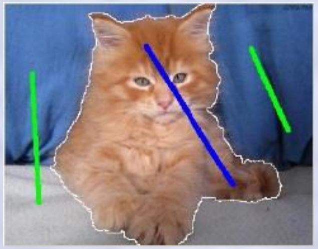
Implementation of Video Colorization and Matting [1]

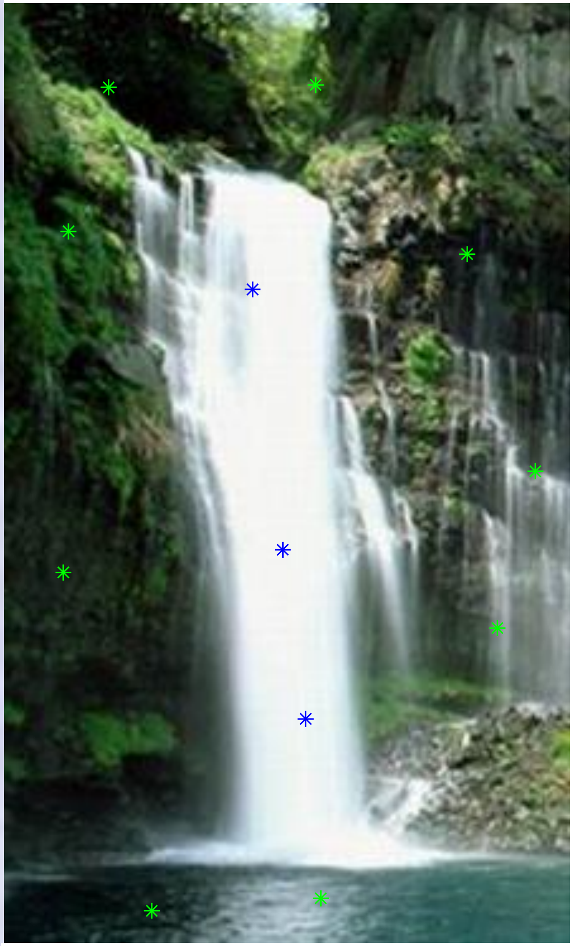
Leonid Bilevich
Alex Bronstein
Shai Avidan

"Scribble points"



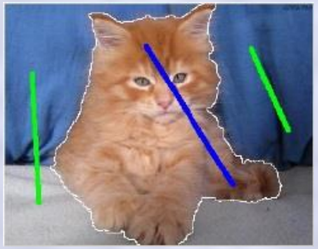
[1]

- Define "scribble points" for foreground and background
- Matlab: getpts



Original image with scribble points

RGB→HSV

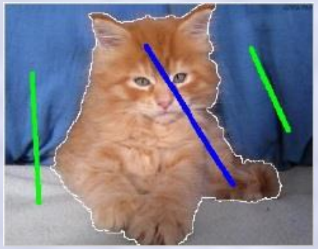


[1]

- **We'll segment the V layer**

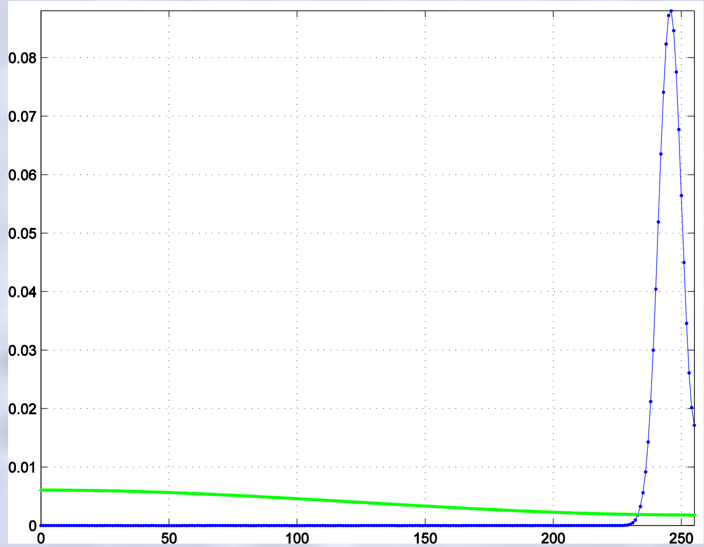
Original image (V layer)

Foreground/Background likelihood



[1]

