calculated as follows:

$$P_{i,j}^{CS} = \phi(\boldsymbol{c}_i, \boldsymbol{c}_j) \left\| \boldsymbol{x}_j - \boldsymbol{u}_i^{cs} \right\|_2 , \quad (6)$$

where the function $\phi(\boldsymbol{c}_i, \boldsymbol{c}_j)$ is used to calculate the similarity of two colors $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$ and is defined as $\exp(-\frac{\|\boldsymbol{c}_i - \boldsymbol{c}_j\|_2^2}{2\delta_c^2})$, whereas $\boldsymbol{u}_i^{cs} = \sum_{j=1}^{M} \phi(\boldsymbol{c}_i, \boldsymbol{c}_j)\boldsymbol{x}_j$ defines the weighted mean position of color $\boldsymbol{c}_i$. The parameter δ_c is set to 20 in our implementation. We omit the depth compactness prior in our method since the depth map contains only a few dozen depth levels and their spatial distributions can be very random. Experimental results also show that the addition of the depth compactness does not strongly affect the final results.

In addition to the color compactness prior, we also introduce a background prior, which leverages the fact that salient objects are less likely to be located close to the image boundaries. We first extract N_b regions along the image boundaries as pseudo-background regions. Then, the color and depth contrasts with respect to these pseudo-background regions are calculated in a manner similar to Eq. 3 and Eq. 5. In our experiment, the number of superpixels N is set to 1024, and N_b is set to 160.

B. Hyper-feature extraction via a CNN

Given the obtained saliency feature vectors, we then propose a CNN architecture to automatically combine them into unified and representative features. We formulate saliency detection as a binary logistic regression problem, in which the saliency feature vectors of a superpixel (reshaped to dimensions of 32 × 32 × 6) are taken as inputs and the probabilities of this superpixel belonging to a salient or non-salient region are produced as outputs. For each superpixel P_i, all seven saliency feature vectors Γ_{P_i} are integrated into a multiple-channel image (with dimensions of $32 \times 32 \times 6$) as follows:

(1) Reshape the N-length vectors (Θ^{CL}, Θ^{CG}, Θ^{DL}, Θ^{DG} and Θ^{CS}) to dimensions of 32×32 to form the first five channels.

(2) Perform zero padding on the N_b-length vectors Θ^{CB} and Θ^{DB} to a length of $N/2$ and then concatenate and reshape them into dimensions of 32×32 to form the sixth channel.

As shown in Figure 2, our network consists of three convolutional layers followed by a fully connected layer and a logistic regression output layer with a sigmoid nonlinear function. Following the first and second convolutional layers, we add an average pooling layer for translation invariance. We adopt the sigmoid function as the nonlinear mapping function for the three convolutional layers, whereas Rectified Linear Units (ReLUs) are applied in the last two layers. A dropout procedure is applied after the first fully connected layers to avoid overfitting.

For simplicity, we use $conv(N, K)$ and $fc(N)$ to indicate a convolutional layer and a fully connected layer, respectively, with N outputs and kernel size K. $pool(T, K)$ indicates a pooling layer of type T and kernel size K. sig and $relu$ represent the sigmoid function and the ReLUs. Thus, the architecture of our CNN can be described as $conv1(6, 5) - sig1 - pool1(MEAN, 2) - conv2(12, 5) - sig2 - pool2(MEAN, 2) - conv3(24, 3) - sig3 - fc4(200) - relu4 - dropout4 - fc5(2)$. This proposed CNN was trained via back-propagation using the stochastic gradient descent (SGD) method.

C. Laplacian propagation

Because the saliency values are estimated for each superpixel individually, the CNN proposed in Section III-B may fail to retain spatial consistency, leading to noisy output. Figure 4c shows two examples of the saliency maps produced by our CNN for RGBD images. These results indicate that our CNN omits some salient regions and incorrectly detects some background regions as salient. Despite these misdetected regions, most of the regions identified as having a high probability of being salient are correct, robust, and reliable. The same situation also occurs for probabilities of non-saliency in the background (Figure 4d). As a consequence, these high confident regions are used as guidance, and they are employed in a Laplacian propagation framework [28] to obtain a more spatially consistent saliency map. The key to the Laplacian propagation procedure lies in propagating the saliency from regions with high probability to more ambiguous regions by considering two criteria: (1) neighboring regions are more likely to have similar saliency values, and (2) regions within the same manifold are more likely to have similar saliency values.

Given a set of superpixels $\mathcal{P} = \{P_1, P_2, ..., P_N\}$ of an input image $\mathcal{I}$ and a label set $\mathcal{L} = \{1, 2\}$, we let w^{sal} and w^{non_sal} denote the saliency and non-saliency probabilities generated by the proposed CNN. The superpixels in $\mathcal{P}$ are labeled as 1 if $w^{sal} > \tau_1$ or as 2 if $w^{non\_sal} > \tau_2$. The goal of Laplacian propagation is to predict the labels of the remaining regions.

Let $F = [\boldsymbol{f}_1^T, \boldsymbol{f}_2^T, ..., \boldsymbol{f}_N^T]^T$ denote an $N \times 2$ non-negative matrix that corresponds to the binary classification results of $\mathcal{P}$, and let each region P_i be assigned a label $y_i = \arg\max_{k=\{1,2\}} f_{ik}$, where $\boldsymbol{f}_i = \{f_{i1}, f_{i2}\}$. An indicator matrix is defined as $Y = [y_{ik}]_{N\times 2}$, where $y_{ik} = 1$ if region P_i is labeled as k and $y_{ik} = 0$ otherwise. We further employ the color and depth information to form the affinity matrix $A = [a_{ij}]_{N\times N}$:

$$a_{ij} = \exp(-\frac{\|\boldsymbol{c}_i - \boldsymbol{c}_j\|_2^2}{2\delta_1^2})\exp(-\frac{|d_i - d_j|^2}{2\delta_2^2}), \tag{7}$$

where the first term defines the color distance between superpixel regions P_i and P_j and the second term defines the relative depth distance. Most of the elements of the affinity matrix A are zero, except for neighboring P_i and P_j pairs. To better leverage local smoothness, we use a two-hierarchy neighboring connection model; i.e., each region is connected not only to its neighboring regions but also to regions that share the same boundaries as its neighboring regions. We set $a_{ii} = 0$ to avoid self-reinforcement. Then, the Laplacian propagation problem can be formulated as the problem of solving the following optimization functions:

