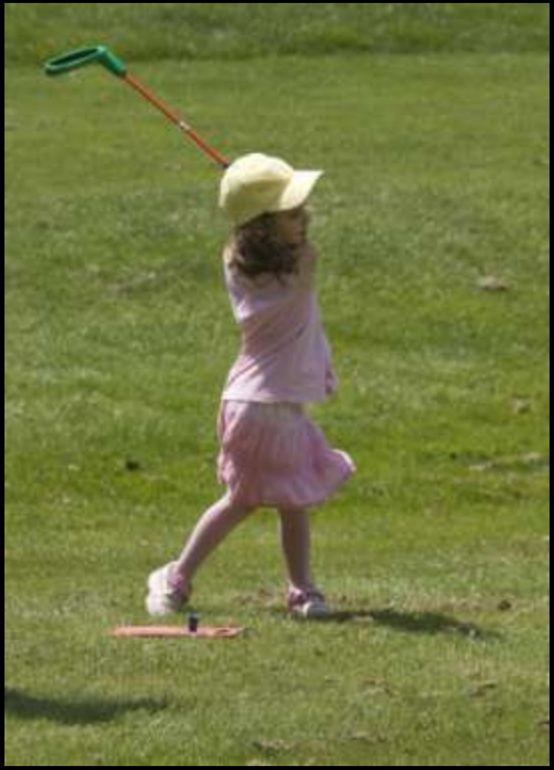
GPU TECHNOLOGY CONFERENCE

Fast High Quality Image and Video Background Removal with CUDA

Timo Stich | NVIDIA

Motivation



Take a Picture

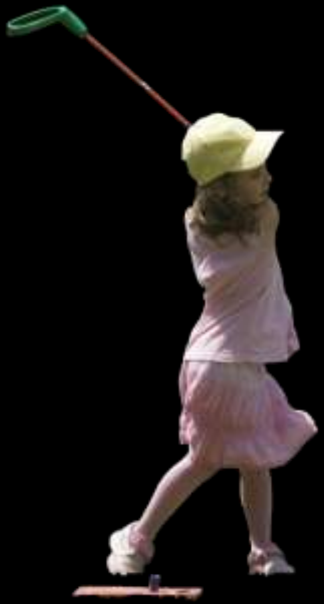
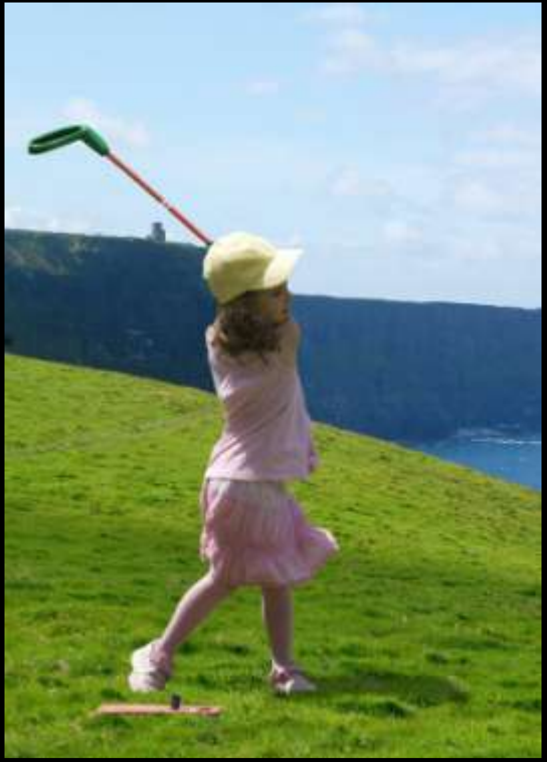


Image Segmentation
(This Talk)



Play Golf in Ireland!

