Compressive Sensing & Visual Saliency

A Ken, Ravi, Calvin, and Mike Project

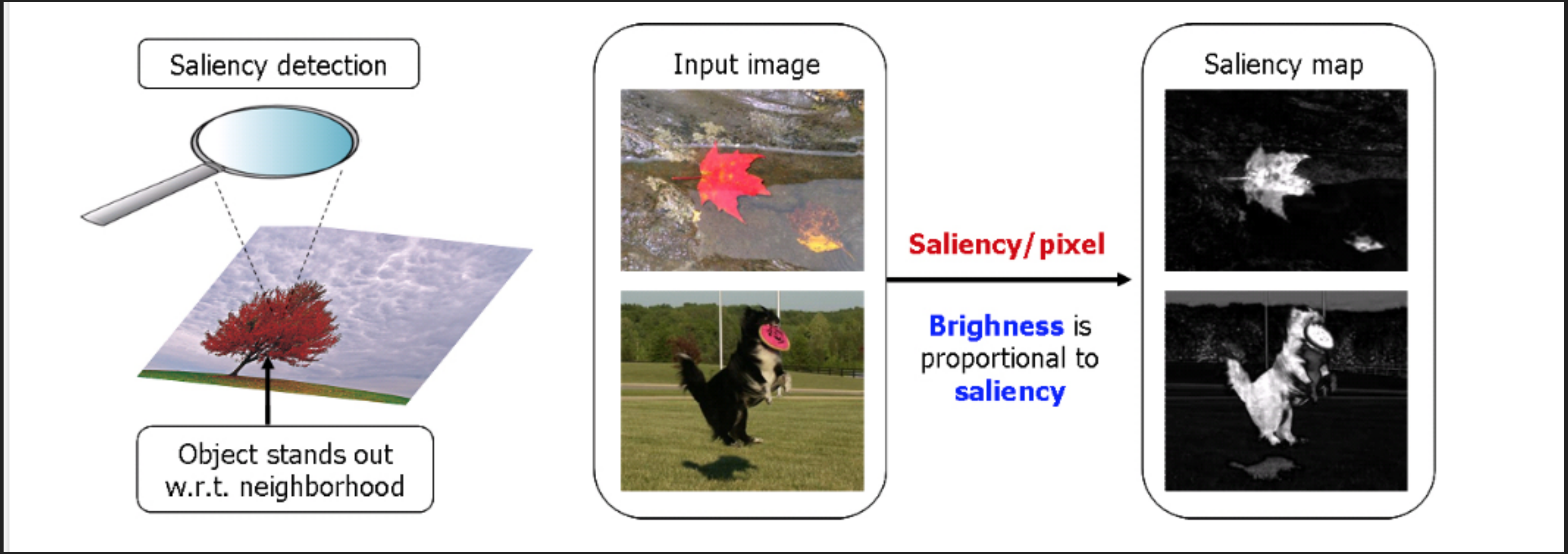
Visual Saliency Motivation

Spans many research domains (e.g. neuroscience, biology, computer vision)

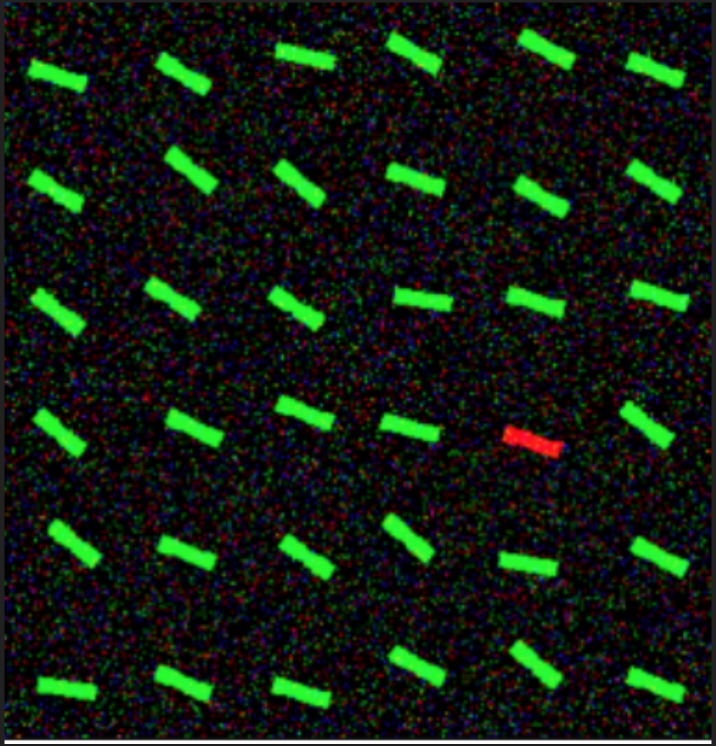
Signal Processing, Pattern Recognition, and Salient Object Detection can be considered Visual Saliency tasks

