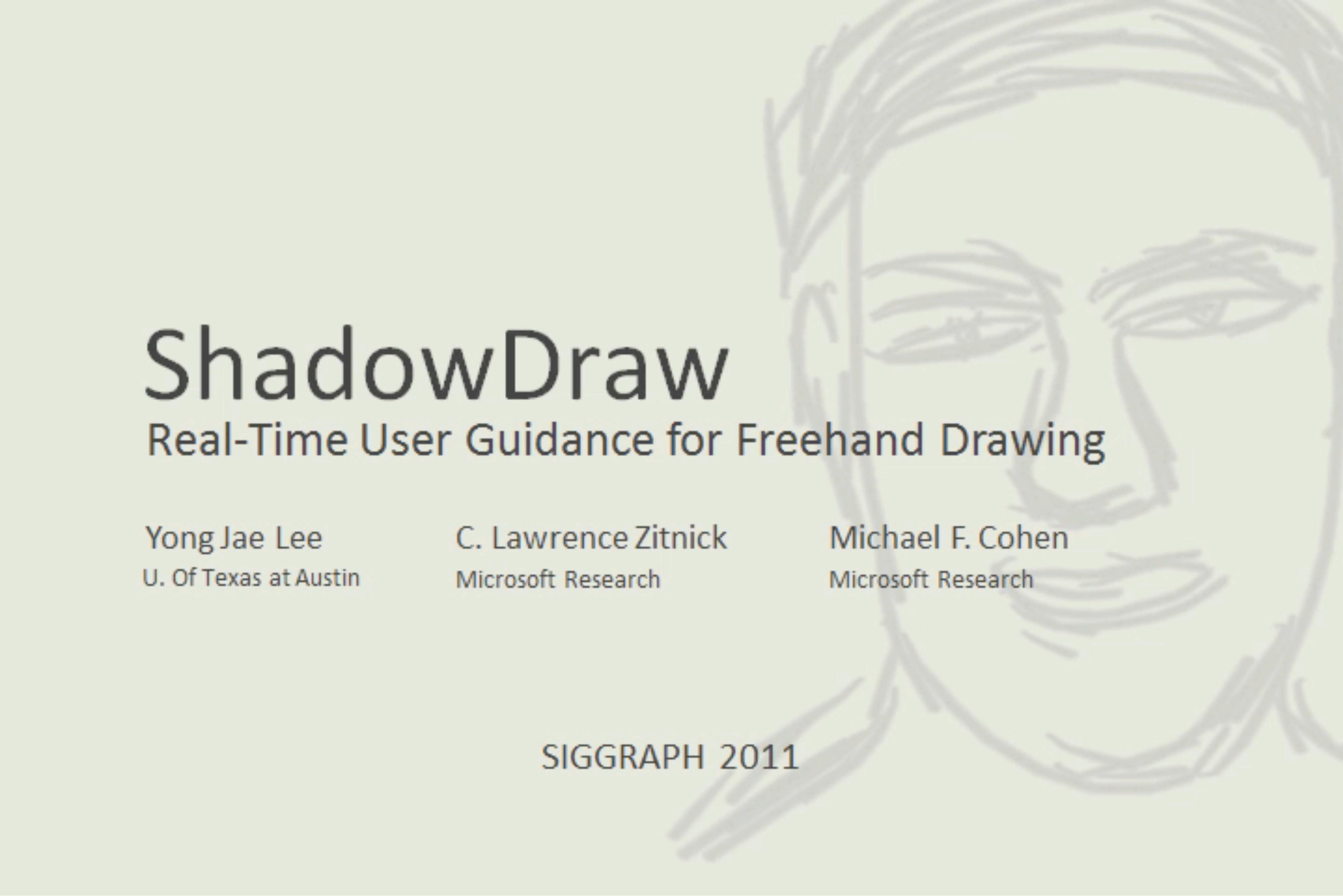
ShadowDraw

Real-Time User Guidance for Freehand Drawing

Harshal Priyadarshi

Demo


ShadowDraw
Real-Time User Guidance for Freehand Drawing
Yong Jae Lee
U. Of Texas at Austin
C. Lawrence Zitnick
Microsoft Research
Michael F. Cohen
Microsoft Research
SIGGRAPH 2011

Components of Shadow-Draw

- **Inverted File Structure for indexing**
  - **Database of images**
  - **Corresponding Edge maps**

- Query method
  - Dynamically retrieves matching images
  - Aligns them to evolving drawing
  - Weighs them based on matching score, to form shadow

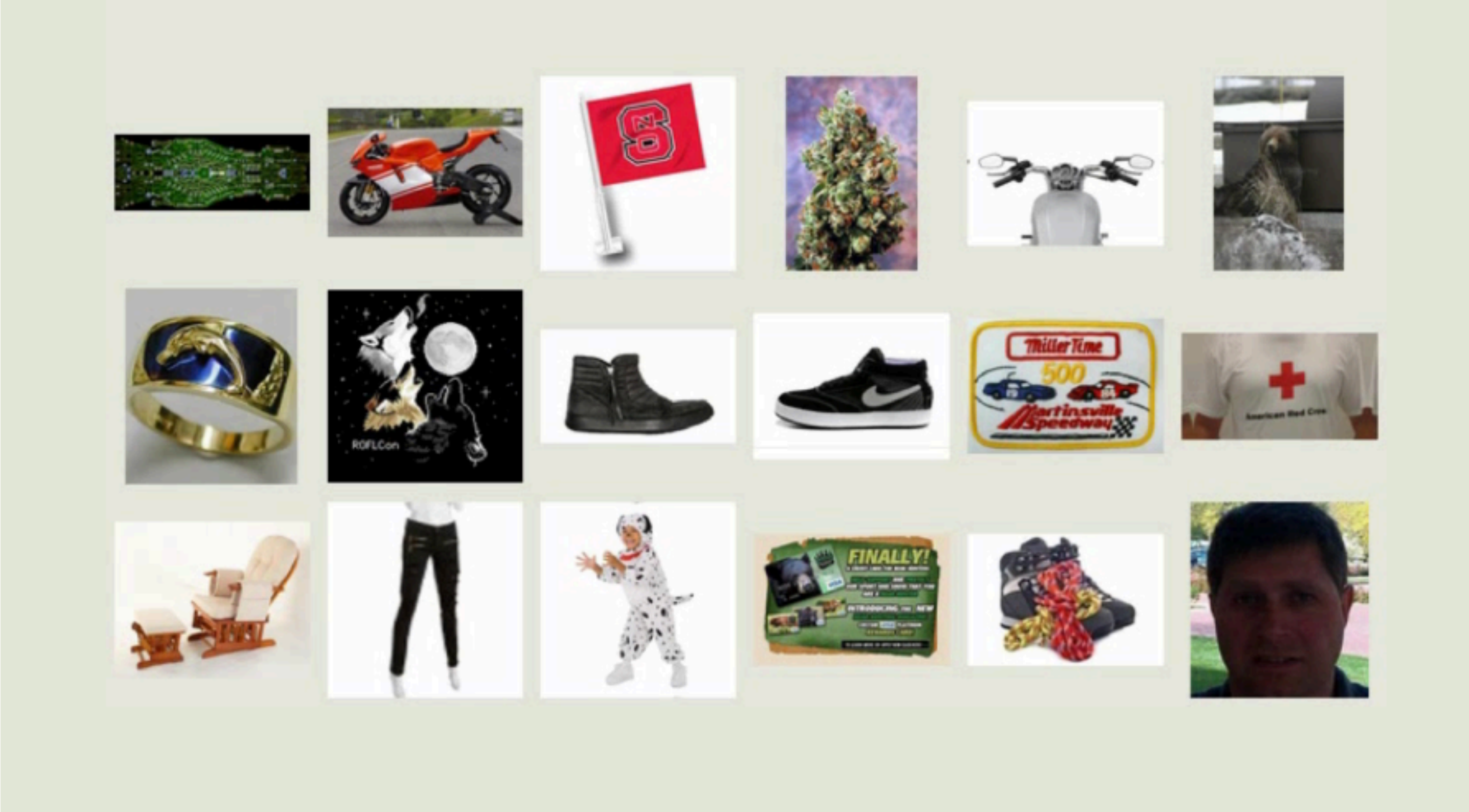
- UI
  - Displays a shadow of weighted edge maps beneath the user’s drawing

