Constrained Parametric Min-Cuts for Automatic Object Segmentation

Sanmit Narvekar
Department of Computer Science
The University of Texas at Austin

September 28, 2012

Outline

- Introduction
- Method Overview
- Phase I: Generate Pool of Segments
- Phase II: Rank Segments
- Experiments
- Analysis
- Conclusion

Object Segmentation

Image credit: Carreira & Sminchisescu (PAMI 2012)

Object Segmentation

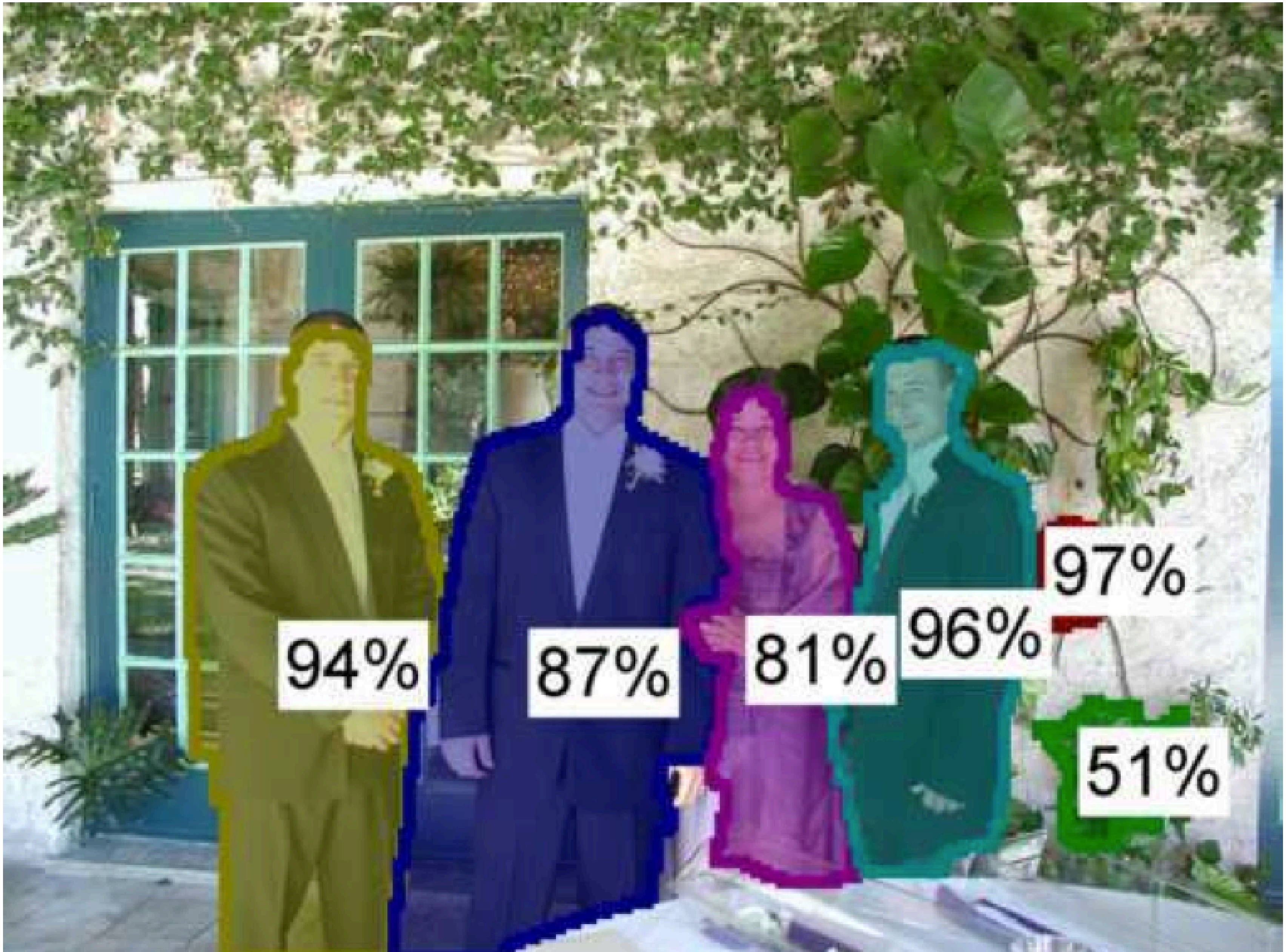


Image credit: Carreira & Sminchisescu (PAMI 2012)

