Article

Motion Saliency Detection for Surveillance Systems Using Streaming Dynamic Mode Decomposition

Thien-Thu Ngo , VanDung Nguyen , Xuan-Quy Pham , Md-Alamgir Hossain and Eui-Nam Huh *

Department of Computer Science and Engineering, Kyung Hee University Global Campus, Yongin 17104, Korea; thu.ngo@khu.ac.kr (T.-T.N.); ngvandung85@khu.ac.kr (V.N.); pxuanqui@khu.ac.kr (X.-Q.P.); alamgir@khu.ac.kr (M.-A.H.)

* Correspondence: johnhuh@khu.ac.kr; Tel.: +82-031-201-3778

Received: 4 August 2020; Accepted: 20 August 2020; Published: 21 August 2020

Abstract: Intelligent surveillance systems enable secured visibility features in the smart city era. One of the major models for pre-processing in intelligent surveillance systems is known as saliency detection, which provides facilities for multiple tasks such as object detection, object segmentation, video coding, image re-targeting, image-quality assessment, and image compression. Traditional models focus on improving detection accuracy at the cost of high complexity. However, these models are computationally expensive for real-world systems. To cope with this issue, we propose a fast-motion saliency method for surveillance systems under various background conditions. Our method is derived from streaming dynamic mode decomposition (s-DMD), which is a powerful tool in data science. First, DMD computes a set of modes in a streaming manner to derive spatial-temporal features, and a raw saliency map is generated from the sparse reconstruction process. Second, the final saliency map is refined using a difference-of-Gaussians filter in the frequency domain. The effectiveness of the proposed method is validated on a standard benchmark dataset. The experimental results show that the proposed method achieves competitive accuracy with lower complexity than state-of-the-art methods, which satisfies requirements in real-time applications.

Keywords: motion saliency; dynamic mode decomposition; surveillance systems

1. Introduction

Nowadays, intelligent surveillance systems are gaining attention due to the demand for safety and security in critical infrastructures, such as military surveillance, home security, public transportation, etc. In these systems, video information acquired from sensors in devices is analyzed to assist in speeding up computer vision tasks like object tracking and vehicle detection. Therefore, the pre-processing method becomes an essential step that requires fast computation and high accuracy. One of the well-known pre-processing techniques is saliency detection. There are many studies on saliency detection from different aspects, such as object detection [1], object segmentation [2], video coding [3], image re-targeting [4], image quality assessment [5], and image compression [6]. The concept of saliency was inspired by neuroscience theory in which the human eye tends to focus on particular regions of the scene that stand out from their neighbors. The terms visual saliency or image saliency were first introduced in Itti et al. [7]. In these terms, the saliency model provides a mechanism to highlight the significant objects or regions that are most representative of a scene, while disposing of insignificant information retrieved from the surroundings. A saliency map is a grayscale image in which each pixel is mapped with an intensity value representing how much it differs from its surroundings. Preliminary research on visual saliency focused on still images. Various works have been proposed, and have achieved good performance, such as the graph-based model [8],

the Bayesian-based model [9], the super-pixel-based model [10], histogram-based contrast [11], the frequency-based model [12], the patch-based local-global mixture approach [13], low-rank matrix recovery [14], context-awareness [15], and spectral residuals [16]. These approaches are divided into two categories: local-based approaches and global-based approaches. The first category employs low-level cues from small regions to obtain the saliency map. Itti et al. [7] decomposed images into a set of multi-scale features, and the saliency map was obtained through center-surround contrast in different scales. Harel and Perona [8] introduced graph models to compute the saliency map based on Itti et al. [7]. Zhang et al. [9] integrated the advantages of the Bayesian framework and local self-information to improve performance. Jiang et al. [10] introduced a super-pixel-based method by formulating saliency detection via the Markov chain framework. In the second category, global feature-based approaches were introduced [11–16]. Cheng et al. [11] used color statistics to compute a regional color histogram, and then measured its color contrast with other regions as a saliency value. Achanta et al. [12] analyzed the efficiency of color and luminance features in the frequency domain. Yeh et al. [13] incorporated patch-based local saliency with background/foreground seed likelihood in order to generate the saliency map. In [14], Shen formulated an image saliency problem as low-rank sparse decomposition in the feature space, and then, the salient region was indicated by the sparse matrix. Goferman et al. [15] measured the distinctiveness of every pixel by considering its appearance with the most similar surrounding patch. Although they achieved successful performance, using a multi-scale framework or image segmentation only added more complexity to their models.

Despite these image saliency models, saliency for videos is more complicated because they contain more information than still images. Video saliency considers not only spatial information within a frame but also the temporal information between consecutive frames. In a surveillance system, temporal information such as motion cues or flicker gains a lot of attention from viewers. For example, a specific region considered important in a still image becomes less important in a video when objects move across the scene. Notably, in surveillance videos, moving objects catch more attention than other regions, so the salient regions can be people walking or cars moving. As a result, when applied to videos, traditional image saliency becomes less useful for highlighting these regions. Therefore, temporal information has been exploited in saliency models to make use of existing image saliency for videos [17–26]. Although these methods are robust and versatile, they demand high computational costs with complex models not fast enough to use as a pre-processing algorithm in surveillance systems.

To cope with these issues, we introduce a fast-motion saliency method for surveillance videos. When compared with existing approaches, the proposed method is more practical in real-time applications, because feature extraction is an important and time-consuming step in saliency models. Our proposed method rapidly extracts spatial-temporal features from streaming data. The spatial and temporal information is represented by eigenvectors and eigenvalues of equation-free systems. This process is updated incrementally when a new frame is available, which allows our method to run in a streaming manner. The main contributions are summarized as follows.

- We introduce a new approach to generating motion saliency for surveillance systems, which is fast and memory-efficient for applications with streaming data.
- The spatial-temporal features from video are generated from a sparse reconstruction process using streaming dynamic mode decomposition (s-DMD).
- We compute a motion saliency map from the refinement process using a difference-of-Gaussians (DoG) filter in the frequency domain.

The remainder of the paper is organized as follows. Section 2 reviews the existing saliency detection methods. Section 3 introduces the background to dynamic mode decomposition. We describe the algorithms of the proposed methodology in Section 4. Experiment results are discussed in Section 5, and the conclusion is given in Section 6.

2. Related Works

There have been numerous studies into video saliency detection over the past two decades. Based on the applications, we classify these methods into the fusion strategy approach [17–23] and the direct-pipeline approach [24–26].

In the first category, several works added temporal information into image saliency models. Zhang et al. [17] extended SUN models [9] to videos by introducing a temporal filter, and used a generalized Gaussian distribution to estimate the filter response. Zhong et al. [18] added optical flow to the existing graph-based visual saliency (GBVS) [8]. Besides, Mauthner et al. [19] encoded a color histogram structure, and estimated the local saliency to different scales using foreground and background patches. Wang et al. [20] used geodesic distance to estimate a spatiotemporal saliency map based on motion boundaries, edges, and colors. Yubing et al. [21] generated static saliency based on face detection and low-level features, with motion saliency calculated based on a motion vector analysis of the foreground region, and then both maps were weighted by a Gaussian function. In [22], motion trajectories were learned via sparse coding frameworks, and a sparse reconstruction process was developed to capture regions with high center-surround contrast. Chen et al. [23] defined spatial saliency via color contrast, with motion guide contrast computed to define temporal saliency.

The second category includes various works that generate spatial-temporal saliency directly from the pipeline. Xue et al. [24] used low-rank and sparse decomposition on video slices where sparse components represent the salient region. Bhattacharya et al. [25] obtained spatial features based on video decomposition, and identified salient regions using the sparsest features. Wang et al. [26] considered spatial-temporal consistency over frames by using gradient flow field and energy optimization. All of these methods achieved good results; however, the performance heavily relies on how good the fusion strategy is [23,24], or demands high complexity in the models [25–27]. Therefore, these works have to deal with execution time to satisfy the requirements of pre-processing methods in surveillance systems.

To solve the complexity issue, some models have been proposed recently to speed up the calculations. Cui et al. [28] extended spectral residual models [16] onto the temporal domain to achieve computational efficiency. However, the plausibility of spectrum analysis in saliency detection is still not clear. Recently, Alshawi [29] explored the relation between QR factorization and saliency detection owing to the capability of hardware accelerators for matrix factorization in terms of processing speed. These methods were mainly designed for images, and lack motion features when applied to videos. In contrast to the above methods, our proposed model does not require hardware acceleration, is very fast, and is more specifically concerned with motion saliency. In Table 1, we summarize the state-of-the-art video saliency methods. Please see [30–32] for details and comparisons of these studies.

Table 1. Overview of various state of the art video saliency methods.

Models	Features	Type	Description
Zhong et al. [18]	color, orientation, texture, motion features	Fusion model	Dynamic consistent optical flow for motion saliency map
Mauthner et al. [19]	color, motion features		Encoding-based approach to approximate joint feature distribution
Wang et al. [20]	spatial static edges, motion boundary edges		Super-pixels based, geodesic distance to compute the probability for object segmentation
Yubing et al. [21]	color, intensity, orientation motion vector field		Motion saliency and stationary saliency are merged with Gaussian distance weights
Z. Ren et al. [22]	sparse representation, motion trajectories		Patch-based method, learning the reconstruction coefficients to encode the motion trajectory for motion saliency
C. Chen et al. [23]	motion gradient, color gradient		Guide fusion low level saliency map using low-rank coherency

Table 1. Cont.

