A Fast Method for Inferring
High-Quality Simply-Connected Superpixels

Oren Freifeld, Yixin Li, and John Fisher
Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Laboratory
Sensing, Learning and Inference Group

ICIP 2015
September 30, 2015
Introduction
  1. Superpixels
  2. Simple connectivity
  3. Shape flexibility
  4. Problem statement

The proposed approach:
  - Modifying a method by Chang, Wei and Fisher
  - Improved modeling
  - Faster Inference

Results and comparisons
Superpixels$^1$

Coherent image patches; boundaries well aligned with image edges

Results of the proposed method for different values of $K$

$K$: #superpixels
$N$: #pixels
$K \ll N$

$^1$[Ren & Malik, ICCV ’03]
**Superpixels**

Coherent image patches; boundaries well aligned with image edges (but not too irregular)

Results of the proposed method for different values of $K$

$K$: #superpixels

$N$: #pixels

$K \ll N$

---

$^1$[Ren & Malik, ICCV ’03]
Superpixels: a Compact Intermediate Representation

A useful pre-processing for many tasks

Image

Borders

Average colors
Introduction

Superpixels: a Compact Intermediate Representation

A useful pre-processing for many tasks

Image

Borders

Average colors
Simple Connectivity ("Connected Without Holes")

Usually a desired property for superpixels

- Each of the small white regions is simply connected
- Their union is disconnected
- Blue region: connected but not simply connected

Superpixel approaches to simple connectivity

<table>
<thead>
<tr>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ignore it &amp; post-processing heuristics (results not directly related to the model)</td>
</tr>
<tr>
<td>Respect it &amp; very restricted (e.g., convex polygons)</td>
</tr>
<tr>
<td>Respect only (non-simple) connectivity (not always an issue)</td>
</tr>
<tr>
<td>Respect it &amp; more free-form; typically:</td>
</tr>
</tbody>
</table>

[SLIC], [Duan et al.], [Veksler et al.], [SEEDS], [Turbopixels], [Chang et al.]
Simple Connectivity ("Connected Without Holes")

*Usually a desired property for superpixels*

- Each of the small white regions is simply connected
- Their union is disconnected
- Blue region: connected but not simply connected

### Superpixel approaches to simple connectivity

<table>
<thead>
<tr>
<th>Ignore it &amp; post-processing heuristics</th>
<th>(results not directly related to the model)</th>
<th>[SLIC]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Respect it &amp; very restricted (e.g. convex polygons)</td>
<td>[Duan et al.]</td>
<td></td>
</tr>
<tr>
<td>Respect only (non-simple) connectivity (not always an issue)</td>
<td>[Veksler et al.], [SEEDS]</td>
<td></td>
</tr>
<tr>
<td>Respect it &amp; more free-form; typically:</td>
<td></td>
<td>[Turbopixels], [Chang et al.]</td>
</tr>
<tr>
<td>slow</td>
<td></td>
<td></td>
</tr>
<tr>
<td>need a large $K$ for good results</td>
<td></td>
<td></td>
</tr>
<tr>
<td>favor isotropic superpixels</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Flexibility in the Overall Shape of a Superpixel

- Many methods: a strong bias to almost-isotropic\(^2\) superpixels
- Poorly suited for elongated structures

\(^2\)In effect, circle-like

But don’t want “too much” flexibility

Adapt \(K\) locally? Slow, \(K\) may be too large
Flexibility in the Overall Shape of a Superpixel

- Many methods: a strong bias to almost-isotropic\(^2\) superpixels
- Poorly suited for elongated structures

\[^2\text{In effect, circle-like}\]

SLIC

SLIC

Proposed method

Proposed method

But don’t want “too much” flexibility

Adapt $K$ locally? Slow, $K$ may be too large
Flexibility in the Overall Shape of a Superpixel

Many methods: a strong bias to almost-isotropic\(^2\) superpixels
Poorly suited for elongated structures

