Introduction to Video Colorization and Matting [1]

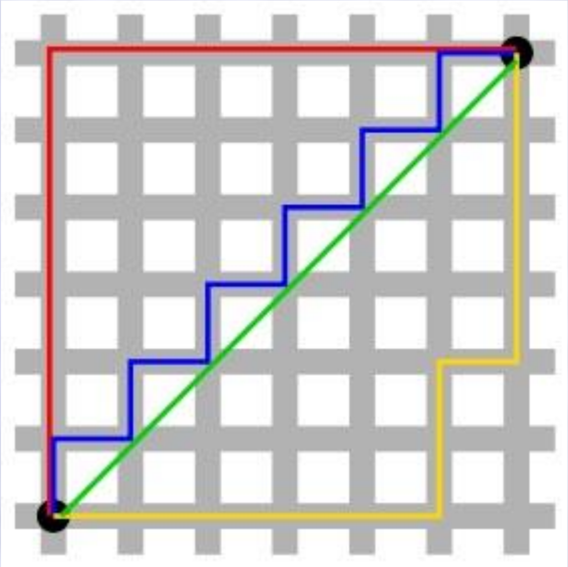
Leonid Bilevich

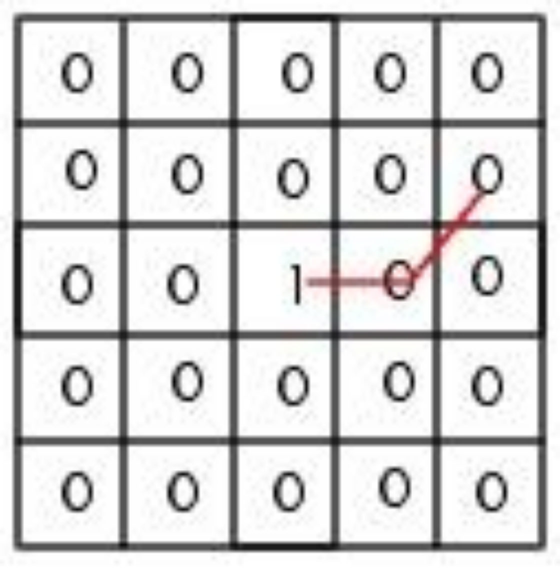
Alex Bronstein

Shai Avidan

Distance

- Euclidean
- Cityblock