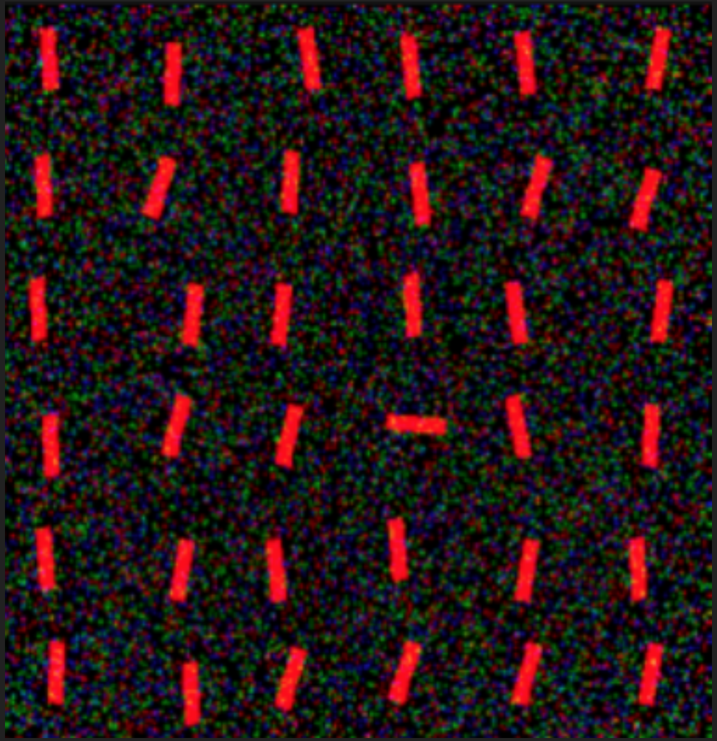
Optimizing the computation of image and video saliencies leads to dramatic performance boosts in these tasks

Visual Saliency Motivation

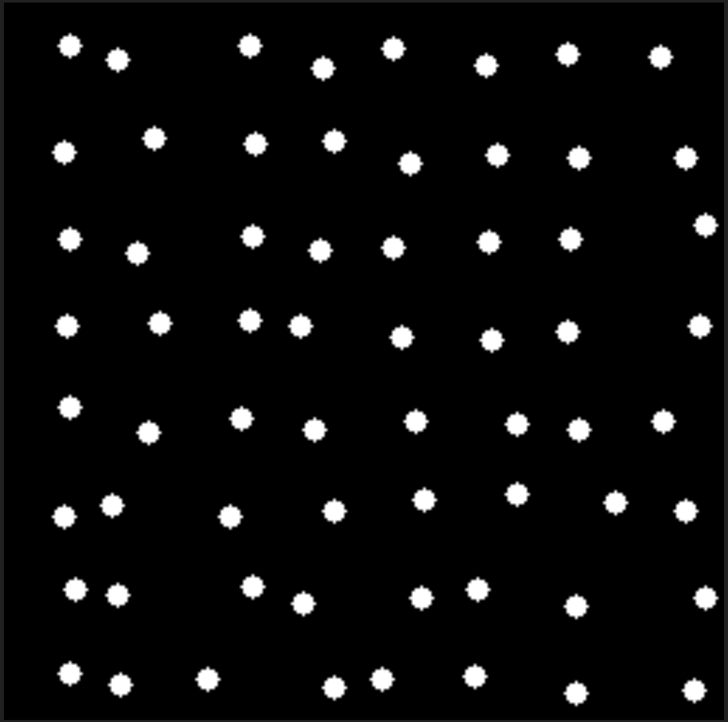


Visual Saliency Motivation





Visual Saliency Motivation



Overview of Project

Two Models for Consideration: Hou and Low-Rank Sparse-Matrix Decomposition (LSMD)

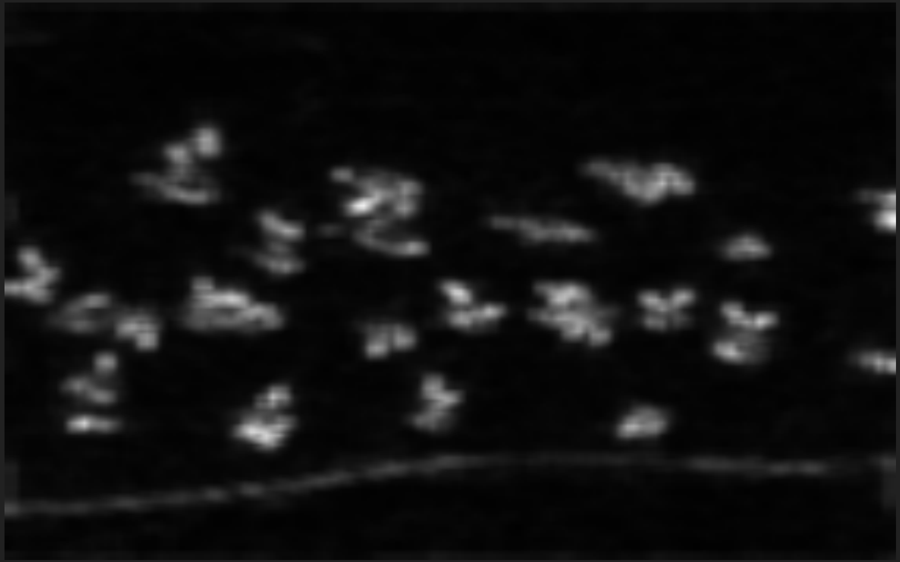
Implement both for image and video, and evaluate the accuracy of results

Data Sets

We tested our two models using a collection of short videos that contained a salient object and a static background to see the differences in their resulting saliency maps.

