Previous Work

- Maximize mutual information (MI) of pixel with region labeling

$$I(G(c); L_{\mathcal{C}}(c))$$

- Approximate MI by using a Kernel Density Estimate to find the PDFs

$$\hat{p}_{\pm}(c) = \frac{1}{|R_{\pm}|} \int_{R_{\pm}} K(G(c) - G(x))dx$$

- Minimize the energy functional in level set methods

$$E(c) = -|\Omega| \hat{I}(G(c); L_{\mathcal{C}}(c)) + \alpha \oint_{\mathcal{C}} ds \qquad \forall c \in \mathcal{C}$$

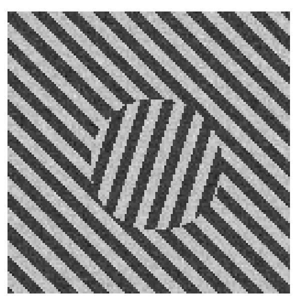
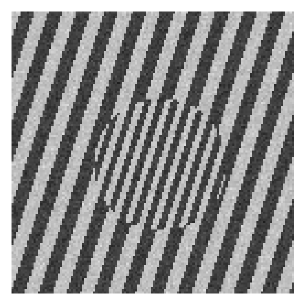
- Find the flow for the zero level set curve

$$\vec{V}(c) = \left[log \frac{\hat{p}_{+}(G(c))}{\hat{p}_{-}(G(c))} - \alpha\kappa \right] \vec{N} \qquad \forall c \in \mathcal{C}$$

Previous Work

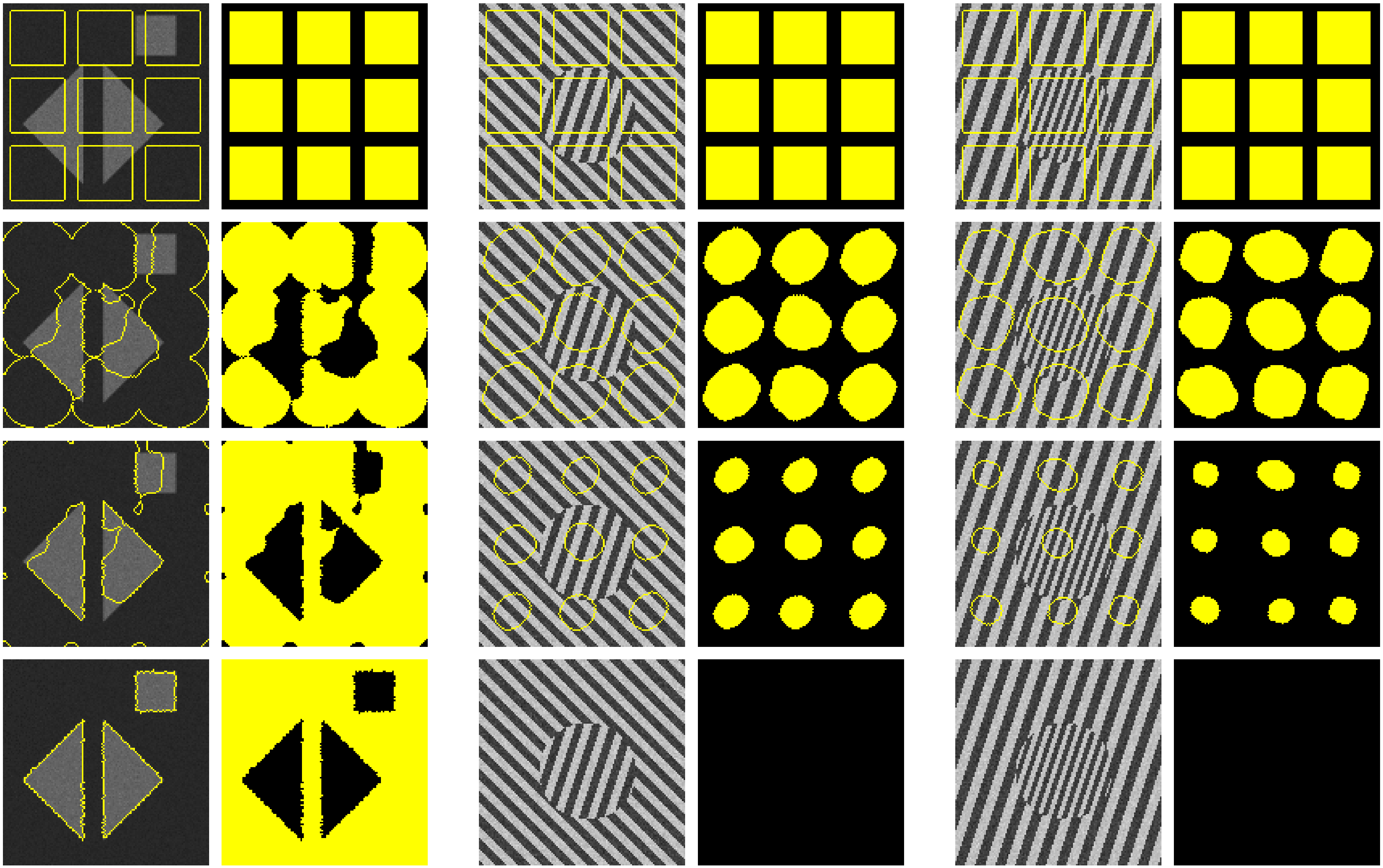
- Why is Junmo Kim's Scalar Segmentation Algorithm Good?
  - Segmentation based on non-parametric statistics
  - No training required
  - Can segment two regions with the same mean and variance
  - Performs very well on a large class of grayscale images