$$F^* = \arg\min_F \frac{\mathcal{Q}(F)}{2}, \tag{8}$$

$$\mathcal{Q}(F) = \sum_{i,j=1}^{N} a_{ij}\left\|\frac{\boldsymbol{f}_i}{\sqrt{m_{ii}}} - \frac{\boldsymbol{f}_j}{\sqrt{m_{jj}}}\right\|_2^2 + \mu\sum_{i=1}^{N}\|\boldsymbol{f}_i - \boldsymbol{y}_i\|_2^2, \tag{9}$$

where the parameter μ controls the balance between the smoothness constraint (the first term) and the fitting constraint (the second term) and the m_{ii} are the elements of the degree matrix M derived from the affinity matrix A, where $m_{ii} = \sum_j a_{ij}$. This designed smoothness constraint not only considers local smoothness but also constrains regions within the same manifold to have the same label by constructing a smooth classifying function. This classifying function can vary sufficiently slowly along the coherent structure revealed by the original image [28].

The optimization function expressed in Eq. 8 can be solved using an iterative algorithm, as shown in [28], or it can be reformulated into a linear system. For efficiency, we set the derivative of $\mathcal{Q}(F)$ to zero and obtain the optimal solution to Eq. 8 by solving the following linear equation:

$$(I - \alpha S)F^* = Y, \tag{10}$$

where I is the identity matrix and $\alpha = 1/(1 + \mu)$. We further adopt the conjugate gradient method with a preconditioner to solve this linear equation for rapid convergence.

After propagation from the high-probability salient and non-salient regions, the final saliency map is normalized to [0,1]; the resulting map is denoted by $S = \overline{F^*}$. Two examples of the proposed propagation technique are shown in Figure 4. The incorrectly estimated regions in Figure 4b and Figure 4c are corrected in the final saliency maps produced via Laplacian propagation. In our implementation, the parameters τ_1 and τ_2 are adaptively determined using the Otsu method [52].

IV. Experimental evaluations

In this section, we evaluate the proposed method on three datasets: the NLPR RGBD salient object detection dataset [12], the NJUDS2000 stereo dataset [13], and the LFSD dataset [54].

NLPR dataset [12]. The NLPR RGBD salient object detection dataset [12] contains 1000 images captured by the

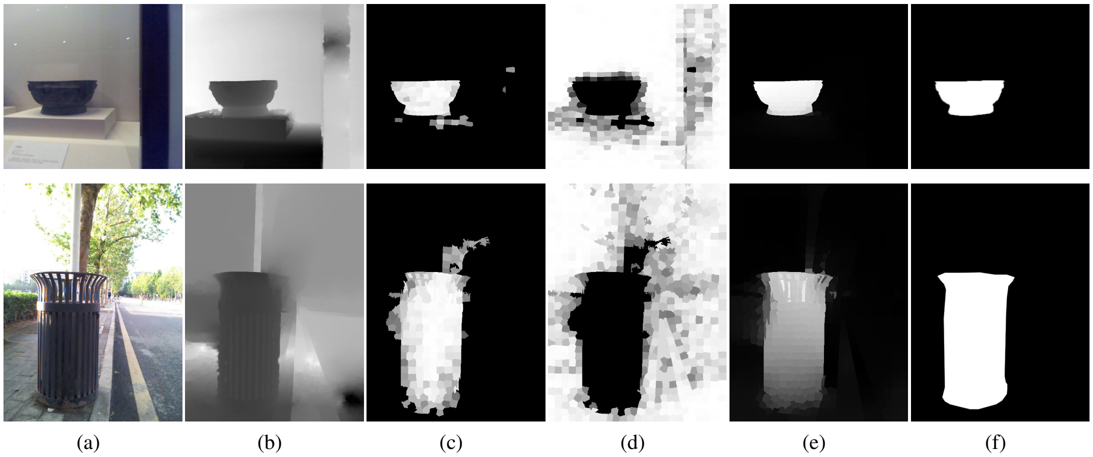


Fig. 4: Examples of the proposed Laplacian propagation technique. (a) RGB images. (b) Depth images. (c) Saliency probabilities produced by the proposed CNN. (d) Background (non-saliency) probabilities produced by the proposed CNN. (e) Refined saliency maps obtained using (c) and (d) as guidance. (f) Ground-truth saliency maps.

Microsoft Kinect in different indoor and outdoor scenarios. We randomly split this dataset into two parts: 750 images for training and 250 for testing.

NJUDS2000 dataset [13]. The NJUDS2000 dataset contains 2000 stereo images as well as the corresponding depth maps and manually labeled ground truths. The depth maps are generated using an optical flow method. We also randomly split this dataset into two parts: 1000 images for training and 1000 for testing.

LFSD dataset [54]. The LFSD dataset [54] contains 100 images with depth information and manually labeled ground truths. The depth information was captured using the Lytro light field camera. All images in this dataset were used for testing.

Evaluation metrics. We compute the precision-recall (PR) curve, the mean precision and recall, and the F-measure score to evaluate the performances of different saliency detection methods. The PR curve indicates the mean precision and recall of the saliency map at various thresholds. The F-measure is defined as $F_\beta = \frac{(1+\beta^2)\times precision \times recall}{\beta^2 \times precision + recall}$, where β^2 is set to 0.3 [34].

A. Implementation details

We used the 750 randomly sampled training images from the NLPR dataset [12] and the 1000 randomly sampled training images from the NJUDS2000 dataset [13] to train our deep learning framework. This randomly selected training dataset includes images of more than 1000 kinds of common objects under different circumstances. The remaining NLPR, NJUDS2000, and LFSD datasets were used to verify the generalizability of the proposed method.

The proposed method was implemented using MATLAB. We set the momentum in our network to 0.9 and the weight decay to 0.0005. The learning rate of our network was gradually decreased from 1 to 0.001. Because of the “data-hungry” nature of CNNs, the existing training data were insufficient for training; therefore, in addition to the dropout procedure, we also employed data augmentation to enrich our training dataset. Similarly to [55], we adopted two different image augmentation operations: the first one consists of image translations and horizontal flipping, and the other involves altering the intensities of the RGB channels. These data augmentations greatly enlarged our training dataset and made it possible for us to train the proposed CNN without overfitting. It took approximately $5 \sim 7$ days for training to converge.

