Learning to Diffuse: A New Perspective to Design PDEs for Visual Analysis

Risheng Liu, *Member, IEEE,* Guangyu Zhong, Junjie Cao, Zhouchen Lin, *Senior Member, IEEE,* Shiguang Shan, *Senior Member, IEEE,* and Zhongxuan Luo

Abstract—Partial differential equations (PDEs) have been used to formulate image processing for several decades. Generally, a PDE system consists of two components: the governing equation and the boundary condition. In most previous work, both of them are generally designed by people using mathematical skills. However, in real world visual analysis tasks, such predefined and fixed-form PDEs may not be able to describe the complex structure of the visual data. More importantly, it is hard to incorporate the labeling information and the discriminative distribution priors into these PDEs. To address above issues, we propose a new PDE framework, named learning to diffuse (LTD), to adaptively design the governing equation and the boundary condition of a diffusion PDE system for various vision tasks on different types of visual data. To our best knowledge, the problems considered in this paper (i.e., saliency detection and object tracking) have never been addressed by PDE models before. Experimental results on various challenging benchmark databases show the superiority of LTD against existing state-of-the-art methods for all the tested visual analysis tasks.

Index Terms—Visual diffusion, PDE governed combinatorial optimization, submodularity, saliency detection, object tracking

1 Introduction

The partial differential equation (PDE) is a system involving unknown functions of multiple variables and their partial derivatives. In the past several decades, this mathematical tool has led to an entire new field in image processing and shown its power for many applications, such as image restoration, smoothing, inpainting, segmentation and multiscale representation. We refer to the monographs [2], [3] and the references therein for an overview of these work. The success of PDE based methods on low-level image processing is mainly because that the theoretical analyses on these problems have already been accomplished in areas such as mathematics and physics. For example, the scale space theory [4] proved that the multiscale representations of images are indeed solutions of the heat equation with different time parameters.

In general, conventional PDEs design methodologies can be roughly divided into two main categories: direct and variational methods. For direct methods, such as anisotropic diffusion [5] and curve evolutions [6], [7], [8], PDEs are directly written down based on some mathematical understandings on the physical natures (e.g., heat flow) or the geometric properties (e.g., curvature) of the problems. In contrast, variational methods, such as Tikhonov [3] and total variation (TV) [9] functionals, first define an energy to collect the desired properties of the problem and then derive PDEs by the Euler-Lagrange equation or its associated flows. Though many efforts have been made in literatures, it is still challenging to utilize above ways to design PDEs for complex vision tasks. The main reasons can be attributable to the following three factors.

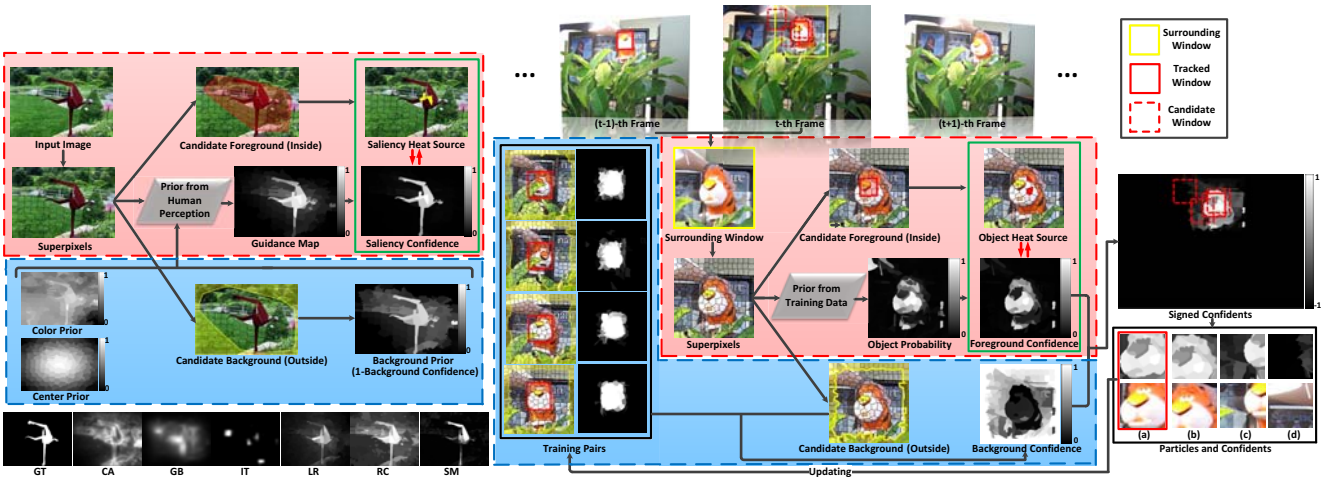
First, good mathematical skills and deep domain knowledge are required for designing PDEs. This is because for a given vision problem, we have to choose appropriate PDE formulation, predict the effect of each derivative term and check whether the final PDEs can meet our goal. So one may fail to acquire effective PDEs when there is no enough intuition for the vision problem. Second, in existing PDEs, the governing equations are predesigned and just some parameters will be tuned. Furthermore, the boundary conditions are only deduced by some simple intuitions (e.g., initial values [4] and well-posed guarantees [2]). Therefore, it is hard to use these PDEs to propagate high-level prior knowledge (extracted by human perception or from training data), which is the core for many complex visual analysis tasks. Third, modeling supervised information and discriminant structure is a big challenge to all existing PDEs because the labels and geometries of training (or previously processed) data cannot be incorporated into the generally designed, fixed partial differential system.

Recently, Liu et al. [10], [11] combined fundamental differential invariants up to second order as general PDEs and determined the combination coefficients by training image pairs for different low-level image processing problems. As a preliminary investigation, this work partially addressed the first issue in above discussions, i.e., provided a straightforward way to design PDEs for image processing. However, due to the complex evolutional formulation, this system suffers from huge computational cost and the optimality of its solution cannot be guaranteed. More importantly, the training mechanism in that work (i.e., only penalizing differential operators for the governing equation) makes it difficult to incorporate high level prior

- *R. Liu (Corresponding author) and Z. Luo are with Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, School of Software Technology, Dalian University of Technology. E-mail: rsliu@dlut.edu.cn*
- *G. Zhong and J. Cao are with School of Mathematical Sciences, Dalian University of Technology.*
- *Z. Lin is with the Key Laboratory of Machine Perception (MOE), School of Electronic Engineering and Computer Science, Peking University, and Cooperative Medianet Innovation Center, Shanghai Jiaotong University.*
- *S. Shan is with Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS.*

A preliminary version of this work was published in [1].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2522415, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) Saliency Detection (b) Object Tracking

Fig. 1. The pipelines of LTD for (a) saliency detection and (b) object tracking, respectively. The pink regions illustrate the core components of LTD framework. The blue regions show how to incorporate different kinds of prior knowledge into the diffusion learning process. In (a) the priors are collected by human perception for saliency detection, while in (b) we learn priors from training data for object tracking. We also shows the ground truth (GT for short) salient region and saliency maps computed by some state-of-the-art saliency detection methods on the bottom row of subfigure (a).

knowledge from human's perceptions and/or labeled training data into the PDEs.

The motivation of our study is trying to provide a simple way to incorporate prior knowledge (from human perception and/or training data) to design PDE system for real world vision problems. More precisely, different from the work in [10], [11], which combines 17 fundamental differential invariants to build coupled PDEs for low-level image processing, we focus on providing a unified diffusion learning framework to address both the *generative* and *discriminative* vision problems (the examples are illustrated in Fig. 1). The key idea in LTD is to assume that both the governing equation and the boundary condition of PDEs should be learned from the visual data. So we propose a PDE governed combinatorial optimization model to incorporate both the generative and discriminative criteria for diffusion learning. Then the stable temperature of our learned diffusion can be used to extract the structure of the data set. *Notably, at least two characteristics of LTD seem to challenge common wisdoms in building vision PDEs: The boundary conditions of PDEs are determined using data and the learned PDEs reveal not only the generative distribution, but also the discriminative category information.* To summarize, the main contributions of this work are threefold:

1) We provide an anisotropic diffusion system with adaptive boundary conditions to formulate general visual analysis tasks. We then develop LTD, a combinatorial optimization framework, to learn PDEs from data for visual diffusions. We also prove the submodularity of the system, which leads to a simple but efficient numerical scheme for LTD.
2) We first introduce a loss function to extract the distribution (generative structure) of the data set for diffusion design. By further considering the information gain based regularizer, LTD can also successfully identify the category information (discriminative structure) for the diffusion. Note that such supervised structure has not been captured by any existing PDE methods before.
3) Both the image based saliency detection and the video based object tracking problems can be addressed within LTD framework. To our best knowledge, this work is the first to use PDEs to solve saliency detection and object tracking[1]. Extensive experiments on different benchmark data sets and comparisons with many state-of-the-art methods show that both of these problems can be efficiently addressed by LTD.

2 A Brief Review of Image Diffusion

In physics, the diffusion equation is a powerful tool to describe density dynamics of physical transport processes. Koenderink [13] and Witkin [14] first built the connection between diffusion equations and multiscale image representations, which enable us to look at solving the isotropic diffusion as a means of constructing a linear and space-invariant transformation of the image. Then Perona and Malik [5] proposed a slight modification to the diffusion process by modeling the flux as a function of edge-strength in the image, thereby giving us "anisotropy" for image diffusion. The above two pioneer work drew great interests on image diffusion methods and various diffusion equations have been considered for image processing problems in the past decades. For instance, as a specific diffusion equation, Poisson equation arose in many image processing tasks, especially gradient domain image analysis (e.g., tone mapping [15], seamless image editing [16] and image matting [17]). This is because Poisson equation can be used to modify image gradients to approximate some given vector fields.

1. Please notice that the object tracking considered in this paper is fundamentally different from the sequence segmentation task, which has been addressed by variational PDEs [12]. This is because the former aims to track the sate of the object for a video sequence while the latter is only to segment the image frame by frame.

Besides designing PDEs from physical perspective, the variational PDEs have also been commonly used for image diffusion. For example, Mumford-Shah (MS) functional [18] is designed for segmentation and TV functional can be used for restoration. To cope with the non-convexity in MS model, the work in [19] established a convex formulation and proved conditions under which MS can achieve global optimum using that convex formulation. As for TV, it was initially motivated by the bounded variation space theory [9] and has been extensively used in imaging sciences. By considering TV energy within the compressive sensing framework, very recent studies proved its guarantees for signal recovery [20].

The above diffusions are performed on regularly distributed image pixels and the differential operators are locally defined on a Cartesian grid of the image domain. So these diffusions can only reflect local interactions on the image. Recently, nonlocal derivatives have been proposed in the context of image processing. The corresponding nonlocal PDEs have shown their efficiency to better preserve fine and repetitive image structure than local ones. For example, Kindermann et al. [21] interpreted the nonlocal means filter and the neighborhoods filter as nonlocal regularization functionals. Guilboa and Osher [22] proposed a nonlocal functional, based on weighted differences. These work can be regarded as the nonlocal analogues of TV models for image processing.

As stated above, diffusion PDEs have been widely used for low-level image processing, such as denoising, segmentation, inpainting and more. However, it is still a challenging task to formulate complex visual analysis problems (e.g., saliency detection and object tracking) using existing PDEs. This is mainly because that modern vision tasks are often defined on more topologically complex domains. For example, the visual data are modeled by collections of feature vectors on irregularly shaped domains (e.g., manifolds). More importantly, human perceptions and labeling information often play very important role in these vision tasks. But unfortunately, we cannot incorporate such priors into conventional PDEs.