Models	Features	Type	Description
Y. Xue et al. [24]	low rank, sparse decomposition	Direct-pipeline model	Stack the temporal slices along X-T and Y-T plane.
Bhattacharya et al. [25]	spatiotemporal features, color cues		Weighted sum of the sparse features along three orthogonal directions determines the salient regions
W. Wang et al. [26]	gradient flow field, local, global contrasts		Gradient flow field incorporates intra-frame and inter-frame information to highlight salient regions.
H.Kim et al. [27]	low level cues, motion distinctiveness, temporal consistency, abrupt change		Random walk with restart is used to detect spatially and temporally salient regions

3. Dynamic Mode Decomposition Background

The fluid mechanics community has been gaining interest in DMD [33,34] as a data-driven method. It approximates a set of dynamic modes that represent non-linear dynamics of the experimental data. Originally, DMD was designed for data collected in regular space–time intervals. Moreover, the DMD framework is described from an equation-free perspective, where the method relies on a sequence of snapshots generated from a dynamic system over time. It uses batch processing that combines dimensionality-reduction techniques and frequency domain techniques.

Let x_i be the vector observed from n data points taken at time t_j , $j = 1, 2, \dots, m$. We have a dynamic system with two matrices, $\mathbf{X}$ and $\mathbf{Y} \in \mathbb{R}^{n \times m}$, for m snapshots, as follows:

$$\mathbf{X} = \begin{bmatrix} | & | & | & \dots & | \\ x_1 & x_2 & x_3 & \dots & x_{m-1} \\ | & | & | & & | \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} | & | & | & \dots & | \\ x_2 & x_3 & x_4 & \dots & x_m \\ | & | & | & & | \end{bmatrix}$$

Assume that a linear mapping, $\mathbf{A}$, connects the data, x_i , to the subsequent data, x_{i+1} , and the relation between two matrices is given by the following equation:

$$\mathbf{Y}^m = \mathbf{A}\mathbf{X}^{m-1} \quad (1)$$

DMD determines eigenvectors and eigenvalues of $\mathbf{A}$, which are considered DMD modes and DMD eigenvalues. In case n is large, and solving the best-fit $\mathbf{A}$ is computationally expensive, companion matrix $\mathbf{S}$ is introduced as follows:

$$\mathbf{A}\mathbf{X}^{m-1} = \mathbf{Y}^m \approx \mathbf{X}^{m-1}\mathbf{S} \# \quad (2)$$

In [29], a robust solution using SVD decomposition is applied to $\mathbf{X}$, so equation (2) can be rewritten as:

$$\mathbf{Y}^m \approx \mathbf{U}\Sigma\mathbf{V}^*\mathbf{S} \quad (3)$$

where $\mathbf{S}$ is obtained as follows:

$$\mathbf{S} \approx \mathbf{V}\Sigma^{-1}\mathbf{U}^*\mathbf{Y}^m \quad (4)$$

The full-rank matrix, $\tilde{\mathbf{S}}$, is derived via similarity transformation of matrix $\mathbf{S}$. It defines the low-dimensional linear model of the system. After computing the eigen-decomposition of $\tilde{\mathbf{S}}$, we have:

$$\tilde{\mathbf{S}}\mathbf{W} = \mathbf{W}\mathbf{\Lambda} \quad (5)$$

where columns of $\mathbf{W}$ are eigenvectors of $\mathbf{S}$, and $\mathbf{\Lambda}$ is a diagonal matrix that contains the corresponding eigenvalues, σ_j . The eigen-decomposition of $\tilde{\mathbf{S}}$ can be related to eigenvalues and eigenvectors of $\mathbf{A}$. Then DMD modes are given by columns of Φ :

$$\Phi = \mathbf{Y}^m \mathbf{V} \Sigma^{-1} \mathbf{W} \quad (6)$$

DMD eigenvectors and DMD eigenvalues provide the spatial information and temporal information, respectively, of each mode. This information is able to capture the dynamics of $\mathbf{A}$.

The frequency of DMD modes is computed as follows:

$$\omega_j = \frac{\ln(\sigma_j)}{\delta t} \quad (7)$$

where δt is the time interval between snapshots. The low rank and sparse components are given by:

$$\mathbf{X}_{\text{DMD}} = \mathbf{X}_{\text{DMD}}^{\text{low-rank}} + \mathbf{X}_{\text{DMD}}^{\text{sparse}} \quad (8)$$

The power of DMD was recently analyzed in various domains, such as image and video processing [35–40]. Grosek and Kutz [35] considered DMD modes with a frequency near the origin as background, with other modes as foreground, as described in Equation (8). Bi et al. [36] determined video boundaries based on the amplitudes of foreground and background modes. Besides, Sikha and colleagues [37,38] adapted DMD on different color channels for image saliency.

4. The Proposed Methodology

In general, the proposed method includes two main phases: (1) generate a raw saliency map based on sparse reconstruction, and (2) apply a coarse- to fine-motion refinement process. Figure 1 shows the architecture overview of the proposed model. For decomposition, we used s-DMD [40] for fast computation on video. Then, we use a difference-of-Gaussians filter on the frequency domain to refine the map.

Figure 1. The architecture overview of the proposed model.

4.1. Motion Saliency Generation Based on s-DMD

Surveillance systems require rapid response and intelligent analysis [39]; therefore, our target is to develop a method to extract features quickly in a relatively reliable way. Although batch-processing DMD described in Section 2 performs well, it requires an entire dataset to be known in advance. Therefore, we used an extended version of DMD called s-DMD for this step. s-DMD can exploit the spatial–temporal coherence structure of the video to extract features in a streaming manner.

In our method, each frame of the video is converted to grayscale and transformed into a column vector of two matrices, $\mathbf{X}, \mathbf{Y}$, where $\mathbf{X} = [x_1, x_2, \dots, x_m] \in \mathbf{R}^{n \times m}$ and $\mathbf{Y} = [y_1, y_2, \dots, y_m] \in \mathbf{R}^{n \times m}$. For efficient computation, we resize the frame resolution before creating the data matrix. In order to compute $\tilde{\mathbf{S}}$ from Equation (5), s-DMD reformulates Equation (4) of the original DMD using the Gram–Schmidt process, which helps to update DMD computation incrementally when new frames become available. First, we compute matrix $\mathbf{Q}_X \in \mathbf{R}^{n \times r_X}$ to form the orthonormal basis of $\mathbf{X}$, and the DMD operator is given as follows:

$$\mathbf{S} = \mathbf{Q}_X^T \tilde{\mathbf{S}} \mathbf{Q}_X \quad (9)$$

$\tilde{\mathbf{S}}$ is an $r_X \times r_X$ matrix defined as

$$\tilde{\mathbf{S}} = \mathbf{Q}_X^T \mathbf{Y} \mathbf{X}^+ \mathbf{Q}_X \quad (10)$$

where $\mathbf{Q}_X^T \in \mathbf{R}^{r_X \times n}$, $\mathbf{Y} \in \mathbf{R}^{n \times m}$, $\mathbf{X}^+ \in \mathbf{R}^{m \times n}$, $\mathbf{Q}_X \in \mathbf{R}^{n \times r_X}$, $\mathbf{X}^+$ is the Moore–Penrose pseudoinverse of $\mathbf{X}$, and r_X denotes the rank of $\mathbf{X}$ and $\mathbf{Y}$. The DMD eigenvalues and modes of $\mathbf{S}$ can now be obtained from the much smaller matrix $\tilde{\mathbf{S}}$. For every pair of frames, s-DMD updates the computation to generate a set of DMD modes and DMD eigenvalues. When there is a new pair of frames, the number of columns for $\mathbf{Y}$ and rows for $\mathbf{X}$ increase. Therefore, to compute $\tilde{\mathbf{S}}$ without storing the previous snapshot, we determine the orthonormal bases of $\mathbf{X}$ and $\mathbf{Y}$ as $\mathbf{Q}_X \in \mathbf{R}^{n \times r_X}$, $\mathbf{Q}_Y \in \mathbf{R}^{n \times r_Y}$. The coming pair of snapshots may very large, so they can be projected onto a low-dimensional space given as: $\tilde{\mathbf{X}} = \mathbf{Q}_X^T \mathbf{X}$, $\tilde{\mathbf{Y}} = \mathbf{Q}_Y^T \mathbf{Y}$, and we then define new matrices, $\mathbf{A} = \tilde{\mathbf{Y}} \tilde{\mathbf{X}}^T \in \mathbf{R}^{r_Y \times r_X}$ and $\mathbf{G}_X = \tilde{\mathbf{X}} \tilde{\mathbf{X}}^T \in \mathbf{R}^{r_X \times r_X}$. If the size of $\mathbf{Q}_X$ is larger than the given rank, we apply proper orthogonal decomposition (POD) compression incrementally by introducing new matrix $\mathbf{G}_Y = \tilde{\mathbf{Y}} \tilde{\mathbf{Y}}^T \in \mathbf{R}^{r_Y \times r_Y}$, where r_Y denotes the rank of $\mathbf{Y}$, and compute leading eigenvalues and eigenvectors of $\mathbf{G}_X, \mathbf{G}_Y$. In order to update operator $\tilde{\mathbf{S}}$, we use the identity matrix $\mathbf{X}^+ = \mathbf{X}^T (\mathbf{X} \mathbf{X}^T)^+$, and Equation (10) is rewritten as follows:

$$\tilde{\mathbf{S}} = \mathbf{Q}_X^T \mathbf{Q}_Y \mathbf{A} \mathbf{G}_X^+ \quad (11)$$