Hou Saliency

$$E\{\mathcal{A}(f)\} \propto 1/f.$$



Hou Saliency

2D Fast Fourier Transform

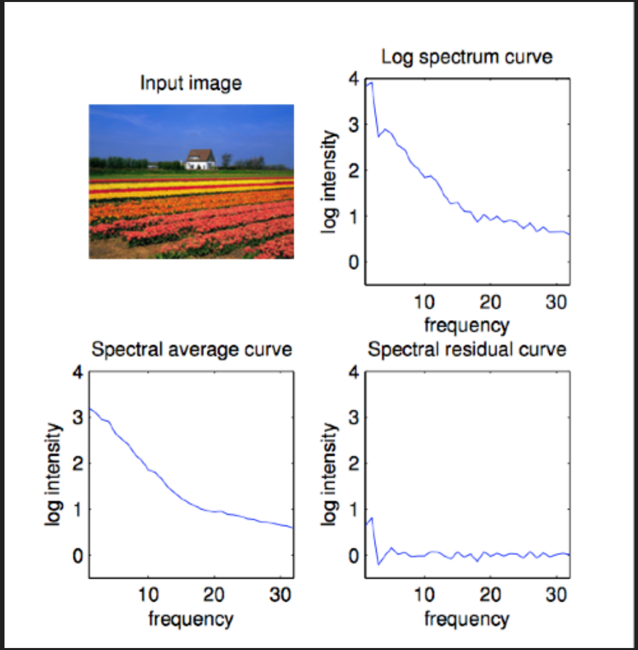
Convert the resulting FFT vectors to intensity and frequency coordinates

Take the log of the magnitude (Log Spectrum)

Average this result using 3*3 blur kernel (Spectral Average)

Remove the average from the log spectrum (Spectral Residual)

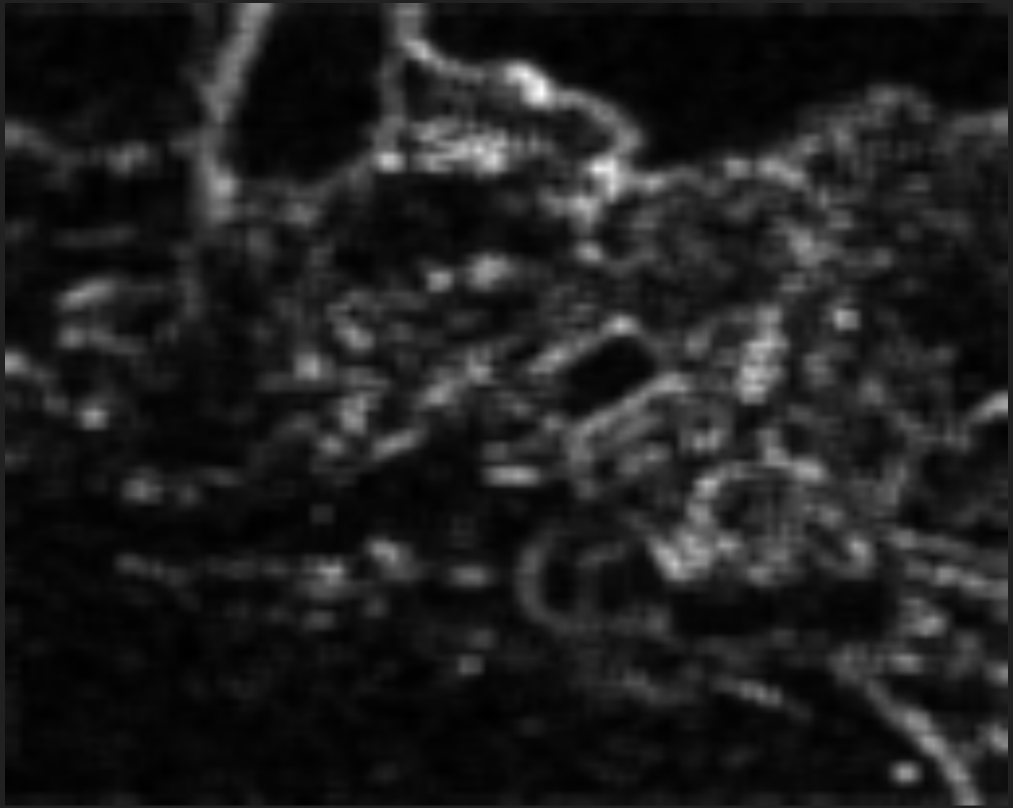
Return the residual intensity as the saliency map



