“Selecting Objects with Freehand Sketches”

By Kar-Han Tan and Narendra Ahuja
at the University of Illinois at Urbana-Champaign

In Proceeding IEEE Int’l Conf. on Computer Vision 2001

Slides by David Anthony Torres
Computer Science and Engineering — University of California at San Diego
Overview of Tool Use:

• User makes an initial selection by drawing a sketch around the object to be selected.
• The tool automatically selects the object.
• User iteratively refines the selection by adding corrective sketches.
Authors take a divide and conquer approach at smart-segmentation.

General Architecture (Overview):

1. Entire Image is coarsely segmented as a preprocessing step.

2. User provides an input sketch [special set of graphical commands] that indicate which segments the user keeps.

3. Perform finer automatic segmentation on these “interesting” segments.

4. Estimate appropriate alpha channel for the foreground object within the finer segments.

5. Allow the user to refine the sketch and iterate the process again.
Detailed Look at Architecture
Step 1: Clustering — Vector Quantization

- Image is segmented using a binary-split vector quantization algorithm (VQ)
- VQ is a lossy data compression method
- Used to quantize a large set of sample vectors into a more tractable set.
- It’s a top-down clustering method (similar to k-means clustering.)
Step 1: Clustering — Vector Quantization

- **VQ input:**
  - a set $T$ of $k$ dimensional feature vectors $T = \{x_1, \ldots, x_n\}$
  - The number $N$ of partitions

- VQ partitions the space $T$ into $N$ parts such that a distance measure $D(\cdot)$ is minimized.

- $D(\cdot)$ is a measure between the sample and the partition centroid.
Step 1: Clustering — Vector Quantization

- VQ Algorithm:
  - Start with one cluster encompassing all data.
  - Cleave cluster into 2 and perform 2-means clustering
  - Cleave clusters into 4 and perform 4-means clustering
  - Continue until $N$ clusters
Step 2: Sketch Processing

- Use sketch elements to mark centroids as either background or foreground.
- 3 sketch elements:
  - Points
  - Lines
  - Regions:
    - Closed-Regions
    - Include-Regions
    - Exclude-Regions
Step 2 : Sketch Processing

- A point labels the segment that it falls on as foreground.
Step 2: Sketch Processing

- Use a line to identify foreground boundaries as follows:
  - First perform a triangulation of the segment centroids.
Step 2: Sketch Processing

- Then classify each vertex/centroid based on what side of the line it's on.
Step 2 : Sketch Processing

- Processing Regions:
  - Let $A_1 \ldots A_N$ be all of our segments.
  - If $S$ is sketch region, then $S \cap A_i$ is the set of pixels both in the sketch region and in the segment.
  - The segment $A_i$ is in the foreground if the majority of its pixels overlap in $S$ that is:
    $$\frac{|S \cap A_i|}{|A_i|} > 0.5$$
Step 3: Object Extraction

- By now we have labeled each segment as either foreground or background.
- We have a triangulation of all segment centroids.
- We try to approximate the object boundary by partitioning the problem as follows:
Step 3: Object Extraction

- Identify all triangles with vertices from both background and foreground.
- These “boundary triangles” define the border between the object and its background.
- Process only boundary triangles.
Step 3: Object Extraction — Acting on Boundary Triangles

- Label each pixel in the triangle according to the coarse segmentation
- For each pixel form a 3-tuple of position and class [f.g. or b.g.] as \((x, y, c)\)
- Compute Fisher’s Linear Discriminant. This gives us a direction of the best boundary separating the classes
Step 3: Object Extraction — Acting on Boundary Triangles

• We let the physical boundary pass through the point:

\[
\frac{n_b}{n_f + n_b} \mu_f + \frac{n_f}{n_f + n_b} \mu_b
\]

f.g. centroid \quad b.g. centroid
Step 3: Object Extraction — Acting on Boundary Triangles

- Perform one more round of segmentation on the boundary triangle.
- Classify the new segments based on the linear discriminant.
Step 4: Alpha Estimation

- With fine grained segments in place we estimate the alpha channel for the object boundary using Ruzon and Tomasi’s algorithm:
Ruzon and Tomasi’s Algorithm

• As you go from background to foreground the PDF of one region smoothly morphs into the PDF of the other.
Ruzon and Tomasi’s Algorithm

• If you want to find the alpha-value of some pixel $Q=(r,g,b,x,y)$ you find that transitional-PDF (characterized by $t$ in $[0,1]$) that maximizes the probability of $Q$. 
Summary of the Steps

• Coarsely segment the image using VQ
• User specifies segments of interesting through an input sketch
• Triangulate the segment centroids and find boundary triangles which define the object’s border
• Segment boundary triangles once more and classify them by the Linear Discriminant. 
• Send these segments into alpha-channel estimator.
• Combine segmented alpha-masks into final selection.
Results
Comparison With other Selection Methods
Critique—One key criterion is minimal user interaction.

- Two user inputs:
  - Segmentation stop-point, \( N \), to VQ
  - User sketches

- Some of the utility of this method comes from having a nice coarse clustering (e.g. using points)
  - May run the risk of users needing to fiddle with the parameter \( N \), which would increase user interaction

- User must be aware of segments.
  - Not as nice having the user select the object with no knowledge of the underlying architecture.

- User must learn a sketch vocabulary
  - Better not to have the user deal with point, line, and region operation primitives.
  - Use of only inclusion/exclusion selection primitives would be preferred since they already exist in selection paradigms.
Comparing GrabCut with “Freehand Sketches”

Freehand Sketch

GrabCut