B. Performance comparison

In this section, we compare our method with four state-of-the-art methods designed for RGB images (S-CNN [41], BSCA [16], MB+ [53], and LEGS [42]) and four RGBD saliency methods designed especially for RGBD images (LMH [12], ACSD [13], GP [14], and LBE [15]).

The results of these different methods were either provided by the authors or generated using the publicly available source codes. Qualitative comparisons of the different methods on various scenes are shown in Figure 5. In the first and fifth rows of Figure 5, the salient objects show high color contrast with the backgrounds, and thus, RGB saliency methods are able to detect these salient objects correctly. However, when the salient objects share similar colors with the backgrounds, e.g., in the sixth, seventh, and eighth rows of Figure 5, it is difficult for the existing RGB models to correctly extract the saliency results. With the help of depth information, the salient objects can be easily detected by the proposed RGBD method. Figure 5 also shows that the proposed method consistently outperforms all other RGBD saliency methods (LMH [12], ACSD [13], and GP [14]).

Quantitative comparisons on the NLPR test set, the NJUDS2000 test set, and the LFSD dataset are shown in Figure 6 and Table I. Figure 6 and Table I show that the proposed method performs favorably compared with the existing algorithms, with higher precision, recall and F-measure values on all three datasets. The NLPR dataset is challenging because most of the salient objects share similar colors with the backgrounds. Consequently, RGB saliency methods perform worse than RGBD saliency methods on this dataset in terms

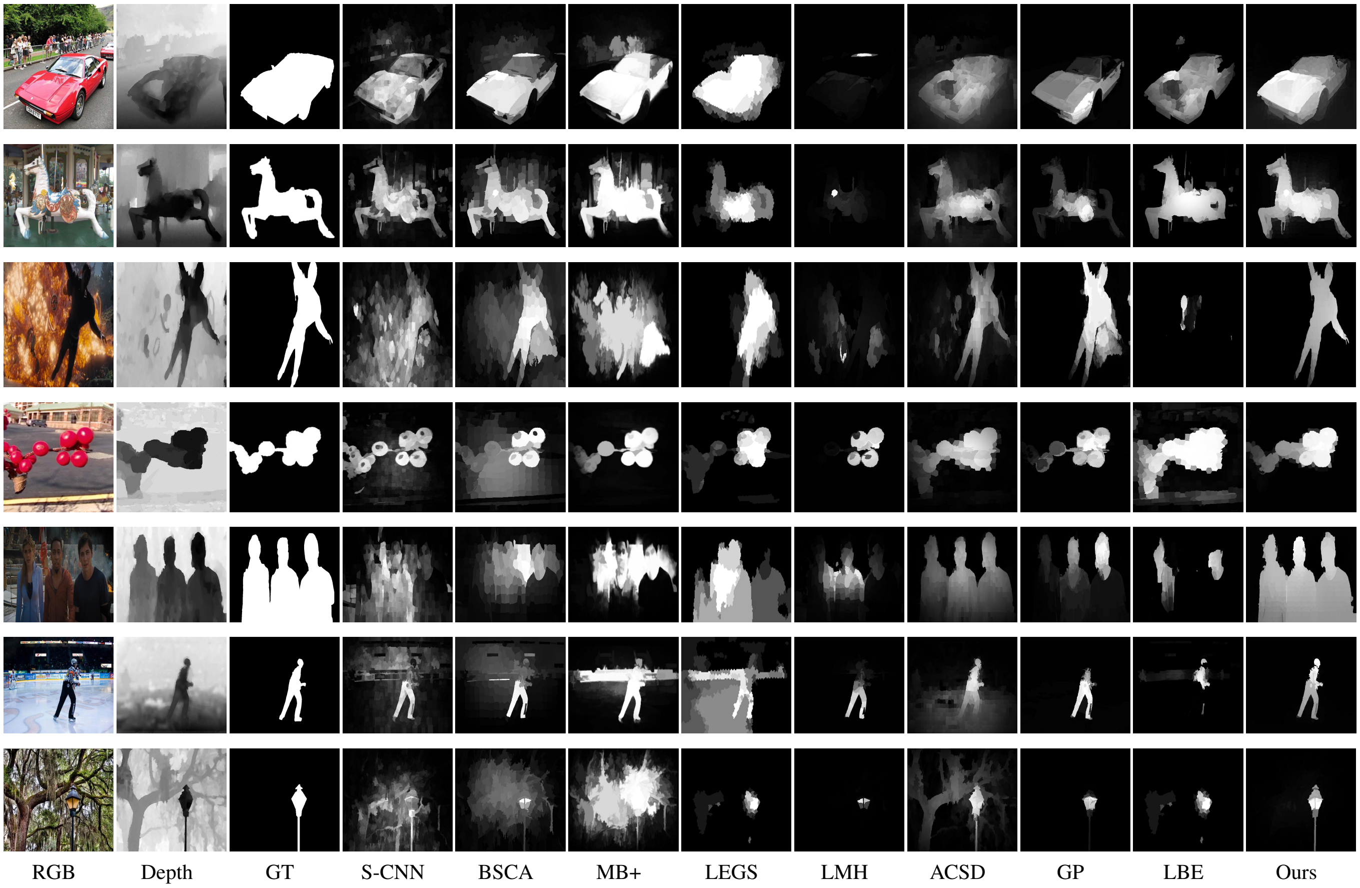


Fig. 5: Visual comparisons of the results of the proposed deep fusion framework with those of four RGB saliency methods and four RGBD saliency methods. The saliency maps generated by S-CNN [41], BSCA [16], MB+ [53], and LEGS [42] are obtained from RGB images, whereas the saliency maps generated by LMH [12], ACSD [13], GP [14], and LBE [15] are obtained from RGBD images.

TABLE I: The F-measure scores for various approaches on three datasets.