3 PRELIMINARIES

3.1 Notations and Definitions

We use lowercase bold letters (e.g., $\mathbf{p}$) to represent vector points and capital calligraphic ones (e.g., $\mathcal{V}$) to denote sets of points. $|\mathcal{V}|$ is the cardinality of $\mathcal{V}$. For any $\mathcal{S} \subset \mathcal{V}$, we denote the complement of $\mathcal{S}$ as $\mathcal{V} \backslash \mathcal{S}$. $\mathbf{1}$ is the all one vector. $\|\cdot\|$ denotes the ℓ_2 norm. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph, where $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a finite set of edges. We denote the neighborhood set of $\mathbf{p}$ on $\mathcal{G}$ as $\mathcal{N}_{\mathbf{p}}$. Suppose f is a real value function on $\mathcal{V}$. For a given point $\mathbf{p} \in \mathcal{V}$ with neighborhood set $\mathcal{N}_{\mathbf{p}}$, we denote ∇f as the gradient of f and discretize it as $\nabla f = [f(\mathbf{p}) - f(\mathbf{q}_1), \cdots, f(\mathbf{p}) - f(\mathbf{q}_{|\mathcal{N}_{\mathbf{p}}|})]$. Similarly, let $\mathbf{v}$ be a vector field on $\mathcal{V}$ and denote $\mathbf{v}_{\mathbf{p}} \in \mathbb{R}^{|\mathcal{N}_{\mathbf{p}}|}$ as the vector at $\mathbf{p}$. Then we denote the divergence of $\mathbf{v}$ as $\mathrm{div}(\mathbf{v})$ and discretize it at $\mathbf{p}$ as $\mathrm{div}(\mathbf{v}_{\mathbf{p}}) = \frac{1}{2} \sum_{\mathbf{q} \in \mathcal{N}_{\mathbf{p}}} (\mathbf{v}_{\mathbf{p}}(\mathbf{q}) - \mathbf{v}_{\mathbf{q}}(\mathbf{p}))$, where $\mathbf{v}_{\mathbf{p}}(\mathbf{q})$ is the vector element corresponding to $\mathbf{q} \in \mathcal{N}_{\mathbf{p}}$[2]. Based on above definitions, we can also discretize the Laplace-Beltrami operator on the graph, i.e., $\Delta f = \mathrm{div}(\nabla f)$.

2. Similar discretization scheme is also used for nonlocal total variation [22].

3.2 Problem Statement

In this subsection we outline the problems for which this work is relevant. That is, we provide a physical viewpoint, named visual diffusion (VD), to understand and model visual data analysis tasks. Specifically, for a set of visual elements $\mathcal{V}$ (extracted from images or videos), the goal of VD is to propagate a specific real value function $f(\mathbf{p}) : \mathcal{V} \to \mathbb{R}$ (i.e., temperature) from the most representative subset $\mathcal{S} \subset \mathcal{V}$ (i.e., heat source) to all the other nodes to extract the latent intrinsic structure of $\mathcal{V}$. Indeed, the heat source can be considered as the "basis" of the data set and the temperatures of other nodes should be understood as their relevances to the heat source.

Actually, many visual analysis tasks can be (re)formulated as the problem of VD. For example, in image domain, segmentation aims to divide an image into different disjoint regions such that image elements have high similarity within each region and high contrast between regions. More complex tasks, such as scene understanding, saliency or object detection, would like to further identify image regions with specific properties. In VD framework, all these problems could be considered as a temperature propagation process. Specifically, for each image element, we define a temperature function f on it to measure its specific property (e.g., local similarity, semantic information, saliency confidence or objectness). Then the problem reduces to that of simultaneously identifying the most *representative* image elements (i.e., heat source $\mathcal{S}$) with respect to the specific property and propagating the temperature to extract the *relevance* between heat source and other image elements. Finally, the intrinsic structure of the image can be obtained using propagated temperature. Furthermore, by incorporating temporal information into the propagation (e.g., propagating temperatures through the sequence), VD could also be suitable for video analysis, such as temporal structure (e.g., event and action) detection, motion segmentation and object tracking.

The fundamental challenge in VD is the "chicken-and-egg" problem. That is, if the heat source $\mathcal{S}$ is already recognized, propagating the temperature f can be performed by solving standard PDEs. While, if f has been propagated to all the nodes in $\mathcal{V}$, the representative subset $\mathcal{S}$ then can be directly identified. So the heart of VD is how to effectively handle the coupling between the heat source $\mathcal{S}$ and the temperature f. Unfortunately, the existing predefined PDEs with fixed governing equation and boundary condition cannot simultaneously obtain $\mathcal{S}$ and f, thus may fail to recover the structure of $\mathcal{V}$. In this work, we would like to develop an adaptive leaning based PDE framework, named learning to diffuse (LTD), to extend conventional diffusion equations for the VD problem.

It will be shown in the following that visual analysis problems on both images (e.g., saliency detection) and videos (e.g., object tracking) can be formulated as specific cases of VD and efficiently addressed by LTD.

4 LEARNING TO DIFFUSE (LTD)

This section first develops a linear elliptic PDE system with Dirichlet boundary condition to formulate VD and then presents a combinatorial optimization framework to optimize diffusions for visual analysis. The necessary numerical and theoretical analysis for LTD will be addressed at the end of this section.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2522415, IEEE Transactions on Pattern Analysis and Machine Intelligence

4.1 An Anisotropic Diffusion System with Adaptive Boundary Condition

For given $\mathcal{V}$, our goal is to simultaneously identify the heat source $\mathcal{S}$ and propagate the temperature f on $\mathcal{V}$. In general, this problem can be mathematically modeled as an evolutionary PDE system with unknown f and $\mathcal{S}$:

$$\frac{\partial f(\mathbf{p},t)}{\partial t} = F(f,\nabla f),\ f(\mathbf{g})=0,\ f(\mathbf{p})=s_{\mathbf{p}},\ \mathbf{p}\in\mathcal{S}, \quad (1)$$

where $\mathbf{g}$ is an environment point with zero temperature (outside $\mathcal{V}$) and $s_{\mathbf{p}}$ is the temperature corresponding to the node $\mathbf{p}$. In general, the governing equation F in (1) can be any smooth functions with respect to f and its derivative ∇f. But in VD framework, our goal is to build a prior guided diffusion system to address various visual analysis tasks. Therefore, F is specified as follows. We first introduce an anisotropic diffusion term $\mathrm{div}(\mathbf{K}\nabla f)$, in which $\mathbf{K}$ is an inhomogeneous metric tensor to control the diffusivity. To further incorporate high-level priors into our diffusions, we define a fidelity term $\mathcal{P}_{\mathcal{V}\backslash\mathcal{S}}(f-u)$, where u is a map to guide the diffusion (can be learned by either human perception or collected training data) and $\mathcal{P}_{\mathcal{V}\backslash\mathcal{S}}$ is the projection on $\mathcal{V}\backslash\mathcal{S}$[3], i.e.,

$$\mathcal{P}_{\mathcal{V}\backslash\mathcal{S}}(f-u)(\mathbf{p}) = \begin{cases} f(\mathbf{p})-u(\mathbf{p}), & \mathbf{p}\in\mathcal{V}\backslash\mathcal{S}, \\ 0, & \mathbf{p}\in\mathcal{S}. \end{cases} \quad (2)$$

Overall, the governing equation is defined as:

$$F(f,\nabla f) = \mathrm{div}(\mathbf{K}\nabla f) + \lambda\mathcal{P}_{\mathcal{V}\backslash\mathcal{S}}(f-u), \quad (3)$$

where $\lambda \geq 0$ is a parameter to control the trade-off between the rate of diffusion and the fidelity to the guidance.

If only caring about the stable situation (i.e., no heat can be further propagated) of this evolution, we omit time t and simplify the PDE system as:

$$F(f,\nabla f)=0,\ f(\mathbf{g})=0,\ f(\mathbf{p})=s_{\mathbf{p}},\ \mathbf{p}\in\mathcal{S}, \quad (4)$$

which is a linear elliptic system with Dirichlet boundary.

In most conventional PDEs, to simplify the computational scheme, the boundary condition (i.e., heat source $\mathcal{S}$) is always predefined and fixed during the diffusion process. But unfortunately, such strategy may significantly reduce the flexibility of the diffusion system. To address this limitation, in our system we will also consider f as a set function[4] with respect to the heat source, i.e., $f(\mathcal{S}): 2^{\mathcal{V}} \to \mathbb{R}$.

4.2 Learning Diffusion by PDE Governed Combinatorial Optimization

Our diffusion learning actually consists of two goals: estimating the stable temperature f and selecting the optimal heat source $\mathcal{S}$. Now we provide a unified optimization model to jointly solve these two problems. For the temperature, it is obvious that f can be directly solved by (4) with selected $\mathcal{S}$. As for the heat source, our first observation is that due to the significant redundancy in the data set, not every node in $\mathcal{V}$ is equally informative. So we first generate a compact and representative subset $\mathcal{F}\subset\mathcal{V}$ and then choose heat source from $\mathcal{F}$[5]. Then we present the following criteria to optimize heat source for our diffusions.

Generative loss: Given $\mathcal{F}$, we tend to select the heat source from it with the highest stable temperature on $\mathcal{V}$ because higher overall temperature indicates better representative ability of the heat source. This criterion is formulated by maximizing the temperature calculated by (4):

$$L(\mathcal{S}) = \sum_{\mathbf{p}\in\mathcal{V}} f(\mathbf{p};\mathcal{S}). \quad (5)$$

In operations research, this objective function can be viewed as the uncapacitated facility location loss [23], which is to select a set of potential facilities (i.e., $\mathcal{S}$) and assign customers (i.e., $\mathcal{V}\backslash\mathcal{S}$) to them in a cost effective and efficient manner (i.e., maximum the utility L).

Discriminative regularizer: The discriminative relationships often play very important role in visual analysis. For example, with the category information correctly identified from the training data, it can extract more accurate data structure from $\mathcal{V}$. However, as all the components of existing PDEs are fixed, we cannot do this for the conventional diffusion system. Fortunately, in LTD, this issue can be efficiently addressed by introducing a discriminative regularizer in the combinatorial formulation. Actually, given nodes from different categories, the goal of discriminative PDEs learning is to utilize training data to help select heat source with homogeneous category label for a particular diffusion. To do this, for the category c, we collect a set of training data $\mathcal{T}_c := \{h_{\mathbf{p}}, p_c(\mathbf{p})\}$, where $h_{\mathbf{p}}$ is the feature vector and $p_c(\mathbf{p})$ is the probability of $\mathbf{p}$ belonging to this category, respectively. By training a regressor on $\mathcal{T}_c$ and applying it to $\mathcal{F}$, we estimate a mapping $p_c(\mathbf{p})$ on $\mathcal{F}$ to measure the probability of candidate heat source belonging to the given category. Then we define two entropies:

$$\begin{cases} E(\mathcal{F}\backslash\mathcal{S}) = -\sum\limits_{\mathbf{p}\in\mathcal{F}\backslash\mathcal{S}} p_c(\mathbf{p})\log p_c(\mathbf{p}), \\ E(\mathcal{F}\backslash\mathcal{S}|\mathcal{S}) = -\sum\limits_{\mathbf{p}\in\mathcal{F}\backslash\mathcal{S},\mathbf{q}\in\mathcal{S}} p(\mathbf{p},\mathbf{q})\log p_t(\mathbf{q}|\mathbf{p}), \end{cases} \quad (6)$$

where $p_t(\mathbf{q}|\mathbf{p})$ is the transition probability (i.e., normalized affinity) from $\mathbf{p}\in\mathcal{F}\backslash\mathcal{S}$ to $\mathbf{q}\in\mathcal{S}$ in the feature space and $p(\mathbf{p},\mathbf{q}) = p_c(\mathbf{p})p_t(\mathbf{q}|\mathbf{p})$. Intuitively, the larger $E(\mathcal{F}\backslash\mathcal{S})$ tends to seek nodes with high probabilities p_c for $\mathcal{S}$ (i.e., select heat source belonging to category c). While the smaller $E(\mathcal{F}\backslash\mathcal{S}|\mathcal{S})$ makes category c be easily differentiated from others (i.e., enhance the discrimination between $\mathcal{S}$ and $\mathcal{F}\backslash\mathcal{S}$). So we define our discriminative regularizer as the following information gain:

$$R(\mathcal{S}) = E(\mathcal{F}\backslash\mathcal{S}) - E(\mathcal{F}\backslash\mathcal{S}|\mathcal{S}). \quad (7)$$

Based on above analysis, we define $H(\mathcal{S}) = L(\mathcal{S}) + \gamma R(\mathcal{S})$ with parameter $\gamma \geq 0$ and formally formulate LTD as the following PDE governed combinatorial optimization:

$$\max_{f,\mathcal{S}\in\mathbb{M}^n} H(\mathcal{S}), \quad s.t. \begin{cases} \mathrm{div}(\mathbf{K}\nabla f) + \lambda\mathcal{P}_{\mathcal{V}\backslash\mathcal{S}}(f-u) = 0, \\ f(\mathbf{g})=0,\ f(\mathbf{p})=s_{\mathbf{p}},\ \mathbf{p}\in\mathcal{S}, \end{cases} \quad (8)$$

3. Here we do not enforce constraints on $\mathcal{S}$ as the temperatures of nodes in $\mathcal{S}$ are specified by the boundary condition for each diffusion.