- What needs to be improved?
  - Likelihood depends solely on pixel intensity, no underlying image structure is taken advantage of
  - Only supports grayscale image segmentation
  - Textured images can not be segmented

Previous Work

Junmo Kim's Scalar Segmentation Algorithm



Vector Segmentation

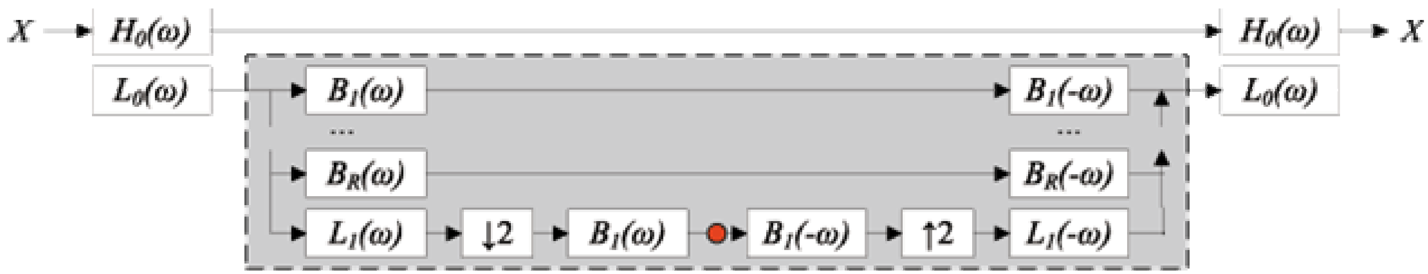
- If each pixel had a vector representation of texture, we could easily extend the formulation… (notice the **bold** vector **c**, instead of the scalar c)

$$\overrightarrow{V}(\mathbf{c}) = \left[log\frac{\hat{p}_+(G(\mathbf{c}))}{\hat{p}_-(G(\mathbf{c}))} - \alpha\kappa\right]\overrightarrow{N} \qquad \forall\mathbf{c} \in \mathcal{C}$$