In our case, rank r_X, r_Y is much less than m , which is the number of snapshots in the video, so $\tilde{\mathbf{S}}$ can be updated incrementally. Moreover, we consider giving more weight to the recent frames by introducing weight parameter p while updating matrices $\mathbf{A}, \mathbf{G}_X, \mathbf{G}_Y$. DMD modes and DMD eigenvalues can be derived from eigenvector and eigenvalues of $\tilde{\mathbf{S}}$ according to Equation (5) in a streaming manner. The s-DMD mode is computed according to [31]:

$$\boldsymbol{\Phi} = \mathbf{Q}_X \mathbf{W} \quad (12)$$

The DMD approximation data can be reconstructed as follows:

$$\mathbf{X}_{\text{DMD}}(t) = \boldsymbol{\Phi} \exp(\Omega t) \mathbf{b} = \boldsymbol{\Phi}_p \exp(\omega_p t) \mathbf{b}_p + \sum_{j \neq p} \boldsymbol{\Phi}_j \exp(\omega_j t) \mathbf{b}_j \quad (13)$$

where $\mathbf{b}_j$ is the initial amplitude of each mode, $\boldsymbol{\Phi}$ is a matrix where columns are DMD eigenvectors, $\boldsymbol{\Phi}_j$, and Ω is a diagonal matrix where the entries are eigenvalues, ω_j . Stationary regions are related to DMD modes with frequency $\omega_j \approx 0$, and these modes represent a region that slowly varies in time. Moving regions are selected from the remaining frequencies. Based on this calculation, the approximate sparse components are computed as follows:

$$\mathbf{X}_{\text{DMD}}^{\text{sparse}} = \mathbf{X}_{\text{DMD}} - |\mathbf{X}_{\text{DMD}}^{\text{Low-Rank}}| \quad (14)$$

According to Equation (13), s-DMD decomposes the video sequence into three matrices: DMD mode matrix $\boldsymbol{\Phi}$, singular values matrix Λ , and amplitude matrix $\mathbf{b}$. The mode matrix represents the relative spatial and temporal information of the scene over time. The singular values matrix is the feature of these regions in the video. The amplitude matrix represents the weighted features of these

modes in each frame, or how much these regions have changed in the video. When objects move across the scene, this model captures the energy of temporal modes corresponding to moving regions through the sparse reconstruction process. Therefore, s-DMD can be used to extract the salient region from the video.

4.2. From Coarse to Fine Motion Saliency Map

The sparse components of the video generated in Section 4.1 are subjected to the refinement process. To suppress non-salient pixels falsely detected in the sparse components, the saliency map is subjected to the difference-of-Gaussians filter in the frequency domain. The proposed coarse-to fine-motion process can suppress interference effectively. The DoG filter is known as a feature enhancement that preserves spatial information that lies within the range of frequencies. It is a combination of low-pass filtering and high-pass filtering. Given an image, $\mathbf{f}$, the DoG applied on $\mathbf{f}$ is defined as:

$$\mathbf{r}_\sigma = \mathbf{f} * \mathbf{g}_{\sigma_1} - \mathbf{f} * \mathbf{g}_{\sigma_2} \text{ with } \sigma_1 > \sigma_2 \quad (15)$$

where $g(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-(x^2)/2\sigma^2}$ is the Gaussian kernel with standard deviation σ , $*$ represents the convolution of the image with the Gaussian kernel, and $\mathbf{f}$ denotes the input image. In our case, we observe that falsely detected salient pixels are often distributed on low-frequency components of the raw saliency map. Therefore, we apply DoG to the sparse components derived from Section 4.1 to suppress these false detections. Compared to the state-of-the-art methods, such as Itti [7], GBVS [8], and spectral residual (SR) [16], they perform low-pass filtering using very low-frequency contents of the image in the spatial domain. Our method applies the DoG in a different way. First, we apply DoG to the frequency domain using a discrete cosine transform (DCT) in the Fourier transform. Secondly, we only compute the DoG on sparse components of the image. This step is similar to traditional DoG, but considers the information generated by different frequencies in the spectrum of the sparse components. Compared with traditional multi-scale DoG, the result is smoother, more accurate, and more efficient in computation. The Fourier transform of Equation (14) to express the DoG in the frequency domain is as follows:

$$\text{FreS} = \mathbf{F} \left[\int_0^{+\infty} (\mathbf{X}_{\text{DMD}}^{\text{sparse}} * (\mathbf{g}_{\sigma_1} - \mathbf{g}_{\sigma_2})) d\sigma \right] \quad (16)$$

where $\mathbf{F}$ denotes the Fourier transform. We used a DoG with $\sigma_1 = 2$, $\sigma_2 = 10$ in the experiments. The proposed DoG removes falsely detected non-salient pixels, and smooths the result. The final saliency map is obtained as:

$$\text{FinalS} = [\mathbf{F}^{-1}\{\mathbf{R}\}]^2 \quad (17)$$

where $\mathbf{F}^{-1}$ denotes the inverse fourier transform of the image. The overall algorithms of the proposed method are summarized in Algorithm 1 and Algorithm 2. The first algorithm is the modified s-DMD for generating DMD modes and DMD eigenvalues. The second algorithm is to generate and refine the saliency map based on the output of the s-DMD module.

Algorithm 1: s-DMD for motion saliency.

Input: $x_i, y_i, r_0, \text{scale}$
Output: ϕ, Λ
for $i=1:m$ **do**

% generating s-DMD modes.

$x_i = \text{resize}(x_i, \text{scale})$

$y_i = \text{resize}(y_i, \text{scale})$

$e_x = x_i - x_i Q_X Q_X^T$

$e_y = y_i - y_i Q_Y Q_Y^T$

if $\|e_x\| > \varepsilon$ **then**

$Q_X = [Q_X e_x / \|e_x\|]$

if $\|e_y\| > \varepsilon$ **then**

$Q_Y = [Q_Y e_y / \|e_y\|]$

if $r_X > r_0$ or $r_Y > r_0$ **then**

$G_X \leftarrow V_X^T G_X V_X$ V_X : leading eigenvectors of G_X
 $G_Y \leftarrow V_Y^T G_Y V_Y$ V_Y : leading eigenvectors of G_Y
 $A \leftarrow V_Y^T A V_X$
 $Q_X \leftarrow Q_X V_X$
 $Q_Y \leftarrow Q_Y V_Y$

$\tilde{x}_i \leftarrow Q_X^T x_i$

$\tilde{y}_i \leftarrow Q_Y^T y_i$

$G_X \leftarrow G_X * p + (1-p) \tilde{x}_i \tilde{x}_i^T$

$G_Y \leftarrow G_Y * p + (1-p) \tilde{y}_i \tilde{y}_i^T$

$A \leftarrow G_Y * p + (1-p) \tilde{y}_i \tilde{x}_i^T$

$\tilde{S} = Q_X^T Q_Y A G_X^+$

$\tilde{S} W = W \Lambda$ Λ : DMD eigenvalues

$\phi = Q_X W$ ϕ : DMD modes

end

Algorithm 2: Generation of motion saliency map

Input: ϕ, Λ
Output: FinalS
for $i=1:m$ **do**

%generating saliency map

$\omega = \log(\text{diag}(\Lambda))$

$\text{index} = \min(\omega)$

$\omega_{bg} = \omega(:, \text{index})$

$b_{bg} = \omega_{bg} / x_i$

$\phi_{bg} = \phi(\min(\omega))$

$b = \omega / x_i$

$X_{DMD} = \phi \exp(\omega) b$

$X_{DMD}^{\text{Low-Rank}} = \phi_{bg} \exp(\omega_{bg}) b_{bg}$

$X_{DMD}^{\text{sparse}} = X_{DMD} - |X_{DMD}^{\text{Low-Rank}}|$

%refinement process

$FX_{DMD}^{\text{sparse}} = \text{dct2}(X_{DMD}^{\text{sparse}})$

$\text{kernelF} = g_{\text{maxfq}} - g_{\text{minfq}}$

$\text{FinalS} = \text{idct2}(FX_{DMD}^{\text{sparse}} * \text{kernelF})$

$\text{FinalS} = \text{resize}(\text{FinalS}, 1/\text{scale})$

end

5. Experiments Results

We evaluate the performance of the proposed method on the standard Change Detection 2014 (CDNet2014s) dataset [41]. The dataset contains different categories in various environments. We select 12 videos from five categories for details analysis. The salient region labeled from the human is used as ground-truth. In the experiments, we keep the resolution of the saliency maps the same as the original solutions of the frames. Video information for evaluation is summarized in Table 2. All of the tests were run using Matlab R2016a. The computer is equipped with 16 GB of memory.

5.1. Evaluation Metrics

We used various standard performance metrics to evaluate the performance of the algorithms, including precision recall curve (PR curve), mean absolute error (MAE), area under the Receiver Operating Characteristics (ROC) curve (AUC-Borji) [42], structure measure (S-measure) [43], normalized scan path saliency (NSS) [44], and correlation coefficient (CC) [45]. They are defined as follows.