- Chessboard
- Quasi-Euclidean

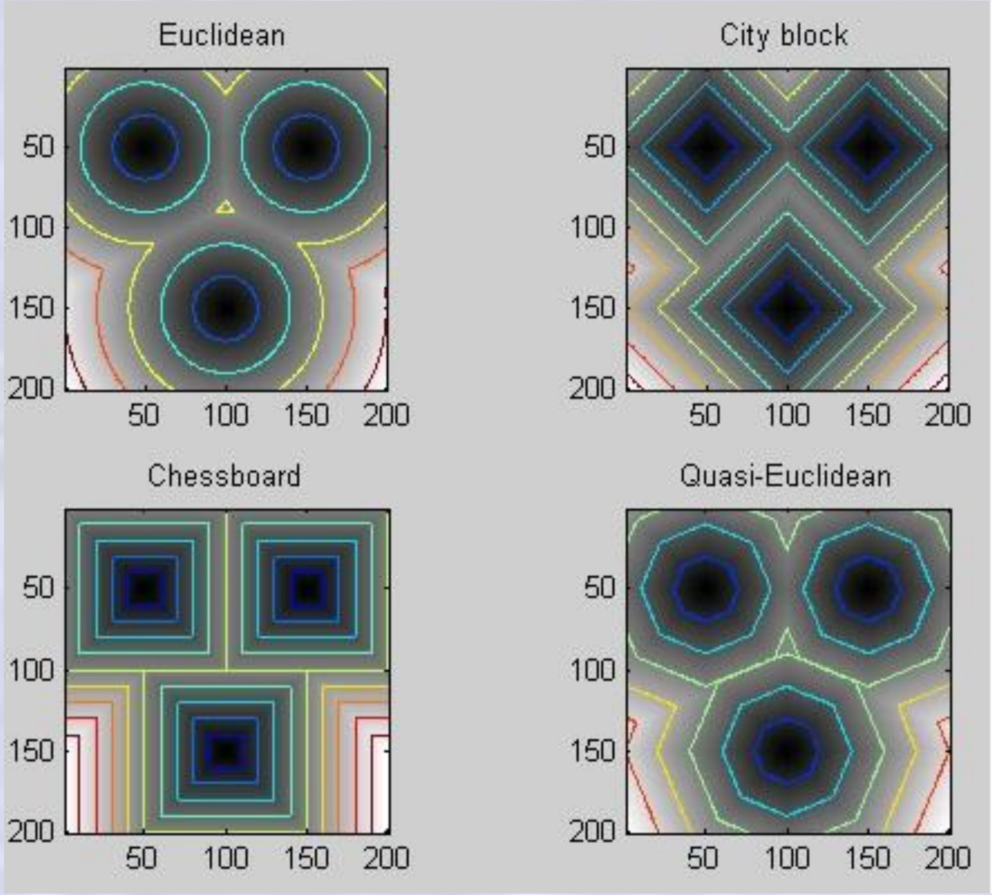




- Matlab: bwdist

Distance Transform

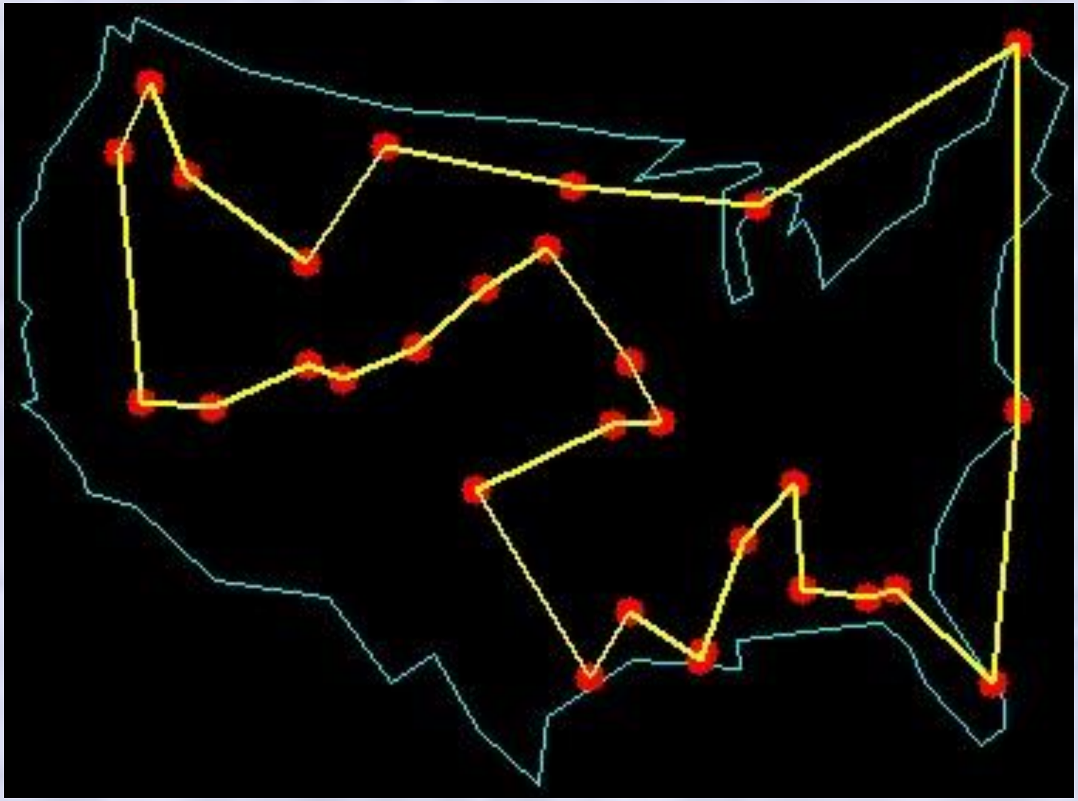
- **Distance between the pixel and the nearest nonzero pixel**