Approaches

"Traditional" Way **VS** **CPMC Way**

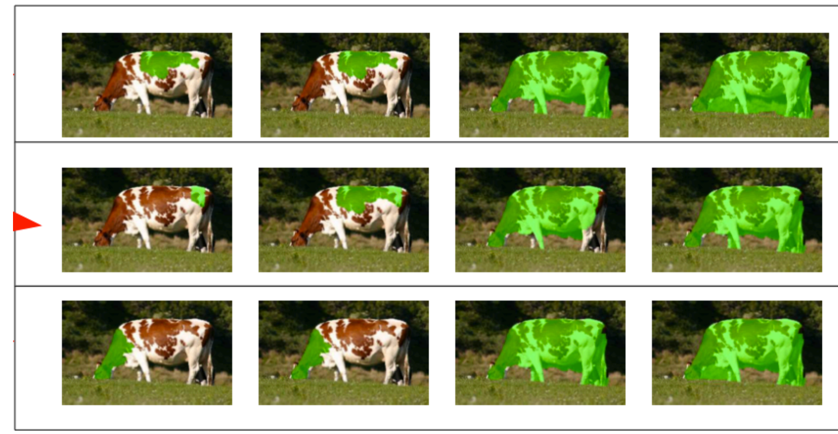
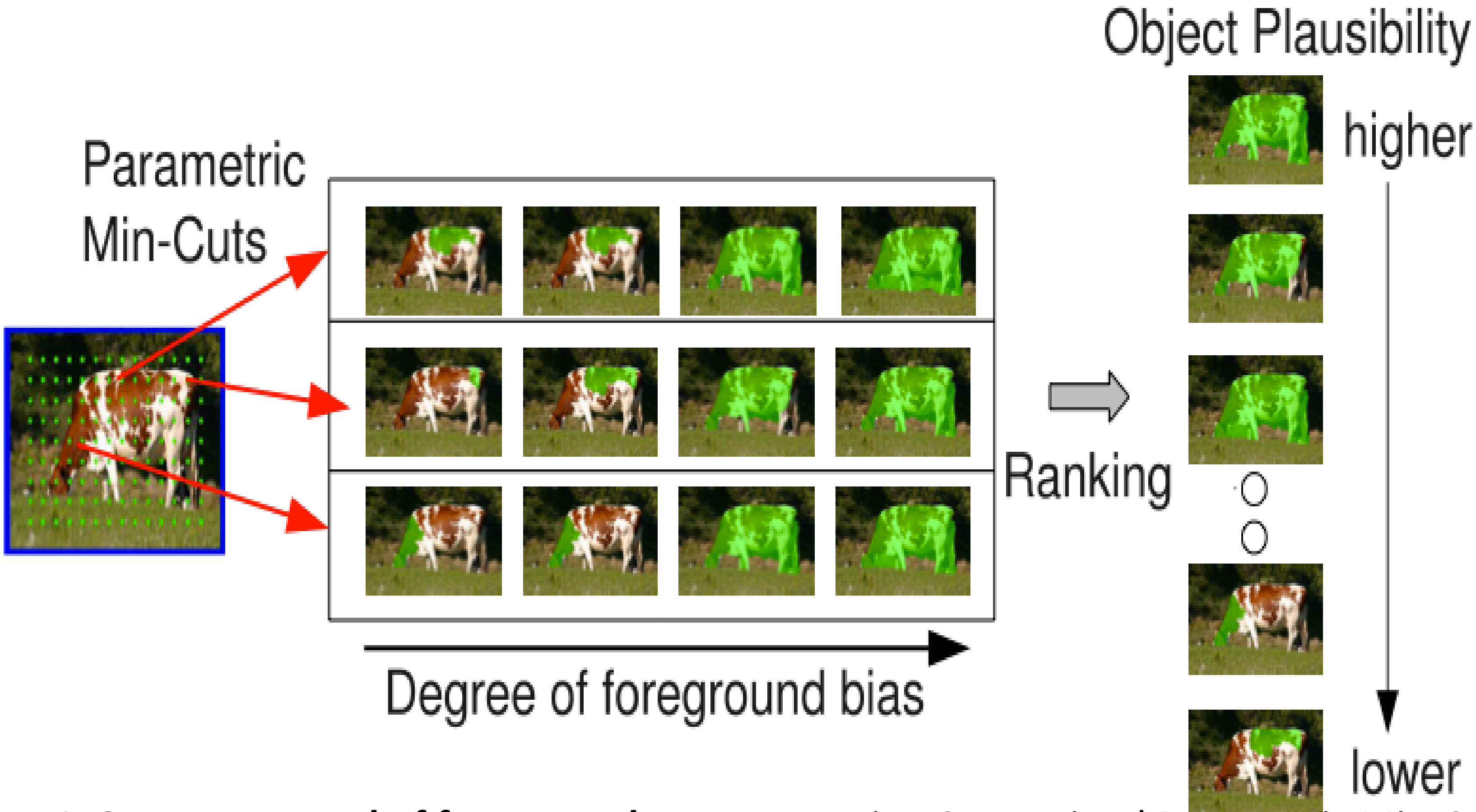


Image credit: Silberman et. al. (ECCV 2012)

Image credit: Carreira & Sminchisescu (CVPR 2010)