Dataset	S-CNN [41]	BSCA [16]	MB+ [53]	LEGS [42]	LMH [12]	ACSD [13]	GP [14]	LBE [15]	Ours
NLPR test set	0.5141	0.5634	0.6049	0.6335	0.6519	0.5448	0.7184	0.7344	**0.7823**
NJUD test set	0.6096	0.6133	0.6156	0.6791	0.6381	0.6952	0.7246	0.7314	**0.7874**
LFSD dataset	0.6982	0.7311	0.7029	0.7384	0.7041	0.7567	0.7877	0.7122	**0.8439**

of precision. By virtue of accurate depth maps (for the NLPR dataset), the LMH [12] and GP [14] methods perform well in both precision and recall. However, they do not perform well when tested on the NJUDS2000 dataset and the LFSD dataset. This is because these two datasets provide only approximate depth information (calculated from stereo images or using a light field camera); consequently, LMH [12] and GP [14] can detect only a small fraction of the salient objects (with high precision but low recall). ACSD [13] does not work as well when the salient object lies in the same plane as the background; see, e.g., the third row of Figure 5 and the poor quantitative results on the NLPR dataset. The Local Background Enclosure features in LBE [15] allow salient structure to be directly measured from depth. However, this approach pays less attention to RGB saliency cues, and its performance is not satisfactory on the LFSD dataset. Both the qualitative and quantitative results show that the proposed method performs better in terms of accuracy and robustness than the methods considered for comparison on RGBD input images.

Saliency maps vs. features. Here, we report a series of experiments conducted to analyze the flexibility of the proposed framework and the effectiveness of Laplacian propagation.

In addition to previous heuristic saliency map merging algorithms [12], [14], we also compare our method with four other saliency map integration methods on the three test datasets [12], [13], [54] to demonstrate the flexibility of fusing different cues at the feature level. These four integration methods are direct linear fusion (LF), fusing in CRF [25], the latest Multi-layer Cellular Automata (MCA) integration algorithm [16], and a CNN-based fusion method (denoted by CNN-F). To investigate the importance of saliency map quality, we tested these saliency map merging methods on two set of inputs. The first set was drawn from seven saliency

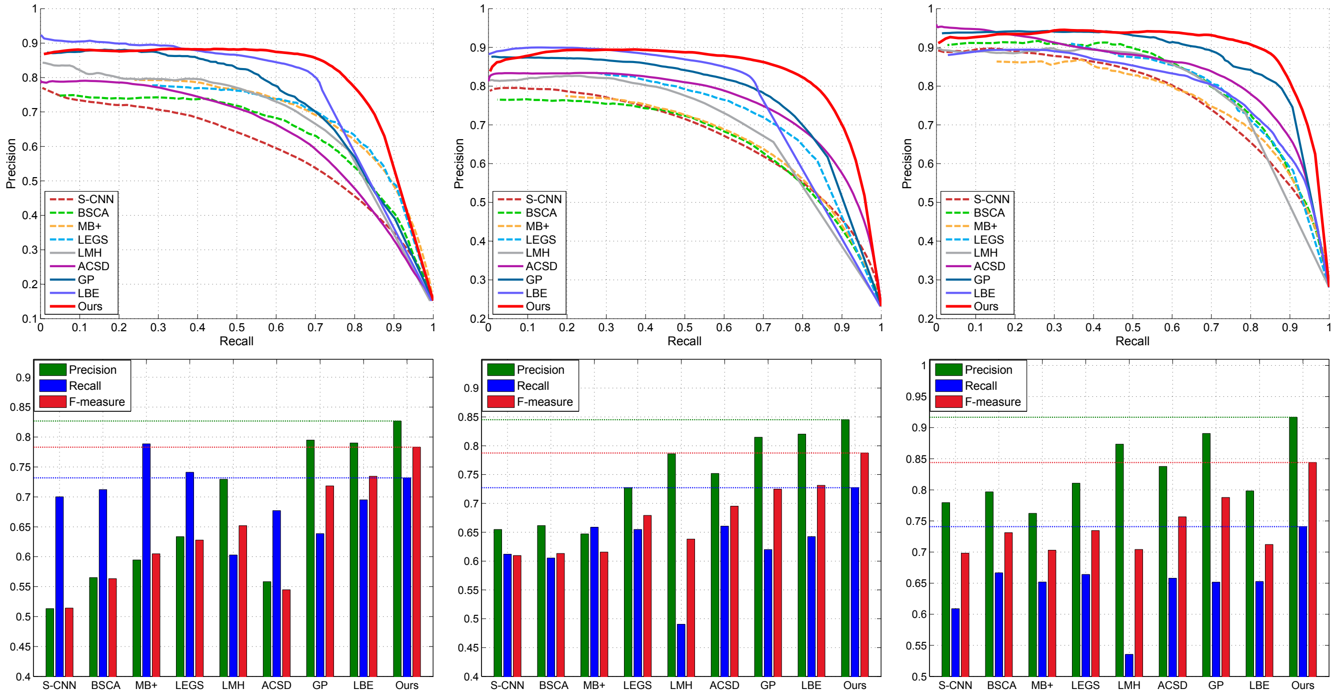


Fig. 6: PR curves and F-measure curves for various methods on three datasets. Left: Quantitative comparisons on the 250 test images from the NLPR dataset [12]. Middle: Quantitative comparisons on the 1000 test images from the NJUDS2000 dataset [13]. Right: Quantitative comparisons on the LFSD dataset [54].

TABLE II: Comparisons of F-measure scores for various saliency map merging approaches with or without Laplacian propagation (LP) on the NLPR test dataset [12].

LP?	Fundamental fusion				Sophisticated fusion				Heuristic fusion		Ours
	LF	CRF	MCA	CNN-F	LF	CRF	MCA	CNN-F	LMH	GP	
No	0.393	0.2991	0.3713	0.4667	0.6352	0.7023	0.7081	0.7254	0.6519	**0.718**	0.7315
Yes	**0.536**	**0.398**	**0.486**	**0.597**	**0.6908**	**0.7462**	**0.7618**	**0.7673**	**0.665**	0.7111	**0.7823**

TABLE III: Comparisons of F-measure scores for various saliency map merging approaches with or without Laplacian propagation (LP) on the NJUD test dataset [13].