- **Matlab: bwdist**

Travelling Salesman

- Matlab's demo



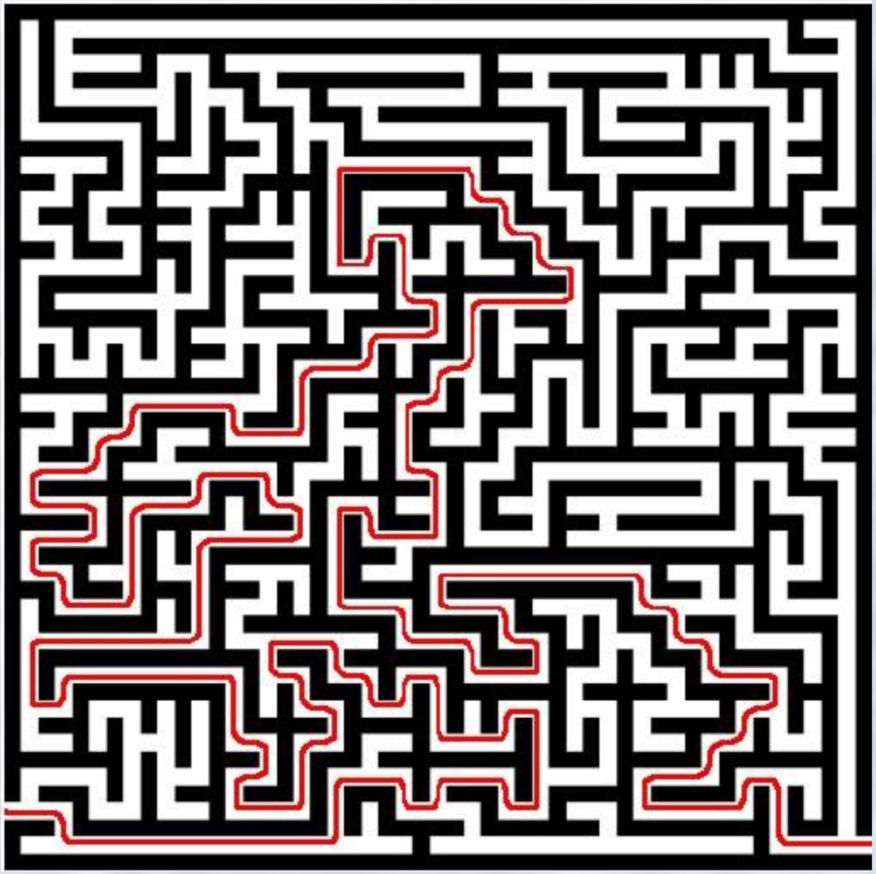
- Matlab: travel

Geodesic

- **Geodesic minimizes local distance**

Maze

- From the blog of Steve Eddins

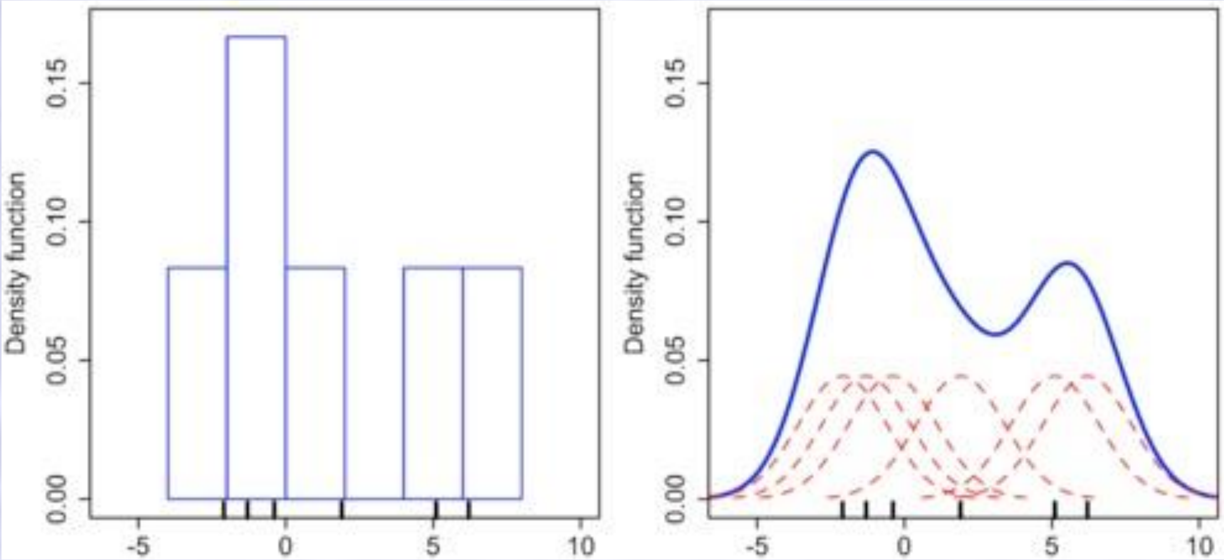