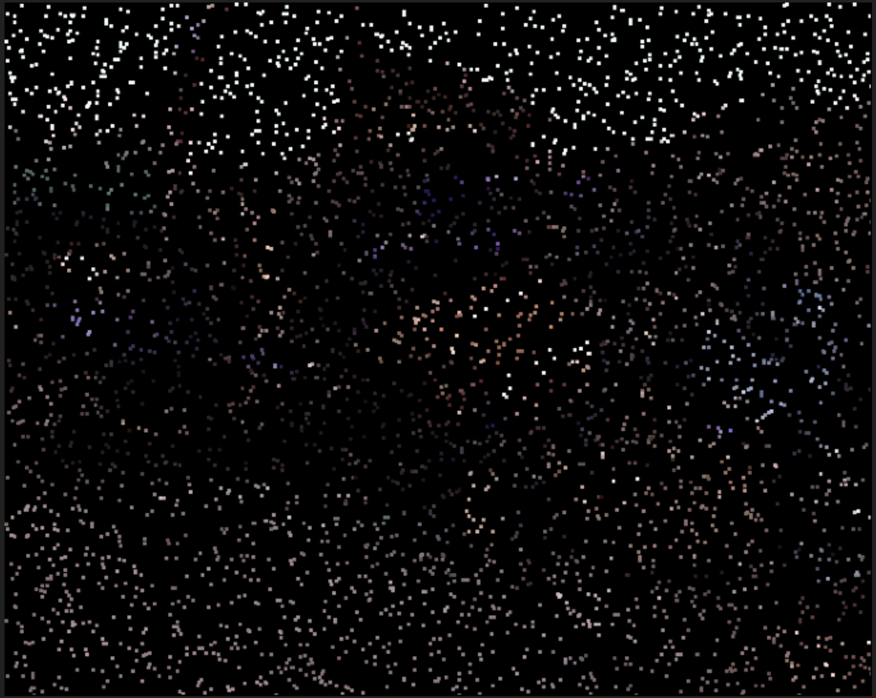
Sample Video

Hou Saliency Model

Hou captures the borders of objects well, because they are natural frequency transition points