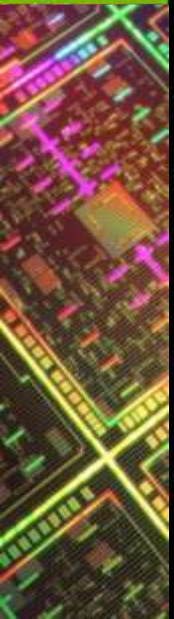
Photo Credits: Golf Girl, Bob Cotter; Cliffs of Moher, Peter Gorman, CC 2.0

Outline

- Interactive Segmentation
- Probabilistic Color Models
  - Gaussian Mixture Model
  - RGB Histogram
- Edge Aware Segmentation
  - Graph Cut (NPP Primitive)
- Summary

Interactive Segmentation

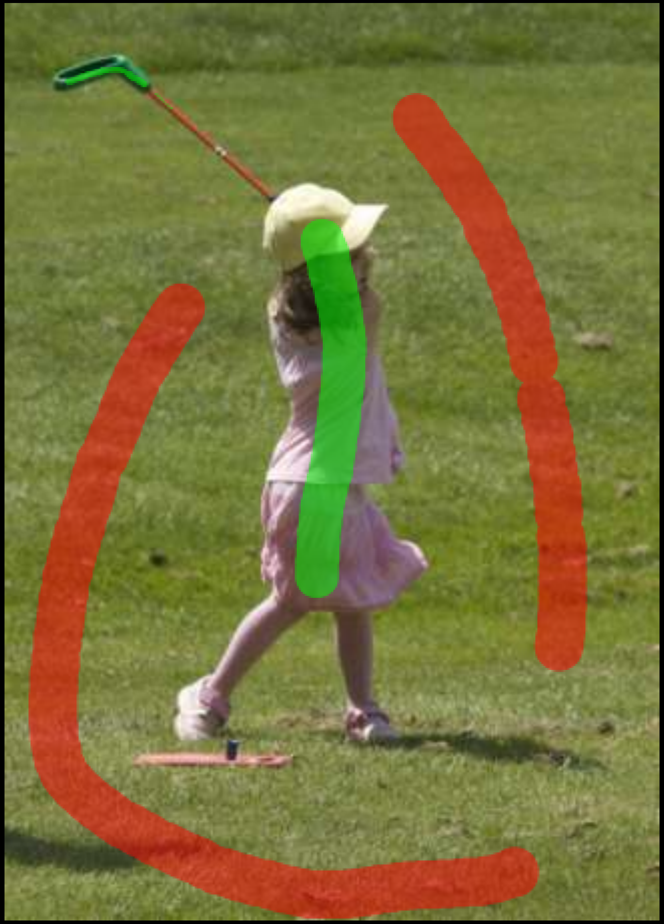
- User draws strokes to highlight FG and BG
- Intelligent computation of segmentation
- User corrects/refines with additional strokes until satisfied



PROBABILISTIC COLOR MODELS

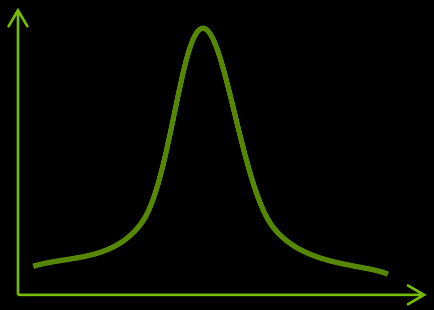
Color Models from User Strokes

- Strokes define sets of pixels that are FG and BG
- Anchor the solution (spatial constraint)
- Sample of the Color Distributions (color constraint)



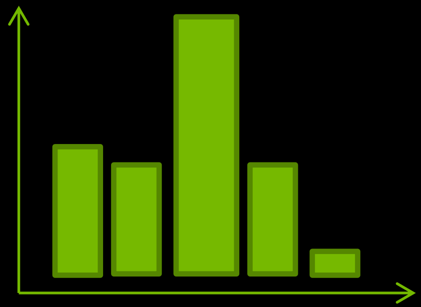
Gaussian Mixture Model

- N Gaussian blobs for each FG/BG
  - Very Accurate
  - Initialization Non-Trivial
  - Picking N wrong can lead to Overfitting
- Implementation Details
  - Compute Largest Eigenvector for Clustering
  - Reductions to compute the Means and Deviations
  - Code available in “GrabCut” SDK Sample