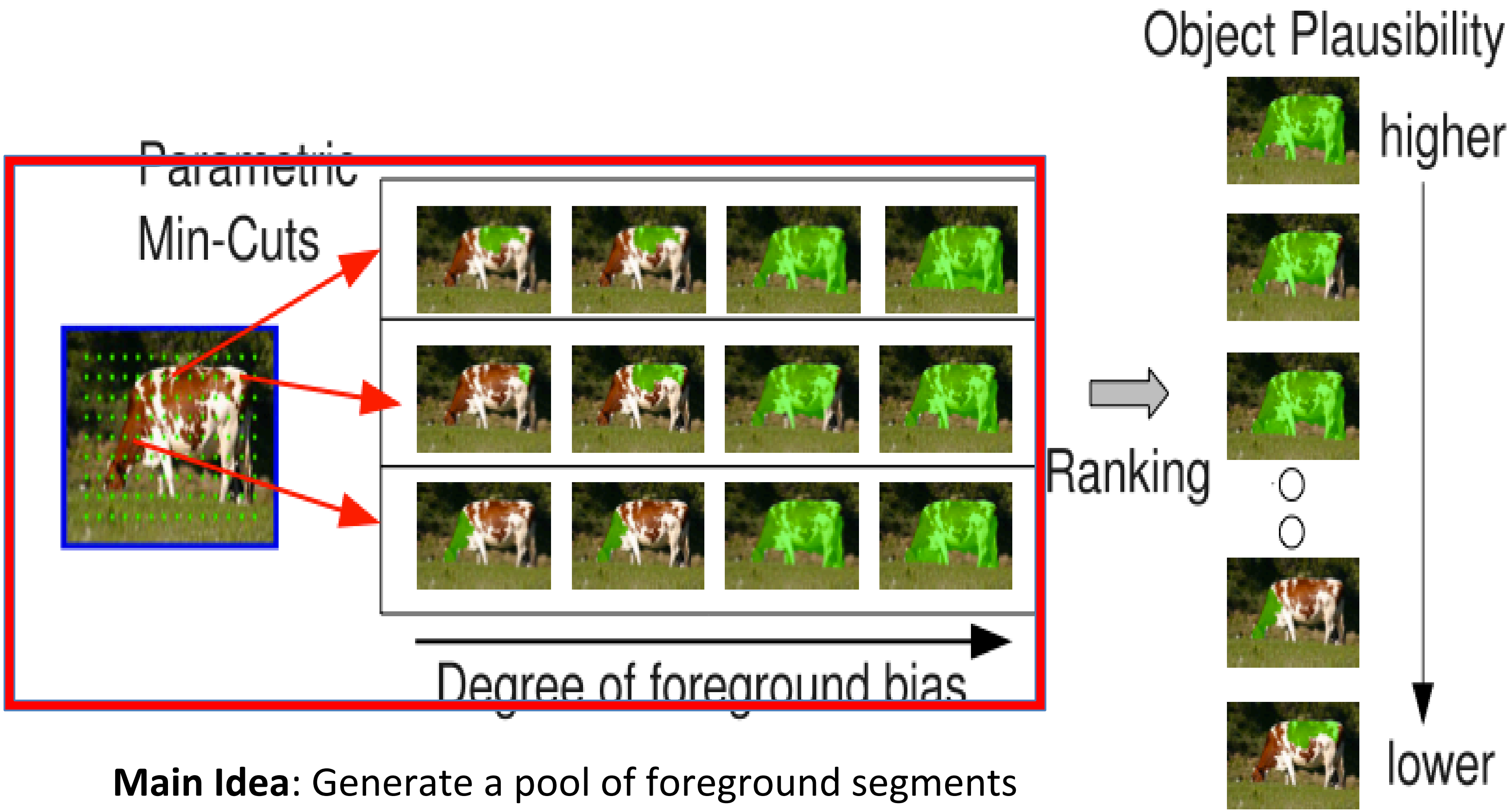
Method Overview



Phase I: Generate **a pool of foreground segments** using Constrained Parametric Min-Cuts

Phase II: **Rank the segments** by learning a random forest regressor

Image credit: Carreira & Sminchisescu (CVPR 2010)



Main Idea: Generate a pool of foreground segments

1. Seed the image-graph with foreground and background seeds
2. Map the image onto a weighted graph
3. Solve the CPMC optimization objective
4. Repeat 1 – 3 with varying seeds and parameters
5. Filter initial candidates with fast rejection

Image credit: Carreira & Sminchisescu (CVPR 2010)

Seeding Policy

- Foreground seeds
  - 5x5 grid approach
- Background seeds
  - Seed along image border
  - Vertical edges on border
  - Horizontal edges on border
  - All but bottom edge

Image credit: Carreira & Sminchisescu (CVPR 2010)

Mapping onto a Weighted Graph

- Map the image onto a weighted graph where:
  - Nodes are pixels
  - Weighted edges represent similarity between pixels
  - Add 2 special nodes: one to foreground, one to background