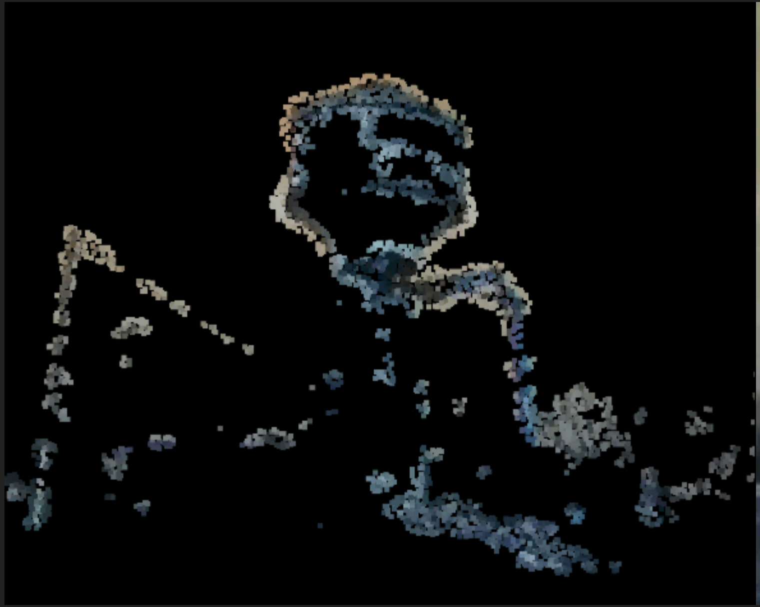


M log(N) Sampling of Hou v. M log(N) Sampling



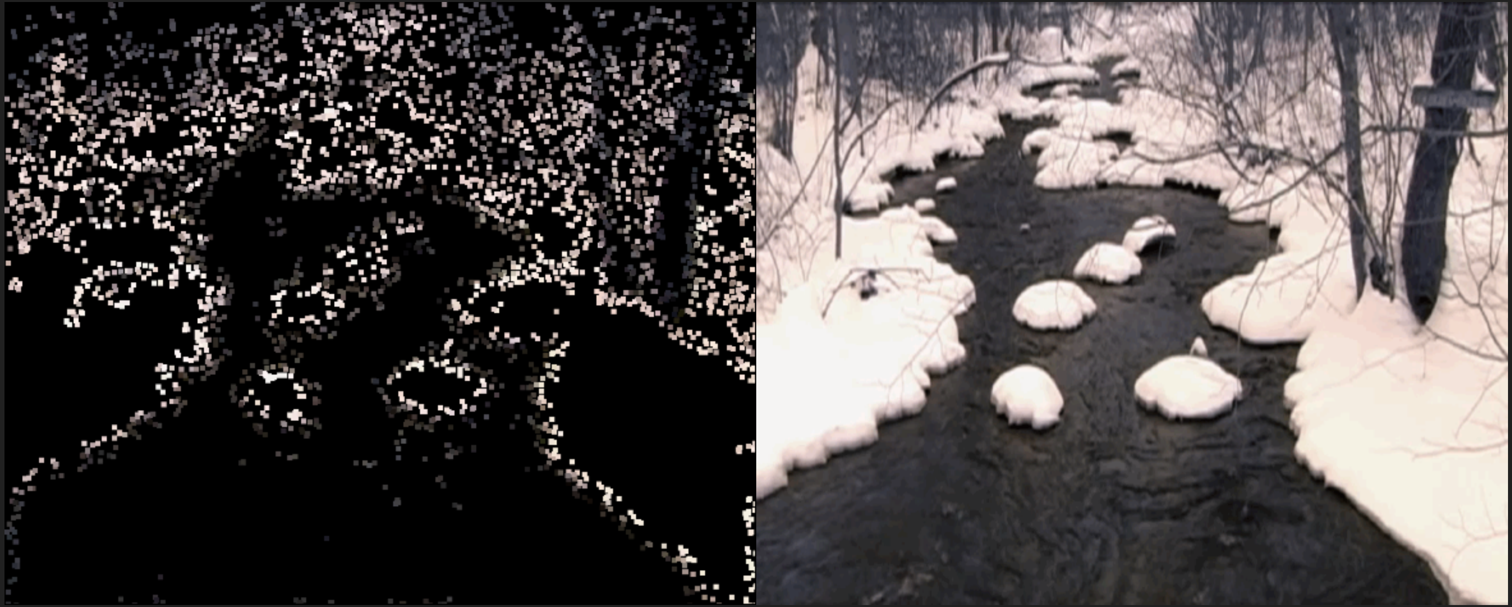
Limitations

Hou has no spatial contextualization capability




Limitations

Frequencies alone do not always represent our focus



LSMD Overview

Image Abstraction:

- Feature Extraction (Color, Edge, Texture)
- Over Segmentation via SLIC Super Pixel model
- Single Feature Matrix Generated

Tree Construction

- Image Prior Compilation (Location, Color, Semantic)
- Structured Index Tree

Low Rank Structured Sparse Matrix Decomposition

Output Saliency

LSMD Background

Two models: Bottom up and Top-down

- Bottom Up:
  - Computed by finding features in the image, fast run time (Hou)
  - Examples: color, texture, and location
- Top-down:
  - Use of priors to modify feature relevance, slow run time
  - Examples: Background priors

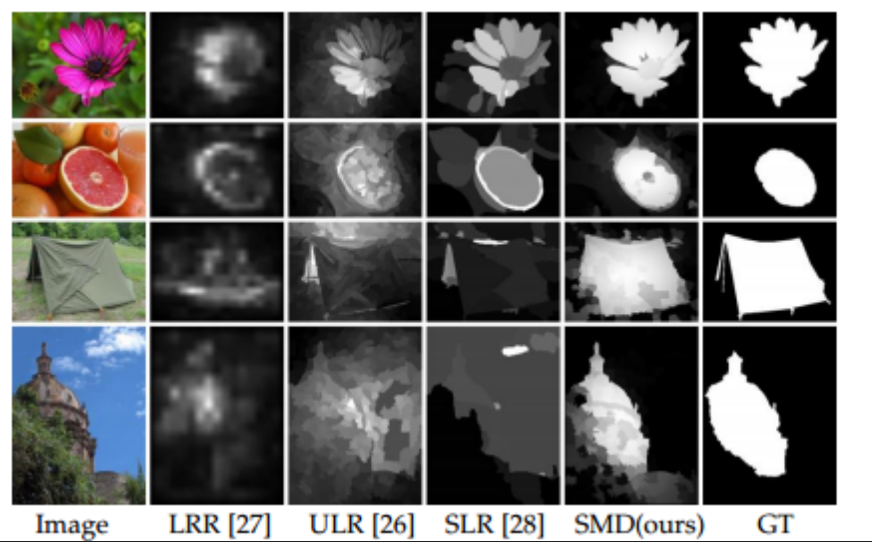
Low-Rank Matrix Recovery (LR) based saliency detection model

- Combines a highly redundant information part (visually consistent background) and a spare salient part (foreground object)

LR-model Limitations

Spatial relations (such as spatial contiguity and pattern consistency) between foreground pixels and patches are ignored

- Causes scatteredness in saliency maps
- Unable to distinguish objects from backgrounds that share similar appearances
- ...ect, causing incompleteness of the detected ...he background more difficult



Low-rank representation (LRR) - top-down priors weighted and combined with multiple features to estimate saliency collaboratively

Unified LR (ULR) - feature transformation that combines traditional low-level features with high-level prior knowledge

SLR - Segmentation priors derived from image background and boundary cues to assist the low-rank matrix recovery

Solution to these Limitations

- Enhance LR with Structured Matrix Decomposition (SMD)
- Tree-structured sparsity inducing norm
  - Constrains the sparse matrix to consider both spatial connectivity and similar features in image patches when performing matrix decomposition
  - l-∞ norm ensures within-object patches share consistent saliency values
- Integrate Laplacian regularization
  - Reduces coherence between low-rank and structured-sparse matrices
  - Takes into account the geometrical structure of the image
  - Enlarges distance between the subspaces induced by the non-salient background and the salient foreground to separate the salient object from the background
    - Encourages patches in the same semantic region to share similar representation

Structured Matrix Decomposition Model

$$\min_{\mathbf{L},\mathbf{S}} \Psi(\mathbf{L}) + \alpha\Omega(\mathbf{S}) + \beta\Theta(\mathbf{L},\mathbf{S}) \quad \text{s.t.} \quad \mathbf{F} = \mathbf{L} + \mathbf{S}$$