RGB Histogram Model

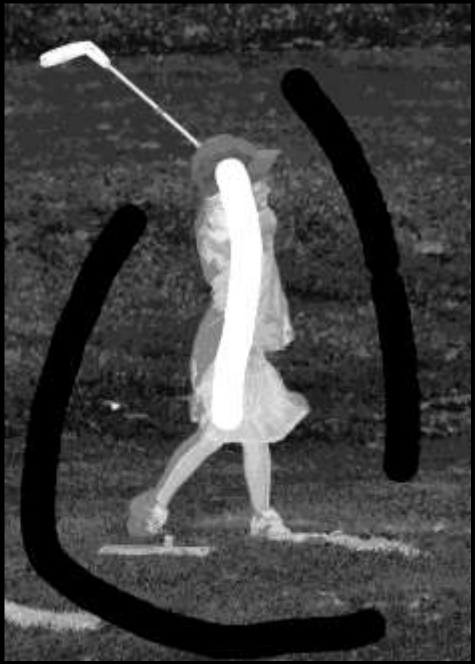
- 32x32x32 RGB Histogram for each FG/BG
  - Less Accurate
  - Simple Implementation
  - Very Robust



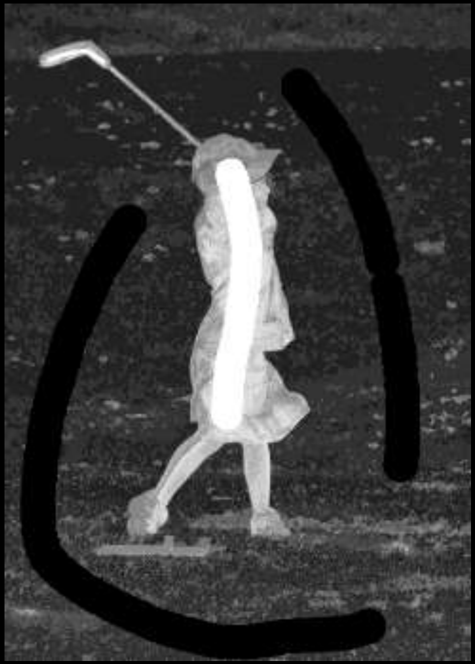
- Implementation Details
  - Accumulate Votes per Warp (__ballot, __popc)
  - GMEM Atomics
  - Multiple Histograms, Reduce in second Kernel

Color Model Comparison

2.8 MP, GTX 680	Accuracy	Robustness	Simplicity	Performance
GMM	+	0	-	16 ms
Histogram	0	+	+	4.7 ms



GMM (N=4)



*

Histogram (32^3)

EDGE AWARE SEGMENTATION

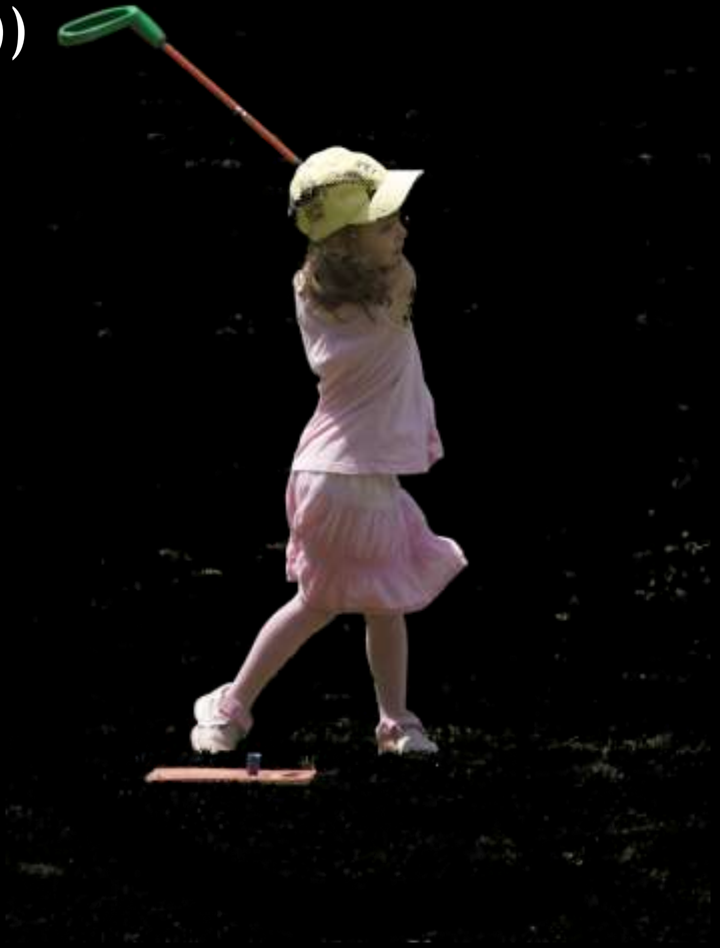
Computing the Segmentation

- What we have:
  - Probabilities for colors to belong to FG / BG
  - Some pixels classified as FG / BG by the user
  - A lot of pixels where we don’t know the class