LP?	Fundamental fusion				Sophisticated fusion				Heuristic fusion		Ours
	LF	CRF	MCA	CNN-F	LF	CRF	MCA	CNN-F	LMH	GP	
No	0.437	0.450	0.458	0.644	0.5582	0.6526	0.7289	0.7378	0.6381	**0.7246**	0.7447
Yes	**0.605**	**0.609**	**0.632**	**0.731**	**0.6908**	**0.7512**	**0.7674**	**0.7682**	**0.6810**	0.7179	**0.7874**

TABLE IV: Comparisons of F-measure scores for various saliency map merging approaches with or without Laplacian propagation (LP) on the LFSD dataset [54].

LP?	Fundamental fusion				Sophisticated fusion				Heuristic fusion		Ours
	LF	CRF	MCA	CNN-F	LF	CRF	MCA	CNN-F	LMH	GP	
No	0.461	0.436	0.558	0.672	0.6409	0.7682	0.8092	0.7209	0.704	**0.7877**	0.8157
Yes	**0.616**	**0.693**	**0.654**	**0.7517**	**0.7515**	**0.7891**	**0.8162**	**0.8205**	**0.718**	0.7830	**0.8439**

maps computed based on widely used features (similar to the seven saliency feature vectors computed in section III-A), and the second set was drawn from seven other representative sophisticated saliency maps (obtained using four state-of-the-art RGBD saliency detection methods [12], [13], [14], [15] and three RGB saliency detection methods [41], [53], [42]).

The original CRF fusion framework presented in [25] was designed to merge three color-based saliency maps. In our implementation, we retrained this CRF framework to merge either the seven adopted fundamental saliency maps or three sophisticated saliency maps [1].

For CNN-F, we utilized the same CNN architecture shown in Fig. 3 to perform CNN-based saliency map fusion, i.e., the same convolutional layers and fully connected layers, with the exception of the input layer. More specifically, we formulated the saliency map merging problem as a binary logistic regression problem, in which several saliency map patches are taken as inputs (with dimensions of $52 \times 52 \times 7$

[1] We adopted the implementation available at http://www.cs.unc.edu/~vicente/code.html for training and testing. This CRF was trained on the NLPR training dataset [12] and the NJUDS2000 training dataset [13]

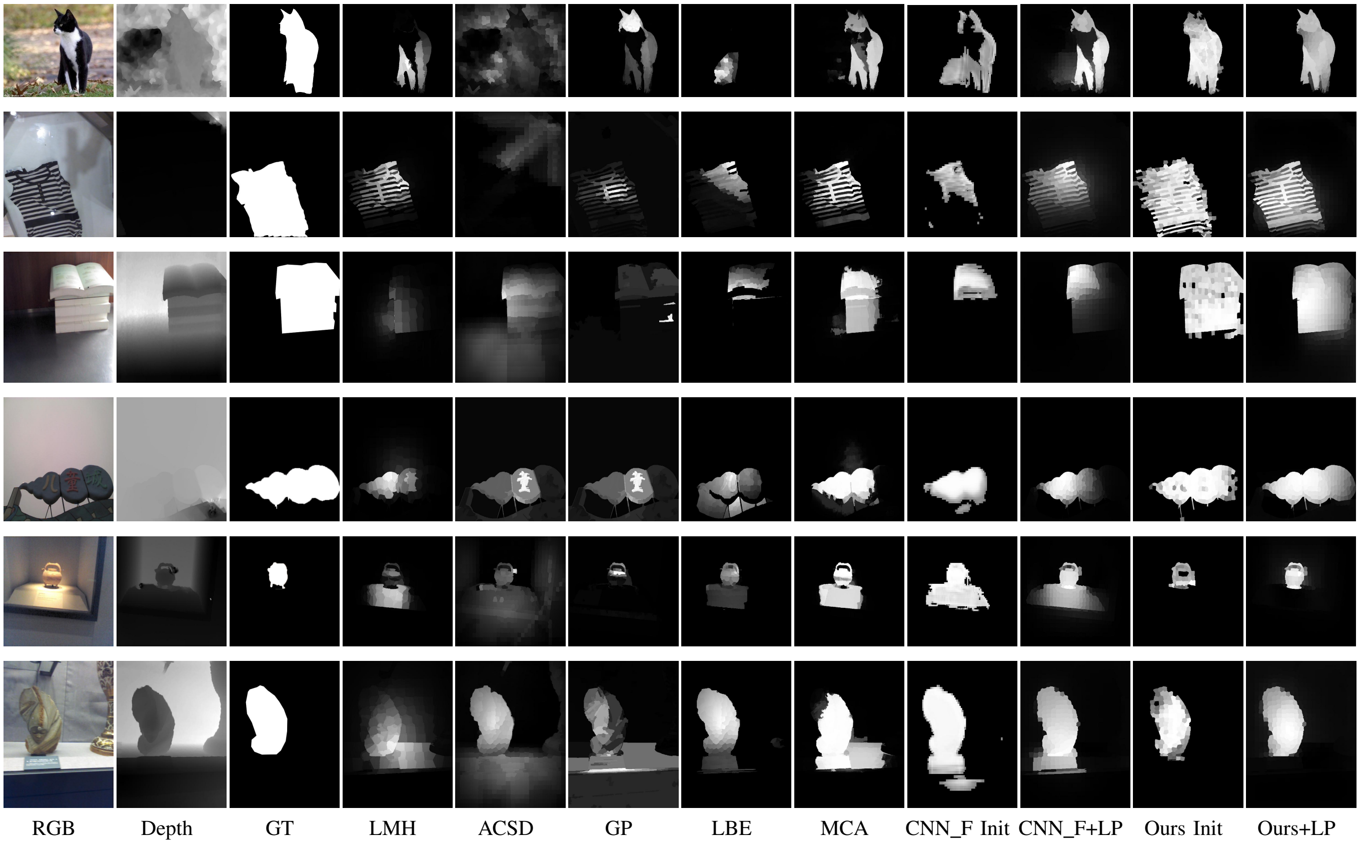


Fig. 7: Additional examples illustrating the problems with various saliency map merging methods. MCA and CNN_F denote the results of sophisticated fusion methods (fusing the results of four RGBD saliency detection methods [12], [13], [14], [15] and three RGB methods [41], [53], [42]). Here, we present the saliency maps generated by four RGBD saliency detection methods (i.e., LMH [12], ACSD [13], GP [14], and LBE [15]). "CNN_F Init" and "Ours Init" denote the initial results of CNN_F and the proposed hyper-feature method without Laplacian propagation, respectively.

for fundamental saliency map merging and $52 \times 52 \times 3$ for sophisticated saliency map merging) and the probabilities of the central pixel being salient or non-salient are produced as outputs. Similar to what is done in reference [42], the CNN-F was trained in a patch-wise manner. We collected training samples by cropping patches with dimensions of 52×52 from each saliency map using a sliding window. We labeled a patch as salient if the central pixel was salient or if 75% pixels in the patch were salient; otherwise, it was labeled as non-salient. The CNN-F was trained on cropped patches from the NLPR training set [12] and the NJUDS2000 training set [13].