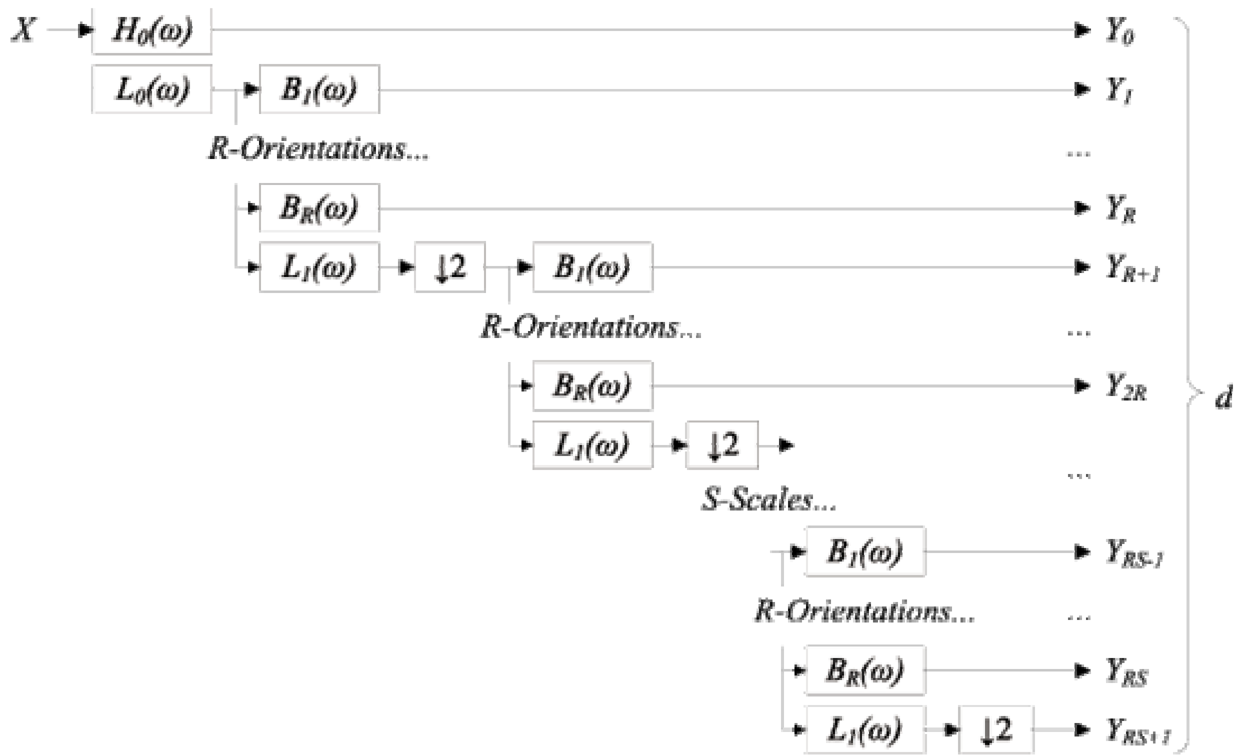
- Color images are now easily segmented

- Vector images can be generated using Steerable Pyramids (Simoncelli et al.)