4. In general, the solution to conventional PDEs with fixed boundary condition is a continuous function with respect to space and/or time variables. While the solution to (4) is inherently combinatorial with respect to the heat source.

5. We will use different strategies to define $\mathcal{F}$ for particular vision tasks.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2522415, IEEE Transactions on Pattern Analysis and Machine Intelligence

where $\mathbb{M}^n = \{\mathcal{S}|\mathcal{S} \subset \mathcal{F} \subset \mathcal{V}, |\mathcal{S}| \leq n\}$ is a uniform matroid [24] to enforce that the cardinality of $\mathcal{S}$ is no more than a given number $n \leq |\mathcal{F}|$. *By solving* (8)*, we can identify the optimal heat source and calculate the stable temperature in a unified framework for the VD system.*

Adaptive penalty: To provide an adaptive way to identify the number of nodes in heat source and further suppress the redundancy in $\mathcal{F}$, we define a confidence function $w(\mathbf{p}) \geq 0$ on $\mathcal{F}$, in which larger $w(\mathbf{p})$ implies that $\mathbf{p}$ has a higher probability of belonging to $\mathcal{F}\backslash\mathcal{S}$ and should be suppressed. Therefore, we maximize another cost function $\hat{H}(\mathcal{S}) = H(\mathcal{S}) - W(\mathcal{S})$ in (8), where $W(\mathcal{S}) = \sum_{\mathbf{p}\in\mathcal{S}} w(\mathbf{p})$. Please notice that subtracting the penalty term $W(\mathcal{S})$ can also be understood as incorporating the cost of opening facilities into the facility location problem.

4.3 Discretization and Optimization

Now we discuss how to discretize and optimize LTD problem.

Discretization: For each $\mathbf{p}$, let $\mathcal{N}_\mathbf{p} = \{\mathbf{q}_1, \cdots, \mathbf{q}_{|\mathcal{N}_\mathbf{p}|-1}, \mathbf{g}\}$ be its neighborhood set, where the first $|\mathcal{N}_\mathbf{p}|-1$ nodes are in the domain $\mathcal{V}$ and the environment point $\mathbf{g}$ outside $\mathcal{V}$ is connected to each node [25]. Then we can specify $\mathbf{K}$ at $\mathbf{p}$ to measure the variance between $\mathbf{p}$ and its neighborhood $\mathcal{N}_\mathbf{p}$, i.e., we define an inhomogeneous metric tensor $\mathbf{K}_\mathbf{p}$ as:

$$\mathbf{K}_\mathbf{p} = \text{diag}\left(k(\mathbf{p}, \mathbf{q}_1), \cdots, k(\mathbf{p}, \mathbf{q}_{|\mathcal{N}_\mathbf{p}|-1}), z_\mathbf{g}\right), \tag{9}$$

where $k(\mathbf{p}, \mathbf{q}) = \exp(-\beta\|h_\mathbf{p} - h_\mathbf{q}\|^2)$ is the Gaussian similarity (with a strength parameter β) between the features of nodes, $h_\mathbf{p}$ is a feature vector at node $\mathbf{p}$, and $z_\mathbf{g}$ is a small constant to measure the dissipation conductance at $\mathbf{p}$. Then we can approximately discretize the PDE formulation as:

$$f(\mathbf{p}) = \begin{cases} \frac{1}{d_\mathbf{p}+\lambda}\left(\sum\limits_{\mathbf{q}\in\mathcal{N}_\mathbf{p}} \mathbf{K}_\mathbf{p}(\mathbf{q}) f(\mathbf{q}) + \lambda u(\mathbf{p})\right), & \mathbf{p} \in \mathcal{V}\backslash\mathcal{S}, \\ s_\mathbf{p}, & \mathbf{p} \in \mathcal{S}, \end{cases} \tag{10}$$

where $\mathbf{K}_\mathbf{p}(\mathbf{q})$ is the diagonal element of $\mathbf{K}_\mathbf{p}$ corresponding to $\mathbf{q}$ and $d_\mathbf{p} = \sum_{\mathbf{q}\in\mathcal{N}_\mathbf{p}} \mathbf{K}_\mathbf{p}(\mathbf{q})$.

Optimization: It is easy to check that (10) is indeed a linear system, thus can be easily solved. However, the optimization of (8) without knowing any further properties can be extremely difficult (e.g., trivially worst-case exponential time or even inapproximable [26]). Fortunately, we can prove the following theory to exploit some good properties (i.e., monotonicity and submodularity[6]) for LTD.

Theorem 1: [7] Let f be the stable temperature, H and $\hat{H}$ be the objectives to (8). Then by considering them as set functions with respect to $\mathcal{S}$, the following assertions hold:

1) $f(\mathcal{S})$ is monotone and submodular.
2) $H(\mathcal{S})$ is monotone and submodular.
3) $\hat{H}(\mathcal{S})$ is submodular and $\hat{H}(\emptyset) = 0$.

The monotonicity and submodularity of H together with the uniform matroid constraint in (8) imply that using a greedy algorithm to solve (8) yields a $(1 - 1/e)$-approximation [30]. Due to the non-monotone nature, we cannot have the same theoretical guarantee for $\hat{H}$. But in practice, by adding the stopping criterion $\hat{H}(\mathcal{S} \cup \{\mathbf{p}\}) \leq \hat{H}(\mathcal{S})$, the maximization process for $\hat{H}$ can be automatically stopped and then the optimal seed set is obtained accordingly. We have experimentally found that a greedy algorithm with this stopping criterion is efficient for maximizing $\hat{H}$ in all the tested problems. The complete LTD optimization framework is summarized in Algorithm 1.

Algorithm 1 The LTD Optimization Framework

Input: Given $\mathcal{V}$ and necessary parameters.
Output: Stable f^* and optimal $\mathcal{S}^*$.

```
Calculate p_c and p_t for p ∈ F, K and g for p ∈ V.
Initialize heat source S ← ∅.
while |S| ≤ n do
    for p ∈ V\S do
        Solve (10) with S ∪ {p} for f.
        Obtain the gain ΔH(p) = H(S ∪ {p}) − H(S),
        or ΔĤ(p) = Ĥ(S ∪ {p}) − Ĥ(S).
    end for
    p* = arg max_{p∈V\S} ΔH(p) or arg max_{p∈V\S} ΔĤ(p).
    if Ĥ(S ∪ {p*}) ≤ Ĥ(S) (only for Ĥ) then
        Break.
    end if
    S ← S ∪ {p*}.
end while
Solve (10) with optimal S* to obtain stable f*.
```

5 LTD for Visual Analysis

In this section, we consider two example applications of LTD on images and sequences, respectively.

5.1 LTD on Images for Saliency Detection

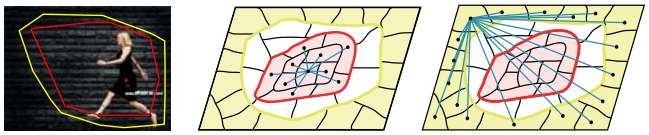
We first apply LTD for saliency detection, which is a typical visual analysis task on the image domain. Given visual scenes, saliency detection is to find the regions which are most likely to capture human's attention. We show that this task can be formulated as a specific case of VD. That is, we first define the saliency confidence as a temperature function and assume that our attention is firstly attracted by some most representative salient image elements (considered as heat source). Then the saliency confidence will be propagated from the heat source to all salient regions on the image. In this view, we define $\mathcal{V}$ as the discrete image domain (i.e., a set of points corresponding to all image elements) and consider f as the saliency confidence function on $\mathcal{V}$. Thus detecting salient regions reduces to the problem of leaning a particular diffusion system for f.

Fig. 1 (a) shows the pipeline of LTD based saliency detector on an example image. This visual comparison together with more sufficient experimental results in Section 6.2 show that by incorporating priors from human perception (e.g., color, location and background) for diffusion learning, the PDE (4) with properly specified governing equation and boundary condition can successfully model the saliency diffusion, thus achieves better saliency detection results than state-of-the-art approaches. In the following, we discuss the details of this process.

6. Submodularity is an important property for discrete set function and has far-reaching applications in operations research, machine learning and computer vision [23], [27], [28], [29].

7. Please see Appendix for necessary definitions and proofs.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2522415, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) (b) (c)

Fig. 2. Illustration of the shift convex hull strategy in (a) and connection relationship in (b)-(c). The red and yellow polygons in (a) denote $\mathcal{C}$ and $\mathcal{C}'$, respectively. The red and yellow regions in (b)-(c) represent $\mathcal{F}_c$ and $\mathcal{B}_c$, respectively. Lines in (c) indicate that all nodes in $\mathcal{B}_c$ are connected.

5.1.1 Determining the Governing Equation

For a given image, we generate superpixels[8] to build the image elements set $\mathcal{V} = \{\mathbf{p}_1, \cdots, \mathbf{p}_{|\mathcal{V}|}\}$ and define feature vectors $\{h_{\mathbf{p}}, \mathbf{p} \in \mathcal{V}\}$ as the means of the superpixels in the CIE LAB color space. The image structure information is extracted as follows. Suppose the image domain $\mathcal{V}$ consists of two parts: the candidate foreground $\mathcal{F}_c$ (salient regions, may also contain some spurious image elements) and the candidate background $\mathcal{B}_c$ (non-salient regions). We utilize a shift convex hull strategy to approximately estimate these two subsets from the input image. Specifically, we use Harris operator [32] to roughly detect the corners and contour points and estimate a convex hull $\mathcal{C}$ based on these points [33]. Then $\mathcal{F}_c$ can be obtained by collecting nodes inside $\mathcal{C}$. To further identify pure background nodes, we define an expended hull $\mathcal{C}'$ by adding adjacent nodes to $\mathcal{C}$. Then $\mathcal{B}_c$ is obtained by collecting all nodes outside $\mathcal{C}'$. Please see Fig. 2 (a) for an example of $\mathcal{C}$ and $\mathcal{C}'$.