Database Creation

- **Image Acquisition**
- Edge Extraction
- Patch Description
- Min-hash Encoding

Image Acquisition

- 30k images spanning 40 categories obtained from internet
- Scaled to obtain 300 x 300 images



Problems Handled:
1. Direct Sketch images are not abundant
2. Diverse Background – Still has good edges you might want to draw
3. Photographer bias to rescue

Database Creation

- Image Acquisition
- **Edge Extraction**
- Patch Description
- Min-hash Encoding

Edge Extraction (Step 1)

- Compute the local edge magnitude (p_m) and orientation (p_o)at each pixel using steerable filters

Input

Output

Edge Extraction (Step 2)

- Normalize the edge magnitude
- Need to detect long, coherent edges even when faint (i.e., not simply edges with strong magnitude)

Input

$$\hat{p}_m = \frac{p_m - \mu_w}{\sigma_w + \epsilon},$$

Output

Edge Extraction (Step 3)

- Message Passing for length estimation

Input

$$m_0^t(p) = \sum_q w_\alpha(q) \cdot w_\theta(q) \cdot (\hat{q}_m + m_0^{t-1}(q)),$$

$$m_1^t(p) = \sum_q w_\alpha(q) \cdot w_\theta(q) \cdot (\hat{q}_m + m_1^{t-1}(q)).$$

$$w_\theta(q) = exp(-(p_\theta - q_\theta)^2 / 2\sigma_\theta^2),$$

$$e^l(p) = m_0^t(p) + m_1^t(p) + \hat{p}_m$$

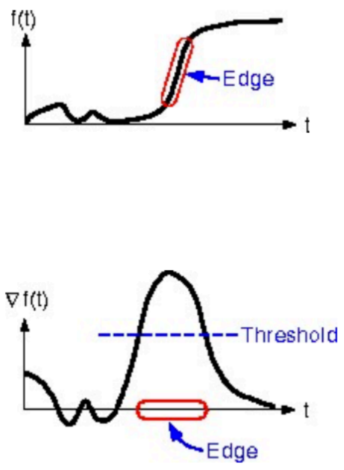
$$e^o(p) = p_o$$

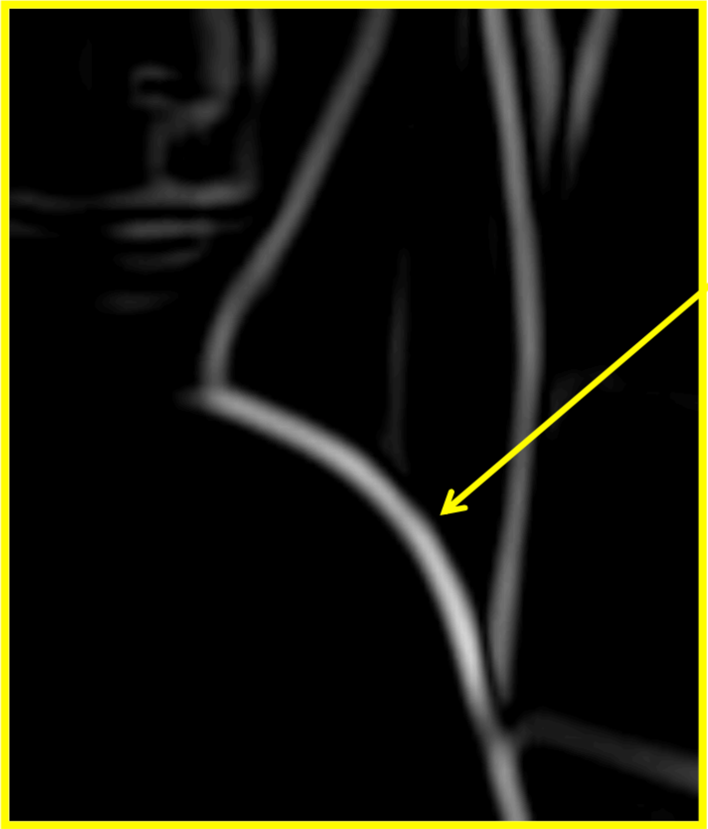
Output

See the difference

Canny Edge Detection (Non Max Suppression)

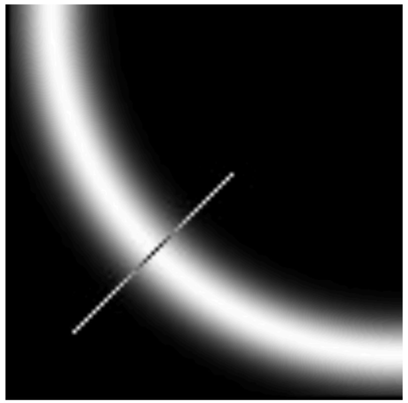
The Canny edge detector

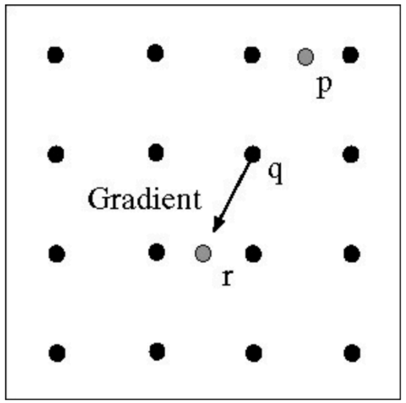




How to turn these thick regions of the gradient into curves?