But don’t want “too much” flexibility

Adapt \(K\) locally? Slow, \(K\) may be too large

\(^2\)In effect, circle-like
Flexibility in the Overall Shape of a Superpixel

- Many methods: a strong bias to almost-isotropic\(^2\) superpixels
- Poorly suited for elongated structures

\[\text{SLIC} \quad \text{SLIC} \quad \text{Proposed method} \quad \text{Proposed method}\]

- But don’t want “too much” flexibility
- Adapt \(K\) locally? Slow, \(K\) may be too large

\(^2\)In effect, circle-like
Want a model that
- respects simple connectivity
- allows for some (inexpensive) flexibility in the shapes of the superpixels
# A Model That Respects Simple Connectivity [Chang et al.]

## Connectivity-Constrained Gaussian Mixture Model (CC-GMM)

Observed pixel \( i \): \( x_i = (l_i, c_i) = (\text{location, color}) \)

Latent label: \( z_i = j \iff \text{pixel } i \in \text{superpixel } j \)

GMM: \( x_i \sim \sum_{j=1}^{K} w_j N(x_j; \mu_j, \Sigma_j) \) \( \Sigma_j \) is block diagonal

\[
\Pr(z_1, \ldots, z_N) = \begin{cases} 
0 & \text{if at least one of the superpixels is not simply connected} \\
\text{const} > 0 & \text{otherwise}
\end{cases}
\]

- Similar to \([\text{SLIC}^3, \text{Blobworld}]\) except the \( z_i \)'s are dependent
- Good results but slow inference
  - Sequential label updates (simple connectivity)
  - Expensive splits and merges (shape flexibility)

\(^3\)If all covariances are identical to each other and isotropic
Introduction

A Model That Respects Simple Connectivity [Chang et al.]

Connectivity-Constrained Gaussian Mixture Model (CC-GMM)

- Observed pixel $i$: $x_i = (l_i, c_i) = \text{(location, color)}$
- Latent label: $z_i = j$ $\iff$ pixel $i \in \text{superpixel } j$
- GMM: $x_i \sim \sum_{j=1}^{K} w_j N(x_j; \mu_j, \Sigma_j)$  \quad $\Sigma_j$ is block diagonal

$$\text{Pr}(z_1, \ldots, z_N) = \begin{cases} 0 & \text{if at least one of the superpixels is not simply connected} \\ \text{const} > 0 & \text{otherwise} \end{cases}$$

- Similar to [SLIC$^3$, Blobworld] except the $z_i$’s are dependent
- Good results but slow inference
  - Sequential label updates (simple connectivity)
  - Expensive splits and merges (shape flexibility)

$^3$If all covariances are identical to each other and isotropic
Introduction

A Model That Respects Simple Connectivity [Chang et al.]

Connectivity-Constrained Gaussian Mixture Model (CC-GMM)

Observed pixel $i$: $x_i = (l_i, c_i) =$ (location, color)

Latent label: $z_i = j \iff$ pixel $i \in$ superpixel $j$

GMM: $x_i \sim \sum_{j=1}^{K} w_j N(x_j; \mu_j, \Sigma_j)$ \hspace{1cm} $\Sigma_j$ is block diagonal

Pr($z_1, \ldots, z_N$) = \begin{cases} 0 & \text{if all least one of the superpixels is not simply connected} \\ \text{const} & \text{otherwise} \end{cases}

- Similar to [SLIC$^3$, Blobworld] except the $z_i$’s are dependent
- Good results but slow inference
  - Sequential label updates (simple connectivity)
  - Expensive splits and merges (shape flexibility)

\footnote{If all covariances are identical to each other and isotropic}
A Model That Respects Simple Connectivity [Chang et al.]