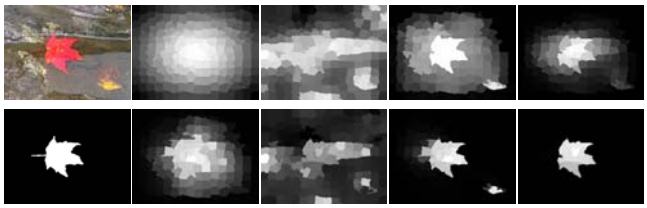
We construct an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to reveal the connection relationships (i.e., $\mathcal{N}_{\mathbf{p}}$ for each $\mathbf{p}$) in the image domain, where $\mathcal{E}$ is a set of undirected edges corresponding to the nodes set $\mathcal{V}$. Specifically, we first connect each node with its 2-ring neighbors to exploit the local spatial relationship (Fig. 2 (b)). Then all the nodes in $\mathcal{B}_c$ are connected to each other to enforce the smoothness of background (Fig. 2 (c)). As there may exist spurious image elements, we do not further connect nodes in $\mathcal{F}_c$. Finally, all the nodes are connected to an environment node $\mathbf{g}$.

Now we are ready to determine the governing equation F. First, $\mathbf{K}_{\mathbf{p}}$ can be calculated by (9) using the graph connection (i.e., $\mathcal{N}_{\mathbf{p}}$) and the features (i.e., $h_{\mathbf{p}}$). To incorporate high-level priors into the governing equation, u is defined in the following way. By assuming that the distribution of background is significantly different from that of foreground, we perform a simplified diffusion with $\lambda = 0$ in (4) to compute a temperature f_b with respect to the background confidence score, where the boundary is chosen as the union of $\mathcal{B}_c$ (with temperature 1) and an environment node $\mathbf{g}$ (with temperature 0). It is easy to check that the solution to this background diffusion is a harmonic function, thus $f_b(\mathbf{p}) \in [0, 1]$[9]. So the elements in f_b can be viewed as probabilities of nodes belonging to the background. In this view, we have the probability of a node belonging to the foreground as $u_f(\mathbf{p}) = \mathbf{1} - f_b(\mathbf{p})$. Then the final guidance map u is obtained by combining u_f with two standard saliency priors, namely the color and center maps (denoted by u_c and u_l, respectively) from [34], using multiplication [35]:

$$u(\mathbf{p}) = u_f(\mathbf{p}) \times u_c(\mathbf{p}) \times u_l(\mathbf{p}). \tag{11}$$

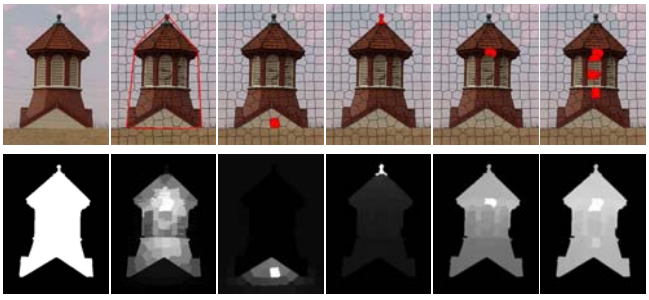
8. Generally, any edge-preserving superpixel methods can be used and SLIC algorithm [31] is adopted in this paper to generate image elements.

9. Based on the maximum/minimum principles of harmonic functions.



(a) (b) (c) (d) (e)

Fig. 3. Saliency diffusion with different guidance maps. (a) input image and GT salient region. (b)-(e) center prior u_l, color prior u_c, background diffusion prior u_f, final guidance map u (top) and their corresponding saliency maps (bottom), respectively.



(a) (b) (c) (d) (e) (f)

Fig. 4. Saliency diffusion with different heat source. (a) input image and GT salient region. (b) $\mathcal{F}_c$ (inside red polygon) and u. (c)-(e) diffusion results using one candidate seed in $\mathcal{F}_c$: (c) background ($L = 10.6175$), (d) bad foreground ($L = 1.6818$) and (e) good foreground ($L = 31.7404$). (f) optimal seeds ($L = 43.8589$) and final saliency map. Here we report L values using the original saliency maps but normalize them for visual comparison.

5.1.2 Selecting the Boundary Condition

Due to the following two reasons, we do not use all nodes in $\mathcal{F}_c$ as heat source. First, the convex hull may not adequately suppress background nodes in $\mathcal{F}_c$ (Fig. 4 (c)). Second and more importantly, it is observed that the heat source with extremely high local contrast to its neighbors (e.g., nodes near object boundary and bright or dark nodes on the object) may also lead to a bad saliency map (Fig. 4 (d)). So it is necessary to select *the most representative foreground nodes* in $\mathcal{F}_c$ to define the boundary conditions.

The goal of our diffusion system is to propagate the temperature of heat source $\mathcal{S}$ to the whole image domain $\mathcal{V}$. So here we only maximize the loss L (i.e., the sum of scores f with respect to all image elements in $\mathcal{V}$) when the saliency diffusion is stable, that is, we solve the discrete optimization problem (8) with $\gamma = 0$. As the saliency confidence can be considered as the relevances between nodes and the salient heat source, the maximum criterion in (8) naturally tends to choose heat source in relatively larger connected subgraphs (thus is

more representative). Therefore, the nodes in $\mathcal{F}_c$ with high local contrast (i.e., less connections and paths to other nodes) will be removed from $\mathcal{S}$. One may concern that incorporating background nodes will also lead to a large L as they may connect to nodes outside $\mathcal{F}_c$. Fortunately, our guidance map u can enforce very small saliency scores (in most case near zero) in background regions (u in Fig. 4 (b)). So background nodes in $\mathcal{F}_c$ still result in a relatively small L value and cannot be included in $\mathcal{S}$ (Fig. 4 (c)). Here we also use u to define the scores of saliency seeds, i.e., $s_{\mathbf{p}} = u(\mathbf{p})$, for $\mathbf{p} \in \mathcal{S}$.

In general, the performance of (8) is depend on the maximum number of heat source (i.e., n). By specifying $W(\mathcal{S})$ in (8), we provide an adaptive way to identify n and further suppress background nodes in $\mathcal{F}_c$. Specifically, we define $w(\mathbf{p}) = 1/(\epsilon + u(\mathbf{p})^2)$ on $\mathcal{F}_c$, where ϵ is a small positive constant. Here the larger $w(\mathbf{p})$ implies that $\mathbf{p}$ has a higher probability of belonging to the background and should be suppressed. Then we maximize the loss function $\hat{L}(\mathcal{S}) = L(\mathcal{S}) - W(\mathcal{S})$ in (8), where $W(\mathcal{S}) = \sum_{\mathbf{p}\in\mathcal{S}} w(\mathbf{p})$.

5.2 LTD on Sequences for Object Tracking

Now we address object tracking using LTD. It is to illustrate that LTD can also be used for sequential visual analysis. Object tracking is one of the most fundamental components in video analysis. Given the initialized object, the goal of tracking is to estimate the states of the target in the subsequent frames. We consider this problem as the task of distinguishing the target object from the surrounding background (i.e., binary classification) in Particle filter framework [36]. Utilizing this viewpoint, we can formulate object tracking as jointly performing temporal and spatial VDs on the sequence. Specifically, we define the observation likelihood as a temperature and propagate it from previously processed (i.e., training) frames to the current frame to establish the object probability (i.e., priors). We also estimate the candidate heat source using the location of the tracked object in the last frame. Then the final object confidence can be calculated by a prior guided propagation on the current frame. Fig. 1 (b) illustrates the pipeline of the LTD based tracker. It can be seen that besides the temperature based generative loss L, we also incorporate prior knowledge learned from training data into the optimization model (8) (i.e., discriminative regularization R) to enforce discriminative constraints for the diffusions.

Specifically, let $\mathcal{Y}_{1:t-1} = \{\mathcal{Y}_1, \cdots, \mathcal{Y}_{t-1}\}$ be the tracked objects from the first to the $(t-1)$-th frame, $\mathcal{Y}_t$ be a candidate object at time t and $\mathbf{x}_t$ be the state variable describing the affine motion parameters of $\mathcal{Y}_t$, respectively. Then we can process $\mathbf{x}_t$ with the following probabilities:

$$p(\mathbf{x}_t|\mathcal{Y}_{1:t}) \propto p(\mathcal{Y}_t|\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathcal{Y}_{1:t-1}) d\mathbf{x}_{t-1}, \tag{12}$$

where $\mathcal{Y}_{1:t} = \{\mathcal{Y}_{1:t-1}, \mathcal{Y}_t\}$, $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ denotes the state transition distribution and $p(\mathcal{Y}_t|\mathbf{x}_t)$ estimates the likelihood of observing $\mathcal{Y}_t$ at state $\mathbf{x}_t$. So the optimal state of the target at time t is obtained by the maximum-a-posteriori (MAP) estimation over m candidates:

$$\mathbf{x}_t = \arg\max_{\mathbf{x}_t^i} p(\mathcal{Y}_t^i|\mathbf{x}_t^i) p(\mathbf{x}_t^i|\mathbf{x}_{t-1}), \ i = 1, \cdots, m, \tag{13}$$

where $\mathbf{x}_t^i$ indicates the i-th candidate state and $\mathcal{Y}_t^i$ is the target image region predicated by $\mathbf{x}_t^i$. Here $p(\mathbf{x}_t^i|\mathbf{x}_{t-1})$ can be simply formulated by random walks. So object tracking reduces to the problem of calculating the observation likelihood $p(\mathcal{Y}_t^i|\mathbf{x}_t^i)$.

5.2.1 Discriminative Object Representation

To setup our tracking system, we first learn a discriminative object representation from a set of d initial frames. For the target state $\mathbf{x}_t$ at the t-th training frame, we denote its tracking window as $\mathcal{A}(\mathbf{x}_t)$. In the following, we will use $|\mathcal{A}(\mathbf{x}_t)|$ to denote its area size. We also define an additional square window $\mathcal{A}'(\mathbf{x}_t)$ at the location of the target with larger area size, i.e., $\mathcal{A}'(\mathbf{x}_t) = \delta\mathcal{A}(\mathbf{x}_t)$, where $\delta > 1$ is a magnifying parameter. In this paper, we always set $\delta = 1.5$ to guarantee that $\mathcal{A}'(\mathbf{x}_t)$ can cover the entire target in the last frame and include sufficient background for better discrimination. It is illustrated in Fig. 1 (b) that the surrounding window and the tracking window for the target are denoted as yellow and red rectangles both with solid lines, respectively. We oversegment the surrounding region $\mathcal{A}'(\mathbf{x}_t)$ to build a set of image elements $\mathcal{V}_t = \{\mathbf{p}_1, \cdots, \mathbf{p}_{|\mathcal{V}_t|}\}$. For each superpixel $\mathbf{p} \in \mathcal{V}_t$, the corresponding feature vector $h_{\mathbf{p}}$ is defined as a normalized histogram in RGB color space. Then we define an object representation for training image elements $\bar{\mathcal{V}} = \{\mathcal{V}_1, \cdots, \mathcal{V}_d\}$ as follows:

$$p_c(\mathbf{p}) = |\mathcal{A}(\mathbf{p}) \cap \mathcal{A}(\mathbf{x}_t)| / |\mathcal{A}(\mathbf{p})|, \quad \forall \mathbf{p} \in \bar{\mathcal{V}}, \tag{14}$$

where $\mathcal{A}(\mathbf{p})$ denotes the area covered by superpixel $\mathbf{p}$ on the frame and $|\mathcal{A}(\mathbf{p}) \cap \mathcal{A}(\mathbf{x}_t)|$ is thus the area size of $\mathbf{p}$ overlapping the target at the t-th training frame. It can be observed that p_c has the property that the larger value indicates the higher confidence to assign $\mathbf{p}$ to the target. Then we define our training data for the target object as $\mathcal{T}_c = \{h_{\mathbf{p}}, p_c(\mathbf{p}) | \mathbf{p} \in \bar{\mathcal{V}}\}$.