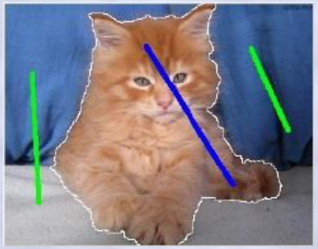
- **Apply KDE program [2] to “scribble points” and compute the histograms for Foreground and Background**



- **Compute likelihood of pixel to belong to Foreground/ Background:**

$$P_{\mathcal{F}}(\vec{c}_x) = \frac{Pr(\vec{c}_x|\mathcal{F})}{Pr(\vec{c}_x|\mathcal{F}) + Pr(\vec{c}_x|\mathcal{B})}$$

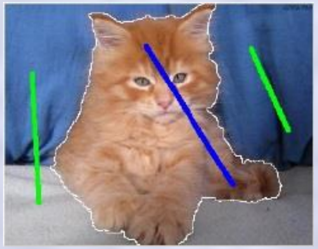
Histogram back-projection


[1]

- **Get the map of Foreground probabilities**
  $P_F(x) = P_F(\vec{c}_x)$


Original image (V layer)

Histogram back-projection



[1]

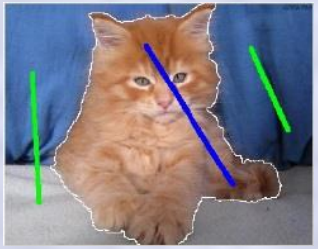
- **Get the map of Foreground/Background probabilities**

$P_F(x) = P_F(\vec{c}_x)$,
$P_B(x) = P_B(\vec{c}_x)$

$P_B(x)$

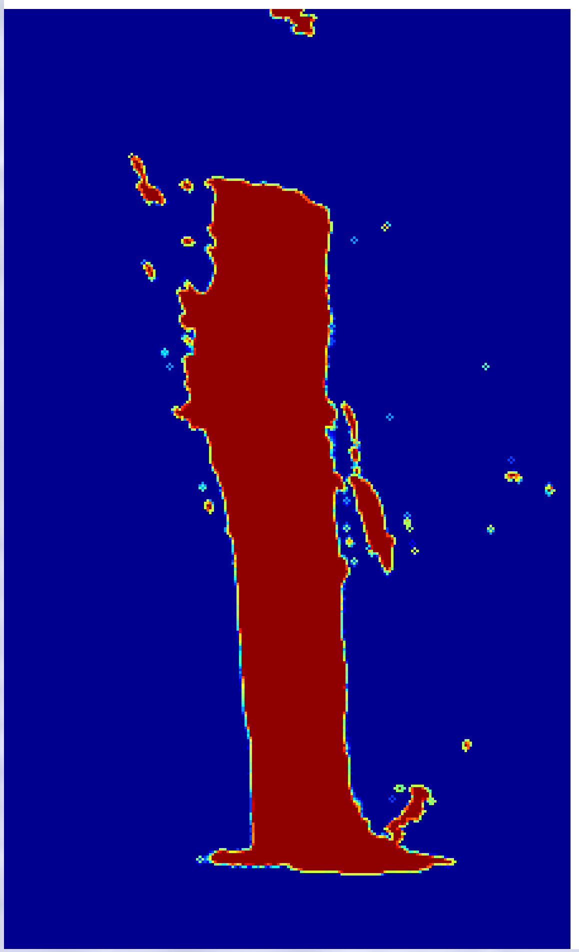
$P_F(x)$

Discrete Weighted Geodesic Distance



[1]

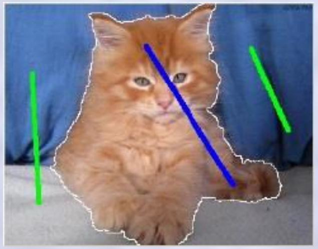
- **Solve with Matlab's 'graydist' that is applied to**
  $|\nabla P_F(x)|$,
  $|\nabla P_B(x)|$.
  Use "scribble points" as seed locations in 'graydist'.