Connectivity-Constrained Gaussian Mixture Model (CC-GMM)

Observed pixel $i$: $x_i = (l_i, c_i) = \text{(location, color)}$

Latent label: $z_i = j \iff \text{pixel } i \in \text{superpixel } j$

GMM: $x_i \sim \sum_{j=1}^{K} w_j \mathcal{N}(x; \mu_j, \Sigma_j)$ \hspace{1cm} $\Sigma_j$ is block diagonal

$$w_j = \text{Pr}(z_i = j)$$

$$\text{Pr}(z_1, \ldots, z_N) = \begin{cases} 0 & \text{if at least one of the superpixels is not simply connected} \\ \text{const} > 0 & \text{otherwise} \end{cases}$$

- Similar to [SLIC$^3$, Blobworld] except the $z_i$'s are dependent
- Good results but slow inference
  - Sequential label updates (simple connectivity)
  - Expensive splits and merges (shape flexibility)

$^3$If all covariances are identical to each other and isotropic
A Model That Respects Simple Connectivity [Chang et al.]

Connectivity-Constrained Gaussian Mixture Model (CC-GMM)

Observed pixel $i$: $x_i = (l_i, c_i) = (\text{location}, \text{color})$

Latent label: $z_i = j \iff \text{pixel } i \in \text{superpixel } j$

GMM: $x_i \sim \sum_{j=1}^{K} w_j N(x; \mu_j, \Sigma_j)$ where $\Sigma_j$ is block diagonal

Segmentation:

$$\Pr(z_1, \ldots, z_N) = \begin{cases} 0 & \text{if at least one of the superpixels is not simply connected} \\ \text{const} > 0 & \text{otherwise} \end{cases}$$

- Similar to [SLIC$^3$, Blobworld] except the $z_i$'s are dependent
- Good results but slow inference
  - Sequential label updates (simple connectivity)
  - Expensive splits and merges (shape flexibility)

$^3$If all covariances are identical to each other and isotropic
Introduction

A Model That Respects Simple Connectivity [Chang et al.]

Connectivity-Constrained Gaussian Mixture Model (CC-GMM)

Observed pixel \( i \): \( x_i = (l_i, c_i) = \) (location, color)

Latent label: \( z_i = j \iff \) pixel \( i \in \) superpixel \( j \)

GMM: \( x_i \sim \sum_{j=1}^{K} w_j \mathcal{N}(x_j; \mu_j, \Sigma_j) \quad \Sigma_j \) is block diagonal

Pr(\( z_1, \ldots, z_N \)) = \begin{cases} 
0 & \text{if at least one of the superpixels is not simply connected} \\
\text{const} > 0 & \text{otherwise}
\end{cases}

Similar to [SLIC$^3$, Blobworld] except the \( z_i \)'s are dependent

Good results but slow inference

Sequential label updates (simple connectivity)

Expensive splits and merges (shape flexibility)

$^3$If all covariances are identical to each other and isotropic
A Model That Respects Simple Connectivity [Chang et al.]

Connectivity-Constrained Gaussian Mixture Model (CC-GMM)

- **Observed pixel** $i$: $x_i = (l_i, c_i) = (\text{location}, \text{color})$
- **Latent label**: $z_i = j \iff \text{pixel } i \in \text{superpixel } j$
- **GMM**: $x_i \sim \sum_{j=1}^{K} w_j N(x_j; \mu_j, \Sigma_j)$
  - $\Sigma_j$ is block diagonal
  - $w_j = \Pr(z_i = j)$

Pr($z_1, \ldots, z_N$) = \begin{cases} 0 & \text{if at least one of the superpixels is not simply connected} \\ \text{const} > 0 & \text{otherwise} \end{cases}

- **Similar to** [SLIC$^3$, Blobworld] except the $z_i$’s are dependent
- **Good results but slow inference**
  - Sequential label updates (simple connectivity)
  - Expensive splits and merges (shape flexibility)

$^3$If all covariances are identical to each other and isotropic
Problem Statement

1. Want a model that
   1. respects simple connectivity
   2. allows for some (inexpensive) flexibility in the shapes of the superpixels

