Interactive Video Cutout\textsuperscript{1}

Aaron Eidelson
Jiamin Bai

November 9th, 2011

\textsuperscript{1}Jue Wang, Pravin Bhat, R. Alex Colburn, Maneesh Agrawala, Michael F. Cohen, University of Washington, Microsoft Research
Objective

Given a video we want an easy and accurate way to separate a foreground object from the background.
Previous Work

This paper builds off of a number of previous works

- **Segmentation-based cutout**
  - User specifies some background and foreground regions
  - Segmentation done by extending known regions

- **Boundary-based cutout**
  - Beperating foreground and background by fitting curve to the object

- **Matting**
  - Border matting can estimate the alpha matte and foreground color around borders

- **Video as 3D object**
  - Video can be represented as a cube
System

The system is split up into three distinct stages:

1. Preprocessing
2. Interactive segmentation
3. Post-processing
Preprocessing

There is too much data to run a segmentation algorithm on a pixel-level (74 million pixels for 10 seconds), we need to simplify

1. Hierarchical mean-shift segmentation
   1.1 Cluster pixels on each frame into regions
   1.2 Group regions between frames to make 3D volumes

2. Neighbor determination
   ▶ Create links between each pixel/region/volume and its neighbors at each level of the hierarchy

3. Local statistics
Interactive Segmentation

User Interface

Min-Cut
Interactive Segmentation

User Interface

In 2D (images) all data is immediately visible. In 3D (video) pixels can be occluded.

▶ Represent video as 3d cube

▶ A path can be extruded through the volume and labeled
Interactive Segmentation
Hierarchical Min-Cut

Min-Cut is optimized using preprocessing and user input

- Nodes can be pixel, 2D region in a single frame, or 3D volume
  - If a root is “mixed” it is replaced by children
  - Nodes cover every frame of the video, no overlaps
- Known background nodes are discarded
Post-Processing

Results of interactive segmentation have noisy edges (in both space and time)

- Refinement min-cut
- Spatio-temporal matting
Post-Processing

Refinement min-cut

We perform a pixel-level min-cut at the boundary
- Set data cost based on gaussian foreground color model
- Use 10 pixel width spatially, 1 pixel temporally
Post-Processing
Spatio-temporal matting

Image matting applied frame-by-frame produces noisy results because small errors are incoherent

- Adopted from GrabCut’s border matting
  - Want to solve for the $\Delta$ and $\sigma$ that best matches the border

- Add an additional "smoothness" constraint to minimize large fluctuations in these values between frames
Results

For 100-200 frames:

- Preprocessing: 10-30 min
- Hierarchical min-cut: 5-15 sec
- Post-processing: 30-50 min
Points of Failure

- Shaky video
  - Optical flow
- Foreground similar color to background
  - User-defined curves to aid in the calculations
  - Crop out the rest of the foreground so color models can more tightly focus
- Worst-case, more user interaction
Questions?