The relevant comparisons of the results of our proposed method with those of these saliency map merging methods are shown in Figure 7, Table II, Table III, and Table IV. "Fundamental fusion" refers to the results of the four non-heuristic merging methods applied to the seven fundamental saliency maps. "Heuristic fusion" refers to the results of the two state-of-the-art heuristic saliency map merging methods [12], [14], whereas "Sophisticated fusion" refers to the results of the four non-heuristic merging methods applied to the seven sophisticated saliency maps (calculated using the seven state-of-the-art RGBD and RGB saliency detection methods presented in [12], [13], [14], [15], [41], [53], [42]).

Among the results for "Fundamental fusion" in Table IV, none of the existing saliency map merging methods (including the deep learning framework) can achieve satisfactory performance. Even when fed with state-of-the-art sophisticated saliency maps, these saliency merging methods still perform worse than our saliency feature fusion method without the LP framework (0.8092 vs. 0.8157), which further validates the flexibility of our feature-level fusion approach. Note that this value of 0.8157 was obtained using our initial saliency feature fusion network, which operates only at the pixel level and without considering spatial consistency. Our model achieves superior performance even though the input features are very simple, indicating that even simple features have significant representational power for this problem. Compared with the other methods using similar features ("Fundamental fusion") and using more sophisticated saliency maps ("Sophisticated fusion"), we can observe that the fusion of features is much more interpretable and flexible than the fusion of saliency maps.

Analysis of Laplacian Propagation. We then evaluated the effectiveness of the proposed Laplacian propagation technique and the optimization of the results of existing methods using Laplacian propagation. The F-measure scores for our

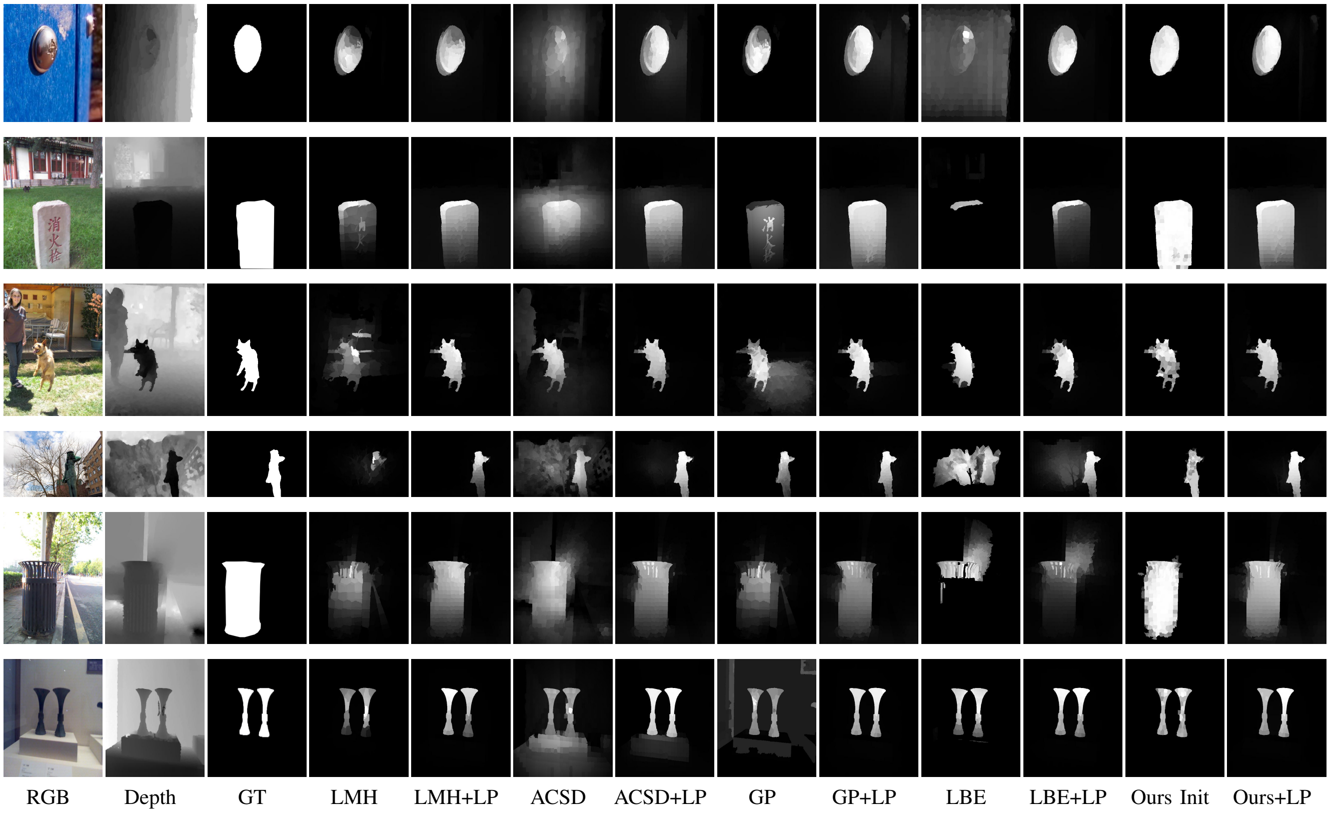


Fig. 8: Examples demonstrating the effectiveness of Laplacian propagation.