PR curve: A precision value is the ratio of all salient pixels distributed correctly to all pixels in the image. Recall is the fraction of detected salient pixels to all ground truth pixels. The saliency map was converted to binary image S using a fixed threshold, which was used to compare against ground truth, G , to compute the precision and recall. PR curves show how reliable the saliency maps are, and how well they assign a salient score:

$$\begin{aligned} \text{Precision} &= \frac{|S \cap G|}{|S|} \\ \text{Recall} &= \frac{|S \cap G|}{|G|} \end{aligned} \quad (18)$$

MAE: Mean absolute error provides a method to measure the difference between the saliency map and ground truth. MAE was normalized to [0, 1], which is defined as follows:

$$\text{MAE} = \frac{1}{W+H} \sum_{x=1}^W \sum_{y=1}^H \|S(x, y) - G(x, y)\| \quad (19)$$

AUC-Borji: The area under the ROC curve (AUC) [42] measures the area under true positive and false positive rates (ROC curve), and ranges between 0 and 1. A perfect model has an AUC of 1.

S-measure: The structure measure [43] evaluates the structure information that pixel-based metrics (precision, recall) do not consider. The S-measure score is expressed as:

$$\text{Smeasure} = \frac{2\bar{x}\bar{y}}{(\bar{x})^2 + (\bar{y})^2} \cdot \frac{2\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2} \cdot \frac{\sigma_{xy}}{\sigma_x\sigma_y} \quad (20)$$

where x and y are vectors of saliency and ground truth values, respectively, $\bar{x}, \bar{y}$ denote mean values, and σ denotes covariance values.

NSS: Normalized scan path saliency [44] measures the average saliency values at fixation pixels in the normalized saliency map. Given saliency map P and binary fixation map Q^B , the NSS score is defined as:

$$\text{NSS}(P, Q^B) = \frac{1}{N} \sum_i \bar{P}_i x Q_i^B \text{ where } N = \sum_i Q_i^B \text{ and } \bar{P} = \frac{P - \mu(P)}{\sigma(P)} \quad (21)$$

CC: The correlation coefficient [45] measures the Euclidean distance between the saliency map and the normalized empirical saliency map. The CC has a large value when two saliency maps have the same magnitude. Given saliency map P and fixation map Q^D , the CC score is defined as:

$$\text{CC}(P, Q^D) = \frac{\sigma(P, Q^D)}{\sigma(P)\sigma(Q^D)} \text{ where } \sigma \text{ is covariance} \quad (22)$$

Table 2. The summarization of videos information in Change Detection 2014 (CDNet2014s) dataset used in performance evaluation.

Category	Video Sequence	No. of Frames	Frame Resolution	Description
Baseline	highway	1700	320 × 240	A mixture of others category
	office	2050	360 × 240	
	pedestrian	1099	360 × 240	
	PETS2006	120	720 × 576	
Dynamic Background	canoe	1189	320 × 240	Strong background motion like waters, trees
	overpass	3000	320 × 240	
Bad Weather	blizzard	7000	720 × 480	Poor weather condition like snow, fog
	skating	3900	540 × 360	
Camera Jitter	badminton	1150	720 × 480	Vibrational cameras in outdoor environment
	traffic	1570	320 × 240	
Intermittent Object Motion	sofa	2750	320 × 240	Some objects move then stop again
	streetlight	3200	320 × 240	

5.2. Comparison Results of Various State-of-the-Art Methods

In our method, we set weighting parameter p to 0.5, the scaling factor parameter to 0.25, and the $\max_rank$ parameter to 100 when performing experiments. The quantitative results of the CDNet2014s dataset are reported in Table 3 for detailed analysis. The proposed method showed the best results with the PETS2006 video. For other videos in the baseline category, MAE score decreased significantly. In other challenging categories, which consist of dynamic or interrupted motion, the proposed method showed comparative performance in terms of accuracy and structure measure.

Table 3. The accuracy performance of the proposed method on CDNet2014s dataset.

Video Sequence	Abbr.	MAE	AUC-Borji	S-Measure	NSS	CC
highway	HIG	0.071	0.801	0.499	2.158	0.561
office	OFF	0.069	0.719	0.464	1.298	0.426
pedestrian	PED	0.130	0.659	0.658	2.245	0.400
PETS2006	PET	0.051	0.842	0.479	3.310	0.457
canoe	CAN	0.194	0.522	0.390	0.505	0.136
overpass	OVE	0.116	0.514	0.357	0.196	0.074
blizzard	BLI	0.017	0.526	0.344	1.094	0.167
skating	SKA	0.136	0.481	0.344	0.511	0.136
sidewalk	SID	0.345	0.483	0.149	0.071	0.111
traffic	TRA	0.054	0.581	0.294	0.859	0.230
sofa	SOF	0.101	0.623	0.459	0.991	0.305
streetlight	STR	0.852	0.500	0.064	0.002	0.010
Average		0.178	0.604	0.375	1.103	0.251

To demonstrate the efficiency of our proposal, we compared the proposed method with various state of the art methods including image saliency methods (ITTI [7], GBVS [8], SUN [9], saliency by self-resemblance SSR [46], fast and efficient saliency (FES) [47], quaternion-based spectral saliency (QSS) [48], high-dimensional color transform (HDCT) [49], principle component analysis (PCA) [50], region stability saliency (RSS) [51]) and video saliency method (consistent video saliency (CVS) [26], random walk with restart (RWRS) [27]), the implementation source code were collected from C.Wloka et al. [52] and the project page of the authors. We keep all parameters of the author's proposal as the default.

Figure 2 provides the performance of the compared algorithms using PR and ROC curves. The green thick dashed lines represents the proposed results. As shown in Figure 2a, our method outperforms others image saliency method in PR curves. The recall values of some image saliency are very small because their saliency maps cannot locate salient points well on salient objects. Moreover,

our method achieves high precision rate that says it can detect salient object well. Figure 2b shows that our method attains higher positive rates for low false positive rates. The area under ROC curves also shows that our method performs slightly better compared with other algorithms.

Figure 2. The comparison of the proposed method on Change Detection 2014 Dataset. (a) Precision and Recall (PR) curve; (b) ROC curve.

Tables 4–8 shows the comparison results of various metrics on CDNet2014s Dataset. The first, second and third-ranked values of the corresponding metrics are highlighted in red, blue and black colors. The obtained results indicate that our method has competitive performance compared with other state-of-the-art algorithms. As shown in Tables 4 and 5, the MAE and AUC-Borji score of the proposed method is always in top four performance in most of the cases. Although RSS model has lower MAE score in many cases, our method significantly outperforms this model in terms of AUC score. Our method achieves the highest AUC score in four videos of baseline category and has slightly lower MAE score than two complex models (CVS, RWRS) in case of *highway* and *office* video. In case the scene is disturbed with complex motions such as dynamic background or bad weather condition, the AUC score of the proposed method decreases slightly but still better than many state-of-the-art models.

Table 4. Mean absolute error (MAE) comparison of the proposed method on CDNet2014s dataset.

Methods	HIG	OFF	PED	PET	CAN	OVE	BLI	SKA	SID	TRA	SOF	STR	Avg.
ITTI	0.200	0.252	0.237	0.208	0.229	0.191	0.135	0.211	0.425	0.233	0.210	0.681	0.268
SUN	0.244	0.219	0.207	0.316	0.349	0.194	0.146	0.274	0.288	0.323	0.251	0.783	0.300
SSR	0.245	0.265	0.249	0.309	0.110	0.251	0.354	0.383	0.360	0.122	0.310	0.777	0.311
GBVS	0.213	0.238	0.206	0.169	0.242	0.252	0.130	0.250	0.417	0.237	0.211	0.677	0.270
FES	0.104	0.093	0.068	0.086	0.051	0.097	0.037	0.078	0.271	0.058	0.105	0.848	0.158
QSS	0.222	0.139	0.134	0.199	0.259	0.224	0.100	0.170	0.342	0.169	0.187	0.784	0.244
HDCT	0.112	0.100	0.079	0.113	0.044	0.110	0.052	0.118	0.240	0.062	0.169	0.781	0.165
PCA	0.241	0.194	0.223	0.116	0.351	0.183	0.085	0.062	0.400	0.167	0.217	0.773	0.251
RSS	0.086	0.087	0.032	0.035	0.025	0.048	0.008	0.036	0.255	0.058	0.082	0.872	0.135
CVS	0.111	0.126	0.079	0.134	0.046	0.135	0.058	0.082	0.263	0.074	0.151	0.750	0.167
RWRS	0.184	0.253	0.116	0.143	0.064	0.152	0.152	0.077	0.232	0.071	0.143	0.731	0.193
Proposed	0.071	0.069	0.130	0.051	0.194	0.116	0.017	0.136	0.345	0.054	0.101	0.852	0.178

Table 5. Area under the ROC curve (AUC)-Borji comparison of the proposed method on CDNet2014s dataset.

