Learning-Based PDE: A New Perspective for PDE Methods in Computer Vision

Risheng Liu
Dalian University of Technology
http://cggi.dlut.edu.cn/people/~rsliu/
rsliu@dlut.edu.cn
Outline

• **Motivation**
• Learning a specified PDE system
• Learning the general PDE system
• Discussions
PDE Methods in Computer Vision

• PDEs have been applied to
  – Denoising, Deblurring, Inpainting, Segmentation, Stereo, ...

• Roughly have two kinds of methodologies for PDE design
  – Direct design: write down PDEs directly using some insights from physics
  – Variational design: energy functional $\rightarrow$ Euler-Lagrange equation

Anisotropic Diffusion: $\frac{\partial u}{\partial t} = \text{div}(c(\|\nabla u\|)\nabla u)$, $u|_{t=0} = f$, $\frac{\partial f}{\partial n}|_{\partial D} = 0$

– Total Variation: $\min_u \int_{\Omega} \|\nabla u\| + \lambda \|f - u\|^2$

Not a hot topic in computer vision society RECENTLY!
Why PDEs fell out of favor?

• PDE is a powerful tool for data analysis
  – PDE is indeed a general regressor to approximate the nonlinear mapping between input and output vision data.
  – Differential system should be more efficient than linear equation for data analysis.

• Unfortunately...
  – It is challenging to design PDE in handcraft way (huge feasible solution set)
    • Deep insights for the problem and the data
    • High mathematical skills for building a PDE system
  – Fixed PDE formulation and/or boundary conditions is lack of flexibility

Beautiful mathematical formulation, **BUT** poor performance and limited application!!

理想很丰满、现实很骨感
New Perspective: PDE + Learning

Two possible ways to incorporate “learning” ideas into PDE designing:

• For a specific kind of PDE system (e.g., heat diffusion), we may adaptively learn its form and boundary for different data and tasks

• For using PDE to address general vision tasks, we may learn the PDE form from the “dictionary of differential operators” for different image processing and analysis tasks
Outline

• Motivation

• **Learning a specified PDE system**

• Learning the general PDE system

• Discussions
Saliency Detection

Saliency detection is a hot topic in computer vision recently.

• **Bottom-up methods: from image features**
  – Feature + Clustering [Itti’98, Cheng’11, Lang’12, Shen’12, Wei’12, Xie’13]
  – Candidates + Refinement [Jiang’13, Yang’13]

• **Top-down methods: from saliency model**
  – CRF [Liu’11, Yang’12].
Learning Heat Diffusion for Saliency Detection

• We assume that visual attention starts from saliency seeds and then propagates to a salient region.

• General heat diffusion system for image processing

\[
\begin{align*}
\frac{\partial f(p, t)}{\partial t} &= F(f, \nabla f), \quad (p, t) \in \Omega \times (0, T], \\
f(p, t) &= g(t), \quad p \in \partial \Omega \times (0, T], \\
f(p, 0) &= I(p).
\end{align*}
\]

• But not for saliency detection!

• Difficulties of PDE methods
  – Hard to describe high-level information
  – Using fixed boundary conditions
\[
\frac{\partial f(p, t)}{\partial t} = F(f, \nabla f), \quad \text{s.t. boundary conditions.}
\]

Propagate visual attention from saliency seeds to other image elements

We need a diffusion equation

\[
F(f, \nabla f) = \text{div}(K_p \nabla f(p)) + \lambda(f(p) - g(p)).
\]

Guidance Map

We need Dirichlet boundary conditions

Steady state

\[
F(f, \nabla f) = 0, \quad \text{s.t.} \quad f(g) = 0, f(p) = s_p, p \in S.
\]

How to determine \( K_p, g(p), \) and \( s_p \)?