- Matlab: bwdistgeodesic

Kernel Density Estimation

- **Kernel Density Estimation can be viewed as "generalized histogram"**

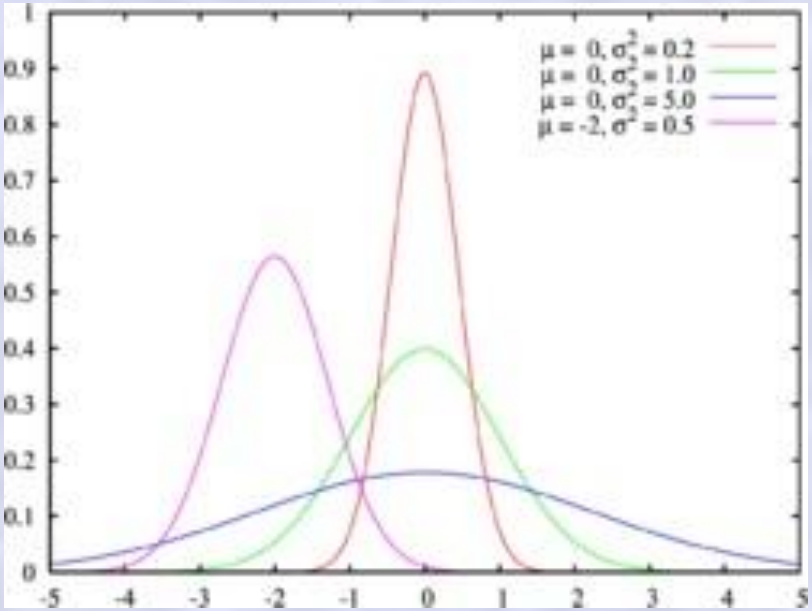
$x_1 = -2.1, x_2 = -1.3, x_3 = -0.4, x_4 = 1.9, x_5 = 5.1, x_6 = 6.2.$