Methods	HIG	OFF	PED	PET	CAN	OVE	BLI	SKA	SID	TRA	SOF	STR	Avg.
ITTI	0.687	0.634	0.613	0.767	0.510	0.437	0.439	0.449	0.471	0.510	0.712	0.499	0.561
SUN	0.629	0.621	0.562	0.677	0.467	0.547	0.423	0.442	0.491	0.499	0.685	0.500	0.545
SSR	0.748	0.695	0.595	0.754	0.530	0.595	0.473	0.483	0.485	0.546	0.762	0.487	0.596
GBVS	0.626	0.667	0.590	0.768	0.490	0.407	0.398	0.434	0.477	0.481	0.676	0.500	0.543
FES	0.567	0.676	0.493	0.745	0.472	0.488	0.262	0.422	0.491	0.499	0.645	0.503	0.522
QSS	0.729	0.670	0.649	0.818	0.527	0.543	0.470	0.480	0.488	0.530	0.743	0.485	0.594
HDCT	0.692	0.718	0.594	0.740	0.521	0.451	0.512	0.471	0.472	0.554	0.732	0.499	0.580
PCA	0.685	0.723	0.581	0.709	0.514	0.449	0.406	0.467	0.471	0.548	0.775	0.488	0.568
RSS	0.549	0.541	0.485	0.589	0.442	0.502	0.430	0.436	0.502	0.502	0.548	0.501	0.502
CVS	0.737	0.634	0.604	0.743	0.513	0.573	0.496	0.457	0.472	0.523	0.693	0.508	0.579
RWRS	0.769	0.700	0.644	0.758	0.526	0.630	0.630	0.488	0.474	0.568	0.756	0.509	0.621
Proposed	0.801	0.719	0.659	0.842	0.522	0.514	0.526	0.481	0.483	0.581	0.623	0.500	0.604

Table 6. S-measure comparison of the proposed method on CDNet2014s dataset.

Methods	HIG	OFF	PED	PET	CAN	OVE	BLI	SKA	SID	TRA	SOF	STR	Avg.
ITTI	0.452	0.400	0.501	0.401	0.382	0.353	0.453	0.378	0.252	0.371	0.473	0.241	0.388
SUN	0.401	0.413	0.444	0.402	0.405	0.357	0.427	0.394	0.138	0.390	0.443	0.144	0.363
SSR	0.447	0.513	0.449	0.401	0.351	0.375	0.298	0.436	0.224	0.303	0.443	0.132	0.364
GBVS	0.431	0.454	0.494	0.413	0.380	0.357	0.452	0.390	0.247	0.373	0.454	0.242	0.391
FES	0.377	0.499	0.455	0.416	0.256	0.323	0.477	0.305	0.079	0.279	0.464	0.078	0.334
QSS	0.434	0.464	0.480	0.419	0.427	0.367	0.483	0.364	0.208	0.304	0.481	0.130	0.380
HDCT	0.422	0.501	0.465	0.360	0.244	0.291	0.299	0.347	0.053	0.276	0.448	0.133	0.320
PCA	0.439	0.528	0.460	0.401	0.466	0.294	0.497	0.286	0.227	0.363	0.442	0.150	0.379
RSS	0.341	0.341	0.442	0.380	0.217	0.307	0.307	0.274	0.078	0.258	0.380	0.052	0.281
CVS	0.501	0.442	0.475	0.342	0.237	0.322	0.290	0.257	0.046	0.247	0.463	0.161	0.315
RWRS	0.468	0.455	0.522	0.363	0.225	0.317	0.317	0.270	0.068	0.261	0.463	0.171	0.325
Proposed	0.499	0.464	0.658	0.479	0.390	0.357	0.344	0.344	0.149	0.294	0.459	0.064	0.375

Table 7. Normalized scan path saliency (NSS) comparison of the proposed method on CDNet2014s dataset.

Methods	HIG	OFF	PED	PET	CAN	OVE	BLI	SKA	SID	TRA	SOF	STR	Avg.
ITTI	0.902	0.527	1.283	1.352	0.323	0.149	1.398	0.154	0.106	0.199	0.994	0.007	0.616
SUN	0.483	0.466	0.414	0.606	0.099	0.096	1.317	0.097	0.002	0.129	0.769	0.007	0.374
SSR	1.035	0.985	0.636	0.988	0.678	0.329	0.982	0.282	0.019	0.303	1.065	0.042	0.612
GBVS	0.641	0.770	1.207	1.718	0.245	0.256	1.196	0.105	0.086	0.373	0.854	0.001	0.621
FES	0.533	1.286	0.576	1.844	0.194	0.049	0.082	0.056	0.051	0.302	0.786	0.016	0.481
QSS	1.030	0.916	1.239	1.721	0.629	0.107	1.856	0.350	0.007	0.265	1.212	0.045	0.781
HDC	0.985	1.361	1.158	1.281	0.468	0.163	0.730	0.291	0.095	0.364	0.975	0.004	0.656
PCA	0.722	1.206	0.871	1.178	0.431	0.121	1.513	0.258	0.096	0.373	1.190	0.041	0.667
RSS	0.639	0.409	0.826	0.641	0.102	0.054	0.634	0.219	0.011	0.406	0.190	0.016	0.346
CVS	1.324	0.962	1.164	1.285	0.342	0.206	0.633	0.167	0.121	0.242	1.025	0.028	0.625
RWRS	1.448	0.852	1.725	1.142	0.475	0.480	0.480	0.383	0.084	0.507	1.196	0.032	0.734
Proposed	2.158	1.298	2.245	3.310	0.505	0.196	1.094	0.511	0.071	0.859	0.991	0.002	1.103

Table 8. Correlation coefficient (CC) comparison of the proposed method on CDNet2014s dataset.

Methods	HIG	OFF	PED	PET	CAN	OVE	BLI	SKA	SID	TRA	SOF	STR	Avg.
ITTI	0.306	0.169	0.256	0.193	0.104	0.013	0.218	0.057	0.170	0.094	0.264	0.031	0.156
SUN	0.142	0.151	0.084	0.081	0.033	0.024	0.192	0.033	0.003	0.057	0.204	0.030	0.086
SSR	0.299	0.325	0.121	0.143	0.166	0.092	0.163	0.090	0.031	0.115	0.288	0.185	0.168
GBVS	0.231	0.254	0.244	0.233	0.089	0.039	0.191	0.047	0.138	0.085	0.230	0.004	0.149
FES	0.187	0.428	0.120	0.244	0.071	0.008	0.017	0.028	0.081	0.124	0.231	0.071	0.134
QSS	0.285	0.304	0.236	0.250	0.145	0.043	0.274	0.101	0.011	0.061	0.324	0.196	0.186
HDCT	0.339	0.449	0.229	0.167	0.138	0.021	0.120	0.110	0.151	0.140	0.279	0.018	0.180
PCA	0.238	0.398	0.176	0.158	0.127	0.003	0.246	0.096	0.153	0.150	0.331	0.178	0.188
RSS	0.180	0.133	0.155	0.094	0.028	0.013	0.096	0.060	0.018	0.103	0.048	0.069	0.083
CVS	0.437	0.314	0.229	0.175	0.111	0.078	0.101	0.060	0.193	0.082	0.286	0.121	0.182
RWRS	0.414	0.277	0.326	0.179	0.135	0.124	0.124	0.119	0.133	0.144	0.329	0.138	0.204
Proposed	0.561	0.426	0.400	0.457	0.136	0.074	0.167	0.136	0.111	0.230	0.305	0.010	0.251

In Table 6, we measure structure similarity score (S-measure) of all methods. This metric demonstrate how well each model generates more completed object. Our method preserves global structure quite well in the baseline category; in other categories, our method shows comparative results.

Moreover, we evaluate the performance of the proposed method using NSS and CC metric. NSS metric uses the absolute saliency values in calculation. NSS is quite sensitive to false positive values; therefore, many low false positive values may contribute to low NSS values. CC metric evaluates the similarity of saliency magnitudes at fixation locations. As shown in Tables 7 and 8, the proposed method achieves best scores in the baseline category and comparative results in another categories compared with other models. This shows that our method achieves relatively reliable accuracy result.

To further demonstrate the effectiveness of the s-DMD core, we compare the computational time for all methods under different resolutions in Table 9. The execution time of twelve algorithms was measured on Matlab 2016a. Although two models CVS and RWRS achieve better accuracy score in some categories, their complex model demands long run time for generating the saliency map. The CVS model requires more than 20 s for computing the optical flow, RWRS requires more than 10 s for the core process. Our method can reach 22 fps in Matlab environment for 320×240 px videos. The proposed method is much faster than these complex models, which satisfies the requirement of pre-processing algorithm in surveillance system.

Table 9. The average time complexity (in seconds) of difference resolutions *.

Frame Size	Ours	Itti	SUN	SSR	GBVS	FES	QSS	CVS	HDCT	PCA	RSS	RWRS
320×240 px	0.043	0.175	1.498	0.680	0.377	0.051	0.029	5.045	3.454	2.014	0.130	10.813
720×480 px	0.130	0.217	8.096	0.841	0.380	0.112	0.054	20.98	7.256	11.308	0.152	16.636

* The implementation codes were performed in Matlab R2016a.

In Tables 10 and 11, we show the visual comparison of our method and others image saliency map in which each column shows saliency maps obtained from each method for various categories in each row. Some image saliency methods do not distribute salient points well on the moving object due to the lack of temporal information in their models. SUN does not perform well in detecting salient objects due to the limitation of using local features. FES and QSS cannot preserve the shape of object well. The salient points of RSS mostly distribute on the edge and the saliency map does not completed.

Table 10. Visual comparison of saliency maps generated of image saliency methods.

	Input	GT	Ours	Itti	SUN	SSR	GBVS	FES	QSS	PCA	HDCT
HIG
OFF
PED
PET
CAN
OVE
BLI
SKA

Table 10. Cont.

	Input	GT	Ours	Itti	SUN	SSR	GBVS	FES	QSS	PCA	HDCT
SID
TRA
SOF

Table 11. Visual comparison of video saliency methods.