5.2.2 Object Tracking via Diffusion on the Video

When the t-th test frame arrives, we also extract its surrounding region $\mathcal{A}'(\mathbf{x}_t)$ centered at the location of the tracked target in the $(t-1)$-th frame (i.e., $\mathcal{A}(\mathbf{x}_{t-1})$) and oversegment $\mathcal{A}'(\mathbf{x}_t)$ to define the image elements $\mathcal{V}_t$. Then we perform a simple linear support vector regression (SVR) [37] on $\mathcal{T}_c$ and apply it to $\mathcal{V}_t$. Now we can define the candidate target at t-th frame (denoted as $\mathcal{F}_c^t$) using the regressed confidence map. That is, we collect nodes in a small square region located at the center of $\mathcal{A}'(\mathbf{x}_t)$ (i.e., the candidate foreground in Fig. 1 (b)) with p_c greater than zero as $\mathcal{F}_c^t$. We also define the candidate background $\mathcal{B}_c^t$ as nodes on the boundary of $\mathcal{A}'(\mathbf{x}_t)$ (i.e., the candidate background in Fig. 1 (b)). To enforce the sequential structure into the current frame, we also include the boundary nodes of the surrounding regions $\{\mathcal{A}'(\mathbf{x}_1), \cdots, \mathcal{A}'(\mathbf{x}_d)\}$ in the training frames to $\mathcal{B}_c^t$ (i.e., the left of the blue region in Fig. 1 (b)). Different from LTD on the image domain (presented in Section 5.1), which only needs to define a single graph, here we construct two different graphs for the sequence. First, to reveal sequential relationships, we define a graph $\mathcal{G}^s$ by connecting k nearest neighbors of each node in RGB color space for $\bar{\mathcal{V}} \cup \mathcal{V}_t$. Meanwhile, to collect spatial information at the current frame, we build a graph $\mathcal{G}^c$ for nodes in $\mathcal{V}_t$.

Based on the regressed confidence map p_c, we define the temperature of heat source $\mathcal{S}_t \subset \mathcal{F}_c^t$ (i.e., $s_{\mathbf{p}} = p_c(\mathbf{p})$ for $\mathbf{p} \in$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2522415, IEEE Transactions on Pattern Analysis and Machine Intelligence

$\mathcal{S}_t$). Also, the discriminant penalty $R(\mathcal{S}_t)$ is defined by p_c using (7). Then we perform LTD on $\mathcal{G}^c$ to learn the foreground (i.e., object) confidence f for $\mathcal{V}_t$ (i.e., the pink region in Fig. 1 (b)). To adaptively determine the number of heat source in (8), we can also use p_c to define W by the same formulation as that in Section 5.1.2[10].

5.2.3 Observation Model

Intuitively, we may define the observation model for candidate $\mathcal{Y}_t^i \subset \mathcal{V}_t$ as $p(\mathcal{Y}_t^i|\mathbf{x}_t^i) = \sum_{\mathbf{p}\in\mathcal{Y}_t^i} f(\mathbf{p})$. But unfortunately, due to the following two reasons, such simple strategy may not work well in practice. First, until now the diffusion is only performed on the current frame, thus there is no sequential information used in the observation model. Moreover, the calculated f is in the range $[0, 1]$. So the MAP estimation will always tend to find the one with the larger area as the target object, which is definitely not the case in most frames. For example, it can be seen in Fig. 5 (c) that the state estimated by f (i.e., green rectangle) improperly includes the top left player, which is not our target object.

To address above issues, we further define a temperature f_b to measure the confidence of nodes belonging to the background and set $f_b(\mathbf{p}) = 1$ for $\mathbf{p} \in \mathcal{B}_c^t$. Then we can perform a diffusion on $\mathcal{G}^s$ to propagate f_b from $\mathcal{B}_c^t$ to the other nodes. In this way, we achieve a background confidence map f_b on $\mathcal{V}_t$ (illustrated at the bottom right of the blue region in Fig. 1 (b)). It is also clear that simply calculating the confidence map by f_b (e.g., define $u_f = \mathbf{1} - f_b$) still cannot correctly identify our target (see the blue rectangle in Fig. 5 (d)). So we try to combine the foreground and background confidences for the final observation model. One possible idea is to utilize the multiplication strategy (i.e., calculating the confidence by $f \times u_f$), which has been used for saliency detection. However, such confidence without negative components will still make us choose object with larger area size (see the yellow rectangle in Fig. 5 (e)). So we would like to define a confidence map, in which the target region should have high positive value, while the background region must have high negative value. To achieve this goal, we define a signed confidence map by $f - f_b$, which is in the range $[-1, 1]$. It can be seen in Fig. 5 (f) that such signed confidence can successfully identify the target from complex background, thus lead to the optimal tracking window (i.e., the red rectangle).

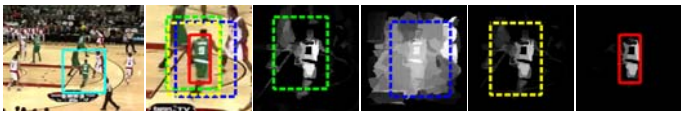
For each candidate state $\mathbf{x}_t^i$, we normalize its tracking window into canonical size (denoted as $\hat{\mathcal{A}}(\mathbf{x}_t^i)$)[11]. Let $v(x, y)$ be the value at location (x, y) on $\hat{\mathcal{A}}(\mathbf{x}_t^i)$. Then we accumulate v to obtain the confidence:

$$c(\mathbf{x}_t^i) = a(\mathbf{x}_t^i, \mathbf{x}_{t-1}) \sum_{(x,y)\in\hat{\mathcal{A}}(\mathbf{x}_t^i)} v(x, y), \tag{15}$$

where $a(\mathbf{x}_t^i, \mathbf{x}_{t-1}) = |\mathcal{A}(\mathbf{x}_t^i)|/|\mathcal{A}(\mathbf{x}_{t-1})|$ is an adaptive scale weight. It is easy to check that this confidence value does not take scale change into account. Finally, we normalize $c(\mathbf{x}_t^i)$ into $[0, 1]$ to compute the likelihood $p(\mathcal{Y}_t^i|\mathbf{x}_t^i)$ for all candidate targets $\{\mathbf{x}_t^i, i = 1 \cdots, m\}$.

10. Here note that we do not introduce guidance map for tracking.

11. The canonical size for the t-th frame is defined as the size of the tracked target in the $(t-1)$-th frame.



(a) (b) (c) (d) (e) (f)

Fig. 5. Confidence maps calculated by different strategies. (a) A new frame at time t and the surrounding window $\mathcal{C}'(\mathbf{x}_t)$. (b) Zoomed-in surrounding window and different candidate tracking windows (rectangles with dotted and solid lines). (c)-(f) are confidence maps defined by f, u_f, $f \times u_f$ and $f - f_b$, respectively. The corresponding tracking windows are also plotted on these maps.

6 Experiments

This section presents the evaluation of LTD for visual analysis. We first perform a simple image segmentation task to verify and compare the mechanism of our LTD model against conventional PDEs. We then apply LTD on image domain for saliency detection. Based on LTD saliency detection results, we further show that the performance of other vision tasks (e.g., image retargeting) can also be improved. Finally, we test LTD on videos for object tracking. Please notice that neither saliency detection nor object tracking has been addressed by PDEs. In each task, we compare the performance of LTD against many state-of-the-art methods on different benchmark data sets. In this paper, all experiments are run on the same PC with an Intel Core i7-3770 3.4GHz CPU that has 4 cores and 16GB memory, running Windows 7 (64-bit) and Matlab (Version 8.2). We also suggest readers to refer to Appendix for the evaluation methodology and more comprehensive experimental results. Please notice that this work is to develop a theoretical framework with insights for various vision tasks, not to provide a system to achieve best performance in each problem.

6.1 Image Segmentation (Model Verification)

We first design a simple image segmentation experiment to verify the mechanism of LTD and demonstrate the superiority of our framework over conventional PDEs for image analysis. For a given test image, we first show the results of two conventional image diffusion based PDEs (i.e., level set [38] and active contour [39]) on the bottom left of Fig. 6. It can be seen that the evolution of level set is very sensitive to the textures in the background. Though active contour diffusion can reduce the influence from background, parts of the foreground are not correctly segmented.

For LTD, we first consider the unsupervised loss L. It is shown in the pink region of Fig. 6 (denoted as "L") that LTD with L can segment both two cows from the background. However, such strategy cannot absolutely remove the background[12]. To verify the mechanism of the supervised LTD formulation, we introduce the information gain based regularizer R to the object function. Then we collect another image (shares similar object information with the test one) and manually label the foreground/background to generate our training image pairs (the

12. Please notice that actually the guidance map defined in Section 5.1.2 can easily address this problem. However, as the goal of this experiment is to verify our objective functions, here we do not introduce such guidance map for LTD.

rightmost column in Fig. 6). Here we generate two different label masks for either the two cows or a single cow and the corresponding discriminative regularizer (defined by (7)) are denoted as R_2 and R_1, respectively. By incorporating supervised information to LTD (respectively denoted as "$L+R_2$" and "$L + R_1$"), we can successfully segment the image following the priors learned from training data. Finally, we introduce an adaptive penalty W to the object function. The third column in the blue region of Fig. 6 illustrates that LTD with W (denoted as "$L + R_1 + W$") automatically chooses one node as heat source (we set the maximum number of nodes in heat souse as 3 in this experiment) and achieves the same segmentation results as that by "$L + R_1$". In fact, this experiment verified that our LTD can successfully incorporate discriminative label information from training data for diffusion learning.

6.2 Saliency Detection (LTD on Images)

In this subsection, we consider saliency detection and perform experiments on four benchmark image sets which are generated from three public databases, i.e., MSRA [40], ECSSD [41] and Berkeley [42]. We first conduct experiments on the widely used subset of MSRA with 1000 images, which is provided by [43] (MRSA-1000). Then the comparison is performed on the whole MSRA database with 5000 images (MSRA-5000). We also evaluate saliency detection performance on the recently released ECSSD database with 1000 images, which includes many semantically meaningful but structurally complex images for evaluation. Finally, we test algorithms on the 300 challenging images in the Berkeley image set. Here the number of superpixels is set to 200 for all the test images. We compare our LTD method with 19 state-of-the-art saliency detectors, such as BL [44], UFO [35], IT [45], AC [46], CA [47], CB [48], FT [43], GB [49], GS [50], LC [51], LR [34], MZ [52], RC [53], SER [54], SF [55], SR [56], SM [28], SVO [57], and XIE [33]. For quantitative comparison, we report the precision, recall and F-measure values for the three image sets, respectively. We also present ground truth (GT) salient regions and the saliency maps for compared methods.

Qualitative results: We first show example saliency maps computed by some typical saliency detectors in Fig. 7. As eye fixation prediction based methods (e.g., IT and GB) can only identify center-surround differences but miss most of the object information, here we do not show their results. The simple low-rank assumption in LR may be invalid when images contain complex structures. RC explores superpixels to highlight the object more uniformly, but the complex background always challenges such methods [47], [49], [53]. In SM, regions inside a salient object which share a similar color with the background will be regarded as part of the background. As a result, they may share the same saliency value with the background region. In contrast, our method can successfully highlight the salient regions and preserve the boundaries of objects, thus producing results that are much closer to GT.