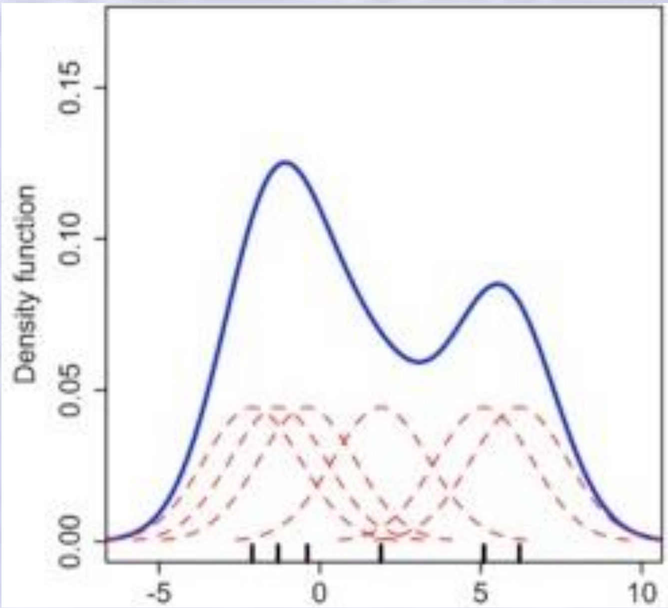


- **Can be computed in O(N) [2]**

Kernel Density Estimation vs Gaussian Mixture Model

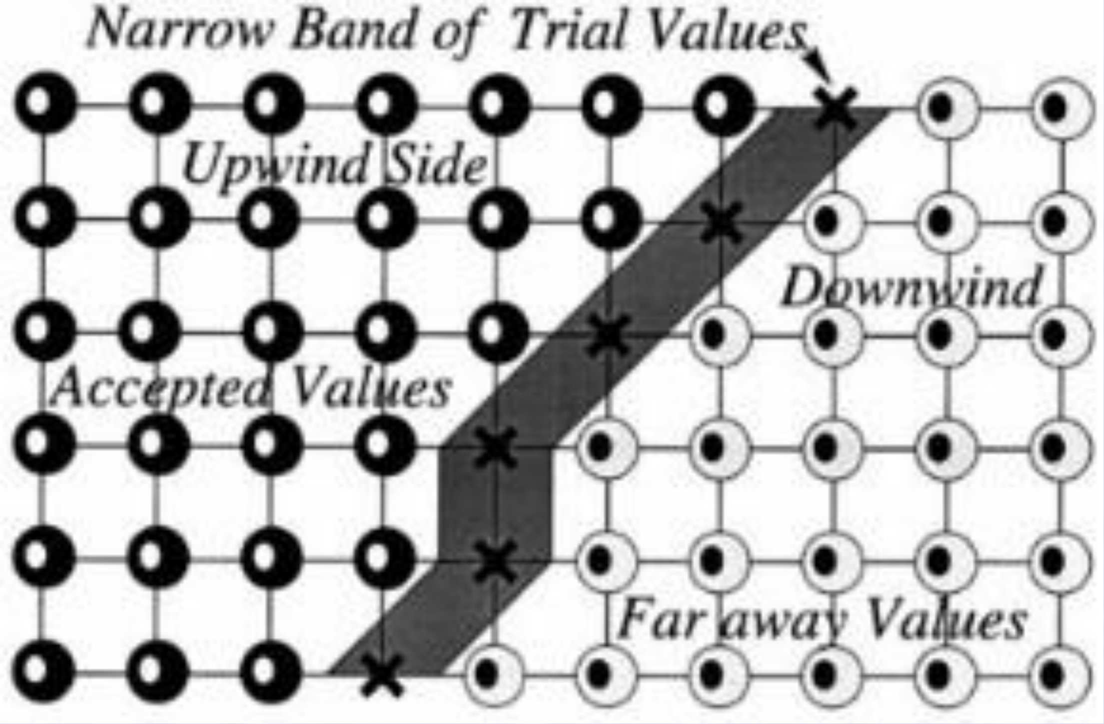
- **GMM is parametric model**
- **KDE is non-parametric model**