	Input	GT	Ours	RSS	CVS	RWRS
HIG
OFF
PED
PET
CAN
OVE
BLI
SKA
SID
TRA
SOF

In order to validate the competitiveness of our proposal with respect to other models, we provided the statistical test in terms of AUC, NSS, and CC metrics. We use Matlab to perform the t -test at $p < 0.05$ for 5% significance as in [53]. The results are illustrated in Figures 3–5. There are two values “1” and “0” in the table, which indicate the statistical significance of the difference between every pair of compared models. If the mean value of the model in the row is larger than the model in the column, it is represented by “1”; otherwise, it is “0”.

	a	b	c	d	e	f	g	h	i	j	k	l
a	0	1	1	1	1	1	1	1	1	1	1	1
b	0	0	1	1	1	1	1	1	1	0	0	1
c	0	0	0	0	0	0	0	0	1	0	0	0
d	0	0	1	0	0	0	0	0	1	0	0	0
e	0	0	1	1	0	1	1	0	1	0	0	1
f	0	0	1	1	0	0	1	0	1	0	0	1
g	0	0	1	1	0	0	0	0	1	0	0	1
h	0	0	1	1	1	1	1	0	1	0	0	1
i	0	0	0	0	0	0	0	0	0	0	0	0
j	0	1	1	1	1	1	1	1	1	0	1	1
k	0	1	1	1	1	1	1	1	1	0	0	1
l	0	0	1	0	0	0	0	0	1	0	0	0

(a)

	a	b	c	d	e	f	g	h	i	j	k	l
a	0	1	1	1	1	1	1	1	1	1	1	1
b	0	0	1	1	1	1	1	1	1	0	1	1
c	0	0	0	0	0	0	0	0	0	0	0	1
d	0	0	1	0	0	0	0	0	0	0	0	1
e	0	0	1	1	0	1	1	0	1	0	0	1
f	0	0	1	1	0	0	1	0	1	0	0	1
g	0	0	1	1	0	0	0	0	1	0	0	1
h	0	0	1	1	1	1	1	0	1	0	0	1
i	0	0	1	0	0	0	0	0	0	0	0	1
j	0	1	1	1	1	1	1	1	1	0	1	1
k	0	0	1	1	1	1	1	0	1	0	0	1
l	0	0	0	0	0	0	0	0	0	0	0	0

(b)

	a	b	c	d	e	f	g	h	i	j	k	l
a	0	1	1	1	1	1	1	1	1	1	1	1
b	0	0	1	1	1	1	1	1	1	1	1	1
c	0	0	0	0	0	0	0	0	1	0	0	1
d	0	0	1	0	0	0	0	0	1	0	0	1
e	0	0	1	1	0	1	1	1	1	0	1	1
f	0	0	1	1	0	0	1	1	1	0	1	1
g	0	0	1	1	0	0	0	0	1	0	0	1
h	0	0	1	1	0	1	0	1	0	1	0	1
i	0	0	0	0	0	0	0	0	0	0	0	1
j	0	0	1	1	1	1	1	1	1	0	1	1
k	0	0	1	1	0	0	1	1	1	0	0	1
l	0	0	0	0	0	0	0	0	0	0	0	0

(c)

	a	b	c	d	e	f	g	h	i	j	k	l
a	0	1	1	1	0	1	0	1	1	1	1	1
b	0	0	0	0	0	0	0	0	1	0	0	1
c	0	1	0	1	0	1	0	1	1	0	0	1
d	0	1	0	0	0	1	0	0	1	0	0	1
e	0	1	1	1	0	1	0	1	1	1	1	1
f	0	0	0	0	0	0	0	0	1	0	0	1
g	1	1	1	1	1	1	0	1	1	1	1	1
h	0	1	0	1	0	1	0	0	1	0	0	1
i	0	0	0	0	0	0	0	0	0	0	0	0
j	0	1	1	1	0	1	0	1	1	0	1	1
k	0	1	1	1	0	1	0	1	1	0	0	1
l	0	0	0	0	0	0	0	0	1	0	0	0

(d)

	a	b	c	d	e	f	g	h	i	j	k	l
a	0	1	0	1	0	1	1	1	1	1	1	1
b	0	0	0	1	0	1	0	1	1	1	0	1
c	0	1	0	1	0	1	1	1	1	1	1	1
d	0	0	0	0	0	1	0	0	1	0	0	1
e	1	1	1	1	0	1	1	1	1	1	1	1
f	0	0	0	0	0	0	0	0	1	0	0	1
g	0	1	0	1	0	1	0	1	1	1	1	1
h	0	0	0	1	0	1	0	0	1	1	0	1
i	0	0	0	0	0	0	0	0	0	0	0	0
j	0	0	0	1	0	1	0	0	1	0	0	1
k	0	1	0	1	0	1	0	1	1	1	0	1
l	0	0	0	0	0	0	0	0	1	0	0	0

(e)

	a	b	c	d	e	f	g	h	i	j	k	l
a	0	1	0	1	0	1	1	1	1	1	1	1
b	0	0	0	1	0	1	0	1	1	1	0	1
c	0	1	0	1	0	1	1	1	1	1	1	1
d	0	0	0	0	0	1	0	0	1	0	0	1
e	1	1	1	1	0	1	1	1	1	1	1	1
f	0	0	0	0	0	0	0	0	1	0	0	1
g	0	1	0	1	0	1	0	1	1	1	1	1
h	0	0	0	1	0	1	0	0	1	1	0	1
i	0	0	0	0	0	0	0	0	0	0	0	0
j	0	0	0	1	0	1	0	1	0	1	0	1
k	0	1	0	1	0	1	0	1	1	1	0	1
l	0	0	0	0	0	0	0	0	1	0	0	0

(f)

	a	b	c	d	e	f	g	h	i	j	k	l
a	0	1	1	1	1	1	1	1	1	1	1	1
b	0	0	1	1	1	0	1	0	1	0	1	1
c	0	0	0	0	0	0	0	0	1	0	0	0
d	0	0	1	0	0	0	1	0	1	0	0	1
e	0	0	1	0	0	0	1	0	1	0	0	1
f	0	1	1	1	1	0	1	0	1	0	1	1
g	0	0	1	0	0	0	0	0	1	0	0	1
h	0	1	1	1	1	1	1	0	1	1	1	1
i	0	0	0	0	0	0	0	0	0	0	0	0
j	0	1	1	1	1	1	1	0	1	0	1	1
k	0	0	1	0	0	0	1	0	1	0	0	1
l	0	0	1	0	0	0	0	0	1	0	0	0

(g)

	a	b	c	d	e	f	g	h	i	j	k	l
a	0	1	1	1	1	1	1	1	1	1	1	1
b	0	0	1	0	0	0	0	1	0	1	1	1
c	0	0	0	0	0	0	0	0	0	0	0	1
d	0	1	1	0	1	0	1	0	1	0	1	1
e	0	0	1	0	0	0	1	0	1	0	1	1
f	0	1	1	1	1	0	1	1	1	0	1	1
g	0	0	1	0	0	0	0	0	1	0	1	1
h	0	1	1	0	1	0	1	0	1	0	1	1
i	0	0	1	0	0	0	0	0	0	0	0	1
j	0	1	1	1	1	1	1	1	1	0	1	1
k	0	0	1	0	0	0	0	0	0	0	0	1
l	0	0	0	0	0	0	0	0	0	0	0	0

(h)

	a	b	c	d	e	f	g	h	i	j	k	l
a	0	1	1	1	1	1	1	1	1	1	1	1
b	0	0	1	0	0	0	1	0	1	0	1	1
c	0	0	0	0	0	0	0	0	0	0	0	1
d	0	1	1	0	1	0	1	1	1	0	1	1
e	0	0	1	0	0	0	1	0	1	0	1	1
f	0	1	1	1	1	0	1	1	1	0	1	1
g	0	0	1	0	0	0	0	0	1	0	1	1
h	0	1	1	0	1	0	1	0	1	0	1	1
i	0	0	1	0	0	0	0	0	0	0	0	1
j	0	1	1	1	1	1	1	1	1	0	1	1
k	0	0	0	0	0	0	0	0	0	0	0	1
l	0	0	0	0	0	0	0	0	0	0	0	0

(i)

	a	b	c	d	e	f	g	h	i	j	k	l
a	0	1	1	1	1	1	1	1	1	1	1	1
b	0	0	0	0	0	0	1	0	1	0	0	1
c	0	0	0	0	1	0	1	0	1	0	0	1
d	0	1	1	0	1	0	1	0	1	1	1	1
e	0	0	0	0	0	0	1	0	1	0	0	1
f	0	1	1	0	1	0	1	0	1	1	1	1
g	0	0	0	0	0	0	0	0	1	0	0	1
h	0	1	1	1	1	1	1	0	1	1	1	1
i	0	0	0	0	0	0	0	0	0	0	0	0
j	0	1	1	0	1	0	1	0	1	0	1	1
k	0	1	1	0	1	0	1	0	1	0	0	1
l	0	0	0	0	0	0	0	0	1	0	0	0

(j)

	a	b	c	d	e	f	g	h	i	j	k	l
a	0	1	1	1	1	1	1	1	1	1	1	1
b	0	0	0	0	0	0	1	0	1	1	1	1
c	0	1	0	1	1	1	1	1	1	1	1	1
d	0	1	0	0	1	1	1	0	1	1	1	1
e	0	0	0	0	0	0	1	0	1	1	1	1
f	0	1	0	0	1	0	1	0	1	1	1	1
g	0	0	0	0	0	0	0	0	1	0	1	1
h	0	1	0	0	1	1	1	0	1	1	1	1
i	0	0	0	0	0	0	0	0	0	0	0	1
j	0	0	0	0	0	0	0	0	1	0	1	1
k	0	0	0	0	0	0	0	0	1	0	0	1
l	0	0	0	0	0	0	0	0	0	0	0	1

Figure 3. The results of statistical test over twelve methods. We abbreviate twelve methods from “a” to “h”: a = OURS, b = CVS, c = FES, d = GBVS, e = HDCT, f = IKN, g = PCA, h = QSS, i = RSS, j = RWRS, k = SSR, l = SUN. (a) HIG-AUC. (b) HIG-NSS. (c) HIG-CC. (d) OFF-AUC. (e) OFF-NSS. (f) OFF-CC. (g) PED-AUC. (h) PED-NSS. (i) PED-CC. (j) PET-AUC. (k) PET-NSS. (l) PET-CC. (m) CAN-AUC. (n) CAN-NSS. (o) CAN-CC.