Non-maximum suppression





Check if pixel is local maximum along gradient direction, select single max across width of the edge

Credits - Kristen Grauman (both images)

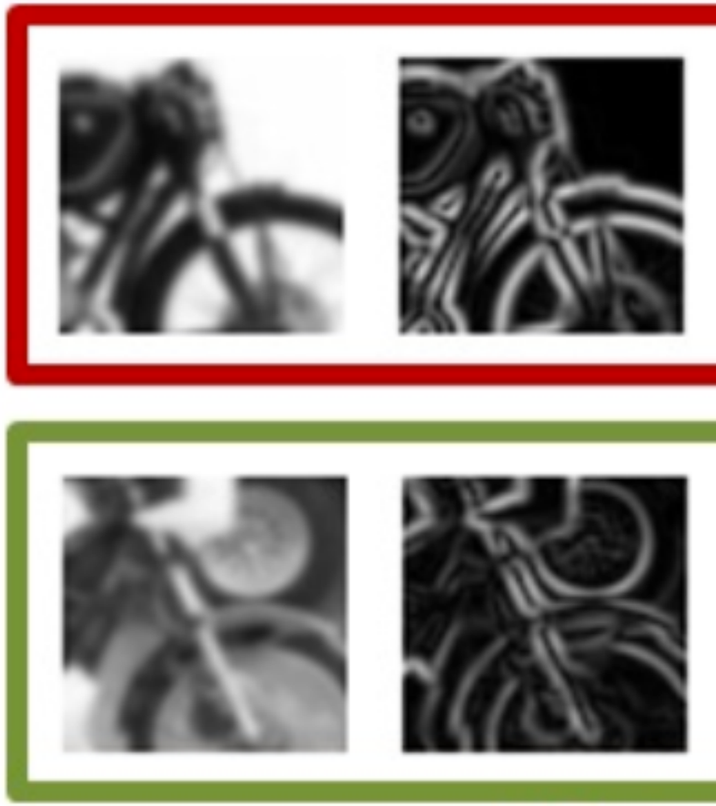
Database Creation

- Image Acquisition
- Edge Extraction
- **Patch Description**
- Min-hash Encoding

Patch Descriptors

- Low Dimensional BiCE descriptor
  - Encodes a histogram of edge positions and orientations
- Done over overlapping 60 x 60 patches with 50 % overlap
  - As mapping an edge image E to incomplete and evolving drawing.
- SIFT / Daisy vs BiCE
  - Former relies on relative strength of edge magnitudes to provide discriminability
  - Thus reduced performance compared to BiCE on our task, where edge strengths are not important.

BiCE descriptor (Steps)



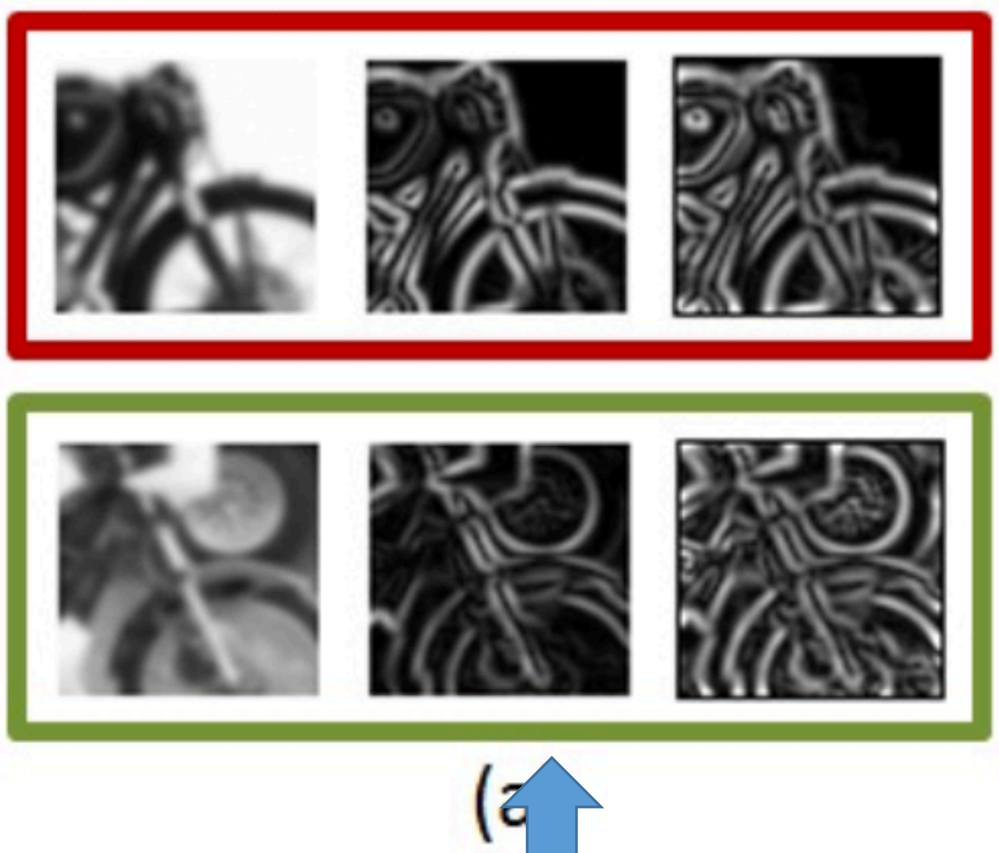
1. Local Normalization of image patch gradients
   - Remove variation in relative gradient magnitudes
2. Binning of normalized gradients
   - using **position, orientation, and local linear length** of image
3. Binarization of normalized gradient histogram
   - Encodes the presence of edges

Presence / absence of Edge is preserved across matching patches,

There magnitude might not.

Step 1

Normalized Gradient (g_cap)



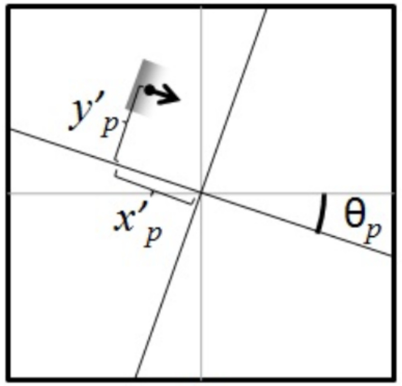
Original Gradient (g)

$$g_p = \|[f_x(p)\ f_y(p)]^{\mathrm{T}}\|_2.$$

$$\bar{g}_p = \sum_{q \in N} g_q \mathcal{N}(q; p, \sigma_s),$$

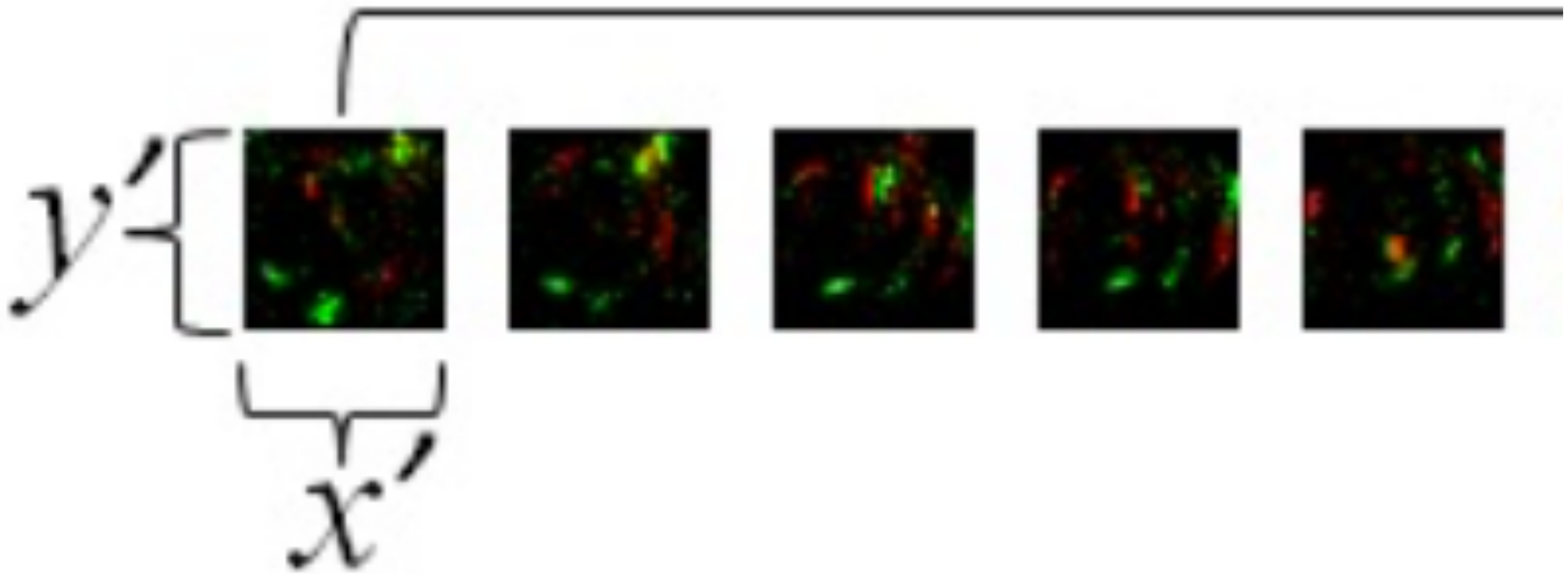
$$\hat{g}_p = \frac{g_p}{\max(\bar{g}_p, \epsilon)},$$

