Localizing 3D Cuboids in Single-view Images

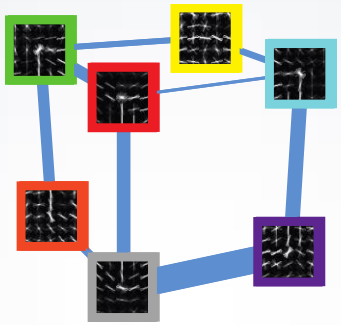
Jianxiong Xiao

Bryan C. Russell

Antonio Torralba

3D Cuboid Detector

Input image



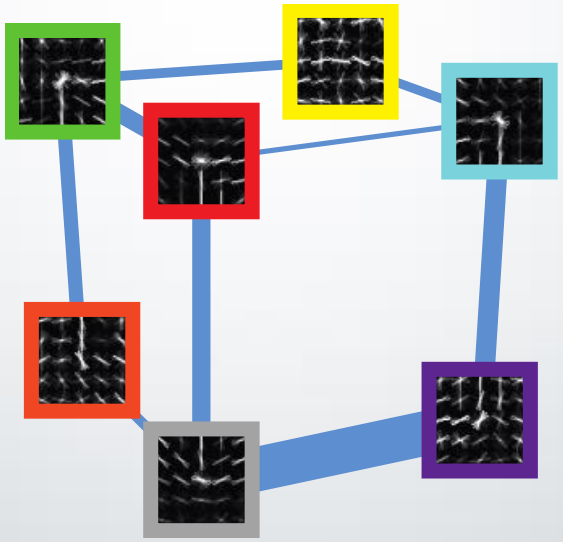
detect

Output detections

Synthesized New Views

Scoring Function

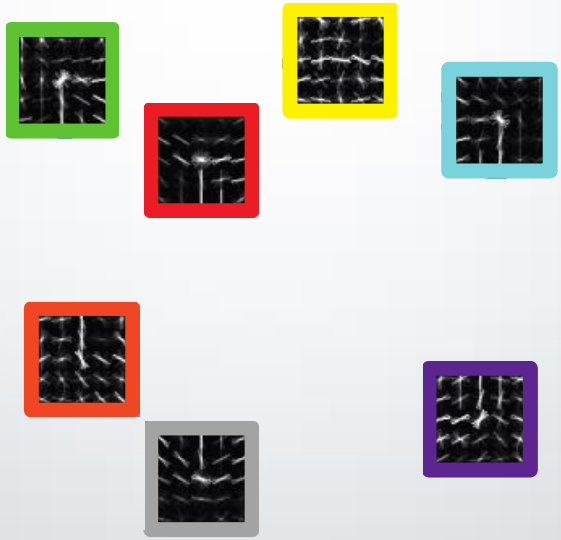
$$S(I,p) = \sum_{i \in \mathcal{V}} w_i^H \cdot \mathrm{HOG}(I, p_i) + \sum_{ij \in \mathcal{E}} w_{ij}^D \cdot \mathrm{Displacement}^{\mathrm{2D}}(p_i, p_j)$$
$$+ \sum_{ij \in \mathcal{E}} w_{ij}^E \cdot \mathrm{Edge}(I, p_i, p_j) + w^S \cdot \mathrm{Shape}^{\mathrm{3D}}(p)$$



Scoring Function

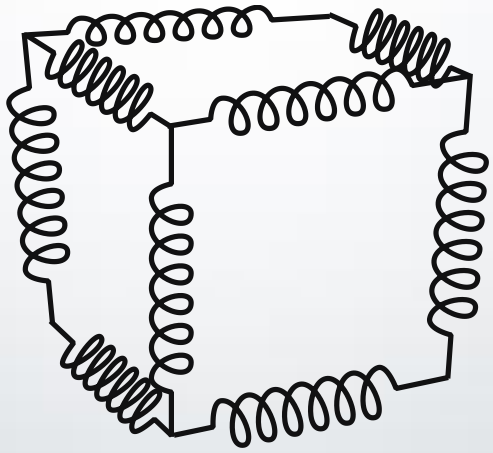
$$S(I,p) = \sum_{i \in \mathcal{V}} w_i^H \cdot \mathrm{HOG}(I, p_i) + \sum_{ij \in \mathcal{E}} w_{ij}^D \cdot \mathrm{Displacement}^{\mathrm{2D}}(p_i, p_j)$$

$$+ \sum_{ij \in \mathcal{E}} w_{ij}^E \cdot \mathrm{Edge}(I, p_i, p_j) + w^S \cdot \mathrm{Shape}^{\mathrm{3D}}(p)$$