Figure 4. Cont.

Figure 4. The results of statistical test over twelve methods. We abbreviate twelve methods from “a” to “h”: a = OURS, b = CVS, c = FES, d = GBVS, e = HDCT, f = IKN, g = PCA, h = QSS, i = RSS, j = RWRS, k = SSR, l = SUN. (a) OVE-AUC. (b) OVE -NSS. (c) OVE -CC. (d) BLI-AUC. (e) BLI-NSS. (f) BLI-CC. (g) SKA-AUC. (h) SKA-NSS. (i) SKA-CC. (j) SID-AUC. (k) SID-NSS. (l) SID-CC. (m) TRA-AUC. (n) TRA-NSS. (o) TRA-CC.

Figure 5. The results of statistical test over twelve methods. We abbreviate twelve methods from “a” to “h”: a = OURS, b = CVS, c = FES, d = GBVS, e = HDCT, f = IKN, g = PCA, h = QSS, i = RSS, j = RWRS, k = SSR, l = SUN. (a) SOF-AUC. (b) SOF-NSS. (c) SOF-CC. (d) STR-AUC. (e) STR-NSS. (f) STR-CC.

Considering the *baseline* category, the proposed model is better than other models in terms of AUC, NSS and CC in most cases. Similar results can be observed in the *bad weather* category for two videos *blizzard* and *skating*. In the *dynamic background* category, our method performs quite well in terms of NSS and CC in two videos *canoe* and *overpass*. In the *camera jitter* category, the proposed method achieves relative performance in the *sidewalk* video; meanwhile, the proposed method and RWRS perform well in *traffic* video without significant difference in terms of NSS and CC. In the *intermittent object motion* category, our method performs better than HDCT, PCA in terms of NSS and CC. From these results, our method is slightly competitive with these advanced models.

5.3. Discussion

From our performance results, we discuss some advantages and disadvantages of our proposal as follows:

First, this paper considers whether matrix decomposition can be used to generate motion saliency effectively in-stream manner, and the results of the experiments have proved our idea. We do not use super-pixels segmentation in pre-processing step or optical flow for generating motion features as other methods. Although our method does not preserve the shape of the salient object well for all cases as in PCA, CVS or HDCT, the total computational time of our method is 80% faster than such models. According to Table 9, it takes on average 43 ms to process a frame, including about 7 ms for the process of down-sampling/up-sampling, 31 ms for s-DMD computation, and 5 ms for the refinement process. Regarding to time complexity of the s-DMD, the input rank also affects the computational cost of the whole process. Since DMD modes and DMD eigenvalues are required for computing the raw saliency after every iterate, therefore the computational cost is $O(nr^2)$ where r is the given rank of matrix (X, Y) and n is the number of pixels in a frame. In our case, the rank is much smaller than n , so this model can speed up the performance and especially computational effective and memory-efficient in real-time applications.

Second, the proposed method achieves better results than its competitors in terms of accuracy metrics (MAE, AUC-Borji, NSS and CC) and structure metrics (S-measure) in stationary videos such as *baseline*, and *bad weather* categories. In the *camera jitter* category where videos recorded by vibrational cameras, or when there is interrupted action in the videos like the *intermittent object motion* category, we achieve top three in the performance. In a challenged category such as *dynamic background*, which contains moving leaves or dynamic waters, the accuracy performance of our proposal decreases slightly but not far from the top three results. When compared with other complex models for video saliency, our method achieve slight better scores in terms of accuracy metrics in some categories.

Thirdly, we discuss failure cases of the proposed method. When the size of the object is too small compared with the frame size, and multiple moving objects may disturb the accuracy of the algorithm. We can see this in *streetlight* videos, only moving cars on the bridge are considered as salient regions in the ground truth. However, the proposed method could not distinguish them from other cars moving on the street. It is because we consider the energy of all temporal modes globally without using local features. Moreover, our proposal does not target on preserving of the shape of the salient object in a complex background scene. Therefore, the S-measure of our method is slightly lower than other methods in these categories.

Finally, it is evidence that s-DMD could help to improve motion saliency performance effectively; however, it has limitations to generate good results in some exceptional cases discussed above. In the future, we could distinguish different moving objects in the scene by differentiate their slow and fast modes to get the finer result in different scales of resolutions. This problem requires an effort to incorporate multi-scale s-DMD towards to more comprehensive model.

6. Conclusions

We have introduced a newly fast motion saliency detection algorithm for surveillance systems. Instead of using optical flow for extracting motion features, we directly extract spatial-temporal features

from the video in streaming manner. Thanks to the power of streaming dynamic mode decomposition, we compute the spatial-temporal modes via low rank and sparse decomposition fastly. These modes represent the spatial-temporal coherence features of the scene over time. We generate the raw saliency map that represents to motion region from the energy of temporal modes. The refinement process utilized the advantage of the difference of Gaussian on frequency domain to suppress background noise. The computational time across various videos is 80% faster than other complicated models. The quality evaluation and statistical validation tests on different categories in Change Detection Dataset 2014 show that our method can balance the performance in terms of accuracy and time efficiency in different video categories.

Although s-DMD could help to improve motion saliency performance effectively, its limitations are to generate good results in distinguishing multiple salient regions in complex scenes. In future works, considering multi-scale modes with respect to different moving objects, we investigate the ability to use multi-scale resolution features from different DMD modes for streaming data to improve the saliency prediction.

Author Contributions: This paper represents the result of collaborative teamwork. Conceptualization, T.-T.N.; Funding acquisition, E.-N.H.; Software, T.-T.N.; Visualization, T.-T.N.; Writing—original draft, T.-T.N.; Writing—review & editing, T.-T.N., X.-Q.P., V.N., M.-A.H. and E.-N.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-01615, Developed digital signage solution for cloud-based unmanned shop management that provides online video advertising)

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, T.; Yuan, Z.; Sun, J.; Wang, J.; Zheng, N.; Tang, X.; Shum, H.Y. Learning to detect a salient object. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 3533–367. [[CrossRef](#)]
2. Liu, Z.; Shi, R.; Shen, L.; Xue, Y.; Ngan, K.N.; Zhang, Z. Unsupervised salient object segmentation based on kernel density estimation and two-phase graph cut. *IEEE Trans. Multimed.* **2012**, *14*, 1275–1289. [[CrossRef](#)]
3. Hadizadeh, H.; Bajić, I.V. Saliency-aware video compression. *IEEE Trans. Image Process.* **2014**, *23*, 19–33. [[CrossRef](#)] [[PubMed](#)]
4. Lei, J.; Wu, M.; Zhang, C.; Wu, F.; Ling, N.; Hou, C. Depth preserving stereo image retargeting based on pixel fusion. *IEEE Trans. Multimed.* **2017**, *19*, 1442–1453. [[CrossRef](#)]
5. Zhang, L.; Shen, Y.; Li, H. VSI: A visual saliency-induced index for perceptual image quality assessment. *IEEE Trans. Image Process.* **2014**, *23*, 4270–4281. [[CrossRef](#)] [[PubMed](#)]
6. Han, S.; Vasconcelos, N. Image compression using object-based regions of interest. In Proceedings of the 2006 International Conference on Image Processing, Atlanta, GA, USA, 8–11 October 2006; pp. 3097–3100.
7. Itti, L.; Koch, C.; Niebur, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 1254–1259. [[CrossRef](#)]
8. Harel, C.K.J.; Perona, P. Graph-based visual saliency. In Proceedings of the Neural Information Processing Systems (NIPS), Vancouver, BC, Canada, 4–7 December 2006.
9. Zhang, L.; Tong, M.; Marks, T.; Shan, H.; Cottrell, G. SUN: A Bayesian framework for saliency using natural statistics. *J. Vis.* **2008**, *8*, 1–20. [[CrossRef](#)]
10. Jiang, B.; Zhang, L.; Lu, H.; Yang, C.; Yang, M.-H. Saliency detection via absorbing Markov chain. In Proceedings of the IEEE International Conference on Computer Vision (2013), Sydney, Australia, 1–8 December 2013; pp. 1665–1672.
11. Cheng, M.; Zhang, G.; Mitra, N.J.; Huang, X.; Hu, S. Global contrast based salient region detection. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 21–25 June 2011; pp. 409–416.
12. Achanta, R.; Hemami, S.; Estrada, F. Susstrunk, frequency-tuned salient region detection. In Proceedings of the Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 1597–1604.
13. Yeh, H.-H.; Liu, K.-H.; Chen, C.-S. Salient object detection via local saliency estimation and global homogeneity refinement. *Pattern Recognit.* **2014**, *47*, 1740–1750. [[CrossRef](#)]