Quantitative results: The quantitative comparisons between our method and other state-of-the-art approaches are performed on MSRA-1000, MSRA-5000, ECSSD and Berkeley, respectively. The average precision, recall, and F-measure values are computed in the same way as in [43], [53], [33], [28]. The precision-recall curves of all 19 methods are presented in Fig. 8. The average precision, recall and F-measure values using an adaptive threshold [43] are shown in Fig. 9. The center-surround contrast based methods, such as IT, GB and CA, can only detect parts of boundaries of salient objects. Using superpixels, recent approaches, such as CB and RC, are capable of detecting salient objects. But they usually fail to suppress background regions and also lead to lower precision-recall curves. In Fig. 8, we observe that GS shares a similar precision with ours when the recall is larger than 0.96. However, the geodesic distance to boundary strategy in that method tends to recognize background parts as salient regions when their colors are significantly different from the boundary. So in most cases, their precision is much lower than ours at the same recall level. It can be seen that overall our LTD saliency detector performs well on most of these challenging image sets and only the recently proposed BL algorithm is comparable to LTD.

These results also verify that the proposed learning strategy can successfully incorporate both bottom-up and top-down information into saliency diffusion. We also report the CPU time of several saliency detectors in Table 1. We observed that the methods with C/C++ implementation (e.g., LC, RC and FT, denoted as "C") achieve a fast speed. Due to the simple formulations, the speed of the eye fixation prediction methods (e.g., IT and SR, denoted as "M") are also fast even with the MATLAB implementation. However, their performance are worse than the object based saliency detectors. Overall, LTD is the fastest detector among methods with MATLAB implementation, including C/C++ library (denoted as "M&C").

6.3 Image Retargeting (Saliency Driven Application)

In this subsection, we evaluate LTD on saliency driven visual analysis problems. To address this issue, we consider the image retargeting task, which is to resize an image by expanding or shrinking the non-informative regions. It is easy to check that retargeting algorithm relies on the availability of saliency map which is used to specify relative importance across image parts. We perform seam carving retargeting technique [58] with saliency maps from CA, RC and LTD on example images. It can be seen from Fig. 10 that our LTD helps produce better retargeting results than CA and RC. This is because image retargeting requires that the entire salient objects should be uniformly highlighted. In Fig. 11, we observe that CA saliency maps only highlight the object boundaries and RC saliency maps fail to distinguish the object and the background. In contrast, LTD can provide more accurate and smooth saliency maps, thus is more suitable for retargeting application.

6.4 Object Tracking (LTD on Sequences)

To test the performance of LTD on sequential data (e.g., videos), we consider the task of object tracking and evaluate our LTD based tracker (proposed in Section 5.2) against other 19 state-of-the-art tracking methods, i.e., DSST [59], TLD [60], ASLA [61], CXT [62], VTD [63], CSK [64], DFT [65], L1APG [66], MTT [67], OAB [68], LOT [69], MIL [70], IVT [36], Frag [71], SPT [72], ORIA [73], CT [74], VR-V [75] and SPOT [76], on the tracking benchmark [77] with 50 challenging video sequences. Here the number of superpixels

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2522415, IEEE Transactions on Pattern Analysis and Machine Intelligence

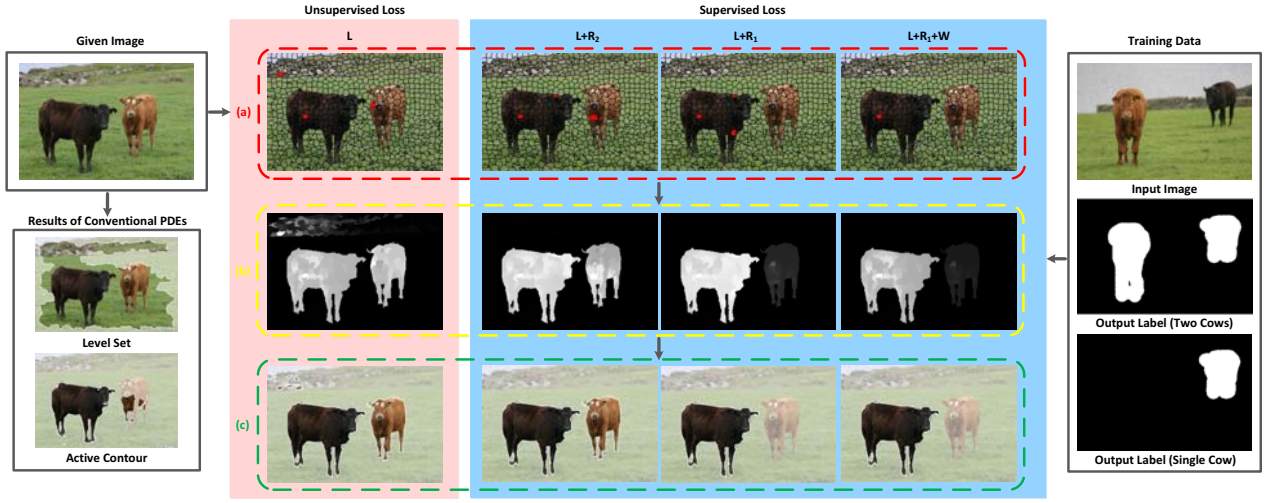


Fig. 6. Comparisons between conventional PDEs (bottom left) and LTD on image segmentation. The results of LTD with unsupervised and supervised losses are presented on the pink and blue regions, respectively. We also use dotted rectangles with different colors to distinguish step results of LTD: (a) heat source determined by different objective functions (red), (b) stable temperature of learned diffusion systems (yellow), and (c) final segmentation results (green). The training images with different labels are also illustrated on the rightmost.

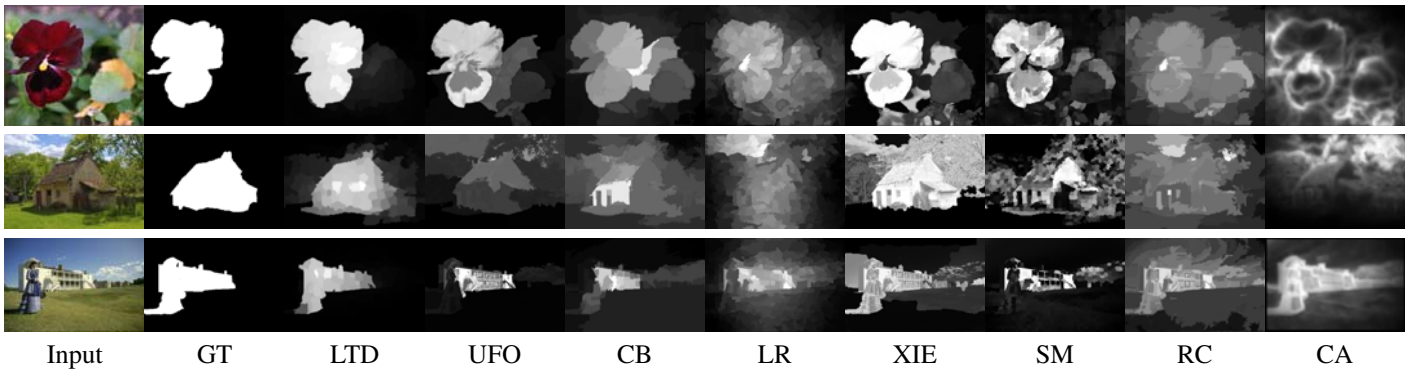


Fig. 7. Qualitative comparisons on images from MSRA (top), ECSSD (middle) and Berkeley (bottom) databases.

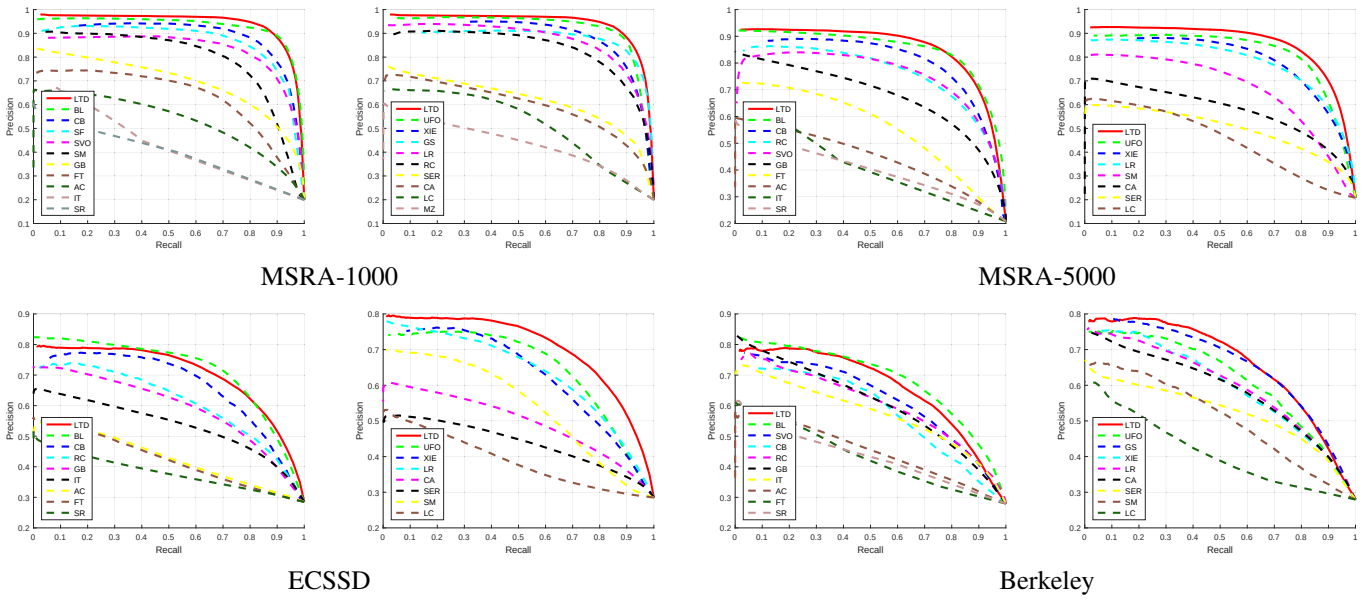
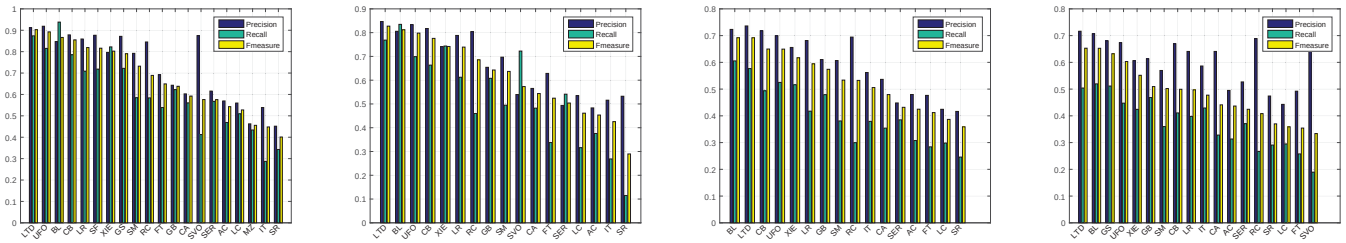


Fig. 8. The average precision-recall curves.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2522415, IEEE Transactions on Pattern Analysis and Machine Intelligence



MSRA-1000 MSRA-5000 ECSSD Berkeley

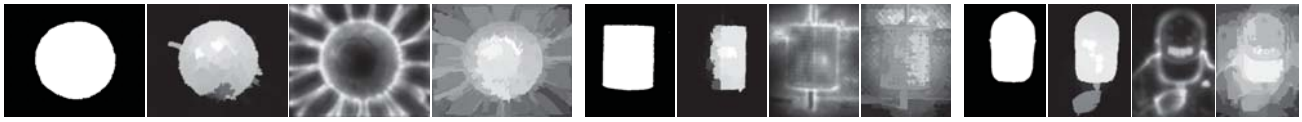
Fig. 9. The average precisions, recalls and F-measures using adaptive thresholding.