- What we want:
  - Classify all pixels as either FG or BG

Naïve Approach

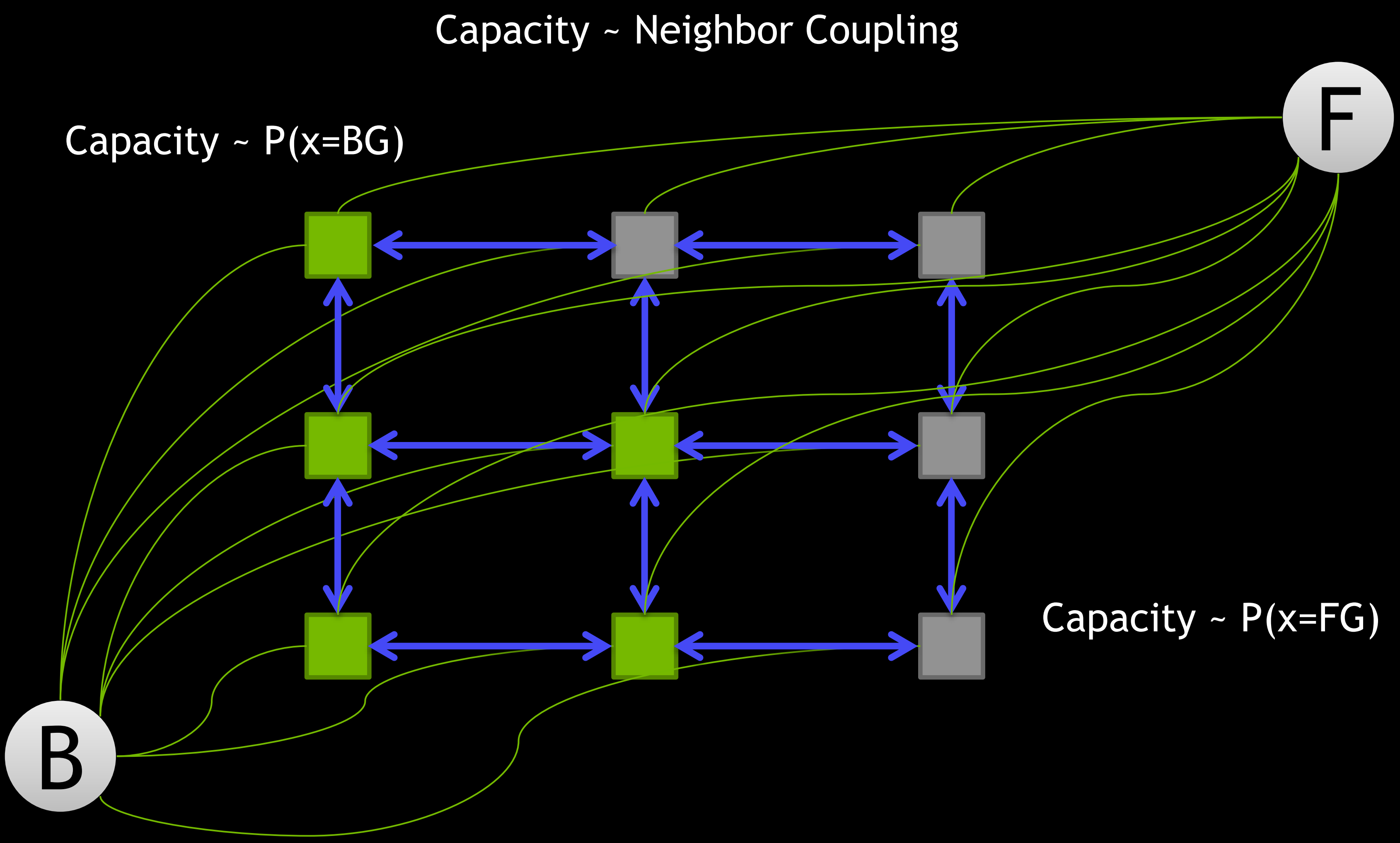
- Classify according to Max Probability
  - L(x) : max(P_{FG}(C(x)), P_{BG}(C(x)))
- Result:
  - Very noisy
  - Not High Quality