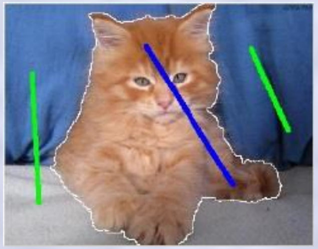
Fast Marching Method

- **Numerical method for solution of Eikonal equation**



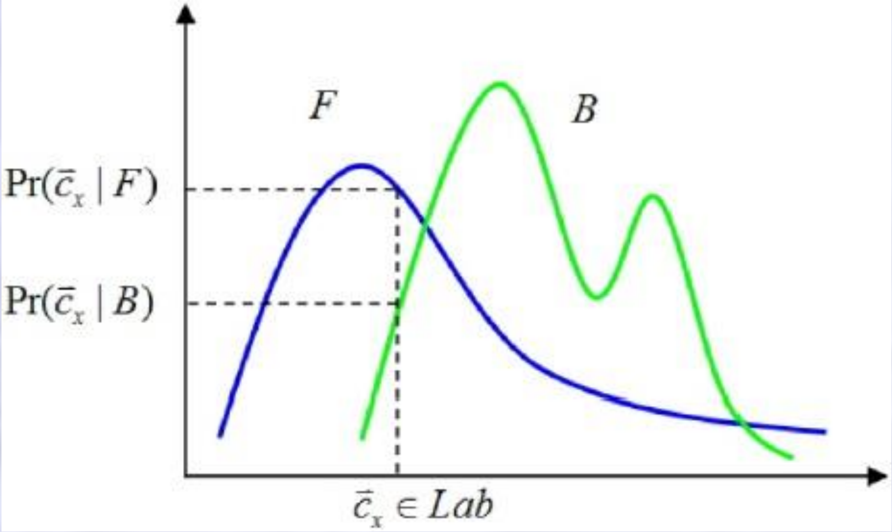
- **Can be computed in O(N) [3]**

Foreground/Background likelihood



[1]

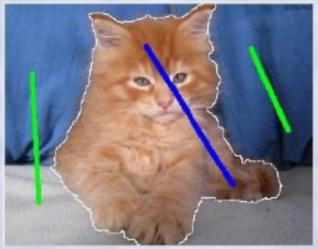
- **Use KDE on scribbles**



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

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

Weighted Geodesic Distance



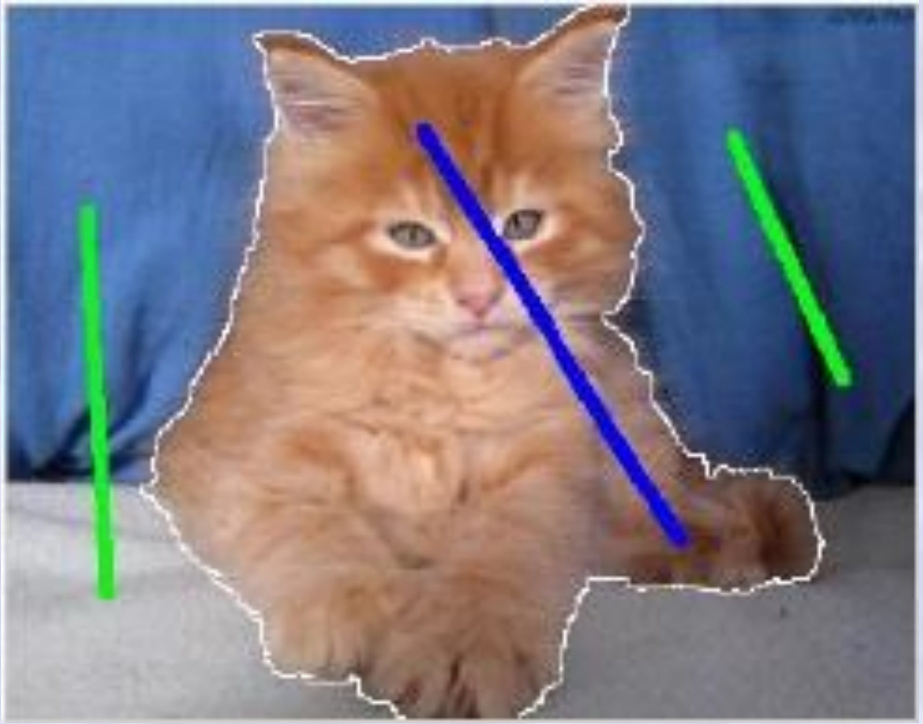
[1]

$$D_l(x) := \min_{s \in \Omega_l} d(s, x), \quad l \in \{\mathcal{F}, \mathcal{B}\},$$

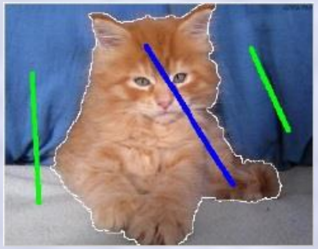
where

$$d(s_1, s_2) := \min_{C_{s_1,s_2}} \int_{s_1}^{s_2} |W(x) \cdot \dot{C}_{s_1,s_2}(x)| dx$$