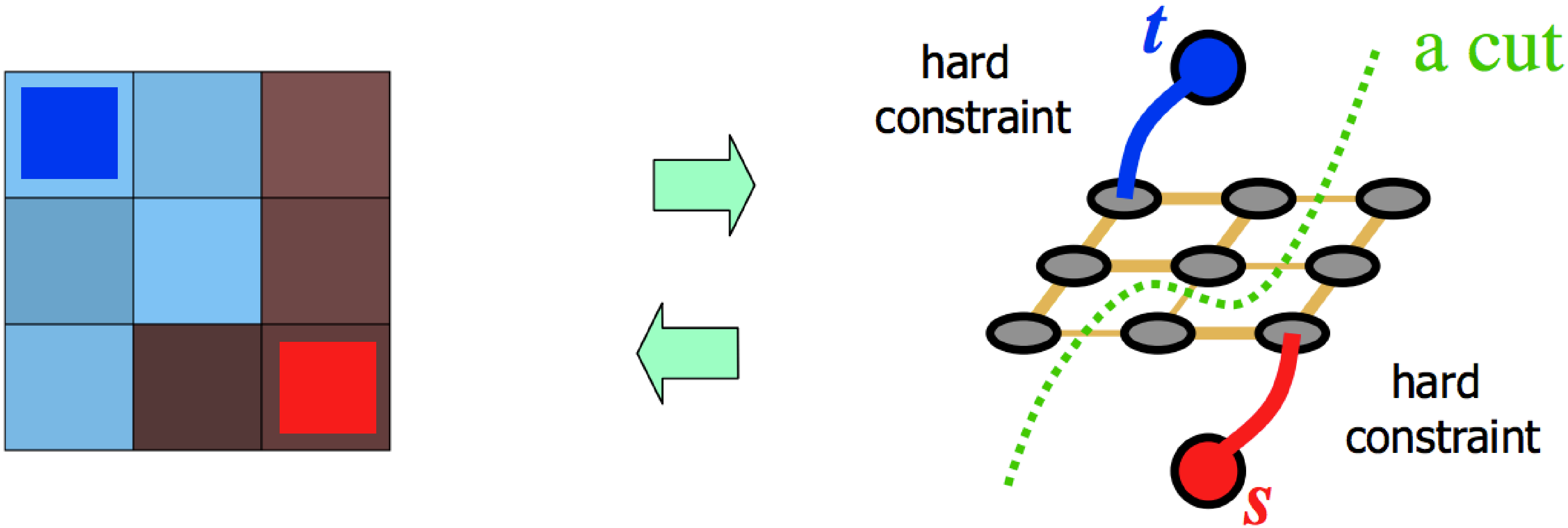


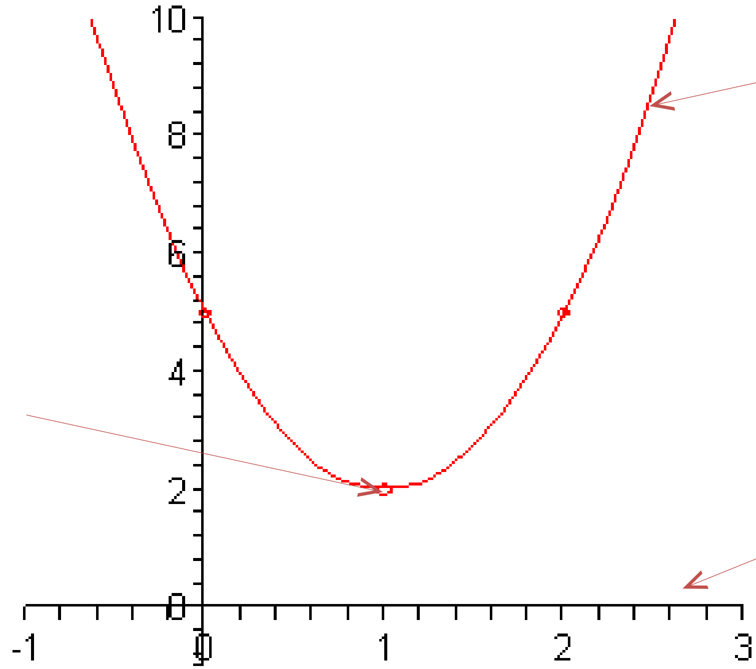
Image credit: Boykov & Jolly (ICCV 2001)

Optimization Objective

- We want to design a function such that

High “energy” for bad labelings

Low “energy” for good labelings (note this will encode our biases of what is good and bad)



Input space is X, a labeling of all pixels in the image

MINIMIZE

$$E^{\lambda}(X) = \sum_{u \in \mathcal{V}} D_{\lambda}(x_u) + \sum_{(u,v) \in \mathcal{E}} V_{uv}(x_u, x_v)$$

Optimization Objective

MINIMIZE

$$E^{\lambda}(X) = \underbrace{\sum_{u \in \mathcal{V}} D_{\lambda}(x_u)} + \sum_{(u,v) \in \mathcal{E}} V_{uv}(x_u, x_v)$$

Penalize on the node-pixel assignment
Determines "foreground bias"

$$D_{\lambda}(x_u) = \begin{cases} \boxed{0 \quad \text{if } x_u = 1, u \notin \mathcal{V}_b} \\ \infty \quad \text{if } x_u = 1, u \in \mathcal{V}_b \\ \infty \quad \text{if } x_u = 0, u \in \mathcal{V}_f \\ \boxed{f(x_u) + \lambda \quad \text{if } x_u = 0, u \notin \mathcal{V}_f} \end{cases}$$

No penalty for labeling as foreground

Prevent labeling background nodes as foreground, and vice versa

Penalizes for labeling as background (controls degree of foreground bias)

$$\begin{cases} f(x_u) = 0 \\ f(x_u) = \ln p_f(x_u) - \ln p_b(x_u) \end{cases}$$

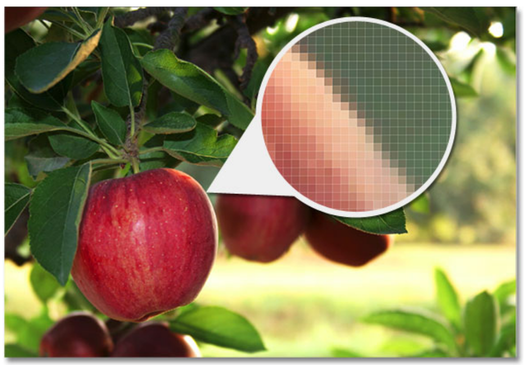
Uniform bias (λ everywhere)

Supplement with color term based on color distributions

Optimization Objective

MINIMIZE

$$E^{\lambda}(X) = \sum_{u \in \mathcal{V}} D_{\lambda}(x_u) + \underbrace{\sum_{(u,v) \in \mathcal{E}} V_{uv}(x_u, x_v)}_{\text{Penalize assigning different labels to “similar” neighbors}}$$



$$V_{uv}(x_u, x_v) = \begin{cases} 0 & \text{if } x_u = x_v \\ g(u,v) & \text{if } x_u \neq x_v \end{cases}$$

Adjacent pixels are usually in the same class, so no penalty

Different labels – penalize based on similarity

$$g(u,v) = \exp\left[-\frac{\max(gPb(u), gPb(v))}{\sigma^2}\right]$$

Measures similarity between *u* and *v*

gPb is the contour detector from Arbelaez et. al.

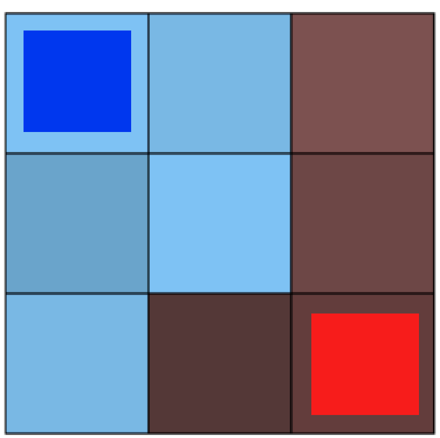
Image credit: Photoshop Essentials

Constrained Parametric Min Cuts (CPMC)

MINIMUM

$$E^{\lambda}(X) = \sum_{u \in \mathcal{V}} D_{\lambda}(x_u) + \sum_{(u,v) \in \mathcal{E}} V_{uv}(x_u, x_v)$$