RGBD method without Laplacian propagation on the three test datasets [12], [13], [54] are shown in blue in Table II, Table III, and Table IV. The initially learned hyper-features already outperform the state-of-the-art approaches, whereas with LP, we achieve F-measures of almost 0.79, 0.79, and 0.84. Figure 8 shows several examples of the optimization of the results of existing methods (LMH [12], ACSD [13], GP [14], and LBE [15]) using Laplacian propagation. These quantitative and qualitative experimental evaluations demonstrate that the proposed Laplacian propagation technique is able to refine the saliency maps of existing methods and thus can be widely adopted as a post-processing step.

Failure cases. Figure 9 presents additional visual results and a failure case of our proposed method on RGBD images. By comparing these two images, we can find that depth information is more helpful when a salient object shows high depth contrast with the background or lies closer to the camera. Our method may fail when the salient object shares very similar color and depth information with the background.

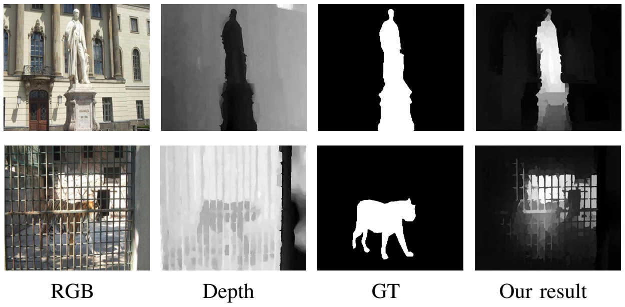


Fig. 9: More visual results and a failure case.

V. Conclusion

In this paper, we propose a novel RGBD saliency detection method. Our framework consists of three different modules. The first module generates various low-level saliency feature vectors from the input image. The second module learns the interaction mechanism of the RGB saliency features and the depth-induced features and produces hyper-features via a CNN. Feeding the CNN with these hand-designed features can guide its learning process toward saliency optimization. In the third module, we integrate a Laplacian propagation framework with the CNN to obtain a spatially consistent saliency map. Both quantitative and qualitative experimental results show that the fused RGBD hyper-features outperform all state-of-the-art methods.

We introduce Laplacian propagation into the saliency literature by adopting it as a two-stage propagation approach for refining the final saliency map, which can be widely adopted as a post-processing step. We demonstrate that an optimized fusion leads to superior performance, and this flexible hyper-feature extraction framework could be further extended by including more saliency cues (e.g., flash cue [33]). In our future work, we intend to explore a deeper and more effective fusion network and extend it to other applications.

References

[1] U. Rutishauser, D. Walther, C. Koch, and P. Perona, "Is bottom-up attention useful for object recognition?" in *CVPR*, vol. 2, 2004, pp. II–37.

[2] V. Mahadevan and N. Vasconcelos, "Saliency-based discriminant tracking," in *CVPR*, 2009, pp. 1007–1013.

[3] G. Sharma, F. Jurie, and C. Schmid, "Discriminative spatial saliency for image classification," in *CVPR*, 2012, pp. 3506–3513.

[4] P. Luo, Y. Tian, X. Wang, and X. Tang, "Switchable deep network for pedestrian detection," in *CVPR*, 2014, pp. 899–906.

[5] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE MultiMedia*, vol. 19, no. 2, pp. 4–10, 2012.

[6] S. B. Gokturk, H. Yalcin, and C. Bamji, "A time-of-flight depth sensor-system description, issues and solutions," in *CVPR*, 2004, pp. 35–35.

[7] A. K. Mishra, A. Shrivastava, and Y. Aloimonos, "Segmenting simple objects using rgb-d," in *ICRA*, 2012, pp. 4406–4413.

[8] H. Fu, D. Xu, S. Lin, and J. Liu, "Object-based rgbd image co-segmentation with mutex constraint," in *CVPR*, 2015, pp. 4428–4436.

[9] D. Banica and C. Sminchisescu, "Second-order constrained parametric proposals and sequential search-based structured prediction for semantic segmentation in rgb-d images," in *CVPR*, 2015, pp. 3517–3526.

[10] S. Alpert, M. Galun, A. Brandt, and R. Basri, "Image segmentation by probabilistic bottom-up aggregation and cue integration," vol. 34, no. 2, pp. 315–327, 2012.

[11] V. Movahedi and J. H. Elder, "Design and perceptual validation of performance measures for salient object segmentation," in *CVPR*, 2010, pp. 49–56.

[12] H. Peng, B. Li, W. Xiong, W. Hu, and R. Ji, "Rgbd salient object detection: a benchmark and algorithms," in *ECCV*, 2014, pp. 92–109.

[13] R. Ju, L. Ge, W. Geng, T. Ren, and G. Wu, "Depth saliency based on anisotropic center-surround difference," in *ICIP*, 2014, pp. 1115–1119.

[14] J. Ren, X. Gong, L. Yu, W. Zhou, and M. Yang, "Exploiting global priors for rgb-d saliency detection," in *CVPRW*, 2015, pp. 25–32.

[15] D. Feng, N. Barnes, S. You, and C. McCarthy, "Local background enclosure for rgb-d salient object detection," in *CVPR*, 2016, pp. 2343–2350.

[16] Y. Qin, H. Lu, Y. Xu, and H. Wang, "Saliency detection via cellular automata," in *CVPR*, 2015, pp. 110–119.

[17] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "Global contrast based salient region detection," in *CVPR*, 2011, pp. 409–416.

[18] Q. Wang, W. Zheng, and R. Piramuthu, "Grab: Visual saliency via novel graph model and background priors," in *CVPR*, 2016, pp. 535–543.

[19] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *ECCV*, 2012, pp. 29–42.

[20] K. Shi, K. Wang, J. Lu, and L. Lin, "Pisa: Pixelwise image saliency by aggregating complementary appearance contrast measures with spatial priors," in *CVPR*, June 2013, pp. 2115–2122.

[21] A. Maki, P. Nordlund, and J.-O. Eklundh, "A computational model of depth-based attention," in *Pattern Recognition*, vol. 4, 1996, pp. 734–739.

[22] C. Lang, T. V. Nguyen, H. Katti, K. Yadati, M. Kankanhalli, and S. Yan, "Depth matters: Influence of depth cues on visual saliency," in *ECCV*, 2012, pp. 101–115.