Ψ(·) is a low-rank constraint to allow identification of the feature subspace of the redundant background patches

Ω(·) is a structured sparsity regularization that captures the spatial and feature relations of patches in S

Θ(·, ·) is an interactive regularization term to enlarge the distance between the subspaces drawn from L and S

α, β are positive tradeoff parameters

Feature matrix F is decomposed into a low-rank matrix part L (redundant information/non-salient background) and a structured salient part S (salient foreground)

Structured Matrix Decomposition Model

$$\min_{\mathbf{L},\mathbf{S}} \Psi(\mathbf{L}) + \alpha\Omega(\mathbf{S}) + \beta\Theta(\mathbf{L},\mathbf{S}) \quad \text{s.t.} \quad \mathbf{F} = \mathbf{L} + \mathbf{S}$$

Low-rank regularization for background

$$\Psi(\mathbf{L}) = rank(\mathbf{L}) = \|\mathbf{L}\|_* + \varepsilon \ ,$$

Structured Sparsity regularization for salient object

$$\Omega(\mathbf{S}) = \sum_{i=1}^{d} \sum_{j=1}^{n_i} v_j^i \|\mathbf{S}_{G_j^i}\|_p \ ,$$

Laplacian Regularization

$$\Theta(\mathbf{L},\mathbf{S}) = \frac{1}{2} \sum_{i,j=1}^{N} \|\mathbf{s}_i - \mathbf{s}_j\|_2^2 \, w_{i,j} = \mathrm{Tr}(\mathbf{S}\mathbf{M}_{\mathbf{F}}\mathbf{S}^T) \ ,$$

Structured Matrix Decomposition Model

$$\min_{\mathbf{L},\mathbf{S}} \Psi(\mathbf{L}) + \alpha\Omega(\mathbf{S}) + \beta\Theta(\mathbf{L},\mathbf{S}) \quad \text{s.t.} \quad \mathbf{F} = \mathbf{L} + \mathbf{S}$$

ADM to minimize the Augmented Lagrangian function

$$\begin{aligned} \min_{\mathbf{L},\mathbf{S}} \quad & \|\mathbf{L}\|_* + \alpha \sum_{i=1}^{d} \sum_{j=1}^{n_i} v_j^i \|\mathbf{S}_{G_j^i}\|_p + \beta \mathrm{Tr}(\mathbf{H}\mathbf{M}_{\mathbf{F}}\mathbf{H}^T) \\ \text{s.t.} \quad & \mathbf{F} = \mathbf{L} + \mathbf{S}, \quad \mathbf{S} = \mathbf{H} \,. \end{aligned}$$

$$\begin{aligned} \mathcal{L}(\mathbf{L},\mathbf{S},\mathbf{H},\mathbf{Y}_1,\mathbf{Y}_2,\mu) = {} & \|\mathbf{L}\|_* \\ & + \alpha \sum_{i=1}^{d} \sum_{j=1}^{n_i} v_j^i \|\mathbf{S}_{G_j^i}\|_p + \beta \mathrm{Tr}(\mathbf{H}\mathbf{M}_{\mathbf{F}}\mathbf{H}^T) \\ & + \mathrm{Tr}(\mathbf{Y}_1^T(\mathbf{F}-\mathbf{L}-\mathbf{S})) + \mathrm{Tr}(\mathbf{Y}_2^T(\mathbf{S}-\mathbf{H})) \\ & + \frac{\mu}{2}\left(\|\mathbf{F}-\mathbf{L}-\mathbf{S}\|_F^2 + \|\mathbf{S}-\mathbf{H}\|_F^2\right) , \end{aligned}$$

Alternating Direction Method (ADM)

Algorithm 1 ADM-SMD.

Input: Feature matrix $\mathbf{F}$, parameters α, β, index tree $T = \{G_j^i\}$ and tree node weight v_j^i (default as 1).

Output: $\mathbf{L}$ and $\mathbf{S}$.

Initialize $\mathbf{L}^0{=}\mathbf{0}$, $\mathbf{S}^0{=}\mathbf{0}$, $\mathbf{H}^0{=}\mathbf{0}$, $\mathbf{Y}_1^0{=}\mathbf{0}$, $\mathbf{Y}_2^0{=}\mathbf{0}$, $\mu^0{=}0.1$, $\mu_{\max} = 10^{10}$, $\rho{=}\ 1.1$, and $k = 0$.