Step 2



$$\begin{bmatrix} x'_p \\ y'_p \end{bmatrix} = \mathbf{R}(\theta_p) \begin{bmatrix} x_p \\ y_p \end{bmatrix},$$

Orientation aligned binning
Robustness to orientation changes



Initial Binning w.r.t. just position and orientation.

Increasing Discriminability
Long Coherent Edges vs Shorter Textured Edges

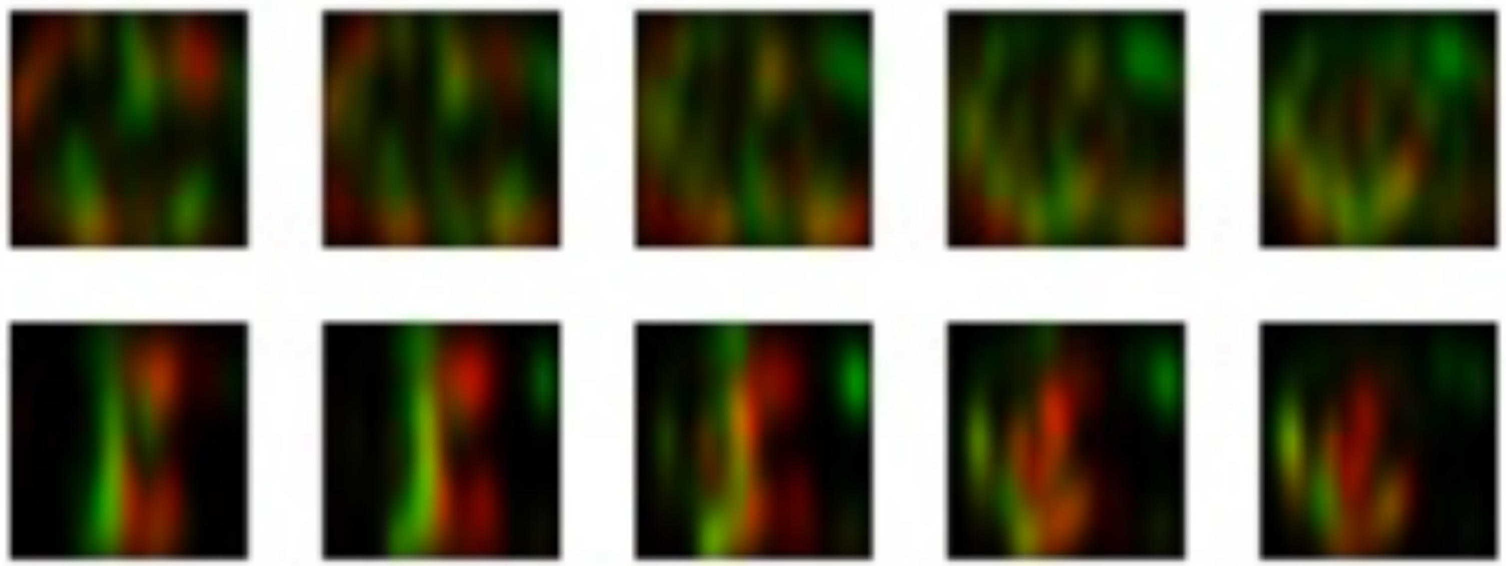
Calculate Edge Length

$$L(x', \theta) = \sum_{y'} H(x', y', \theta).$$

Discretization into 2 bins by weight based normalized gradient splitting

$$\Delta(l_p) = \max(0, \min(1, \frac{l_p - \alpha}{\beta})).$$

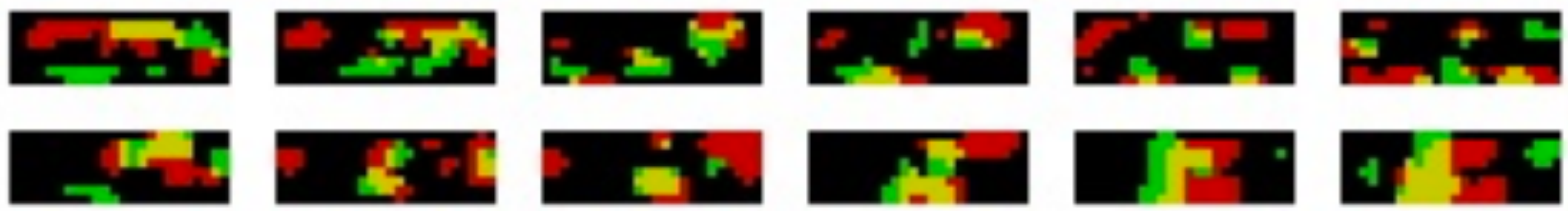
Alpha, beta → Tunable thresholding params



With gaussian Blur along x,y, theta dimension

WHY ??

Step 3 (Subsampling and Binarization)



- Subsample to discrete set of values for x, y, theta and length

- Value = 1 (top T percent of bins with highest frequency)
- Value = 0 (rest)

- **Flatten to get the BiCE desciptor**

Database Creation

- Image Acquisition
- Edge Extraction
- Patch Description
- **Min-hash Encoding**

Retrieval and Clustering Efficiency

- Reduce dimension → Improve Clustering → Improve Retrieval

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Preserves Maximum Jaccard Similarity

Minhashing (I apologize for my terrible animation)

What are these vectors ?