How to determine \( S \)?
Learning PDEs – Graph Representation

- Undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

- Oversegment to superpixels: $\mathcal{V} = \{p_1, \cdots, p_{|\mathcal{V}|}\}$
Learning PDEs – Candidate Foreground and Background

• Use Harris operator to detect corners and contour points and estimate a convex hull \( C \).
  – Candidate foreground \( F_c \) = nodes inside \( C \).
• Add adjacent nodes to \( C \) to form an expanded hull \( C' \).
  – Pure background \( B_c \) = nodes outside \( C' \).
Learning PDEs – Graph Construction

• Undirected graph $G = (V, E)$.
  – Nodes in $F_c$ within distance 2 are connected to each other (Fig. (a)).
  – All nodes in $B_c$ are connected to each other: for smoothness of background (Fig. (b)).

  Defines neighborhood $N_p$.

  – All nodes are connected an environment point $g$.

The red and yellow regions in (a) and (b) represent $F_c$ and $B_c$, respectively. Lines in (b) indicate that all nodes in $B_c$ are connected to each other.
Learning PDEs – Learning $K_p$

- Feature vectors: $\{h(p), p \in \mathcal{V}\}$ – mean of each superpixel in CIELAB color space.

$$K_p = \text{diag}(k(p, q_1), \cdots, k(p, q_{|\mathcal{N}_p| - 1}), \varepsilon_g),$$

where $\mathcal{N}_p = \{q_1, \cdots, q_{|\mathcal{N}_p| - 1}, g\}$ is the neighborhood of $p$,

$$k(p, q) = \exp(-\beta \|h(p) - h(q)\|^2),$$

and $0 < \varepsilon_g \ll 1$ is the dissipation conductance at $p$. 
Learning PDEs – Learning $g(p)$ and $s_p$

$g(p) = f_i(p) \times f_c(p) \times f_f(p),$

and then normalized.

- $f_c(p)$ and $f_i(p)$ are computed according to [Shen’12] and [Judd’09].
- $f_f(p) = 1 - f_b(p)$: probability of belonging to foreground.
  \[ \text{div}(K_p \nabla f_b(p)) = 0, \quad \text{s.t.} \quad f_b(g) = 0, f_b(p) = 1, p \in B_c. \]

\[ s_p = g(p), \text{ for } p \in S. \]

Learning PDEs – Learning $S$

- Optimize saliency seeds by searching for the most representative foreground nodes in $\mathcal{F}_c$ as $S$.

- Maximize the sum of scores $f$ w.r.t. all image elements in $\mathcal{V}$.

$$\max_{S \subset \mathcal{F}_c, |S| \leq n} L(S) \equiv \sum_{p \in \mathcal{V}} f(p),$$

subject to

$$\begin{align*}
\text{div}(K_p \nabla f(p)) + \lambda(f(p) - g(p))) = 0, \\
f(g) = 0, f(p) = s_p, p \in S.
\end{align*}$$

**Theorem:** $L(S)$ is submodular and monotone w.r.t. $S$.

**Greedy algorithm works well.**
The Pipeline of PDE Based Saliency Detection

- **Input image**
- **Superpixel segmentation**
- **Pure background (outside)**
- **Candidate foreground (inside)**
- **Color prior**
- **Center prior**
- **Background prior**
- **Guidance map**
- **Saliency score map**
- **Saliency seeds (yellow regions)**
- **Masked saliency region**
Experiments – Qualitative Comparison

The top three rows are examples in MSRA and the bottom is in Berkeley.
Experiments – Quantitative Comparisons

Results on the **MSRA-5000** image set. (a) Precision-recall curves. (b) Average precision, recall, and F-measure values.
Experiments – Quantitative Comparisons

Results on the Berkeley image set. (a) Precision-recall curves. (b) Average precision, recall, and F-measure values.
Summary

• Propose an adaptive PDE to model the saliency diffusion.
  – The first PDE based method for saliency detection.
  – Learn both the governing equation and the boundary condition of the PDE.

• Optimize saliency seeds by submodular maximization.
  – Saliency seeds make the boundary condition.