$$W(x) = \nabla P_{\mathcal{F}}(x)$$



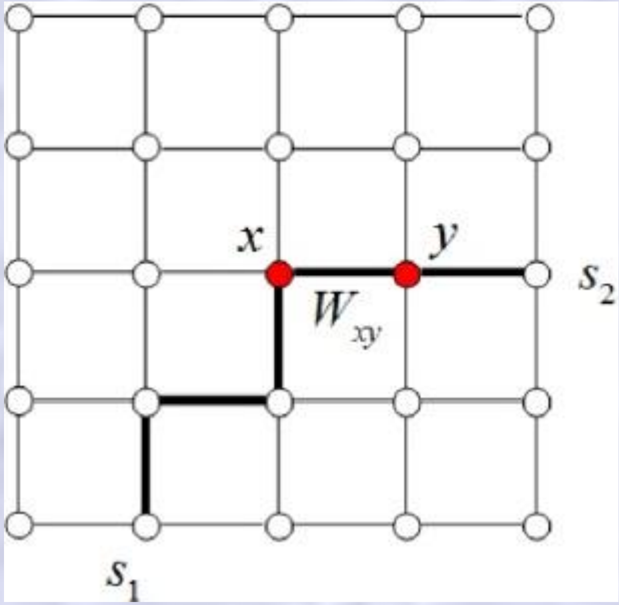
Discrete Weighted Geodesic Distance



[1]

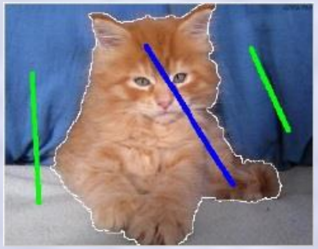
$$d(s_1, s_2) := \min_{C_{s_1,s_2}} \sum_{x,y} W_{xy},$$

$$W_{xy} = |P_{\mathcal{F}}(\vec{c}_x) - P_{\mathcal{F}}(\vec{c}_y)|, \quad x, y \in C_{s_1,s_2}.$$



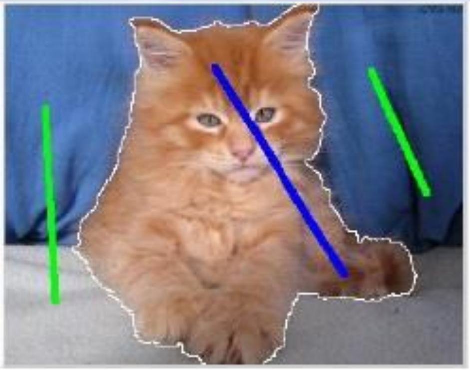
- **Solve with Fast Marching Method**

Boundary



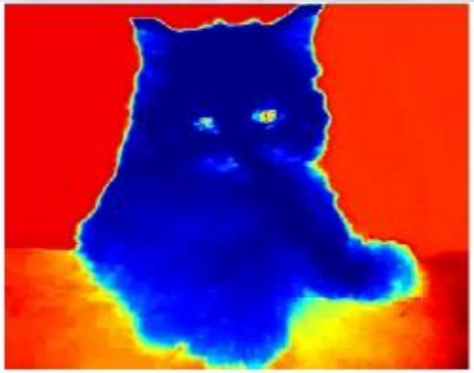
[1]

- **Foreground:** $V_F = \{x: D_F < D_B\}$
- **Background:** $V_B = \{x: D_B < D_F\}$
- **Boundary:** $\Delta = \{x: D_F = D_B\}$