Equivalent to min-cut on graph



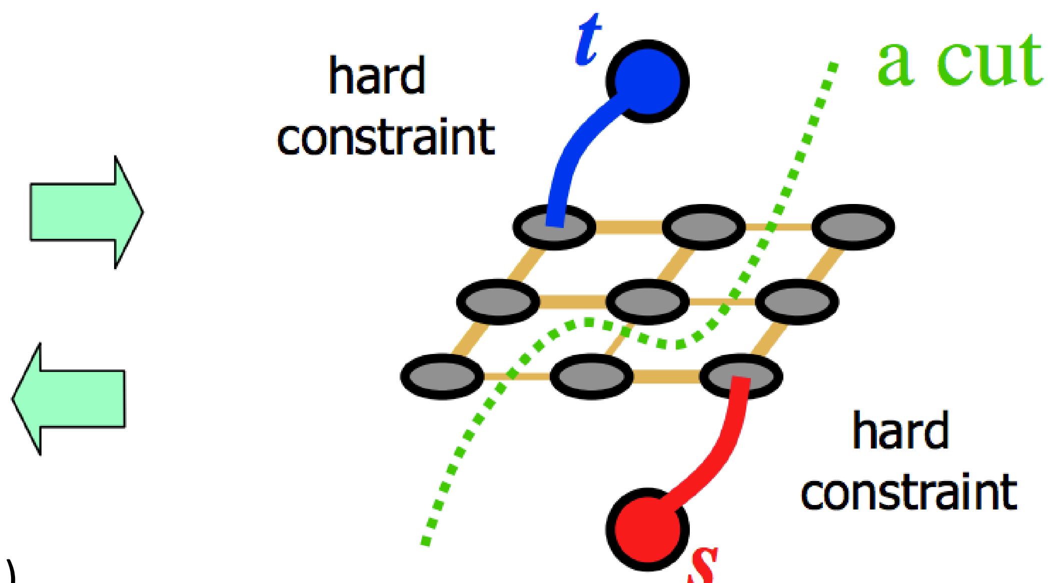
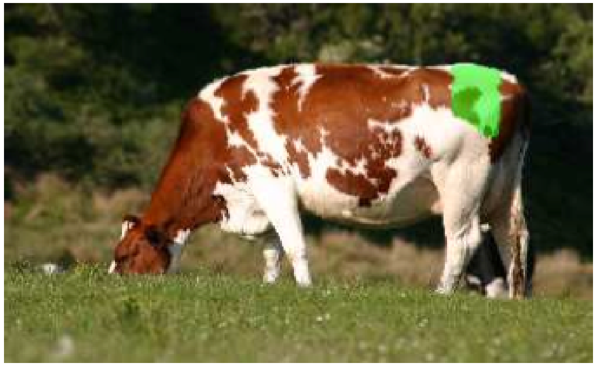


Image credit: Boykov & Jolly (ICCV 2001)

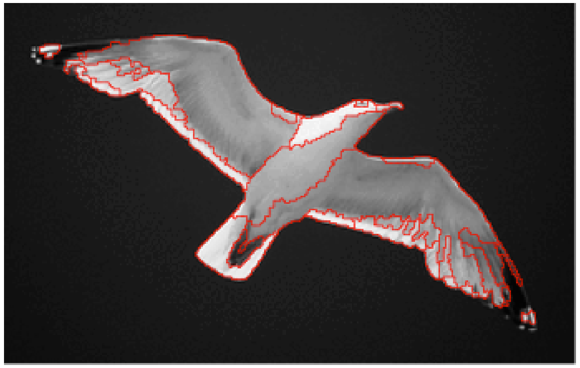
Fast Rejection

- Now we have about 10,000 candidate segments!
  - Need to eliminate some:

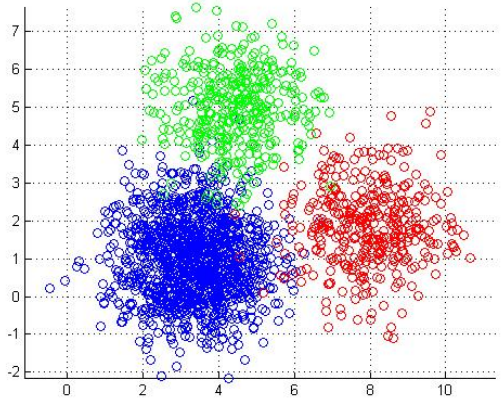
Remove small segments (less than 150 pixels)