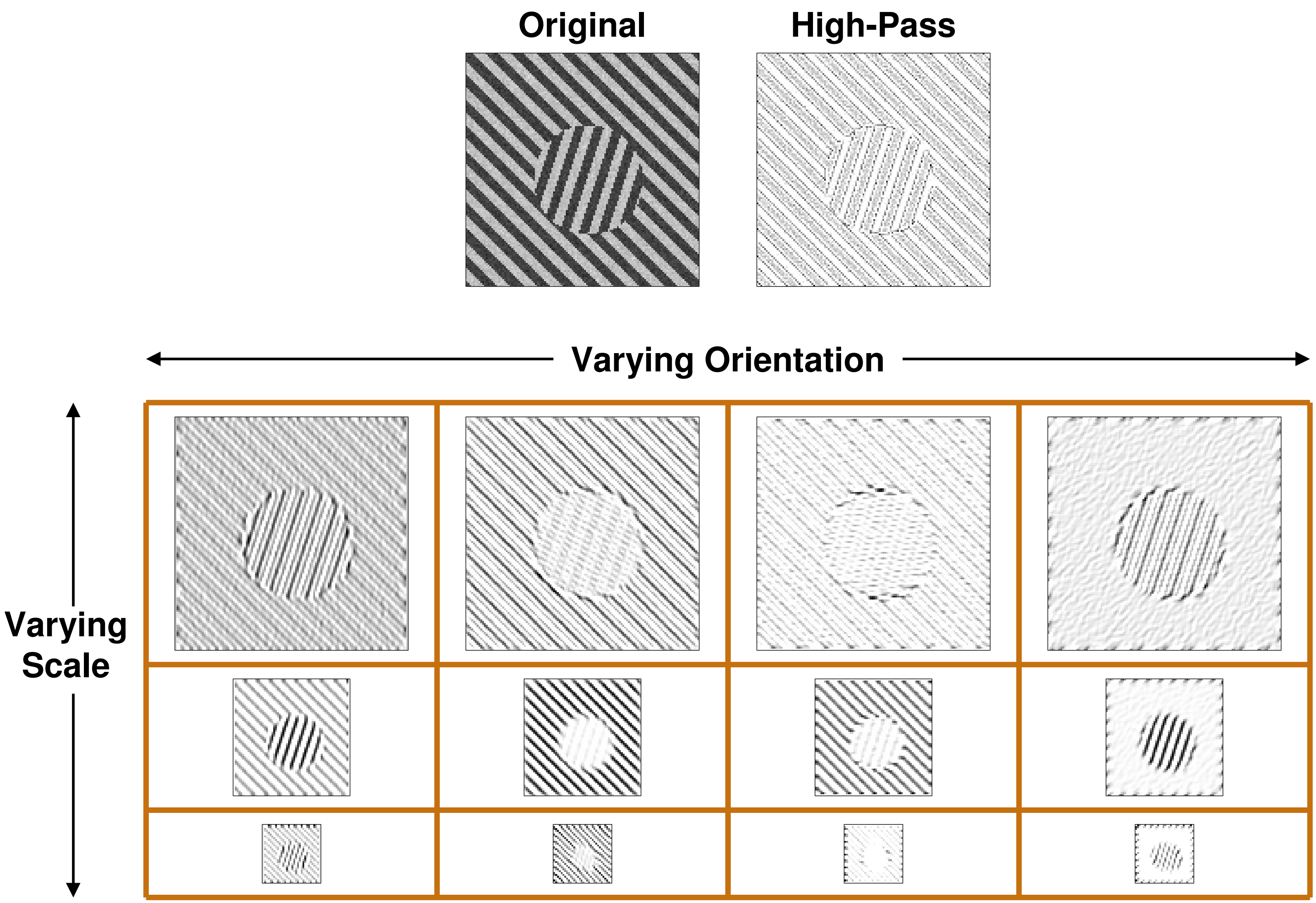
Steerable Pyramid



Recursively replace the gray box at the red dot



Steerable Pyramid



New Algorithm

- Estimating PDFs using the Improved Fast Gauss Transform (Duraiswami et al.) has complexity:

$$\mathcal{O}(d^p(M+N))$$

d – dimensionality   M – # target points   N – # source points

- For high dimensions (like the steerable pyramid output), this computation takes too long to perform

- Perform dimensionality reduction by optimally choosing the K “most contributing” images in the pyramid

Dimensionality Reduction

- By construction of the steerable pyramid, we can reconstruct the original image, $\mathbf{x}$, from the outputs, $\mathbf{y}_i$

$$\mathbf{x} = \sum_{i=0}^{d-1} \Theta_i \mathbf{y}_i$$

- We can approximately reconstruct $\mathbf{x}$ by only using some of the outputs which are chosen by the binary string, $\mathbf{u}$