Index	Vector A	Vector B
1	1	0
2	1	0
3	0	1
4	0	0
5	1	0
6	0	0

H(Index)
4
6
1
5
3
2

Index	Vector A	Vector B
1	5	3

Sketches (n, k) and Inverted Indexing

- To increase precision

Index	Vector A	Vector B
1	5	3
2	2	1
3	1	1
4	4	6
5	3	2
6	1	6

K hash functions

- To increase recall

N sets of
K hash functions

Index	Vector A	Vector B
1	5	3
2	2	1
3	1	1
4	4	6
5	3	2
6	1	6

Components of Shadow-Draw

- Inverted File Structure for indexing
  - Database of images
  - Corresponding Edge maps

- **Query method**
  - **Dynamically retrieves matching images**
  - **Aligns them to evolving drawing**
  - **Weighs them based on matching score, to form shadow**

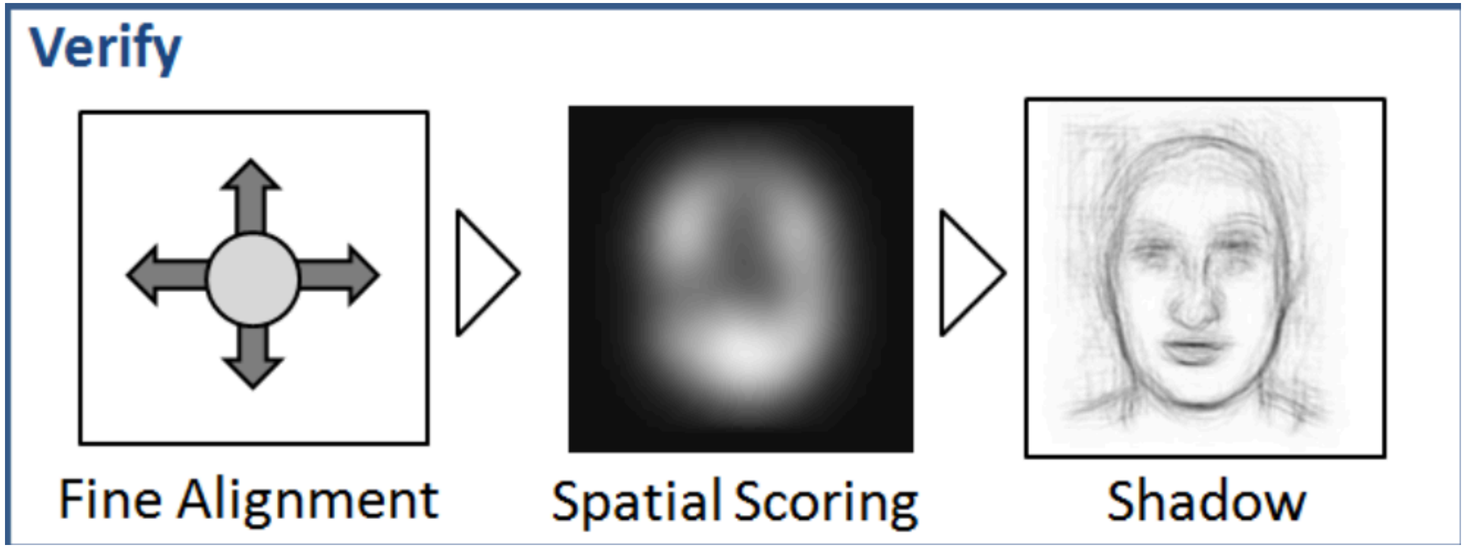
- UI
  - Displays a shadow of weighted edge maps beneath the user’s drawing

Query Steps

- ***Dynamically Retrieving*** **Matching Images**
- *Aligning* Matching Images to Drawn Sketch
- *Shadow Creation* by Weighting

Image Matching

- Obtain Candidate Matches
- Align Candidate Matches with the partially drawn sketch
- Assign weight to each candidate's edge image
- Construct Shadow Image



Candidate Match Finding

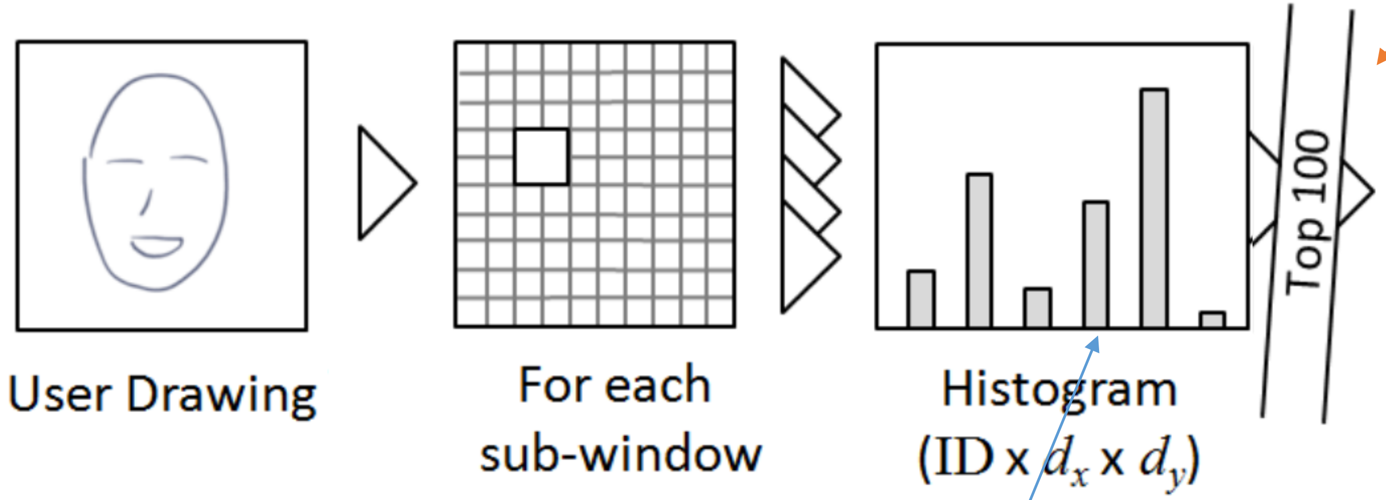
Resultant Candidate Format

(Image Id , patch offset-x direction, patch offset-y direction)

Top 100 images and corresponding offset for the highly voted offset

Note: Not 100 best patches

BiCE sketch descriptor for each patch



Edge Image of Sketch

Vote count for patch for each candidate image

X offset of the candidate patch from the user sketch patch

Query Steps

- *Dynamically Retrieving* Matching Images
- ***Aligning* Matching Images to Drawn Sketch**
- *Shadow Creation* by Weighting