While not converged **do**

$\quad \mathbf{L}^{k+1} = \arg\min_{\mathbf{L}} \mathcal{L}(\mathbf{L}, \mathbf{S}^k, \mathbf{H}^k, \mathbf{Y}_1^k, \mathbf{Y}_2^k, \mu^k)$

$\quad \mathbf{H}^{k+1} = \arg\min_{\mathbf{H}} \mathcal{L}(\mathbf{L}^{k+1}, \mathbf{S}^k, \mathbf{H}, \mathbf{Y}_1^k, \mathbf{Y}_2^k, \mu^k)$

$\quad \mathbf{S}^{k+1} = \arg\min_{\mathbf{S}} \mathcal{L}(\mathbf{L}^{k+1}, \mathbf{S}, \mathbf{H}^{k+1}, \mathbf{Y}_1^k, \mathbf{Y}_2^k, \mu^k)$

$\quad \mathbf{Y}_1^{k+1} = \mathbf{Y}_1^k + \mu^k(\mathbf{F} - \mathbf{L}^{k+1} - \mathbf{S}^{k+1})$

$\quad \mathbf{Y}_2^{k+1} = \mathbf{Y}_2^k + \mu^k(\mathbf{S}^{k+1} - \mathbf{H}^{k+1})$

$\quad \mu^{k+1} = \min(\rho\mu^k, \mu_{\max})$

$\quad k = k + 1$

End While

Return $\mathbf{L}^k$ and $\mathbf{S}^k$.

This algorithm optimizes SMD by breaking the problem down into smaller pieces, making it easier to handle.

Lagrangian function to perform a series of minimizations on L, S, and H so we can retrieve the most optimal L and S after convergence.

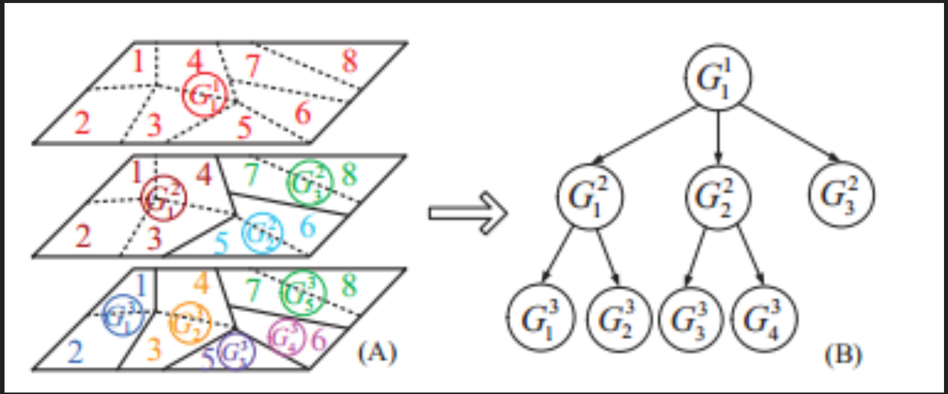
Y1 and Y2 are Lagrange multipliers and μ controls the penalty for violating linear constraints

Tree Algorithm

Algorithm 2 Solving the tree-structured sparsity.

Input: The index tree T with nodes G_j^i $(i = 1, 2, ..., d; j = 1, 2, ..., n_i)$, weight $v_j^i \geq 0$ (default as 1), the matrix $\mathbf{X_S}$, parameters α, and set $\lambda = \alpha/(2\mu^k)$.

1: Set $\mathbf{S} = \mathbf{X_S}$
2: **For** $i = d$ to 1 **do**
3: **For** $j = 1$ to n_i **do**
4: $$\mathbf{S}_{G_j^i}^{k+1} = \begin{cases} \frac{\|\mathbf{s}_{G_j^i}\|_1 - \lambda v_j^i}{\|\mathbf{s}_{G_j^i}\|_1} \mathbf{S}_{G_j^i}, & \text{if } \|\mathbf{S}_{G_j^i}\|_1 > \lambda v_j^i \\ \mathbf{0}, & \text{otherwise} \end{cases}$$
5: **End For**
6: **End For**
7: **Return** $\mathbf{S}^{k+1}$

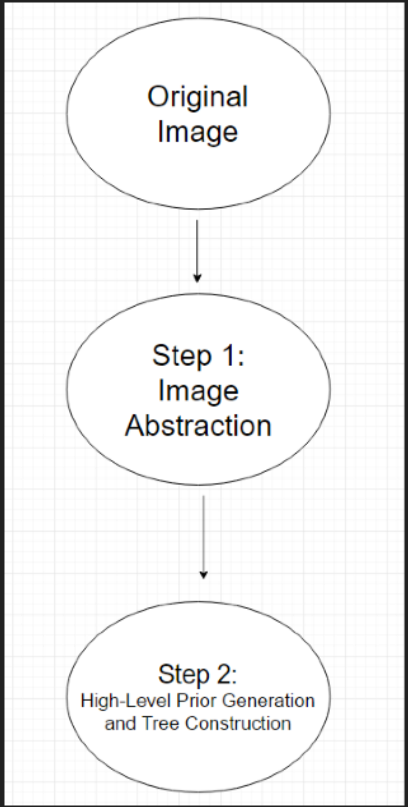


(A) shows the hierarchical segmentation of an image where the digits are the indexes of the patches

(B) shows an index tree constructed over the indices of image patches

Patches within the same group share a similar or identical representation, and the maximum saliency value of patches within the Group G is used to decide whether the group is salient or not

LSMD Process



Step 1: Image Abstraction

Extract low-level features to construct a 53-dimension feature representation

Then perform the simple linear iterative clustering (SLIC) algorithm

Over-segments the image into N superpixels

We get a series of patches P = {P1, P2, …, PN} where each Pi is represented by a feature vector. Together, they form the feature matrix F

