## Introduction:

* Firstly introduced by Xiaofeng Ren and Jitendra Malik in their paper (but superpixel-like segmentation was applied already before):  
  Learning a Classification Model for Segmentation
  + Goal: Classification model for Segmentation accuracy
  + Drew attention through good results and holding its promises
* First algorithms around 2009

## Definition:

* Superpixels group perceptually similar pixels (e.g. colour) to create visually meaningful entities while heavily reducing the number of primitives for subsequent processing steps.
* The high number of pixels in images make then unfeasible computationally and images are a discretization of the continuous reality (why is that a problem though?)
* Often equally treated as oversegmentaion algorithms (but number of Superpixels settable)

## Properties of Superpixels:

* Partition: They should define a partition on the image (disjoint and labelled)
* Connectivity: Connected set of pixels
* Boundary adherence: Preserve image boundaries
* Compactness, Regularity and Smoothness (if no boundaries)
* Efficiency
* Controllable number of Superpixels

## Usage:

1. Tracking
2. Stereo and occlusion
3. 3D-Reconstruction
4. Saliency
5. Object detection
6. Object proposal detection
7. Depth recovery
8. Depth estimation
9. Semantic segmentation
10. Indoor scene understanding
11. Optical flow
12. Scene flow
13. Clothes parsing
14. Basis for CNNs
15. …

## Types of SuperPixel algorithms (Based on their high-level approaches)

1. Watershed-based e.g Compact Watershed, Water Pixels
2. Density-based e.g Edge-Augmented Mean Shift (EAMS), Quick Shift etc - usually classified as over-segmentation algorithms
3. Graph-based e.g Normalised cuts, Constant Intensity Superpixels, ERS
4. Contour-Evolution e.g Turbo Pixels, ERGC (fast)
5. Path-based e.g Path Finder, Topology Preserving Superpixels
6. Clustering-based e.g SLIC (Simple Linear Iterative Clustering), Depth Adaptive Superpixels, Preemptive SLIC, Voxel-Cloud Connectivity Segmentation (VCCS, usually used for point cloud)
7. Energy Optimisation e.g SEEDS (Superpixels extracted via energy-driven sampling), ETPS (Extended Topology Preserving Segmentation), CRS
8. Wavelet-based e.g Superpixels from Edge avoiding wavelets (SEAW)

## Differences

The main differences are in the objective function they minimize and in the optimization technique that performs the minimization. These are typically based on agglomerative clustering in the color domain [9, 12, 14], k-means style energy optimization [1], and coarse-to-fine optimization [7, 8]

## EAMS

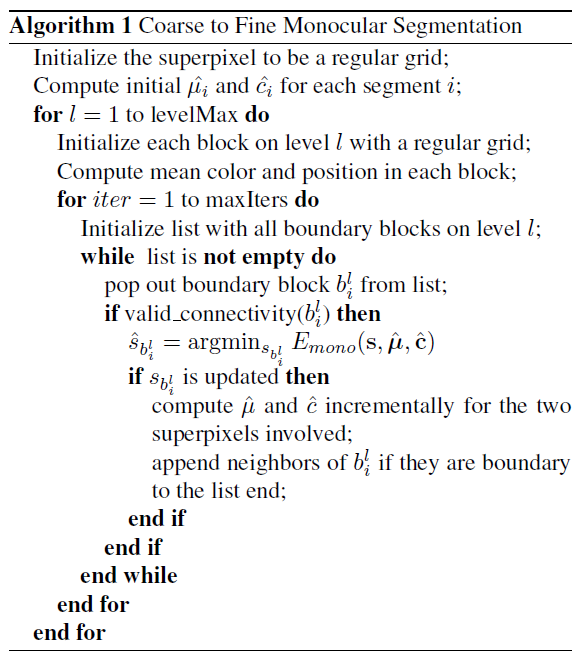
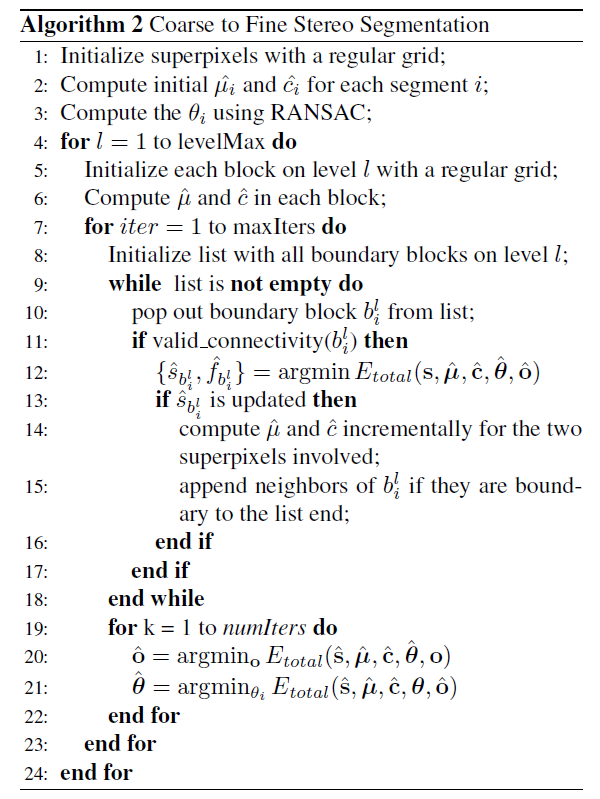
## ERS

SLIC

Performs iterative k-means style clustering, but which does not ensure that the final segmentation is connected.