Sort by ratio cut, and keep top 2000



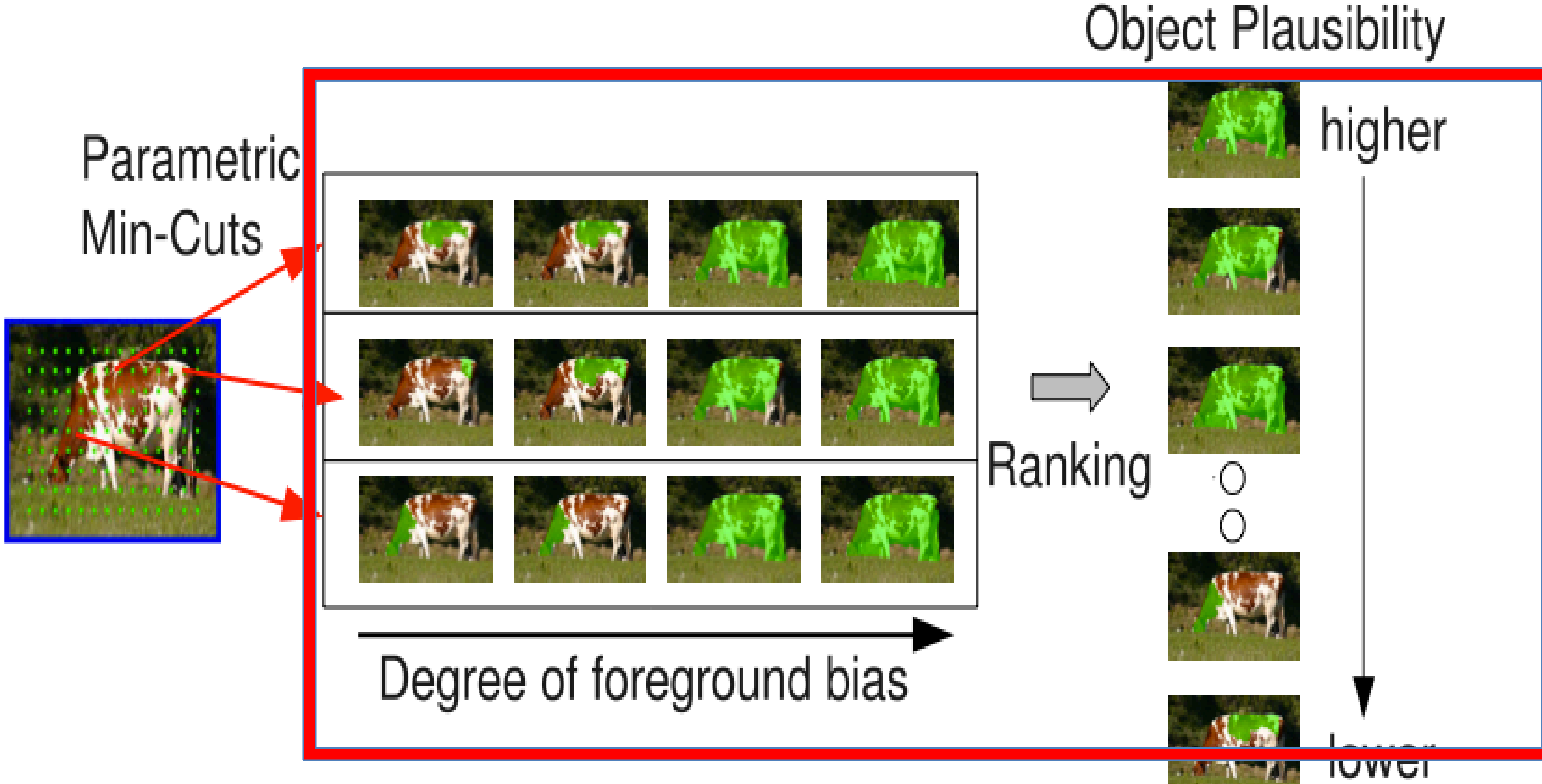
Cluster using overlap, and keep lowest energy segment in each cluster



- Only around 150 candidates left

Image Credit: Carreira & Sminchisescu (CVPR 2010), Wang & Siskind (PAMI 2003), Mathworks

Phase II

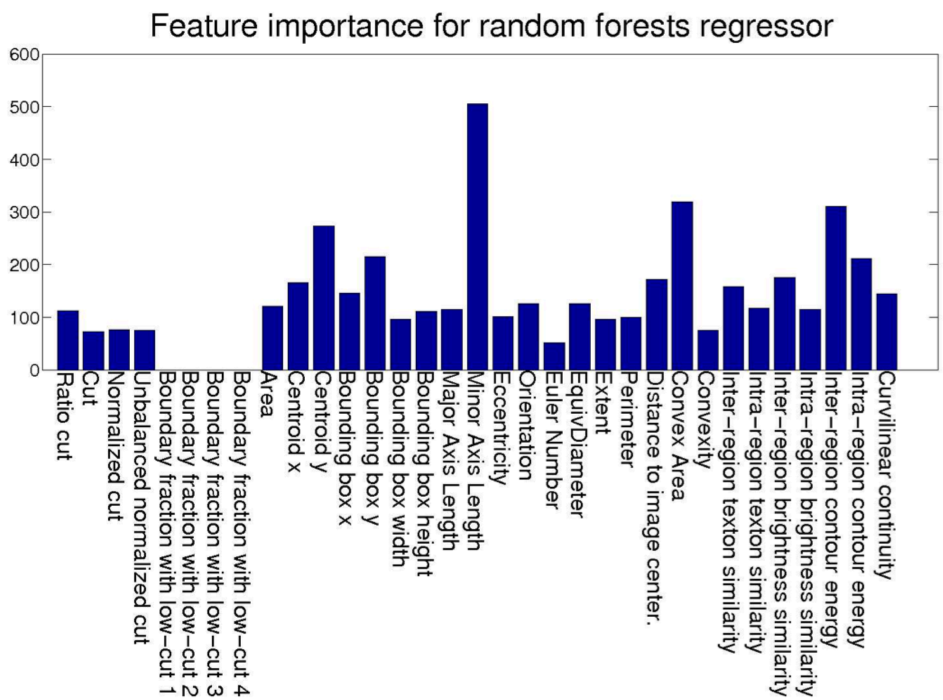


Main Idea: Machine learn which segments are good (i.e. rank them)

1. Generate features that could describe “good” segments
2. Train a Random Forest
3. Diversify the rankings

Image credit: Carreira & Sminchisescu (CVPR 2010)

Segment Features



- Graph Partition Properties (8)
  - Common for segmentation
- Region Properties (18)
  - Location and scale of objects
- Gestalt Properties (8)
  - Mid-level cues (e.g. continuity)

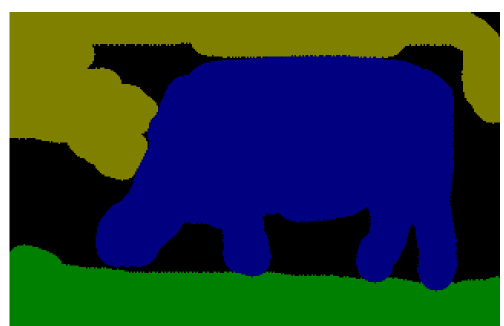
Graph credit: Carreira & Sminchisescu (CVPR 2010)

Random Forest Regression

- Non-linear model that uses several regression trees
- We **maximize** the **pixel-wise overlap** between a segment S. and the ground truth G.