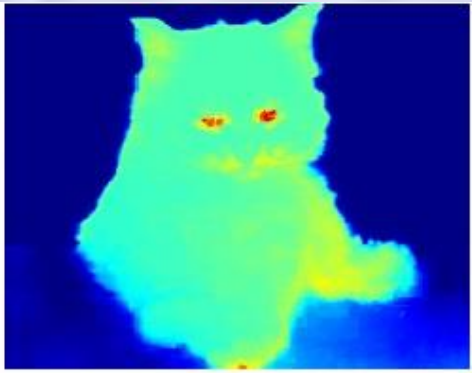


Boundary

$P_{\mathcal{F}}(x)$

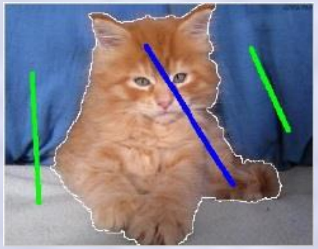


$D_{\mathcal{F}}(x)$



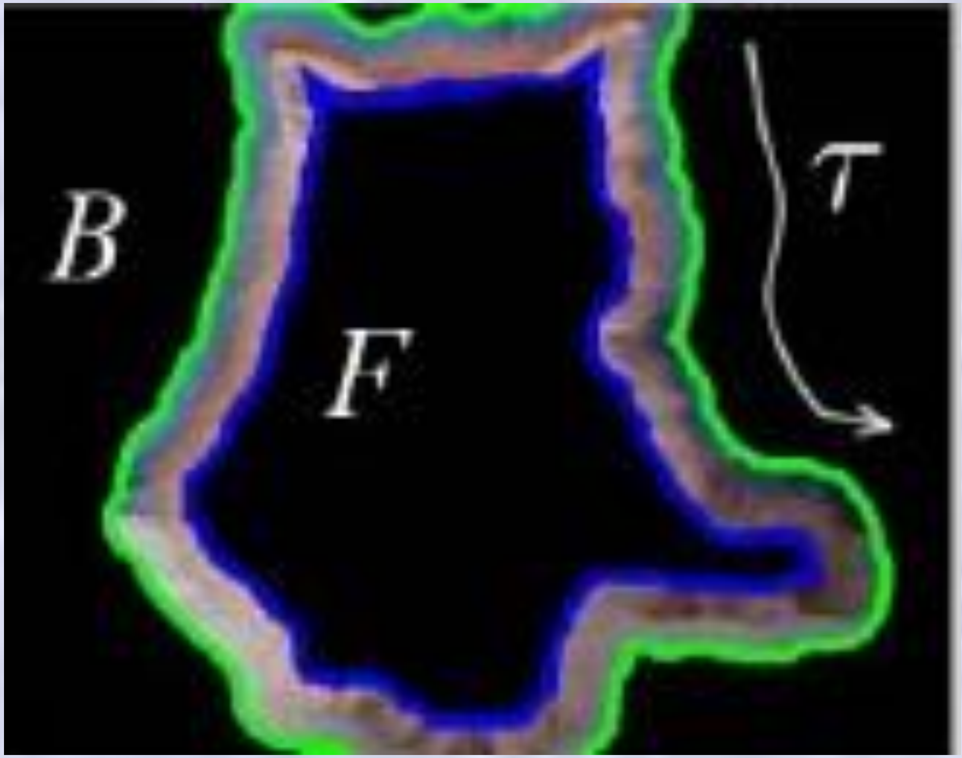
$D_{\mathcal{B}}(x)$

Trimap



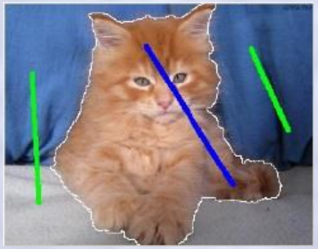
[1]

- **Narrow band:** $B_\rho(\Delta) = \{x : d(x, \Delta) \le \rho\}$



Trimap

Alpha



[1]

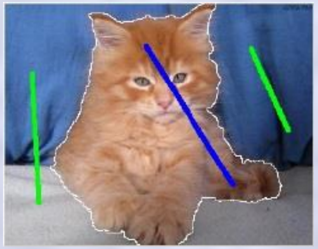
$$\omega_l(x) = D_l(x)^{-r} \cdot P_l(x), \quad l \in \{\mathcal{F}, \mathcal{B}\},$$

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

$$0 \leq r \leq 2$$

Alpha matting (+ segmentation)

Conclusion



[1]

- **Fast segmentation and matting algorithm**
- **Can be computed in O(N)**

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] C. Yang, R. Duraiswami, N. Gumerov, and L. Davis, Improved fast Gauss transform and efficient kernel density estimation. *Proc. ICCV*, pp. 464–471, Nice, France, 2003.

[3] L. Yatziv, A. Bartesaghi, and G. Sapiro, O(n) implementation of the fast marching algorithm, *J. Comput. Phys.*, 212, pp. 393–399, 2006.