## ETPS

“In this paper, we build on [31] (K. Yamaguchi, D. McAllester, and R. Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In ECCV, 2014.) and propose a much more efficient optimization algorithm that results in an order of magnitude less updates (speed-up). Inspired by the SEEDS algorithm [8] our method uses a coarse-to-fine energy update strategy, which allows the optimization to reach better energy minima than [31] when employing even a single iteration.”



Energy function: Objective function similar to kmeans clustering, where we want Superpixels that are coherent in appearance but that have also regular shape. We additionally add constraints on the size of the Superpixel to prevent tiny Superpixels.

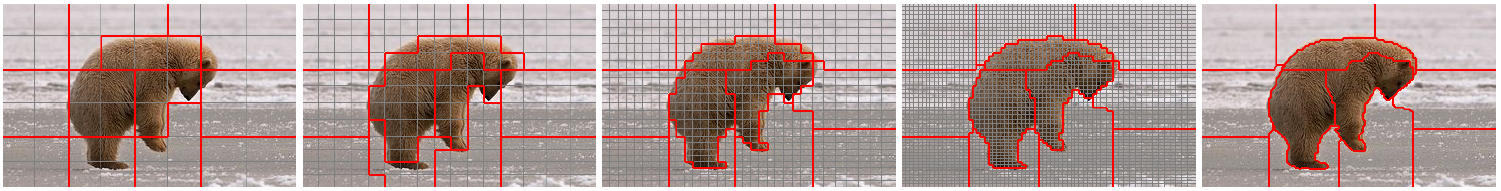
E pos = Shape Regularization (should be regular in shape)  
E col = Appearance Coherence (encourage color homogeneity)  
E b = Boundary Length (encourage small boundary length)  
E topo = Topology Preservation (focuses a connected component)  
E size = Minimum size (size needs to be at least ¼ of their initialization size)

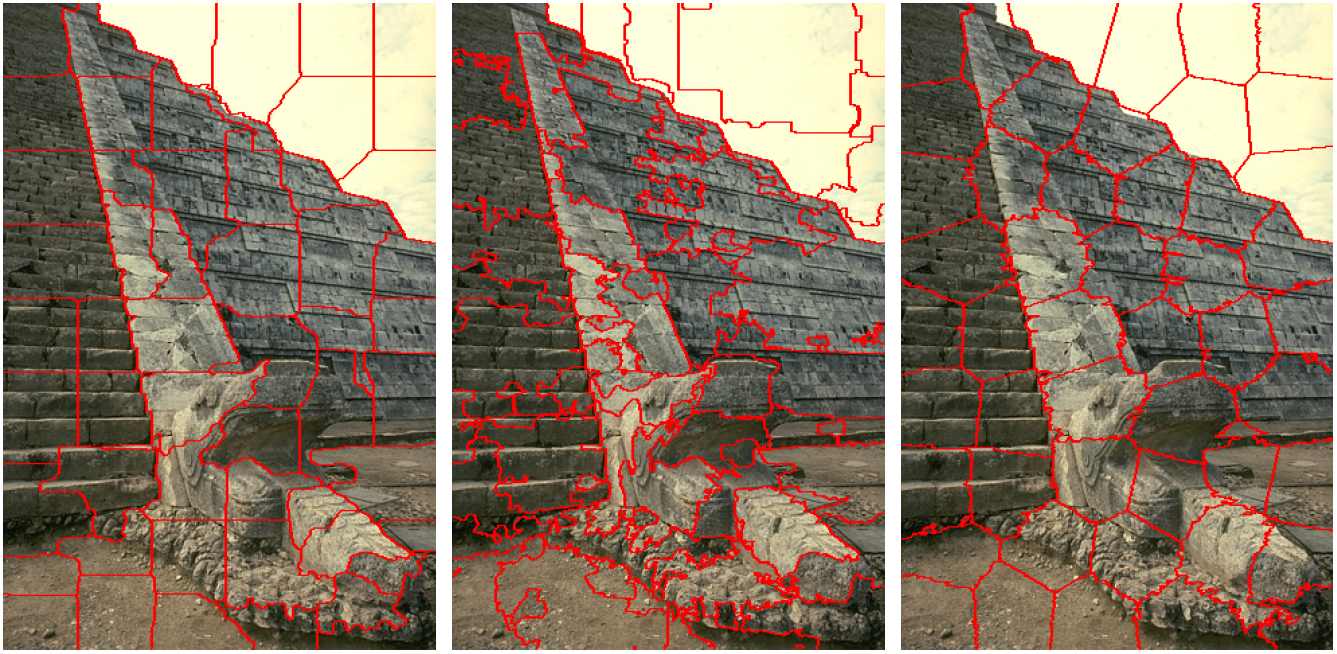
E mono =

1. Rectangular grid
2. Init + For level:
3. Compute mean color (center) and mean position
4. For iter:
5. Get all boundary blocks on level
6. While list != empty:
7. Valid connectivity block i -> minimize Energy function to find block best suitable block
   1. If new block is different from old one:

Update center and means for the two blocks and

Add i’s neighbours to block list end





ETPS, SEEDS, SLIC