$$O(S, G) = \frac{|S \cap G|}{|S \cup G|}$$

- Penalizes on over-segmenting and under-segmenting



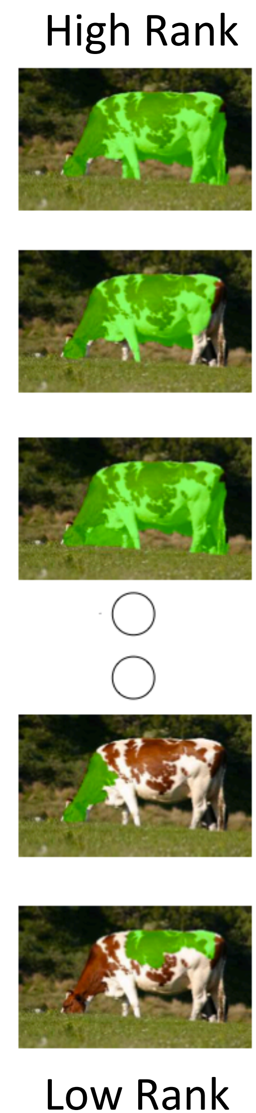


Image credit: Carreira & Sminchisescu (CVPR 2010)

Maximal Marginal Relevance (MMR)

- Rankings returned by Random Forests put similar segments together
- MMR diversifies the rankings
  - After the top segment, each subsequent segment is the original score minus a redundancy measure (the overlap)

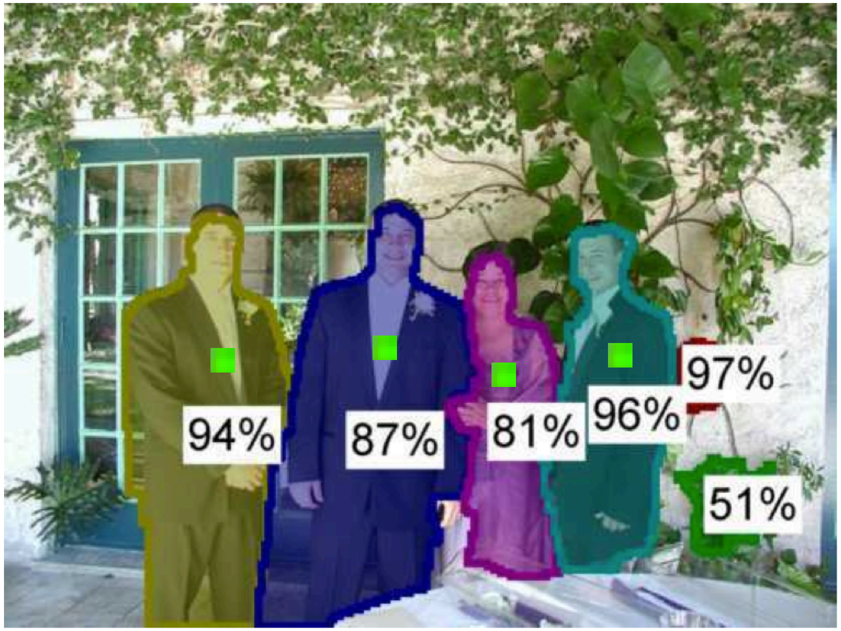


Image credit: Carreira & Sminchisescu (PAMI 2012)

Experiments

- Weizmann's Segmentation Evaluation Database
  - 100 grayscale images
  - One prominent foreground object in each