[23] K. Desingh, K. M. Krishna, D. Rajan, and C. Jawahar, "Depth really matters: Improving visual salient region detection with depth," in *BMVC*, 2013.

[24] Y. Zhang, G. Jiang, M. Yu, and K. Chen, "Stereoscopic visual attention model for 3d video," in *Advances in Multimedia Modeling*, 2010, pp. 314–324.

[25] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," *IEEE TPAMI*, vol. 33, no. 2, pp. 353–367, 2011.

[26] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *CVPR*, 2013, pp. 1155–1162.

[27] L. Zhou, Z. Yang, Q. Yuan, Z. Zhou, and D. Hu, "Salient region detection via integrating diffusion-based compactness and local contrast," *IEEE TIP*, vol. 24, no. 11, pp. 3308–3320, 2015.

[28] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," *NIPS*, vol. 16, no. 16, pp. 321–328, 2004.

[29] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A survey," *arXiv preprint arXiv:1411.5878*, 2014.

[30] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE TPAMI*, vol. 20, no. 11, pp. 1254–1259, 1998.

[31] D. A. Klein and S. Frintrop, "Center-surround divergence of feature statistics for salient object detection," in *ICCV*. IEEE, 2011, pp. 2214–2219.

[32] Y. Xie, H. Lu, and M.-H. Yang, "Bayesian saliency via low and mid level cues," *IEEE TIP*, vol. 22, no. 5, pp. 1689–1698, 2013.

[33] S. He and R. W. H. Lau, "Saliency detection with flash and no-flash image pairs," in *ECCV*, 2014, pp. 110–124.

[34] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *CVPR*, 2009, pp. 1597–1604.

[35] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE TPAMI*, 2014.

[36] X. Shen and Y. Wu, "A unified approach to salient object detection via low rank matrix recovery," in *CVPR*, 2012, pp. 853–860.

[37] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *CVPR*, 2013, pp. 2083–2090.

[38] E. Vig, M. Dorr, and D. Cox, "Large-scale optimization of hierarchical features for saliency prediction in natural images," in *CVPR*, 2014, pp. 2798–2805.

[39] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *CVPR*, 2015, pp. 1265–1274.

[40] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *CVPR*, 2015.

[41] S. He, R. W. Lau, W. Liu, Z. Huang, and Q. Yang, "Supercnn: A superpixelwise convolutional neural network for salient object detection," *IJCV*, pp. 1–15, 2015.

[42] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *CVPR*, 2015, pp. 3183–3192.

[43] G. Li and Y. Yu, "Deep contrast learning for salient object detection," in *CVPR*, June 2016.

[44] S. Jetley, N. Murray, and E. Vig, "End-to-end saliency mapping via probability distribution prediction," in *CVPR*, 2016, pp. 5753–5761.

[45] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, "Saliency detection with recurrent fully convolutional networks," in *ECCV*. Springer, 2016, pp. 825–841.

[46] X. Li, L. Zhao, L. Wei, M.-H. Yang, F. Wu, Y. Zhuang, H. Ling, and J. Wang, "Deepsaliency: multi-task deep neural network model for salient object detection," *IEEE TIP*, vol. 25, no. 8, pp. 3919–3930, 2016.

[47] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015, pp. 3431–3440.

[48] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[49] ——, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *arXiv preprint arXiv:1606.00915*, 2016.

[50] J. Wang, M. P. DaSilva, P. LeCallet, and V. Ricordel, "Computational model of stereoscopic 3d visual saliency," *IEEE TIP*, vol. 22, no. 6, pp. 2151–2165, 2013.

[51] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE TPAMI*, vol. 34, no. 11, pp. 2274–2282, 2012.

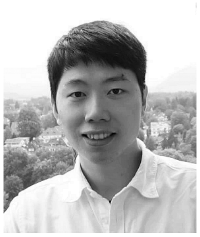
[52] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[53] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Měch, "Minimum barrier salient object detection at 80 fps," in *ICCV*, 2015.

[54] N. Li, J. Ye, Y. Ji, H. Ling, and J. Yu, "Saliency detection on light field," in *CVPR*, 2014.

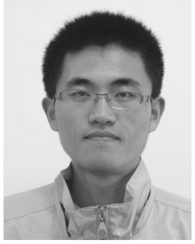
[55] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

Liangqiong Qu received her B.S. degree in automation from Central South University, China, in 2011. She is currently a joint Ph.D. student of the University of the Chinese Academy of Sciences and City University of Hong Kong. Her research interests include illumination modeling, image processing, saliency detection and deep learning.



Shengfeng He obtained his B.Sc. degree and M.Sc. degree from Macau University of Science and Technology and his Ph.D degree from City University of Hong Kong. He is an Associate Professor in the School of Computer Science and Engineering at South China University of Technology. He was a Research Fellow at City University of Hong Kong. His research interests include computer vision, image processing, computer graphics, and deep learning.

Jiawei Zhang received his B.Eng. degree in Electronic Information Engineering from the University of Science and Technology of China in 2011 and his master's degree from the Institute of Acoustics, Chinese Academy of Sciences, in 2014. He is currently a Computer Science Ph.D. student at City University of Hong Kong.



Jiandong Tian received his B.S. Tech. degree in automation from Heilongjiang University, China, in 2005. He received his Ph.D. degree in Pattern Recognition and Intelligent System from the Chinese Academy of Sciences, China, in 2011. He is currently an associate professor in computer vision at the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences. His research interests include pattern recognition and robot vision.

Yandong Tang received B.S. and M.S. degrees from the Department of Mathematics of Shandong University in 1984 and 1987. In 2002, he received his doctorate in applied mathematics from the University of Bremen, Germany. Currently, he is a professor at Shenyang Institute of Automation, Chinese Academy of Sciences. His research interests include robot vision, pattern recognition and numerical computation.

Qingxiong Yang received his B.Eng. degree in electronic engineering and information science from the University of Science and Technology of China, Hefei, China, in 2004, and his Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2010. He is currently the senior director of the Didi Research Institute. He was an Assistant Professor with the Department of Computer Science, City University of Hong Kong. His research interests lie in computer vision and computer graphics. He was a recipient of the Best Student Paper Award at the 2010 International Workshop on Multimedia Signal Processing and the Best Demo Award at the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.