TABLE 1
Average running time (seconds per image) for different methods on MSRA-1000 database.

Method	LTD	UFO	CB	XIE	LR	LC	SR	SM	RC	FT	GB	CA	SER	AC	IT
Code	M&C	M&C	M&C	M	M&C	C	M	M&C	C	C	M&C	M&C	M	M	M
Time (s)	0.38	7.69	0.63	95.91	12.42	0.012	0.02	5.21	0.027	0.011	0.61	33.85	1.88	58.35	0.18

Input LTD CA RC Input LTD CA RC Input LTD CA RC

Fig. 10. Image retargeting results of seam carving [58] with CA, RC and LTD.



GT LTD CA RC GT LTD CA RC GT LTD CA RC

Fig. 11. GT and saliency maps of CA, RC and LTD for input images in Fig. 10.

is set to 250. For k nearest neighbor graph, we set $k = 10$. The number of particles is set to 600 for each frame. The number of initial training frames is set to 4 and the observation probabilities are updated every 25 frames.

TABLE 3
Average FPS for particle filter based trackers on 50 videos in the benchmark [77].

Method	LTD	IVT	ASLA	L1APG	MTT	SPT
Code	M&C	M&C	M&C	M&C	M	M&C
FPS	3.0	16.0	1.8	2.3	0.7	0.5

Qualitative results: For better readability, we first demonstrate qualitative results of LTD together with 11 trackers on 6 representative videos. In Fig. 12 (a)-(c), we show the performance of different trackers in terms of illumination variation, deformation, out-of-plane rotation and background clutters even when the target objects undergo severe occlusion. It can be seen that IVT and L1APG drift away from the target when it undergoes non-rigid shape deformation and large pose change. MTT, ASLA and CXT also do not accurately locate the target all the time. In contrast, our LTD tracker performs well on all these sequences. This is mainly because that LTD is able to exploit both the target and the background appearance thus can alleviate influence from background pixels. Moreover, as we define features in the color space rather than modeling the holistic appearance of objects, LTD is not sensitive to the shape changes, thus can generate the most accurate results. Fig. 12 (d)-(f) show representative results on 3 video sequences which highlight other challenging factors (e.g., out-of-view, motion blur, fast motion and scale variation). It can be seen that our tracker also performs well in these cases.

We observe that SPOT achieves very good performance on "Tiger2" sequence (in Fig. 12 (f)) and the quantitative results are even slightly better than LTD (see Table 2). But unfortunately, it cannot achieve good results (even fail at the beginning of some sequences) on other 5 test videos. This is possibly because SPOT may not handle severe occlusions in Fig. 12 (a)-(c) or the small size of the object in Fig. 12 (d)-(e).

Quantitative results: To assess quantitative performances of these trackers, we first report the overlap rate (OR) and center location error (CLE) in Table 2 for 6 example videos. To further show the overall performances on the whole tracking benchmark, we follow evaluation protocols in [77] to plot the success and precision of all the 19 trackers on 50 video sequences in Fig. 13. The average performance scores are also reported in legends of Fig. 13. The average precision value at threshold 20 pixels for each method is shown in the legend of the precision plot. The legend of the success plot contains the area-under-curve (AUC) score for each tracker. It can be

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2522415, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 2
Average OR (top, higher is better) and CLE (bottom, lower is better) for 6 example videos. The best and the second best results are shown in **bold** and underline fonts, respectively.

Sequence	LTD	SPT	TLD	ASLA	OAB	VTD	MTT	L1APG	CXT	Frag	IVT	SPOT
Basketball	**0.74**	0.68	0.02	0.38	0.03	0.73	0.19	0.23	0.02	0.62	0.11	0.01
	8.11	18.09	213.86	82.63	204.84	**5.62**	106.80	137.53	214.57	13.02	107.11	169.86
Bolt	**0.61**	0.56	0.16	0.01	0.04	0.37	0.01	0.01	0.02	0.13	0.01	0.01
	7.67	8.59	90.92	374.74	253.76	25.16	408.61	408.41	385.49	183.38	397.05	191.11
Jogging	**0.70**	0.61	0.66	0.14	0.42	0.13	0.13	0.15	0.13	0.48	0.14	0.20
	7.72	8.92	13.56	169.86	36.78	122.19	157.12	145.85	139.7	37.54	138.22	72.23
Freeman4	**0.51**	0.08	0.22	0.13	0.11	0.16	0.22	0.34	0.17	0.14	0.15	0.01
	7.59	70.95	39.18	70.24	133.38	61.68	23.55	22.12	65.64	72.27	43.04	108.70
Skiing	**0.48**	0.11	0.07	0.09	0.08	0.07	0.09	0.07	0.09	0.03	0.08	0.02
	6.62	259.82	142.83	266.61	192.54	263.27	256.42	265.87	153.13	270.01	272.36	260.00
Tiger2	0.55	0.15	0.26	0.14	0.15	0.30	0.29	0.24	0.36	0.12	0.09	**0.57**
	19.49	99.74	36.17	84.69	251.97	40.87	48.75	65.16	41.44	113.54	102.47	**17.91**

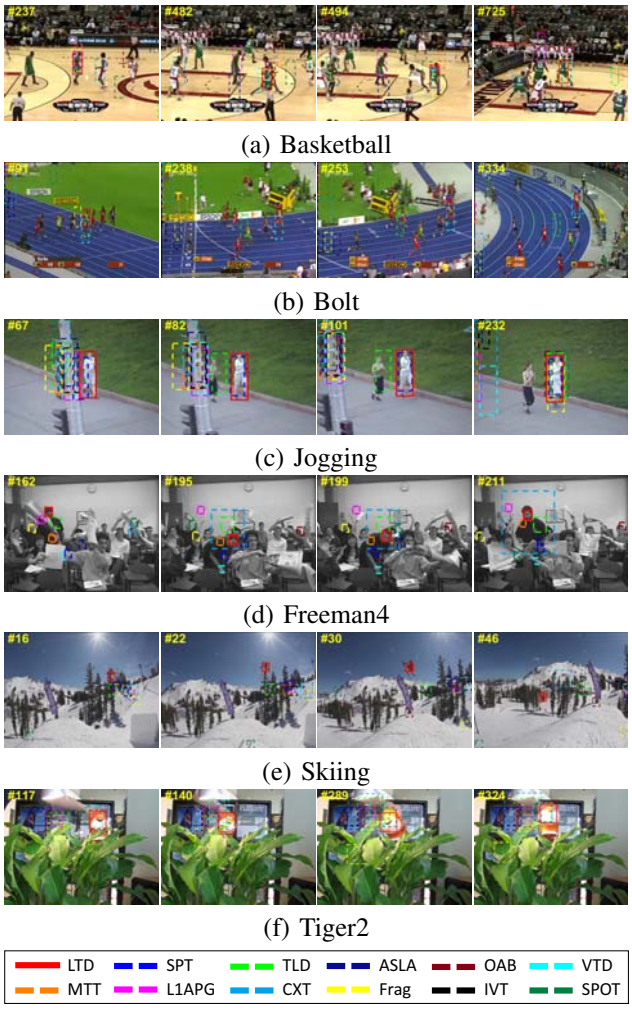


Fig. 12. Sampled tracking results on 6 example videos.

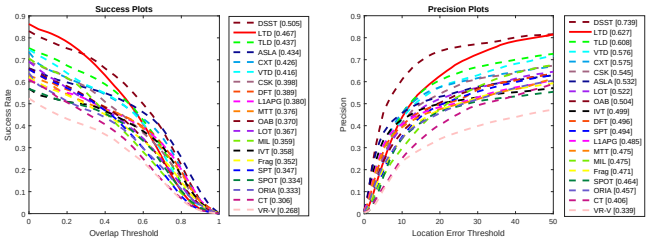


Fig. 13. The success and precision plots over all 50 videos.

observed that our LTD achieves very favorable performance and only the recently proposed DSST tracker performs better than it on this challenging benchmark.

Finally, we compare the speed (i.e., frames per second, FPS) of all particle filter based trackers (i.e., IVT, ASLA, L1APG, MTT, SPT and our LTD) over 50 videos in Table 3. It can be seen that IVT is much faster than other particle filter based trackers as it only involves a simple subspace updating process on each frame. Our LTD is the second fastest among the compared trackers.

7 Conclusions and Future Work

In this paper, we proposed a novel PDE learning framework, called learning to diffuse (LTD), for visual analysis. Within the framework, we extract both the generative data distribution and the discriminative category information for diffusion learning. We verify the proposed model by solving two challenging visual analysis tasks (i.e., saliency detection and object tracking). To our best knowledge, neither of these problems has ever been addressed by PDE based methods before. Comprehensive experimental comparisons with 19 saliency detectors on 4 saliency databases and 19 trackers on 50 tracking benchmark videos demonstrate the efficiencies and effectiveness of LTD on both saliency detection and object tracking.

Our LTD based strategies for particular vision tasks are still rudimentary and several aspects can be improved in the future. First, in saliency detection, the human perceptions (i.e., center and color priors) work well for most test images. But the guidance may occasionally fail to control the visual diffusion when these perceptions are in conflict with the salient structure. For example, as the center prior based guidance cannot highlight saliency around the image boundary, LTD may detect incorrect salient regions (e.g., LTD-1 in Fig. 14). Though redesigned guidance for this image can improve the performance of LTD (e.g., LTD-2 in Fig. 14), we believe more efforts should be made for adaptive guidance learning in real world scenarios. Second, for object tracking, LTD is currently performed in RGB color space. The bottom row of Fig. 14 illustrated that the tracking results on "Tiger2" sequence can be improved if we incorporate histogram-of-oriented-gradient (HOG) feature [78] into LTD framework. Accordingly, the OR and CLE scores are respectively improved from 0.55 and 19.49 to 0.60 and 16.35 on this sequence. It can be seen that LTD with HOG is actually

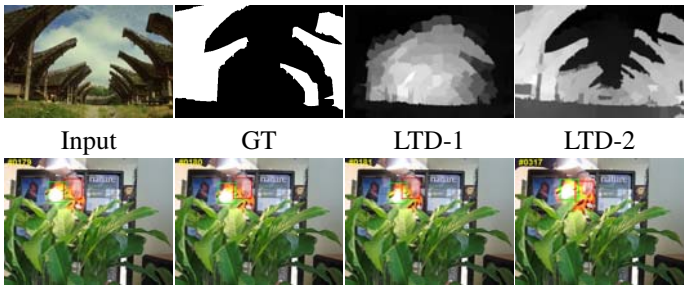


Fig. 14. Failed examples of LTD in saliency detection (top) and object tracking (bottom) and possible improvements. We obtain LTD-1 using the guidance defined by (11) and LTD-2 using a new guidance defined by $\hat{u} = \hat{f}_b \times u_c$, where $\hat{f}_b$ is calculated by background diffusion with a pre-selected boundary condition and u_c is the color prior. The tracking results of LTD and LTD with HOG are denoted by red and green boxes, respectively.

better than the HOG based SPOT tracker, whose OR and CLE are 0.57 and 17.91, respectively. This experiment suggests that the properly designed feature space could give rise to better performance for some particular data and tasks. Therefore, more investigations on feature engineering are necessary when we utilize LTD for more complex vision problems. Third, we observed that scaling techniques (e.g., multi-scale boosting [44] and scale pyramid [59]) achieved good performances in the experimental comparisons. This suggests us to extend LTD for multi-scale diffusions learning to further improve the performance.