2. Want parallel/fast inference despite the topological constraint
Problem Statement

1. Want a model that
   1. respects simple connectivity
   2. allows for some (inexpensive) flexibility in the shapes of the superpixels

2. Want parallel/fast inference despite the topological constraint
The Proposed Approach

Follow [Chang et al.]:
- Model: GMM with a simple-connectivity constraint
- Inference: Alternating between label updates and model-parameter updates

 Modifications:
- Improved modeling of the spatial covariances
  ⇒ better shape flexibility
- Label updates: parallelize over many pixels (despite the constraint)
  ⇒ much faster inference
- Model-parameter updates: parallelize over superpixels
  ⇒ computing time decreases when $K$ increases

$K$: number of superpixels
The Proposed Approach

Follow [Chang et al.]:

- **Model**: GMM with a simple-connectivity constraint
- **Inference**: Alternating between label updates and model-parameter updates

**Modifications:**

1. **Improved modeling of the spatial covariances**  
   ⇒ better shape flexibility
2. **Label updates**: parallelize over many pixels (despite the constraint)  
   ⇒ much faster inference
3. **Model-parameter updates**: parallelize over superpixels  
   ⇒ computing time decreases when $K$ increases

---

$K$: number of superpixels
The Proposed Approach

Model

A Bayesian Take on Spatial Covariances

- Unlike [SLIC] or [Chang et al.] we treat the spatial covariances as *latent* and let them *vary across the superpixels* (similar to [Blobworld])
- Unlike [Blobworld], the proposed approach is Bayesian, using an Inverse-Wishart prior
  - Prior: centered on an isotropic covariance
  - Posterior: centered on an anisotropic covariance
- Better flexibility
- Increased robustness to using the “wrong” amount of boundary regularity
  - Boundaries are usually nicer than, e.g., in [SLIC] or [Chang et al.]
A Bayesian Take on Spatial Covariances

Unlike [SLIC] or [Chang et al.] we treat the spatial covariances as *latent* and let them *vary across the superpixels* (similar to [Blobworld])

Unlike [Blobworld], the proposed approach is Bayesian, using an Inverse-Wishart prior
- Prior: centered on an isotropic covariance
- Posterior: centered on an anisotropic covariance

Better flexibility

Increased robustness to using the “wrong” amount of boundary regularity
⇒ Boundaries are usually nicer than, e.g., in [SLIC] or [Chang et al.]
CC-GMM Inference [Chang et al.]

Q: How to do inference that respects the topological constraint?

A: Restrict single-label updates to simple points\(^4\)

\(^4\)A concept from Digital Topology [Bertrand]
Q: How to do inference that respects the topological constraint?
A: Restrict single-label updates to \textit{simple points}\textsuperscript{4}

\textsuperscript{4}A concept from Digital Topology [Bertrand]
Simple Point

- Consider the binary case: \( K = 2 \)
- Suppose all superpixels are simply connected
- Let say we want to change the label of some pixel
- Will it preserve simple connectivity? (if so, the pixel is called a simple point)
- Can be shown: answer is fully defined by the neighbors in a \( 3 \times 3 \) block
- Can generalize to \( K > 2 \)
Consider the binary case: $K = 2$

Suppose all superpixels are simply connected

- Let say we want to change the label of some pixel
- Will it preserve simple connectivity? (if so, the pixel is called a simple point)
- Can be shown: answer is fully defined by the neighbors in a $3 \times 3$ block
- Can generalize to $K > 2$
Simple Point

- Consider the binary case: $K = 2$
- Suppose all superpixels are simply connected
- Let say we want to change the label of some pixel
  - Will it preserve simple connectivity? (if so, the pixel is called a simple point)
  - Can be shown: answer is fully defined by the neighbors in a $3 \times 3$ block
  - Can generalize to $K > 2$
Simple Point

- Consider the binary case: $K = 2$
- Suppose all superpixels are simply connected
- Let say we want to change the label of some pixel
- Will it preserve simple connectivity? (if so, the pixel is called a simple point)
- Can be shown: answer is fully defined by the neighbors in a $3 \times 3$ block
- Can generalize to $K > 2$
Simple Point