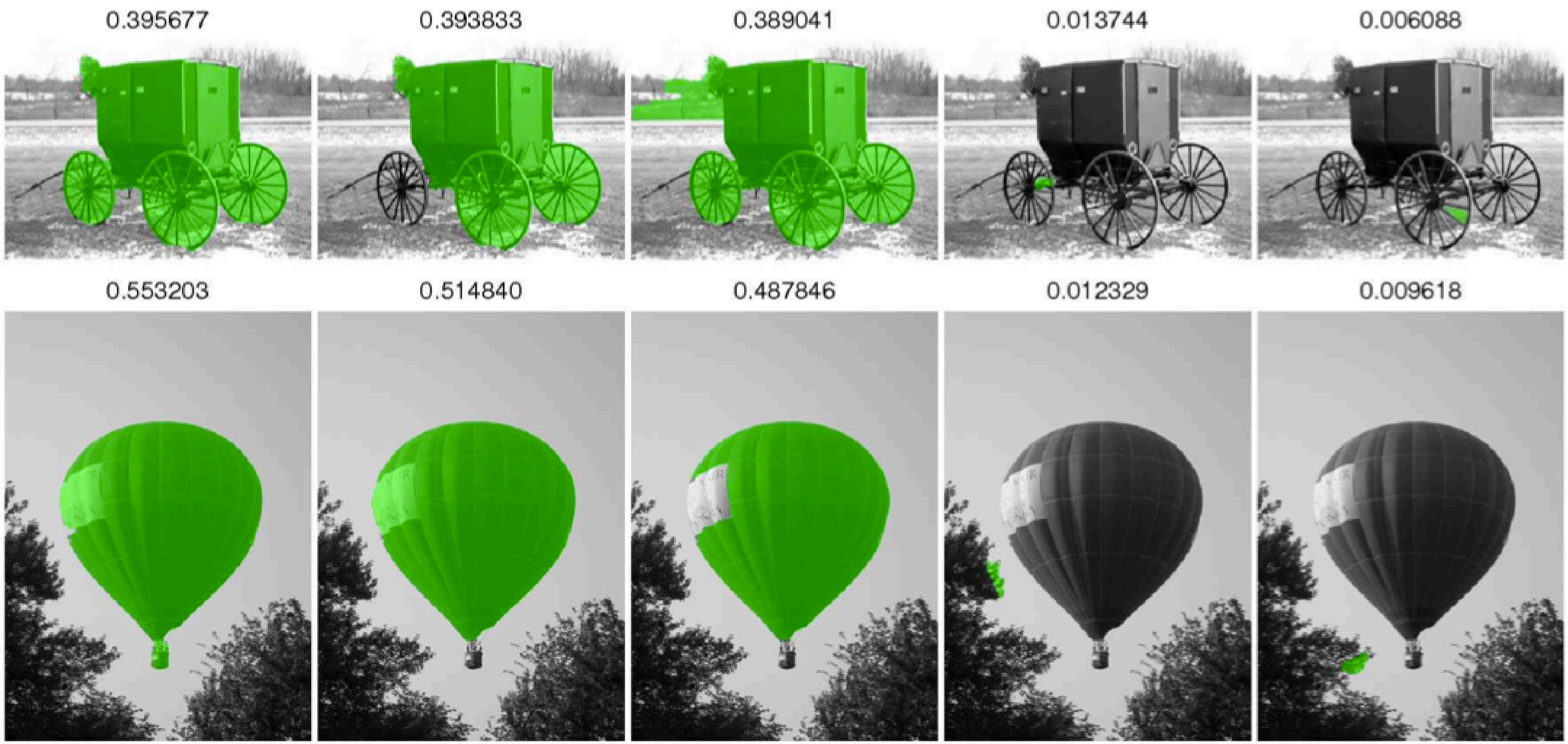


Image credit: Carreira & Sminchisescu (CVPR 2010)

Experiments

- Microsoft Research Cambridge Dataset v2 (MSRC)
  - 591 color images, 23 classes
  - Evaluated as pool of segments, not individual rankings





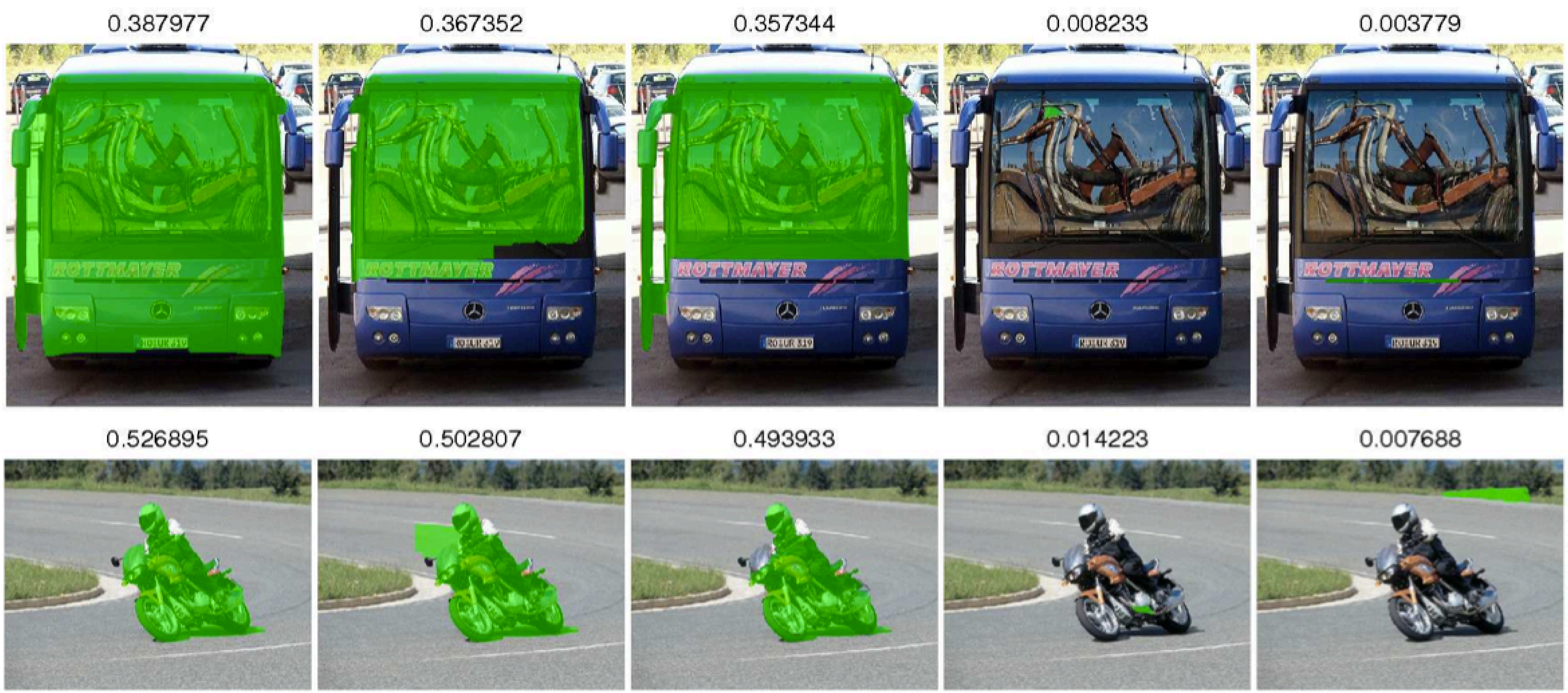


|R|: # pixels in ground truth

Image credit: MSRC

Experiments

- Visual Object Challenge (VOC) 2009
  - 3000 color images, 20 classes
  - Evaluated as pool of segments, not individual rankings



e

|R|: # pixels in ground truth

Image credit: Carreira & Sminchisescu (CVPR 2010)

Analysis

- Strengths
  - Gives multiple possible foreground segments and their scores
  - More likely to represent an object using less segments

- Weaknesses
  - Very small objects
  - Seeding density and hollow objects
  - Partially occluded objects
  - Only “grows” one foreground segment at a time
  - Computationally expensive (too many cuts)

Image credit: Carreira & Sminchisescu (CVPR 2010, PAMI 2012)

Conclusions

- Comparison to related work
  - Arbelaez et. al.
  - Silberman et. al.
- Extensions
  - Multiple object segmentation
  - Applied to object recognition, perhaps in an unsupervised, active setting

Questions?