$D_B(x)$

$D_F(x)$

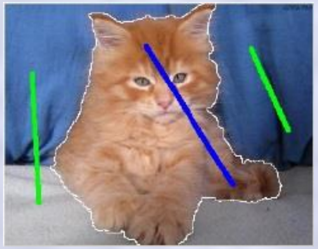
Boundary


[1]

- **Foreground:** $V_F = \{x: (D_F - D_B) \leq 0\}$
- **Boundary:** $\Delta = \{x: \mathrm{bwperim}(V_F) = 1\}$

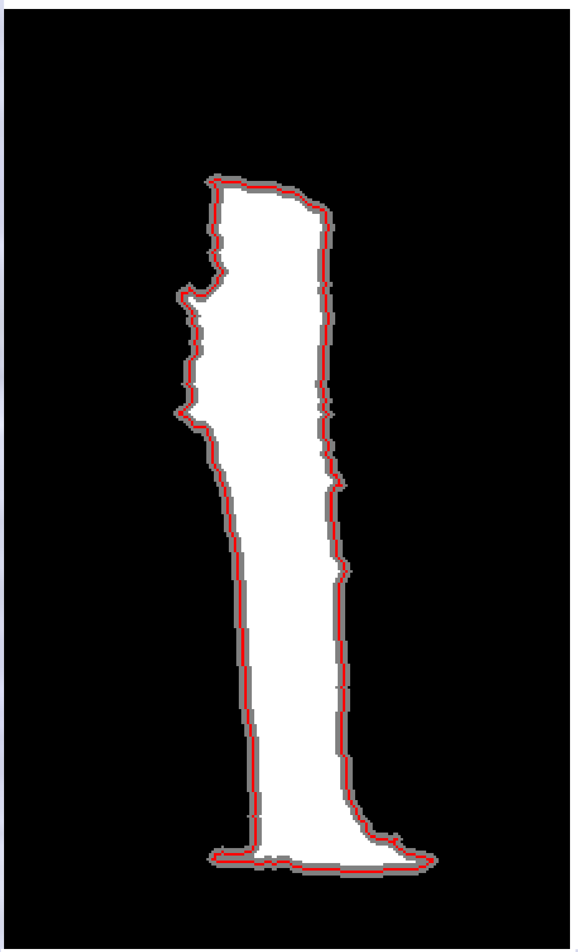

Boundary on Original image

Trimap

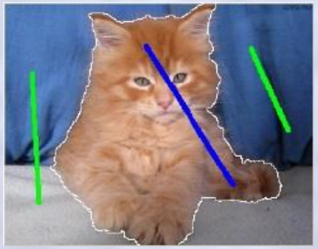


[1]

- **Narrow band:** $B_\rho(\Delta) = \{x: d(x, \Delta) \leq \rho\}$
- **Use dilation on the Boundary with disk of radius ρ**



Trimap and Boundary



[1]

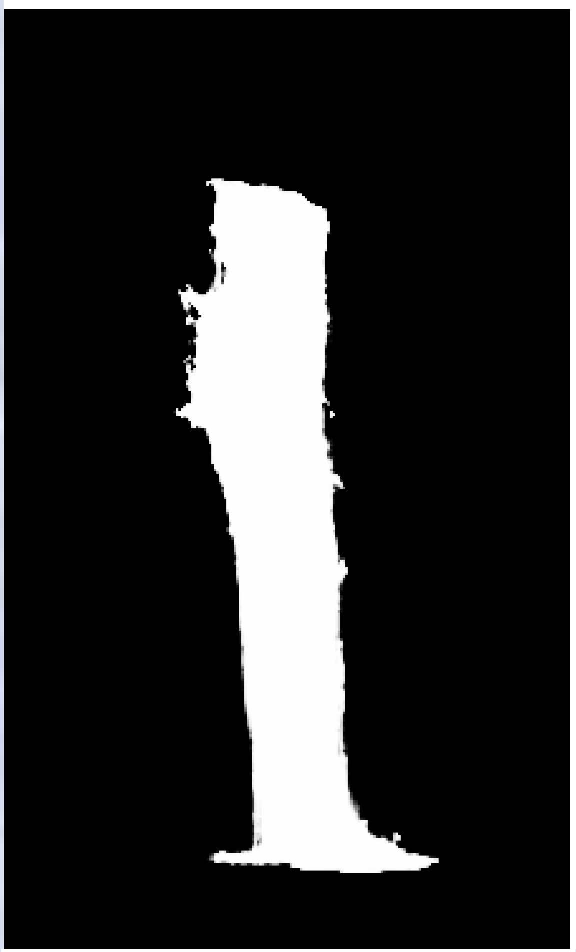
Alpha

- **Geodesic distance weighted with likelihood**

$$\omega_l(x) = D_l(x)^{-r} \cdot P_l(x), \quad l \in \{\mathcal{F}, \mathcal{B}\},$$