Aligning Candidate Matches

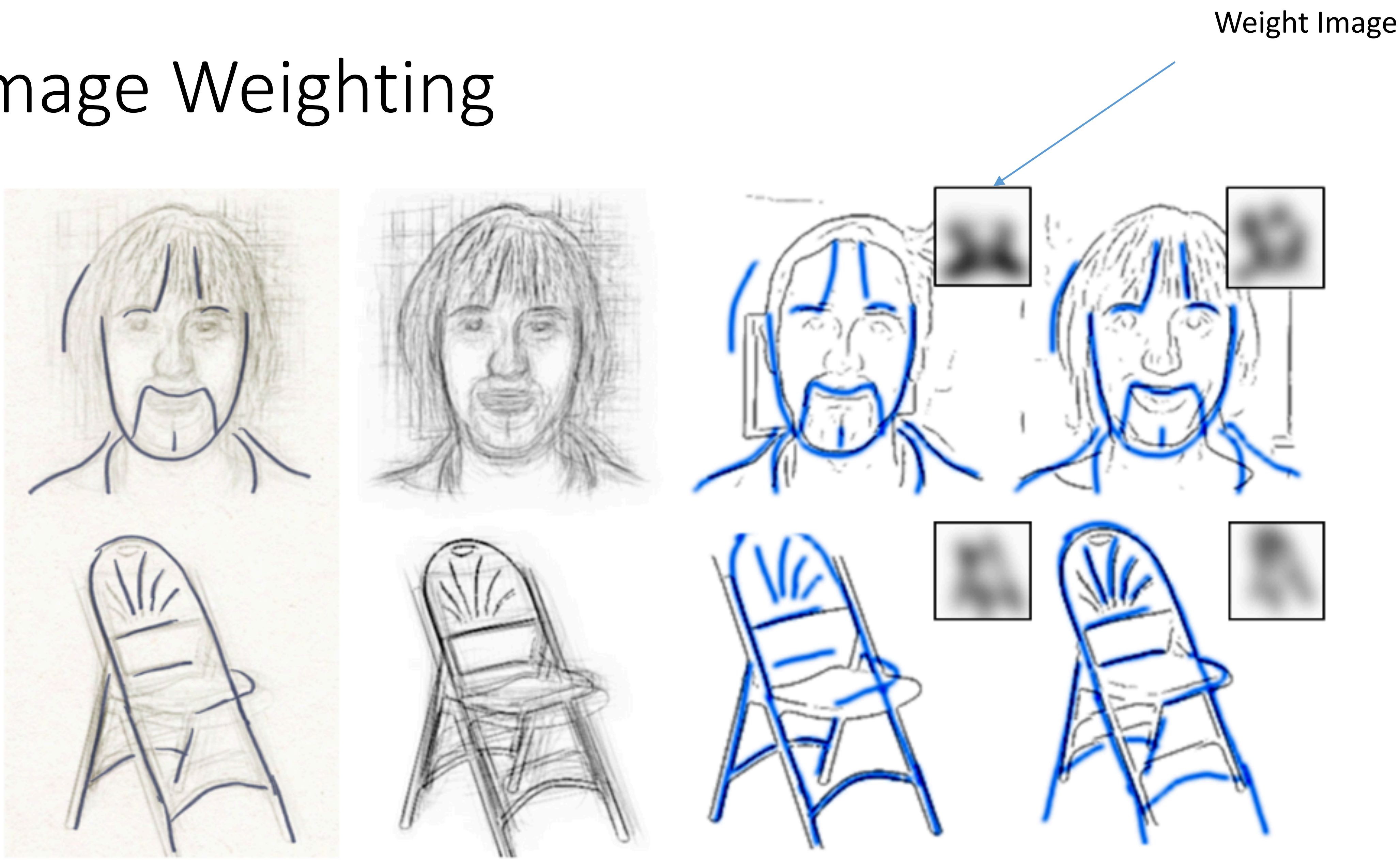
$$T_x(x) = \sum_p \sin(\tilde{\theta}(p_x, p_y))\tilde{E}(p_x, p_y)$$
$$\sin(\theta(p_x + d_x + x, p_y + d_y))E(p_x + d_x + x, p_y + d_y)$$

$$T_y(y) = \sum_p \cos(\tilde{\theta}(p_x, p_y))\tilde{E}(p_x, p_y)$$
$$\cos(\theta(p_x + d_x + x, p_y + d_y))E(p_x + d_x + x, p_y + d_y)$$

Query Steps

- *Dynamically Retrieving* Matching Images
- *Aligning* Matching Images to Drawn Sketch
- ***Shadow Creation* by Weighting**

Image Weighting



Shadow Image Construction

Blending Weight Image

Shadow Image

$$S = \sum_i W_i E'_i,$$

Edge Candidate Image

Global matching term

Spatially varying
match term

$$W_i = \alpha \frac{v_i V_i}{\sum_j v_j V_j}.$$

3 variables:

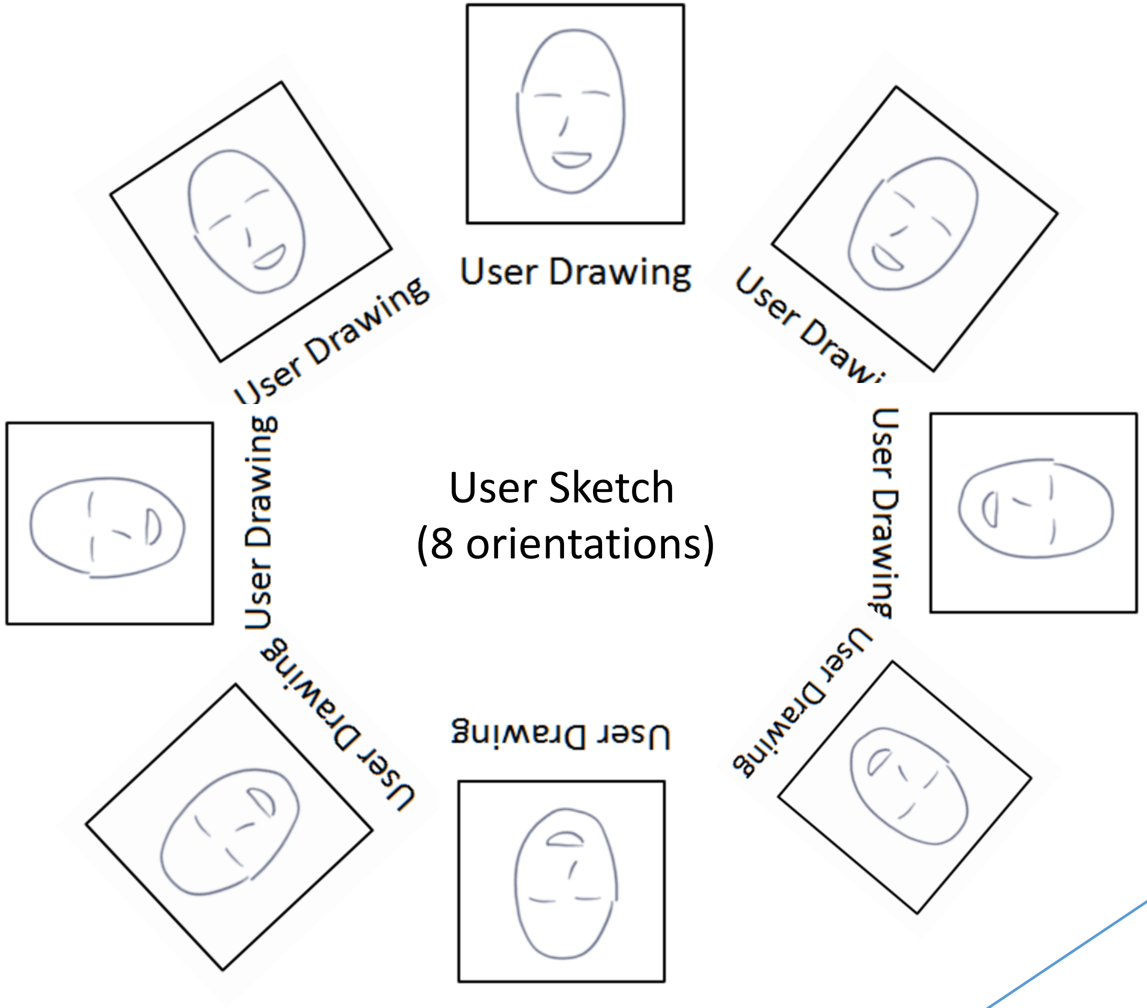
- Global matching term (v)
- Spatially varying matching term (V)
- Visibility Enhancer (alpha)