Scoring Function

$$S(I,p) = \sum_{i \in \mathcal{V}} w_i^H \cdot \mathrm{HOG}(I, p_i) + \sum_{ij \in \mathcal{E}} w_{ij}^D \cdot \mathrm{Displacement}^{\mathrm{2D}}(p_i, p_j)$$
$$+ \sum_{ij \in \mathcal{E}} w_{ij}^E \cdot \mathrm{Edge}(I, p_i, p_j) + w^S \cdot \mathrm{Shape}^{\mathrm{3D}}(p)$$

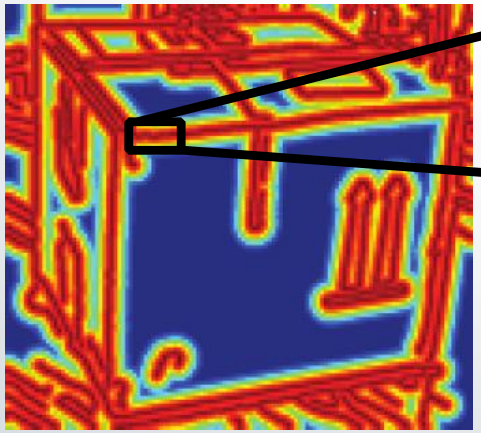


Scoring Function

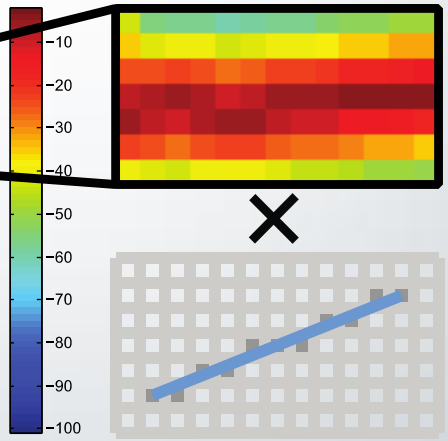
$$S(I,p) = \sum_{i \in \mathcal{V}} w_i^H \cdot \mathrm{HOG}(I,p_i) + \sum_{ij \in \mathcal{E}} w_{ij}^D \cdot \mathrm{Displacement}^{\mathrm{2D}}(p_i,p_j)$$

$$+ \sum_{ij \in \mathcal{E}} w_{ij}^E \cdot \mathrm{Edge}(I,p_i,p_j) + w^S \cdot \mathrm{Shape}^{\mathrm{3D}}(p)$$

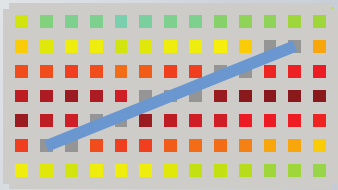
Image



Distance Transformed Edge Map



Pixels Covered by Line Segment



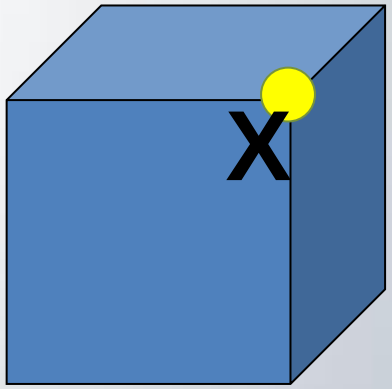
Dot-product is the Edge Term

Scoring Function

$$S(I,p) = \sum_{i \in \mathcal{V}} w_i^H \cdot \mathrm{HOG}(I,p_i) + \sum_{ij \in \mathcal{E}} w_{ij}^D \cdot \mathrm{Displacement}^{\mathrm{2D}}(p_i,p_j)$$
$$+ \sum_{ij \in \mathcal{E}} w_{ij}^E \cdot \mathrm{Edge}(I,p_i,p_j) + w^S \cdot \mathrm{Shape}^{\mathrm{3D}}(p)$$