• Learning a specific PDE system for more vision and graphics tasks (Ongoing)
  – Robust object tracking
  – Mesh segmentation and labeling
Outline

• Motivation
• Learning a specified PDE system
• **Learning the general PDE system**
• Discussions
Can we just consider PDEs as a **BLACK BOX** for computer vision?

- A user only have to prepare input/output training data!
- The same framework for various problems!
- Can we have such a convenient way?

**Possible!!!**

Learning a general PDE system for computer vision

- Design a general “**dictionary**” of differential operators
- Learn combination “**codes**” for specific task and/or data
General PDE Formulation

Most existing evolutionary PDEs for a image \( u \) can be brought to as follows:

\[
\frac{\partial u}{\partial t} - F(u, \nabla u, H_u) = 0,
\]

where \( F \) is the function of \( u \), \( \nabla u \) and \( H_u \).

- The space of PDEs is infinitely dimensional
- The designing of PDEs is heavily rely on intuition
- It is hard to quantify and describe the intuition
- The designed PDEs can only reflect very limited aspects of a vision task
## Dictionary for PDEs in Computer Vision
(Shift/Rotation Invariant Fundamental Differential Invariants)

<table>
<thead>
<tr>
<th>(j)</th>
<th>(\text{inv}_j(u,v))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0,1,2</td>
<td>1, (v, u)</td>
</tr>
<tr>
<td>3,4</td>
<td>(|\nabla v|^2 = v_x^2 + v_y^2, |\nabla u|^2 = u_x^2 + u_y^2)</td>
</tr>
<tr>
<td>5</td>
<td>((\nabla v)^T \nabla u = v_x u_x + v_y u_y)</td>
</tr>
<tr>
<td>6,7</td>
<td>(\text{tr}(H_v) = v_{xx} + v_{yy}, \text{tr}(H_u) = u_{xx} + u_{yy})</td>
</tr>
<tr>
<td>8</td>
<td>((\nabla v)^T \nabla v = v_x^2 v_{xx} + 2v_x v_y v_{xy} + v_y^2 v_{yy})</td>
</tr>
<tr>
<td>9</td>
<td>((\nabla v)^T H_v \nabla v = v_x^2 u_{xx} + 2v_x v_y u_{xy} + v_y^2 u_{yy})</td>
</tr>
<tr>
<td>10</td>
<td>((\nabla v)^T H_v \nabla u = v_x u_x v_{xx} + (v_x u_y + v_y u_x) v_{xy} + v_y u_y v_{yy})</td>
</tr>
<tr>
<td>11</td>
<td>((\nabla v)^T H_u \nabla u = v_x u_x u_{xx} + (v_x u_y + v_y u_x) u_{xy} + v_y u_y u_{yy})</td>
</tr>
<tr>
<td>12</td>
<td>((\nabla u)^T H_v \nabla u = u_x^2 v_{xx} + 2u_x u_y v_{xy} + u_y^2 v_{yy})</td>
</tr>
<tr>
<td>13</td>
<td>((\nabla u)^T H_u \nabla u = u_x^2 u_{xx} + 2u_x u_y u_{xy} + u_y^2 u_{yy})</td>
</tr>
<tr>
<td>14</td>
<td>(\text{tr}(H_v^2) = v_{xx}^2 + 2v_{xy}^2 + v_{yy}^2)</td>
</tr>
<tr>
<td>15</td>
<td>(\text{tr}(H_v H_u) = v_{xx} u_{xx} + 2v_{xy} u_{xy} + v_{yy} u_{yy})</td>
</tr>
<tr>
<td>16</td>
<td>(\text{tr}(H_u^2) = u_{xx}^2 + 2u_{xy}^2 + u_{yy}^2)</td>
</tr>
</tbody>
</table>
Linear Combination of Differential Invariants