Edge Aware Segmentation

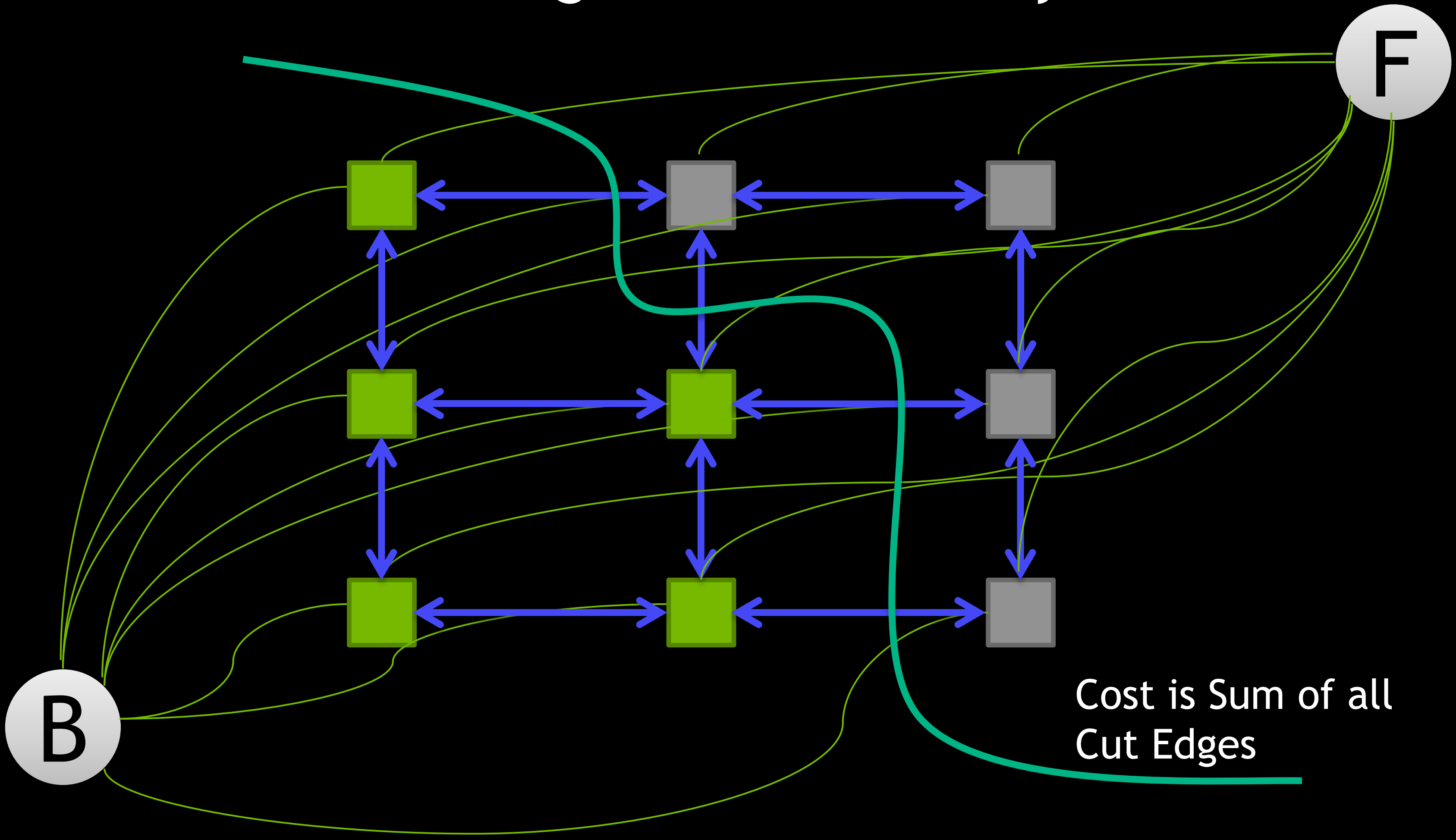
- Problem with Naïve:
  - Each pixel for itself
  - No context information
- Intuition:
  - Segmentation boundaries should be at image edges
- Solution: Decision depends also on neighborhood labels
  - Similar Color - Strong Coupling
  - Different Color - Weak Coupling