Image (2D)

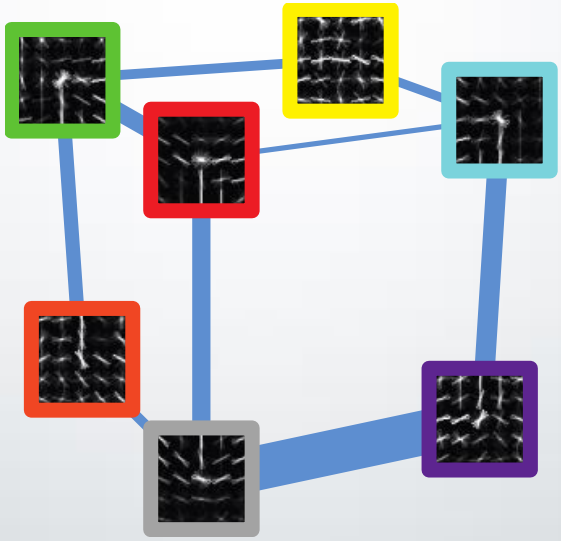
$\|x - PLX\|$



Unit Cuboid (3D)

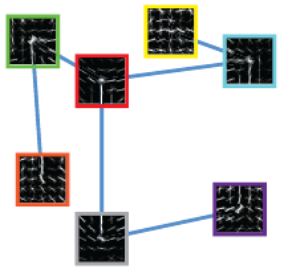
Scoring Function

$$S(I,p) = \sum_{i \in \mathcal{V}} w_i^H \cdot \mathrm{HOG}(I,p_i) + \sum_{ij \in \mathcal{E}} w_{ij}^D \cdot \mathrm{Displacement}^{\mathrm{2D}}(p_i,p_j)$$
$$+ \sum_{ij \in \mathcal{E}} w_{ij}^E \cdot \mathrm{Edge}(I,p_i,p_j) + w^S \cdot \mathrm{Shape}^{\mathrm{3D}}(p)$$



Inference

Select initial cuboid configuration from 2D view



Step 1: Approximation by a tree (dynamic programming + distance transform).

$$S(I,p) = \sum_{i \in \mathcal{V}} w_i^H \cdot \mathrm{HOG}(I,p_i) + \sum_{ij \in \mathcal{T}} w_{ij}^D \cdot \mathrm{Displacement}^{\mathrm{2D}}(p_i,p_j)$$

Step 2: Local search (hill climbing or ICM).

Learning

Learn term weights of score function by supervised learning.

Annotate positive and negative corners and train a structural SVM

Supervised corner-location training with Structural SVM.