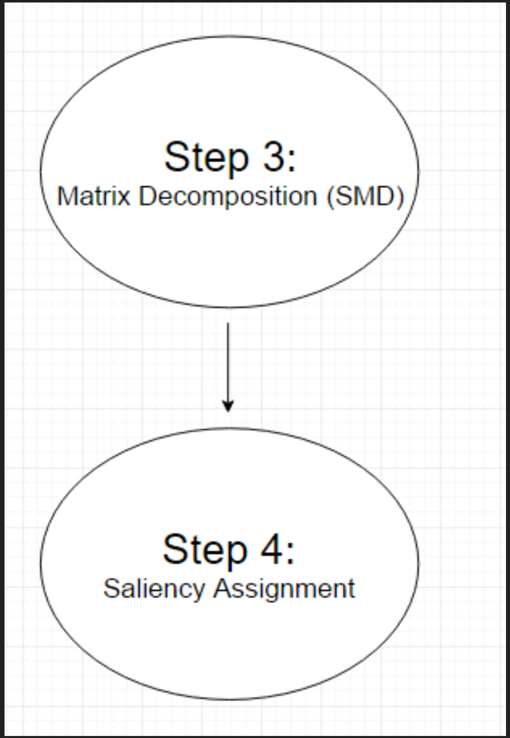
Step 2: Tree Construction

An index tree is constructed to encode structural information via hierarchical segmentation

First compute the affinity of every adjacent patch pair then merge spatially neighboring patches according to their affinity.

This produces a sequence of granularity-increasing segmentations

LSMD Process cont.



Step 3: Matrix Decomposition

Using the feature matrix F and index tree T, we apply the SMD model to decompose F into the structured components L and S.

Step 4: Saliency Assignment

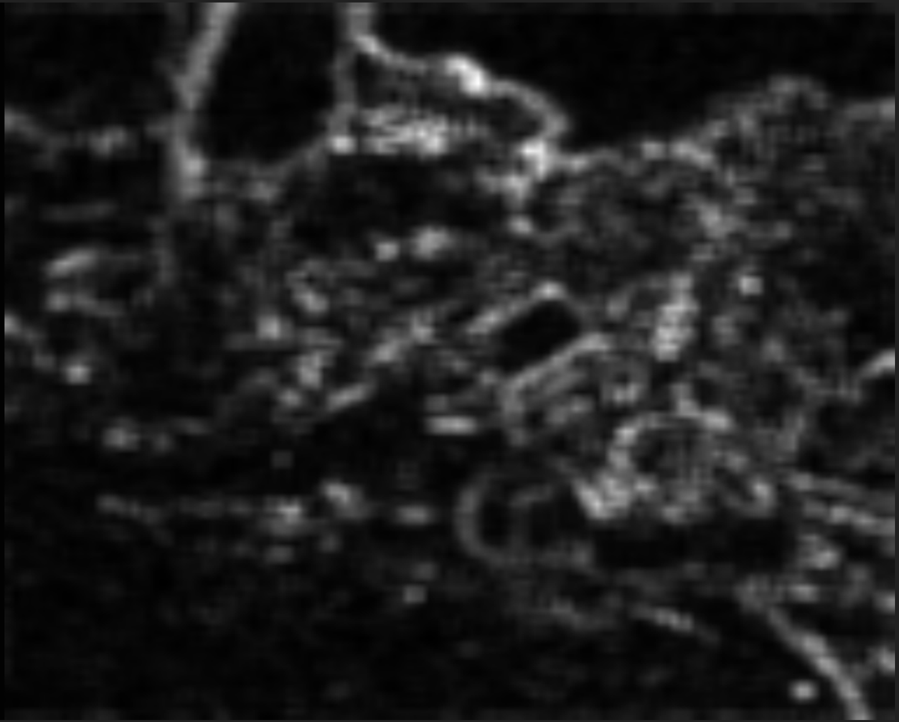
Transfer results from the feature domain to the spatial domain fo $\mathrm{Sal}(P_i) = \|\mathbf{s}_i\|_1$ estimation.

Based on the structured matrix S, we perform the function on each patch in P, where s_i represents the i-th column of S

Finally, after merging all the patches together, we have the final saliency map of the input image

LSMD Sample

LSMD Sample v. Hou Sample



Demo!

Thank you!

Citations

[1] Peng, H., Li, B., Ji, R., Hu, W., Xiong, W., & Lang, C. (2013, July). Salient Object Detection via Low-Rank and Structured Sparse Matrix Decomposition. In *AAAI*.

[2] Peng, H., Li, B., Ling, H., Hu, W., Xiong, W., & Maybank, S. J. (2015). Salient object detection via structured matrix decomposition.

[3] Dao, M., Suo, Y., Chin, S. P., & Tran, T. D. (2014, November). Structured sparse representation with low-rank interference. In *2014 48th Asilomar Conference on Signals, Systems and Computers* (pp. 106-110). IEEE.

[4] X. Hou and L. Zhang (2007). Saliency Detection: A Spectral Residual Approach. IEEE Transactions on Computer Vision and Pattern Recognition (CVPR), p.1-8. Doi: 10.1109/CVPR.2007.383267