Graph Cut Segmentation



Graph Cut Segmentation

- Minimal Cut <-> Edge Aware Boundary



Cost is Sum of all Cut Edges

NPP Graph Cut Primitive

- nppiGraphcut(8)_32(f/s)8u
- Computes Minimum Cut on Regular Graphs
  - One array for edges capacities for each direction (left, right, top bottom + diagonal dirs for 8 nbhd)
  - One array for connections to Source and Sink
    - Terminals = Source – Sink
- Returns 0/1 in 8u array -> BG/FG classification

Graph Cut Performance

- Result:
  - Noise free result
  - Segmentation boundaries align well with image edges

2.8 MP	Quadro 6000[1,2]	GTX 680[1,2]	Core i7 @ 3 GHz[1,3]
4 Nbhd	47 ms	32 ms	460 ms
8 Nbhd	110 ms	72 ms	843 ms

1) Includes time for graph setup
2) NPP 5.0 (10-40% performance boost over 4.1)
3) Maxflow v3.01, Boykov et al, http://vision.csd.uwo.ca/code/maxflow-v3.01.zip

Update the Color Models

- Optionally use computed segmentation to improve the Color Models

Video Segmentation

- Initialize the Color Model once
- Compute Segmentation for each Frame

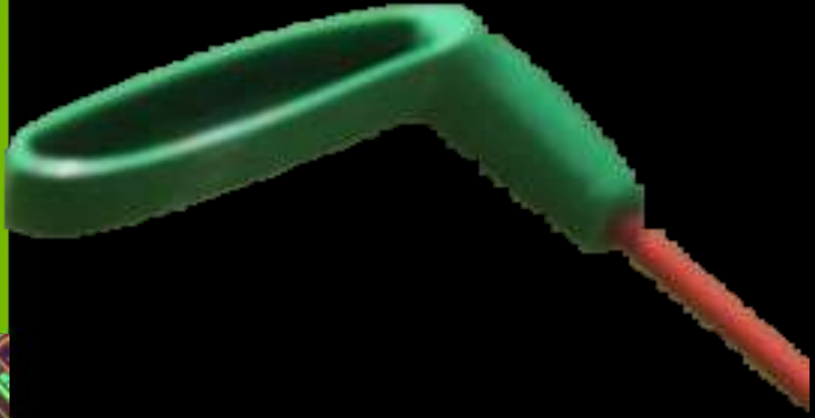
Summary

- Interactive Tool
- Probabilistic Color Model
- Edge Aware Segmentation

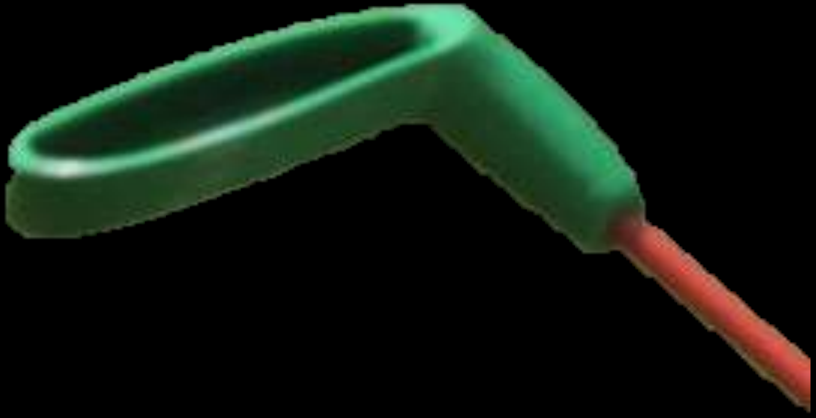
- Code: GrabCut SDK Sample (under CUDALibraries)

BACKUP SLIDES

4/8 Neighborhood Comparison



4 Neighborhood



8 Neighborhood

CUDA Warp Vote Accumulation

```
while(1) {
*smem_warpMaster = laneIdx;
warpMaster = *smem_warpMaster;
if( myKey == smem_keys[warpMaster] ) {
  ballot = __ballot(1);
  myBallot = laneIdx == warpMaster ?
                ballot : 0;
  break;
}
myVote = __popc(myBallot);
```