\[ F_u(u, v, a) = \sum_{j=0}^{16} a_j(t) \text{inv}_j(u, v), \]
\[ F_v(v, u, b) = \sum_{j=0}^{16} b_j(t) \text{inv}_j(v, u). \]

where \( u \) is the target image and \( v \) is an indicator function for collecting large-scale information.
Learning Coefficients by Optimal Control

\[
\min J(\{u_m\}_{m=1}^{M}, \{a_j\}_{j=0}^{16}, \{b_j\}_{j=0}^{16})
\]
\[
= \sum_{m=1}^{M} \int_{\Omega} [u_m(x, y, T) - O_m]^2 d\Omega + \sum_{j=0}^{16} \lambda_j \int_{0}^{T} a_j^2(t) dt + \sum_{j=0}^{16} \mu_j \int_{0}^{T} b_j^2(t) dt,
\]

\[
\begin{align*}
\frac{\partial u_m}{\partial t} - F_u(u_m, v_m, \{a_j\}_{j=0}^{16}) &= 0, \quad (x, y, t) \in Q, \\
&\quad u_m(x, y, t) = 0, \quad (x, y, t) \in \Gamma, \\
&\quad u_m|_{t=0} = I_m, \quad (x, y) \in \Omega,
\end{align*}
\]

\[
\begin{align*}
\frac{\partial v_m}{\partial t} - F_v(v_m, u_m, \{b_j\}_{j=0}^{16}) &= 0, \quad (x, y, t) \in Q, \\
v_m(x, y, t) &= 0, \quad (x, y, t) \in \Gamma, \\
v_m|_{t=0} &= I_m, \quad (x, y) \in \Omega.
\end{align*}
\]

where \((I_m, O_m)\) are training samples, where \(I_m\) is the input image and \(O_m\) is the expected output image, \(m = 1, \cdots, M\).
Experiments

• The same form of PDE for different problems!
  – Only differ in coefficients
• Some of them have never been tried by PDEs!

Risheng Liu, Zhouchen Lin, Wei Zhang and Zhixun Su. Learning PDEs for Image Restoration via Optimal Control. *ECCV’2010*
Denoising (Mixture of Gaussian, Poisson and Salt & Pepper)

Risheng Liu, Zhouchen Lin, Wei Zhang and Zhixun Su. Learning PDEs for Image Restoration via Optimal Control. ECCV’2010
Denoising Performance

Average PSNR

30dB

25dB

20dB

15dB

10dB

Gaussian

Mixture

P-M
ROF
TV-L1
L-PDE

Risheng Liu, Zhouchen Lin, Wei Zhang and Zhixun Su. Learning PDEs for Image Restoration via Optimal Control. *ECCV’2010*
Risheng Liu, Zhouchen Lin, Wei Zhang and Zhixun Su. Learning PDEs for Image Restoration via Optimal Control. *ECCV'2010*
Perceptual Edge Detection

Object Detection (Training)

Object Detection (Testing, Positive)

Object Detection (Testing, Negative)

Outline

• Motivation
• Learning a specified PDE system
• Learning the general PDE system
• Discussions
The spring of Neural Network is coming, but when is PDE?

- Feature learning beats handcraft features
- Can we do similar thing for PDE methods?
  - Building blocks of DNN are linear systems, while PDE considers more powerful differential systems
  - The evolution of PDE is very similar to the multiple layers NN learning process
Take Home Message

• **PDE + Learning** is a promising direction for computer vision
  – Potential connections to deep learning?!
• Learning a specific PDE for particular task
• Learning a general PDE for various tasks
• Build “deep” PDE system for challenging tasks?
References

大连理工大学数字媒体技术团队招生（计算数学、计算机科学与技术和软件工程专业博士、硕士）
欢迎各位老师同学以任何形式与我们开展交流、合作
诚邀各位CV同仁到大工、到软件学院指导、访问
Thanks for attentions

http://cggi.dlut.edu.cn/people/~rsliu/
rsliu@dlut.edu.cn