ACKNOWLEDGMENTS

The authors thank all reviewers for their helpful comments. R. Liu is supported by National Natural Science Foundation of China (NSFC) (Nos. 61300086, 61432003), Fundamental Research Funds for the Central Universities (No. DUT15QY15) and the Hong Kong Scholar Program (No. XJ2015008). J. Cao is supported by NSFC (No. 61363048). Z. Lin is supported by National Basic Research Program of China (973 Program) (No. 2015CB352502), NSFC (Nos. 61272341 and 61231002), and Microsoft Research Asia Collaborative Research Program. S. Shan is supported by NSFC (No. 61222211).

REFERENCES

[1] R. Liu, J. Cao, Z. Lin, and S. Shan, "Adaptive partial differential equation learning for visual saliency detection," in *CVPR*, 2014.
[2] T. Chan and J. Shen, *Image processing and analysis: variational, PDE, wavelet, and stochastic methods*. SIAM, 2005.
[3] O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier, and F. Lenzen, *Variational methods in imaging*. Springer, 2008, vol. 167.
[4] T. Lindeberg, *Scale-space theory in computer vision*. Springer, 1993.
[5] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE T. PAMI*, vol. 12, no. 7, pp. 629–639, 1990.
[6] B. M. ter Haar Romeny, *Geometry-driven diffusion in computer vision*. Kluwer Academic Norwell, MA, 1994.
[7] G. Sapiro, *Geometric partial differential equations and image analysis*. Cambridge University Press, 2006.
[8] F. Cao, *Geometric curve evolution and image processing*. Springer, 2003.
[9] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.
[10] R. Liu, Z. Lin, W. Zhang, and Z. Su, "Learning PDEs for image restoration via optimal control," in *ECCV*, 2010.
[11] R. Liu, Z. Lin, W. Zhang, K. Tang, and Z. Su, "Toward designing intelligent PDEs for computer vision: An optimal control approachn," *Image and Vision Computing*, vol. 31, no. 1, pp. 43–56, 2013.
[12] G. Aubert and P. Kornprobst, *Mathematical problems in image processing: partial differential equations and the calculus of variations*. Springer, 2006, vol. 147.
[13] J. J. Koenderink, "The structure of images," *Biological Cybernetics*, vol. 50, no. 5, pp. 363–370, 1984.
[14] A. P. Witkin, "Scale-space filtering: A new approach to multi-scale description," in *ICASSP*, 1984.
[15] R. Fattal, D. Lischinski, and M. Werman, "Gradient domain high dynamic range compression," *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3, pp. 249–256, 2002.
[16] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 313–318, 2003.
[17] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum, "Poisson matting," *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 315–321, 2004.
[18] L. A. Vese and T. F. Chan, "A multiphase level set framework for image segmentation using the mumford and shah model," *IJCV*, vol. 50, no. 3, pp. 271–293, 2002.
[19] E. S. Brown, T. F. Chan, and X. Bresson, "A convex relaxation method for a class of vector-valued minimization problems with applications to mumford-shah segmentation," *UCLA CAM Report*, pp. 10–43, 2010.
[20] J.-F. Cai and W. Xu, "Guarantees of total variation minimization for signal recovery," *arXiv preprint arXiv:1301.6791*, 2013.
[21] S. Kindermann, S. Osher, and P. W. Jones, "Deblurring and denoising of images by nonlocal functionals," *Multiscale Modeling & Simulation*, vol. 4, no. 4, pp. 1091–1115, 2005.
[22] G. Gilboa and S. Osher, "Nonlocal operators with applications to image processing," *Multiscale Modeling & Simulation*, vol. 7, no. 3, pp. 1005–1028, 2008.
[23] A. Krause and D. Golovin, "Submodular function maximization," *Tractability: Practical Approaches to Hard Problems*, vol. 3, 2012.
[24] G. Calinescuy, C. Chekuri, M. Pal, and J. Vondrak, "Maximizing a monotone submodular function subject to a matroid constraint," *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1740–1766, 2011.
[25] J. Weickert, *Anisotropic diffusion in image processing*. Teubner Stuttgart, 1998, vol. 1.
[26] A. Krause and C. Guestrin, "Beyond convexity: Submodularity in machine learning," in *ICML Tutorials*, 2008.
[27] G. Kim, E. P. Xing, L. Fei-Fei, and T. Kanade, "Distributed cosegmentation via submodular optimization on anisotropic diffusion," in *ICCV*, 2011.
[28] Z. Jiang and L. S. Davis, "Submodular salient region detection," in *CVPR*, 2013.
[29] L. Cao, Z. Li, Y. Mu, and S.-F. Chang, "Submodular video hashing: a unified framework towards video pooling and indexing," in *ACM Multimedia*, 2012.
[30] G. L. Nemhauser and L. A. Wolsey, "Best algorithms for approximating the maximum of a submodular set function," *Mathematics of Operations Research*, vol. 3, no. 3, pp. 177–188, 1978.
[31] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE T. PAMI*, vol. 34, no. 11, pp. 2274–2282, 2012.
[32] J. Van De Weijer, T. Gevers, and A. D. Bagdanov, "Boosting color saliency in image feature detection," *IEEE T. PAMI*, vol. 28, no. 1, pp. 150–156, 2006.
[33] Y. Xie, H. Lu, and M.-H. Yang, "Bayesian saliency via low and mid level cues," *IEEE T. IP*, vol. 22, no. 5, pp. 1689–1698, 2013.
[34] X. Shen and Y. Wu, "A unified approach to salient object detection via low rank matrix recovery," in *CVPR*, 2012.
[35] P. Jiang, H. Ling, J. Yu, and J. Peng, "Salient region detection by ufo: Uniqueness, focusness and objectness," in *ICCV*, 2013.
[36] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *IJCV*, vol. 77, no. 1-3, pp. 125–141, 2008.
[37] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.
[38] C. Li, C. Xu, C. Gui, and M. D. Fox, "Distance regularized level set evolution and its application to image segmentation," *IEEE T. IP*, vol. 19, no. 12, pp. 3243–3254, 2010.
[39] X. Bresson, S. Esedo?lu, P. Vandergheynst, J.-P. Thiran, and S. Osher, "Fast global minimization of the active contour/snake model," *Journal of Mathematical Imaging and Vision*, vol. 28, no. 2, pp. 151–167, 2007.

[40] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," *IEEE T. PAMI*, vol. 33, no. 2, pp. 353–367, 2011.
[41] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *CVPR*, 2013.
[42] V. Movahedi and J. H. Elder, "Design and perceptual validation of performance measures for salient object segmentation," in *CVPR Workshops*, 2010.
[43] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *CVPR*, 2009.
[44] N. Tong, H. Lu, X. Ruan, and M.-H. Yang, "Salient object detection via bootstrap learning," in *CVPR*, 2015.
[45] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE T. PAMI*, vol. 20, no. 11, pp. 1254–1259, 1998.
[46] R. Achanta, F. Estrada, P. Wils, and S. Süsstrunk, "Salient region detection and segmentation," in *Computer Vision Systems*, 2008.
[47] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *IEEE T. PAMI*, vol. 34, no. 10, pp. 1915–1926, 2012.
[48] H. Jiang, J. Wang, Z. Yuan, T. Liu, N. Zheng, and S. Li, "Automatic salient object segmentation based on context and shape prior." in *BMVC*, 2011.
[49] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *NIPS*, 2006.
[50] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *ECCV*, 2012.
[51] Y. Zhai and M. Shah, "Visual attention detection in video sequences using spatiotemporal cues," in *ACM Multimedia*, 2006.
[52] Y.-F. Ma and H.-J. Zhang, "Contrast-based image attention analysis by using fuzzy growing," in *ACM Multimedia*, 2003.
[53] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "Global contrast based salient region detection," in *CVPR*, 2011.
[54] H. J. Seo and P. Milanfar, "Static and space-time visual saliency detection by self-resemblance," *Journal of Vision*, vol. 9, no. 12, 2009.
[55] F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *CVPR*, 2012.
[56] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *CVPR*, 2007.
[57] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai, "Fusing generic objectness and visual saliency for salient object detection," in *ICCV*, 2011.
[58] M. Rubinstein, A. Shamir, and S. Avidan, "Improved seam carving for video retargeting," *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3, p. 16, 2008.
[59] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *BMVC*, 2014.
[60] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," in *CVPR*, 2010.
[61] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *CVPR*, 2012.
[62] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *CVPR*, 2011.
[63] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *CVPR*, 2010.
[64] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *ECCV*, 2012.
[65] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *CVPR*, 2012.
[66] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *CVPR*, 2012.
[67] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *CVPR*, 2012.
[68] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting." in *BMVC*, 2006.
[69] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," in *CVPR*, 2012.
[70] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *CVPR*, 2009.
[71] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *CVPR*, 2006.
[72] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," in *ICCV*, 2011.
[73] Y. Wu, B. Shen, and H. Ling, "Online robust image alignment via iterative convex optimization," in *CVPR*, 2012.
[74] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *ECCV*, 2012.
[75] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE T. PAMI*, vol. 27, no. 10, pp. 1631–1643, 2005.
[76] L. Zhang and L. J. van der Maaten, "Preserving structure in model-free tracking," *IEEE T. PAMI*, vol. 36, no. 4, pp. 756–769, 2014.
[77] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *CVPR*, 2013.
[78] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.

Risheng Liu received the B.Sc and Ph.D degrees both in mathematics from Dalian University of Technology in 2007 and 2012, respectively. He was a visiting scholar in Robotic Institute of Carnegie Mellon University from 2010 to 2012. He is currently an Associate Professor at Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, School of Software Technology, Dalian University of Technology. His research interests include machine learning, computer vision, multimedia and optimization. He was a co-recipient of the IEEE ICME Best Student Paper Award in both 2014 and 2015. He is a member of the IEEE.

Guangyu Zhong received the B.Sc. degrees in mathematics from Dalian University of Technology in 2013. She is currently a PhD student in School of Mathematical Science, Dalian University of Technology. She is a visiting PhD student in Electrical Engineering and Computer Science at University of California, Merced from 2015 to 2016. Her research interests include machine learning and computer vision.

Junjie Cao received the Ph.D degrees in computational mathematics from Dalian University of Technology in 2010. He was a visiting scholar in School of Computing Science, Simon Fraser University in 2015. He is currently an Assistant Professor at School of Mathematical Sciences, Dalian University of Technology. His research interests include computer graphics, computer version and multimedia.

Zhouchen Lin (M'00-SM'08) received the Ph.D. degree in applied mathematics from Peking University in 2000. He is currently a Professor with the Key Laboratory of Machine Perception, School of Electronic Engineering and Computer Science, Peking University. His research interests include computer vision, image processing, machine learning, pattern recognition, and numerical optimization. He is area chairs of CVPR 2014, ICCV 2015, NIPS 2015, AAAI 2016, CVPR 2016, and IJCAI 2016. He is associate editors of the IEEE TPAMI and the IJCV.

Shiguang Shan received the MS degree in computer science from the Harbin Institute of Technology, China, in 1999, and the PhD degree in computer science from the Institute of Computing Technology (ICT), CAS, Beijing, in 2004. He has been with ICT, CAS since 2002 and has been a professor since 2010. His research interests cover image analysis, pattern recognition, and computer vision. He is a Senior member of the IEEE.

Zhongxuan Luo received the B.S. and M.S. degrees from Jilin University in 1985 and 1988, respectively, and the Ph.D. degree from the Dalian University of Technology in 1991, all in computational mathematics. He has been a Full Professor with the School of Mathematical Sciences, Dalian University of Technology, since 1997. He is currently the Dean of the School of Software Technology. His research interests include computational geometry and computer vision.