14. Shen, X.; Wu, Y. A unified approach to salient object detection via low rank matrix recovery. In Proceedings of the Computer Vision and Pattern Recognition (CVPR) 2012, Providence, RI, USA, 16–21 June 2012; pp. 853–860.
15. Goferman, S.; Zelnik-Manor, L.; Tal, A. Context-aware saliency detection. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2376–2383. [\[CrossRef\]](#)
16. Hou, X.; Zhang, L. Saliency detection: A spectral residual approach. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 21–26 July 2007; pp. 1–8.
17. Zhang, L.; Tong, M.; Cottrell, G. SUNDay: Saliency using natural statistics for dynamic analysis of scenes. In Proceedings of the 31st Annual Cognitive Science Conference, Amsterdam, The Netherlands, 29 July–1 August 2009.
18. Zhong, S.-H.; Liu, Y.; Ren, F.; Zhang, J.; Ren, T. Video saliency detection via dynamic consistent spatiotemporal attention modelling. In Proceedings of the National Conference of the American Association for Artificial Intelligence, Washington, DC, USA, 14–18 July 2013; pp. 1063–1069.
19. Mauthner, T.; Possegger, H.; Waltner, G.; Bischof, H. Encoding based saliency detection for videos and images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015, Boston, MA, USA, 7–12 June 2015; pp. 2494–2502.
20. Wang, W.; Shen, J.; Porikli, F. Saliency-aware geodesic video object segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015, Boston, MA, USA, 7–12 June 2015; pp. 3395–3402.
21. Yubing, T.; Cheikh, F.A.; Guraya, F.F.E.; Konik, H.; Trémeau, A. A spatiotemporal saliency model for video surveillance. *Cogn. Comput.* **2011**, *3*, 241–263. [\[CrossRef\]](#)
22. Ren, Z.; Gao, S.; Rajan, D.; Chia, L.; Huang, Y. Spatiotemporal saliency detection via sparse representation. In Proceedings of the 2012 IEEE International Conference on Multimedia and Expo Workshops, Melbourne, Australia, 9–13 July 2012; pp. 158–163. [\[CrossRef\]](#)
23. Chen, C.; Li, S.; Wang, Y.; Qin, H.; Hao, A. Video saliency detection via spatial-temporal fusion and low-rank coherency diffusion. *IEEE Trans. Image Process.* **2017**, *26*, 3156–3170. [\[CrossRef\]](#)
24. Xue, Y.; Guo, X.; Cao, X. Motion saliency detection using low-rank and sparse decomposition. In Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; pp. 1485–1488. [\[CrossRef\]](#)
25. Bhattacharya, S.; Venkatesh, K.S.; Gupta, S. Visual saliency detection using spatiotemporal decomposition. *IEEE Trans. Image Process.* **2018**, *27*, 1665–1675. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Wang, W.; Shen, J.; Shao, L. Consistent video saliency using local gradient flow optimization and global refinement. *IEEE Trans. Image Process.* **2015**, *24*, 4185–4196. [\[CrossRef\]](#) [\[PubMed\]](#)
27. Kim, H.; Kim, Y.; Sim, J.-Y.; Kim, C.-S. Spatiotemporal saliency detection for video sequences based on random walk with restart. *IEEE Trans. Image Process.* **2015**, *24*, 2552–2564. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Cui, X.; Liu, Q.; Zhang, S.; Yang, F.; Metaxas, D.N. Temporal spectral residual for fast salient motion detection. *Neurocomputing* **2012**, *86*, 24–32. [\[CrossRef\]](#)
29. Alshawi, T. Ultra-fast saliency detection using QR factorization. In Proceedings of the 2019 53rd Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 3–6 November 2019; pp. 1911–1915. [\[CrossRef\]](#)
30. Borji, A.; Cheng, M.; Jiang, H.; Li, J. Salient object detection: A benchmark. *IEEE Trans. Image Process.* **2015**, *24*, 5706–5722. [\[CrossRef\]](#)
31. Cong, R.; Lei, J.; Fu, H.; Cheng, M.; Lin, W.; Huang, Q. Review of visual saliency detection with comprehensive information. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *29*, 2941–2959. [\[CrossRef\]](#)
32. Borji, A.; Itti, L. State-of-the-art in visual attention modeling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 185–207. [\[CrossRef\]](#)
33. Schmid, P.J.; Sesterhenn, J.L. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* **2008**. [\[CrossRef\]](#)
34. Tu, J.H.; Rowley, C.W.; Luchtenburg, D.M.; Brunton, S.L.; Kutz, J.N. On dynamic mode decomposition: Theory and applications. *J. Comput. Dyn.* **2014**. [\[CrossRef\]](#)
35. Grosek, J.; Kutz, J.N. Dynamic mode decomposition for real-time background/foreground separation in video. *arXiv preprint*. **2014**, arXiv:1404.7592.

36. Bi, C.; Yuan, Y.; Zhang, J.; Shi, Y.; Xiang, Y.; Wang, Y.; Zhang, R. Dynamic mode decomposition based video shot detection. *IEEE Access* **2018**, *6*, 21397–21407. [[CrossRef](#)]
37. Sikha, O.K.; Kumar, S.S.; Soman, K.P. Salient region detection and object segmentation in color images using dynamic mode decomposition. *J. Comput. Sci.* **2018**, *25*, 351–366. [[CrossRef](#)]
38. Sikha, O.K.; Soman, K.P. Multi-resolution dynamic mode decomposition-based salient region detection in noisy images. *SIViP* **2020**, *14*, 167–175. [[CrossRef](#)]
39. Yu, C.; Zheng, X.; Zhao, Y.; Liu, G.; Li, N. Review of intelligent video surveillance technology research. In Proceedings of the 2011 International Conference on Electronic and Mechanical Engineering and Information Technology, EMEIT 2011, Harbin, China, 12–14 August 2011; pp. 230–233. [[CrossRef](#)]
40. Hemati, M.S.; Williams, M.O.; Rowley, C.W. Dynamic mode decomposition for large and streaming datasets. *Phys. Fluids* **2014**, *26*. [[CrossRef](#)]
41. Wang, Y.; Jodoin, P.-M.; Porikli, F.; Konrad, J.; Benezeth, Y.; Ishwar, P. CDnet 2014: An expanded change detection benchmark dataset. In Proceedings of the IEEE Workshop on Change Detection (CDW-2014) at CVPR-2014, Columbus, OH, USA, 23–28 June 2014; pp. 387–394.
42. Borji, A.; Tavakoli, H.R.; Sihite, D.N.; Itti, L. Analysis of scores, datasets, and models in visual saliency prediction. In Proceedings of the IEEE International Conference on Computer Vision IEEE Computer Society, Sydney, Australia, 1–8 December 2013; pp. 921–928.
43. Fan, D.-P.; Cheng, M.-M.; Liu, Y.; Li, T.; Borji, A. Structure-measure: A new way to evaluate foreground maps. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 4558–4567.
44. Peters, R.J.; Iyer, A.; Itti, L.; Koch, C. Components of bottom-up gaze allocation in natural images. *Vis. Res.* **2005**, *45*, 2397–2416. [[CrossRef](#)] [[PubMed](#)]
45. le Meur, O.; le Callet, P.; Barba, D. Predicting visual fixations on video based on low-level visual features. *Vis. Res.* **2007**, *47*, 2483–2498. [[CrossRef](#)]
46. Seo, H.J.; Milanfar, P. Non-parametric bottom-up saliency detection by self-resemblance. In Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Miami, FL, USA, 20–25 June 2009; pp. 45–52. [[CrossRef](#)]
47. Tavakoli, H.R.; Rahtu, E.; Heikkilä, J. Fast and efficient saliency detection using sparse sampling and kernel density estimation. In Proceedings of the 17th Scandinavian conference on Image analysis (SCIA'11), Ystad, Sweden, 23–27 May 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 666–675.
48. Schauerte, B.; Stiefelhagen, R. Quaternion-based spectral saliency detection for eye fixation prediction. In Proceedings of the 12th European Conference on Computer Vision—ECCV 2012, Florence, Italy, 7–13 October 2012; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7573, pp. 116–129.
49. Kim, J.; Han, D.; Tai, Y.; Kim, J. Salient region detection via high-dimensional color transform and local spatial support. *IEEE Trans. Image Process.* **2016**, *25*, 9–23. [[CrossRef](#)]
50. Margolin, R.; Tal, A.; Zelnik-Manor, L. What makes a patch distinct? In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 1139–1146. [[CrossRef](#)]
51. Lou, J.; Zhu, W.; Wang, H.; Ren, M. Small target detection combining regional stability and saliency in a color image. *Multimed. Tools Appl.* **2017**, *76*, 14781–14798. [[CrossRef](#)]
52. Wloka, C.; Kunić, T.; Kotseruba, I.; Fahimi, R.; Frosst, N.; Bruce, N.D.B.; Tsotsos, J.K. SMILER: Saliency model implementation library for experimental research. *arXiv* **2018**, arXiv:1812.08848.
53. Li, Y.; Mou, X. Saliency detection based on structural dissimilarity induced by image quality assessment model. *J. Electron. Imaging* **2019**, *28*, 023025. [[CrossRef](#)]