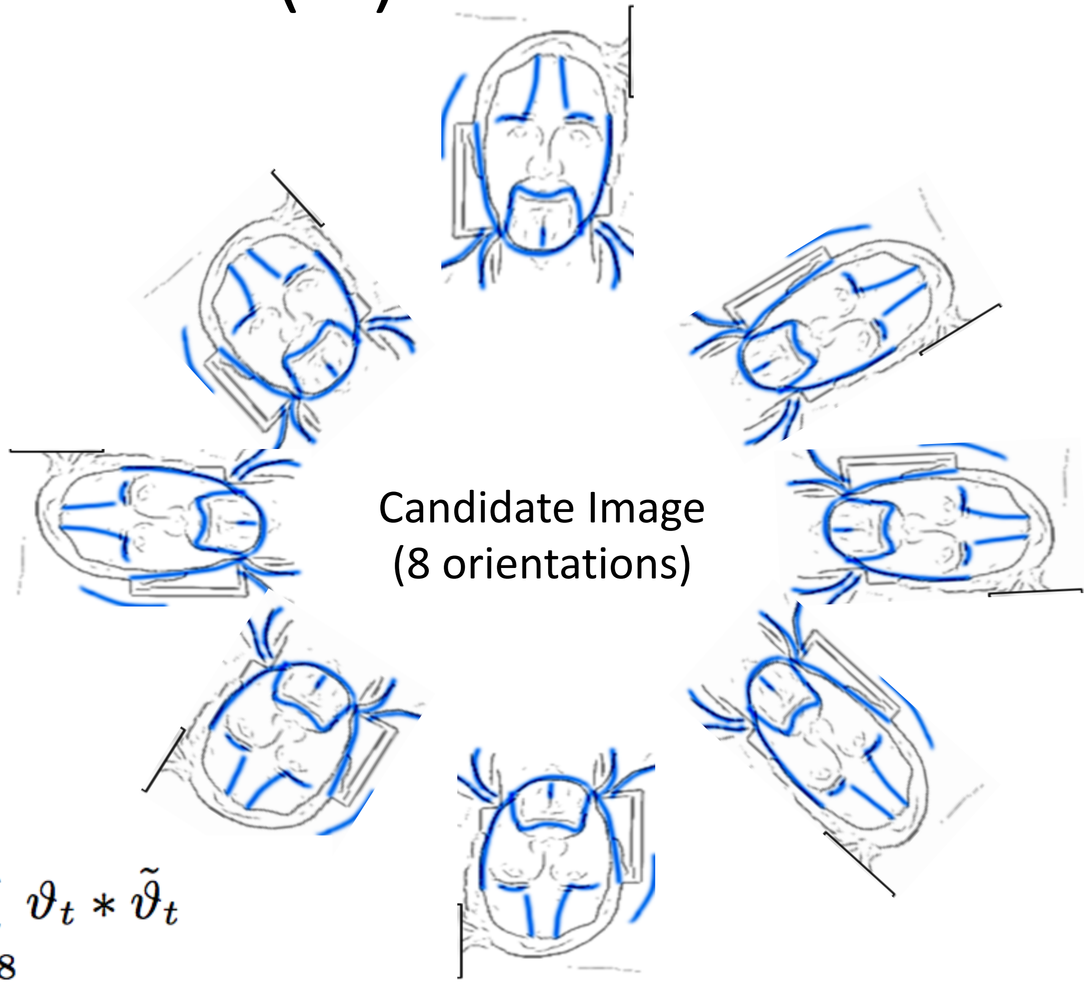
Visibility Enhancer

Normalization Term

Image not oriented, the edges it captures are oriented

Obtaining Global Matching Term (V)





$$\vartheta^{+} = \sum_{t=1,8} \vartheta_t * \tilde{\vartheta}_t$$

$$\vartheta^{-} = \sum_{t=1,8} \vartheta_t * \tilde{\vartheta}_{(t+4)\%8}$$

Negative Correlation

Positive Correlation

Global Matching

Spatial Matching

$$V_i = \mathcal{G}(\vartheta_i^+, 4\varphi)$$

Gaussian Blur on the positive correlation image

$$h_i = \sum_p \vartheta_i^+(p) - \vartheta_i^-(p)$$

Average of 5 highest h_i from the candidate set

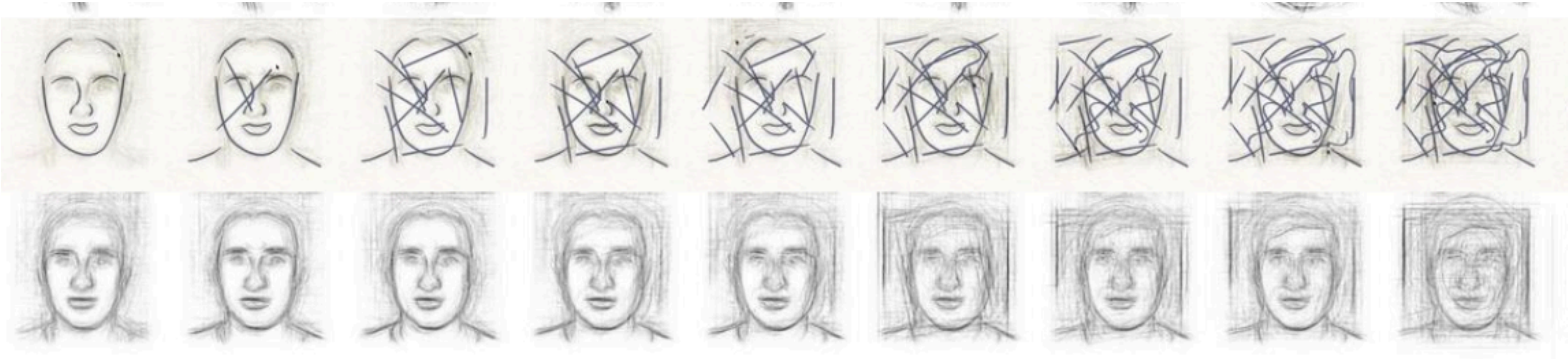
$$v_i = \max\left(0, \left(\frac{h_i - \gamma h^*}{h^* - \gamma h^*}\right)^\kappa\right).$$