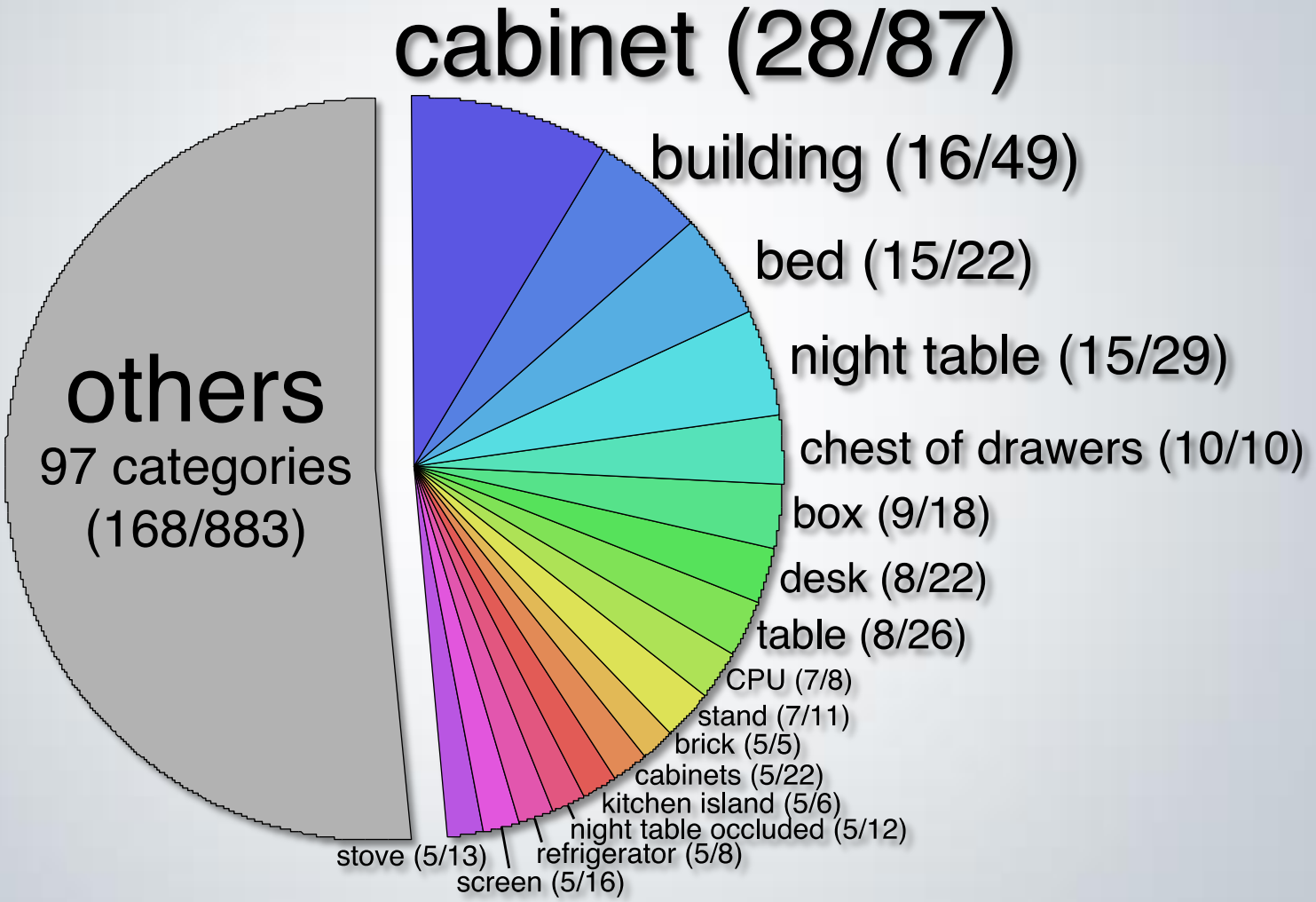
$$\min_{\beta,\xi\geq 0} \quad \frac{1}{2}\beta\cdot\beta + C\sum_n \xi_n$$

$$\forall n \in \text{pos} \quad \beta\cdot\Phi\left(I_n, p_n\right) \geq 1-\xi_n$$

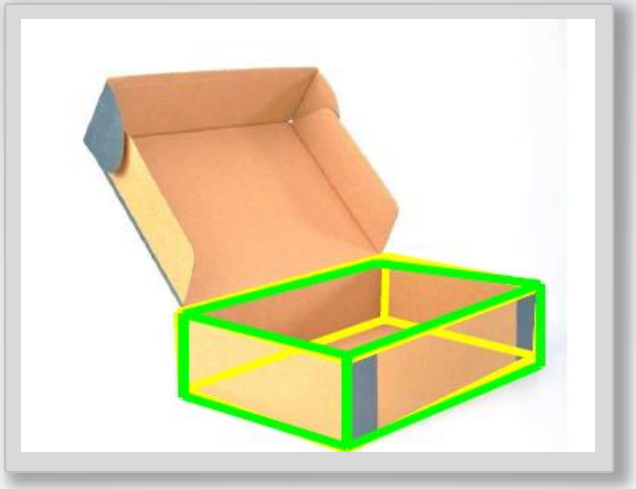
$$\forall n \in \text{neg}, \forall p \in P \quad \beta\cdot\Phi\left(I_n, p\right) \leq -1+\xi_n$$