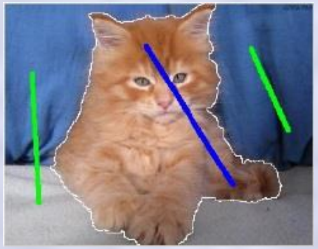
$$\alpha(x) = \frac{\omega_{\mathcal{F}}(x)}{\omega_{\mathcal{F}}(x) + \omega_{\mathcal{B}}(x)},$$

$0 \leq r \leq 2$



Alpha

Segmented Foreground on New Background



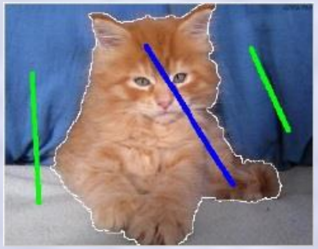
[1]

- **Foreground and Background weighted with Alpha**

$V_x = \alpha_x F_x + (1 - \alpha_x) B_x$

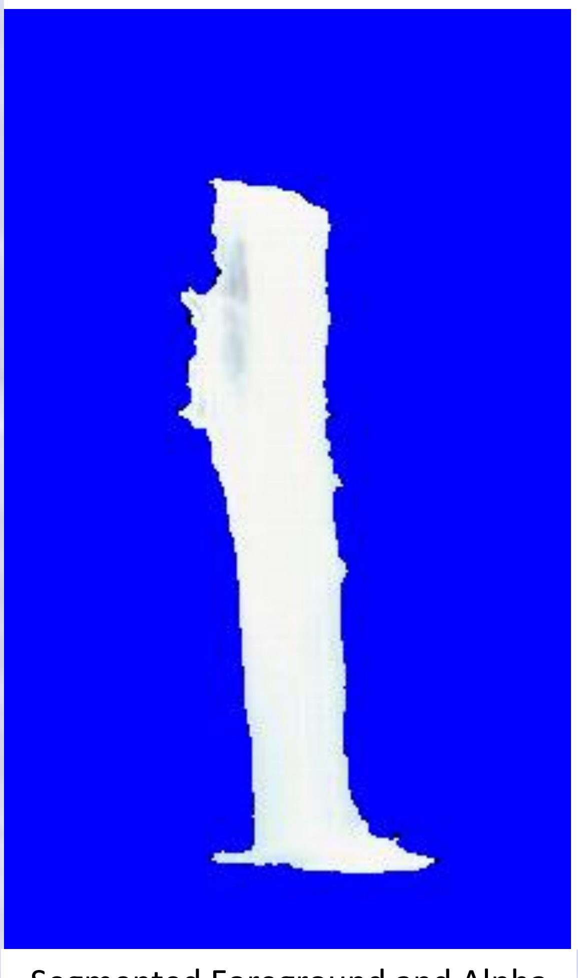
Original Image

Segmented Foreground on New Background

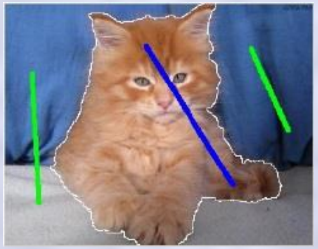

[1]

- **Foreground and Background weighted with Alpha**

$V_x = \alpha_x F_x + (1 - \alpha_x) B_x$


Segmented Foreground and Alpha

Segmented Foreground on New Background

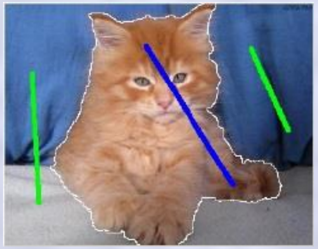


[1]

- **Foreground and Background weighted with Alpha**

$V_x = \alpha_x F_x + (1 - \alpha_x) B_x$

Conclusion



[1]

- **Fast segmentation and matting algorithm**
- **Can be computed in O(N)**
- **Implemented using KDE package [2] and Matlab's 'graydist'**

References

[1] X. Bai and G. Sapiro, Geodesic matting: a Framework for fast interactive image and video segmentation and matting, *Int. J. Comput. Vis.*, 82(2), pp. 113-132, 2009.

[2] Z. Botev, Kernel Density Estimator package. Available at: http://www.mathworks.com/matlabcentral/fileexchange/14034-kernel-density-estimator