$$\hat{\mathbf{x}} = \sum_{i=0}^{d-1} \Theta_i \mathbf{y}_i u_i$$

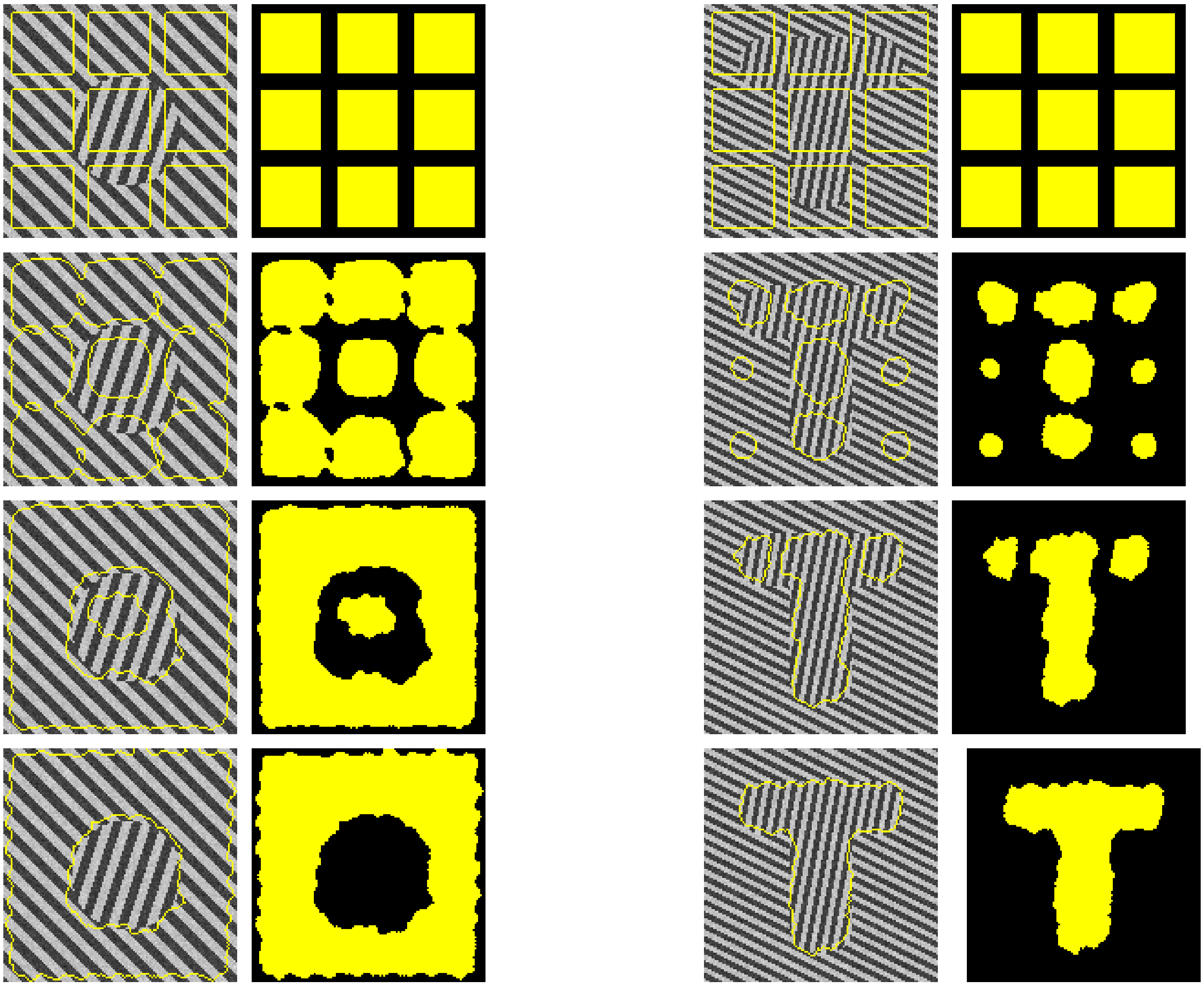
- The error criterion used (not necessarily the best) between $\hat{\mathbf{x}}$ and $\mathbf{x}$ is

$$e(\hat{\mathbf{x}}, \mathbf{x}) = \|\hat{\mathbf{x}} - \mathbf{x}\|_2^2 = (\hat{\mathbf{x}} - \mathbf{x})^T (\hat{\mathbf{x}} - \mathbf{x})$$

- Perform the following optimization

$$\mathbf{u}^* = \arg\min_{\mathbf{u} \in \{0,1\}^d} \left[(1-\beta) e(\hat{\mathbf{x}}, \mathbf{x}) + \beta \sum_{i=0}^{d-1} u_i \right]$$

Preliminary Results



Preliminary Results