- Consider the binary case: $K = 2$
- Suppose all superpixels are simply connected
- Let say we want to change the label of some pixel
- Will it preserve simple connectivity? (if so, the pixel is called a simple point)
- Can be shown: answer is fully defined by the neighbors in a $3 \times 3$ block
- Can generalize to $K > 2$
The Proposed Approach

Inference

Simple Point

- Consider the binary case: $K = 2$
- Suppose all superpixels are simply connected
- Let say we want to change the label of some pixel
- Will it preserve simple connectivity? (if so, the pixel is called a simple point)
- Can be shown: answer is fully defined by the neighbors in a $3 \times 3$ block
- Can generalize to $K > 2$
But What About Doing it in Parallel?

Problem:
- Simultaneous label updates of simple points can break simple connectivity
- Led to sequential single updates in [Chang et al.]

The proposed solution exploits a simple observation:
- Parallelization over a set of simple points is OK provided no two of which are in the same 3×3 block
- We parallelize updates over \( \sqrt{N}/9 \) pixels (subject to simple-points tests)
But What About Doing it in Parallel?

Problem:

- Simultaneous label updates of simple points can break simple connectivity
- Led to sequential single updates in [Chang et al.]

The proposed solution exploits a simple observation:

- Parallelization over a set of simple points is OK provided no two of which are in the same $3 \times 3$ block
- We parallelize updates over $N/9$ pixels (subject to simple-points tests)
Results and Comparisons

SLIC

Turbopixels

Veksler e al.

CC-GMM

CC-GMM-SM

Proposed method
Results and Comparisons

SLIC  
CC-GMM  
CC-GMM-SM

Turbopixels  
Veksler et al.

Proposed method
Quantitative Results on Two Public Benchmarks

1) BRSDS500 benchmark; 2) Chemnitz optical-flow-based Sintel benchmark

- Left to right: the proposed method is, #1, #2, #1 (for most $K$’s) and #1
- In the second case, #1 is the (much) slower CC-GMM + Splits/Merges [Chang et al.]
Recent findings (not shown here):
- SEEDS is between the proposed method and SLIC; in quantitative results, however, the proposed method is uniformly better than SEEDS
- GPU Turbopixels has similar timings to the proposed method
Conclusion

- Improved model and faster inference for a GMM over simply-connected superpixels
- For more visual comparisons and our GPU code\textsuperscript{5}:
  http://people.csail.mit.edu/freifeld/publications
- Some possible extensions:
  - Drop-and-replacement into the video-based pipeline in [Chang et al.]
  - Other features than color
  - Hierarchical models
  - Non-Gaussian distributions for the color/features

\textsuperscript{5}Large parts of the code were written by Yixin Li
Simple Point Test

NCC_{FG} = 1  \quad NCC_{FG} = 1  \quad NCC_{FG} = 3  \quad NCC_{FG} = 1
\quad NCC_{BG} = 1  \quad NCC_{BG} = 1  \quad NCC_{BG} = 2  \quad NCC_{BG} = 3
\checkmark \quad \checkmark \quad \times \quad \times

FG pixel (4-connected)
BG pixel (8-connected)
Why does the computing time decrease with an increase in $K$?

Higher $K \Rightarrow$ smaller superpixels $\Rightarrow$
- less terms to sum in the sufficient statistics
- boundary needs to move a smaller distance $\Rightarrow$ less iterations to converge
Backup Slides

SLIC  
Turbopixels  
Veksler et al.

CC-GMM  
CC-GMM-SM  
Proposed method
Backup Slides

SLIC
Turbopixels
Veksler et al.

CC-GMM
CC-GMM-SM
Proposed method
Backup Slides

SLIC  Turbopixels  Veksler et al.

CC-GMM  CC-GMM-SM  Proposed method

Freifeld, Li and Fisher  (MIT CSAIL SLI)