Visibility Enhancer (alpha)

$$\alpha = \frac{\sum_i v_i}{\epsilon + \sum_i v_i}$$

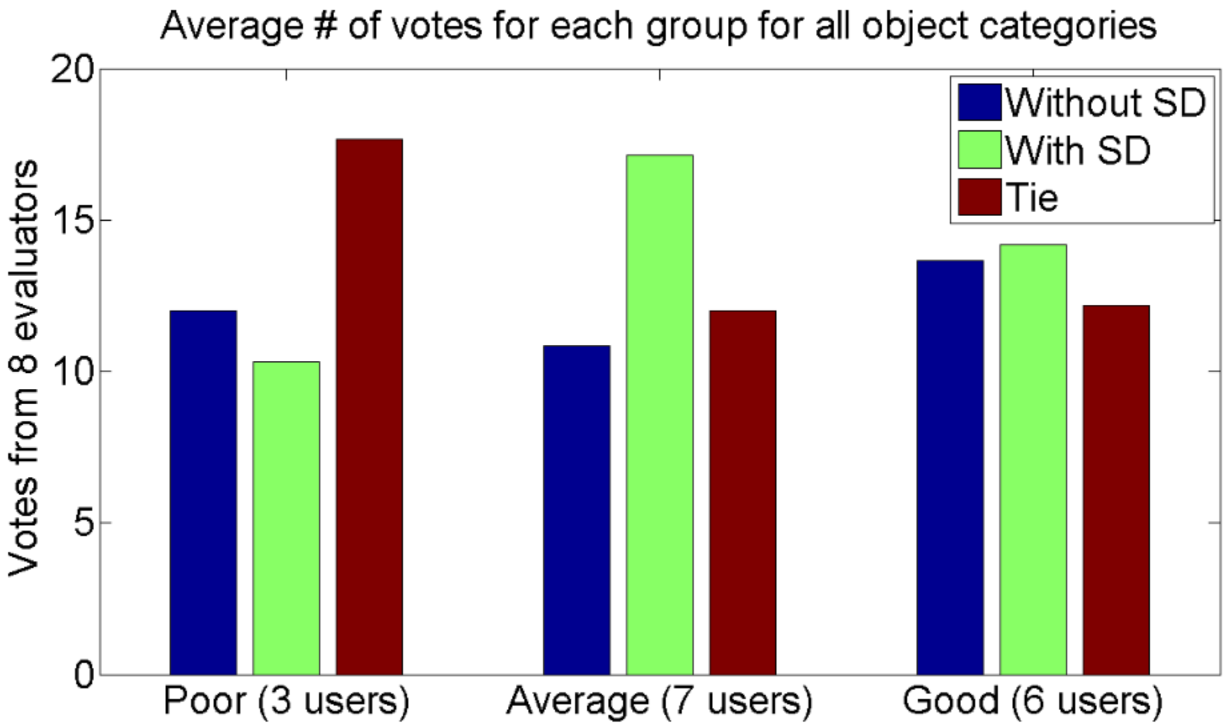
Why is it a visibility enhancer ?

Experimental Findings



Robustness to ________ ????

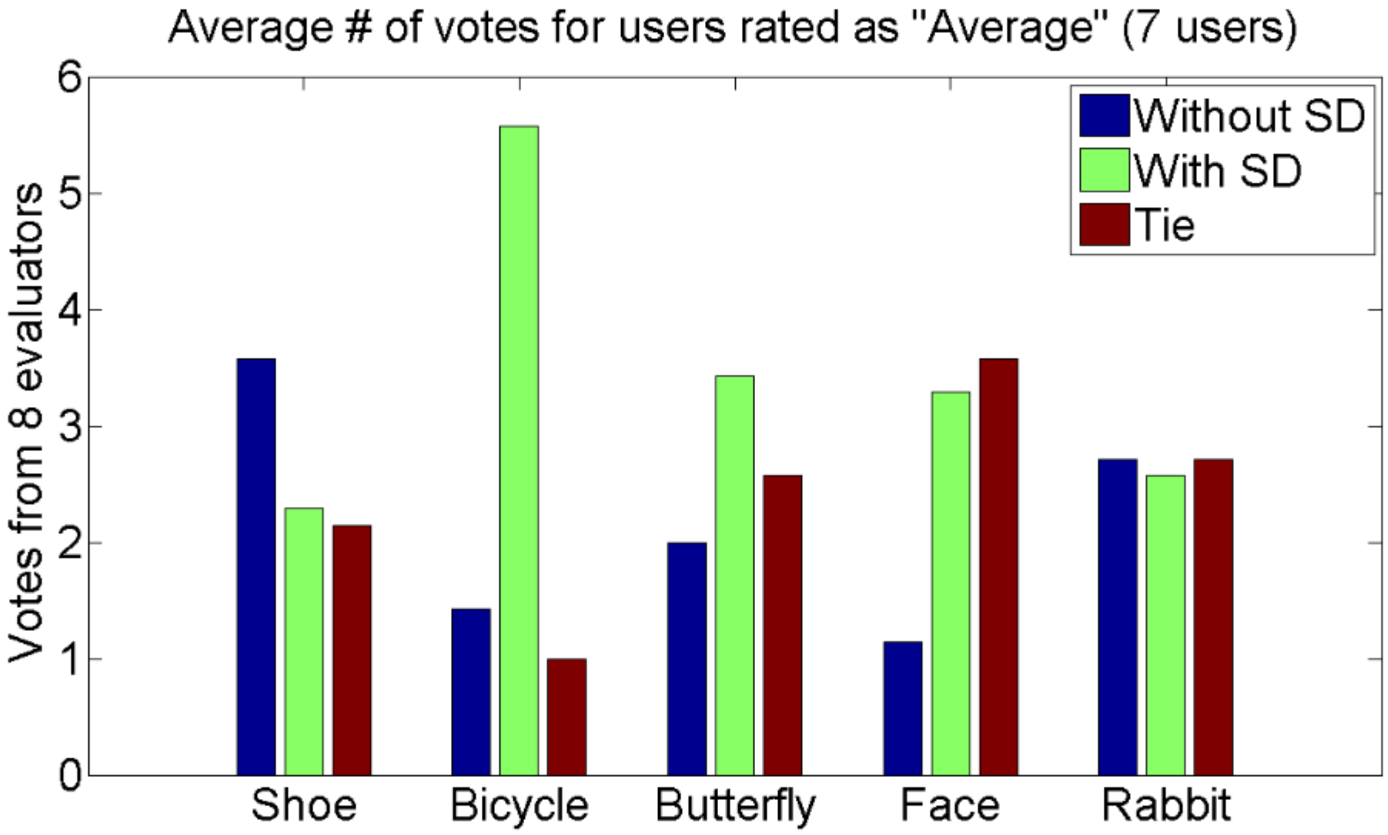
Poor vs Average vs Good



WHY ??

WHY ??

Complexity



Average # of votes for users rated as "Average" (7 users)
Votes from 8 evaluators
0
1
2
3
4
5
6
Without SD
With SD
Tie
Shoe
Bicycle
Butterfly
Face
Rabbit

- Strengths
  - Can help drawing structurally complex objects
  - Helps preserve the unique style of the users
  - Is a real-time algorithm

- Weaknesses
  - Leads to good shadows only if the initial user sketch is not all over the place. Otherwise might confuse the user.
  - A tussle between **guidance and freedom**.
  - **Sketching flow bias** – The way we start drawing the sketch might affect the shadow retrieved, and thus lead to confusion initially, if the user is not very certain of each detail.

References

- Original Paper
  - http://vision.cs.utexas.edu/projects/shadowdraw/ShadowDrawSiggraph11.pdf
- Supporting Papers
  - Long Edge Detector
    - http://grail.cs.washington.edu/projects/gradientshop/demos/gs_paper_TOG_2009.pdf
  - BiCE descriptor
    - http://larryzitnick.org/publication/BiCE_ECCV10.pdf

Thank You