Simple Linear Iterative Clustering - SLIC
What is wrong with pixels?

1) pixels are **unnatural entities**, just a consequence of the discrete representation of images;

2) the **number of pixels is high**; this makes optimization on the level of pixels intractable.

3) pixels are **highly redundant**; neighboring pixels are highly correlated.
What are superpixels

“Superpixels correspond to small, nearly-uniform regions in the image”


“Superpixels are perceptually meaningful atomic regions …. They … provide a convenient primitive from which to compute image features, and greatly reduce the complexity of subsequent image processing tasks.”

Applications of Superpixels

- Body modeling
- Object detection
- Depth estimation
SLIC Algorithm

1. Convert the RGB image to CIELAB color space. The CIELAB color space is *perceptually uniform*, i.e., a change of the same amount in a color value produce a change of about the same visual importance.
SLIC Algorithm

1. Convert the RGB image to CIELAB color space.
SLIC Algorithm

2. Initialize cluster centers $C_k = [l_k; a_k; b_k; x_k; y_k]^T$ by sampling pixels at regular grid steps $S$. 

\[ \sqrt{\frac{N}{K}} = S \]

- Number of pixels in the image
- Desired number of superpixels
SLIC Algorithm

3. Move cluster centers to the lowest gradient position in a $3 \times 3$ neighborhood.
4. Set label $L(i) = -1$ for each pixel $i$. 

$L =$

\[
\begin{array}{cccccc}
-1 & -1 & -1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & -1 & -1 \\
\end{array}
\]
SLIC Algorithm

5. Set distance $d(i) = -\infty$ for each pixel $i$.

\[
D' = \sqrt{\left(\frac{d_c}{m}\right)^2 + \left(\frac{d_s}{S}\right)^2}
\]

\[
d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}
\]

\[
d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2}
\]
6. Set distance $d(i) = -\infty$ for each pixel $i$.

\[
d = D = \sqrt{(d_c)^2 + \left(\frac{d_s}{S}\right)^2} m^2
\]

controls the relative importance of shape and color

large $m$ → favors more compact (lower area to perimeter ratio) superpixels.

small $m$ → favors more adherence to edges.
SLIC Algorithm

7. repeat
   
   for each cluster center \( C_k \) do
     
     for each pixel \( i \) in a \( 2S \times 2S \) region around \( C_k \) do
       
       Compute the distance \( D \) between \( C_k \) and \( i \).
       
       if \( D < d(i) \) then
         
         set \( d(i) = D \)
         
         set \( L(i) = k \)
       
       end if
     
   end for
   
   end for
   
   compute new cluster centers.
   
   compute residual error \( E \).
   
   until \( E < \) threshold.
SLIC Algorithm

Example 1: image size = 735×980 pixels
\[ K = 1333 \text{ superpixels}; \quad m = 40 \]
SLIC Algorithm

Example 2:

Unsupervised Segmentation based on SLIC super-pixels:

David Aldavert - Computer Vision Center
2013-05-29

https://www.youtube.com/watch?v=TGaNkGktTlhQ
SLIC Algorithm

Example 3:

https://www.youtube.com/watch?v=6o2HogjeZkE
Problem with SLIC

“SLIC uses the same compactness ($m$) parameter (chosen by user) for all superpixels in the image. If the image is smooth in certain regions but highly textured in others, SLIC produces smooth regular-sized superpixels in the smooth regions and highly irregular superpixels in the textured regions.”

http://ivrl.epfl.ch/research/superpixels
Problem with SLIC

“SLIC uses the same compactness ($m$) parameter (chosen by user) for all superpixels in the image. If the image is smooth in certain regions but highly textured in others, SLIC produces smooth regular-sized superpixels in the smooth regions and highly irregular superpixels in the textured regions.”

http://ivrl.epfl.ch/research/superpixels
Problem with SLIC

“SLIC uses the same compactness \((m)\) parameter (chosen by user) for all superpixels in the image. If the image is smooth in certain regions but highly textured in others, SLIC produces smooth regular-sized superpixels in the smooth regions and highly irregular superpixels in the textured regions.”

http://ivrl.epfl.ch/research/superpixels
SLICO Algorithm

An adaptive distance function is introduced

\[
D = \sqrt{\left(\frac{d_c}{m}\right)^2 + \left(\frac{d_s}{S}\right)^2}
\]

where \( m \) is computed iteratively for each superpixel as the maximum color distance to the current centroid.
SLICO Algorithm

Example 4:
SLICO Algorithm

Example 5:
SLICO Algorithm

Example 6:
References


- SLIC webpage: http://ivrl.epfl.ch/research/superpixels
Simple Linear Iterative Clustering
SLIC

END