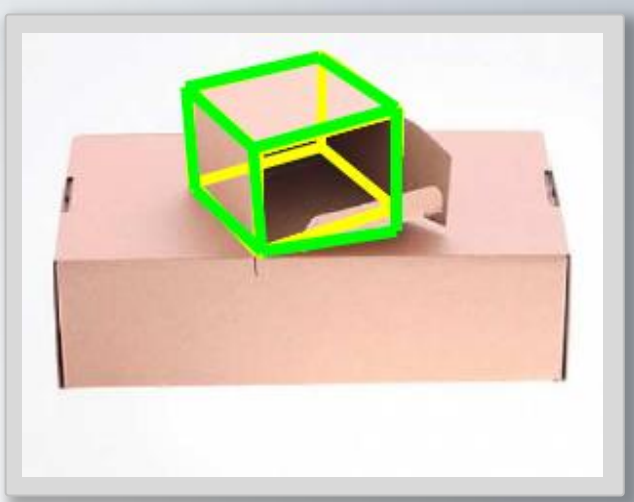

SUN Primitive Database

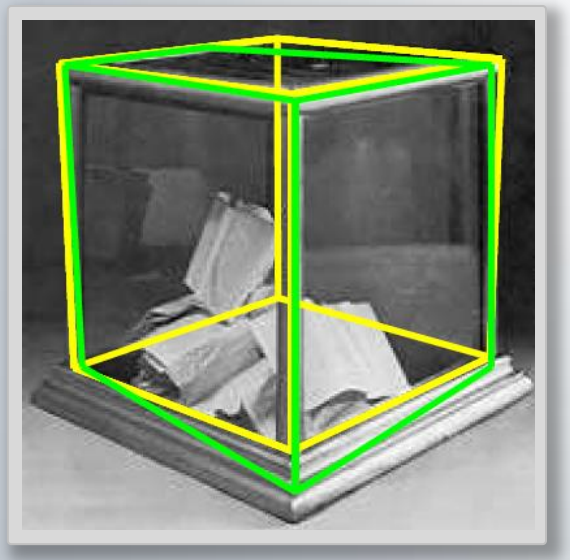
What objects are cuboids?

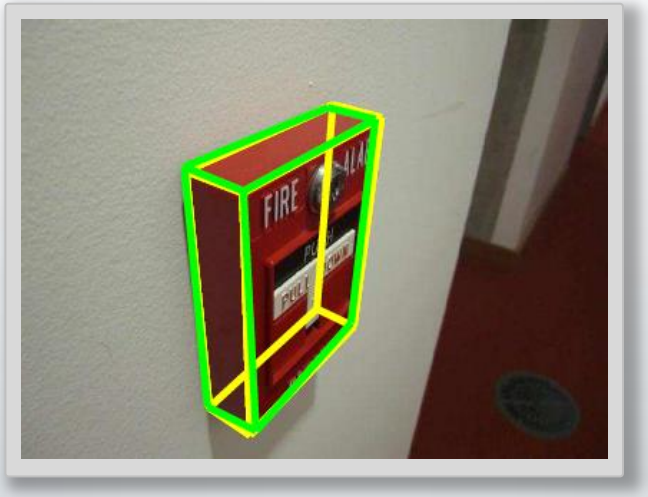


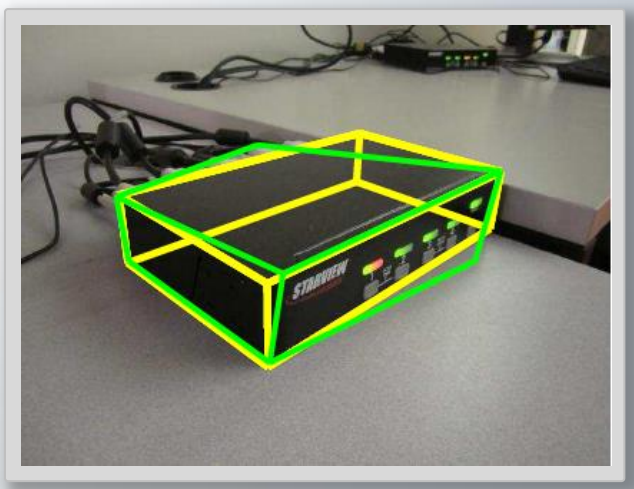
Easy Cases



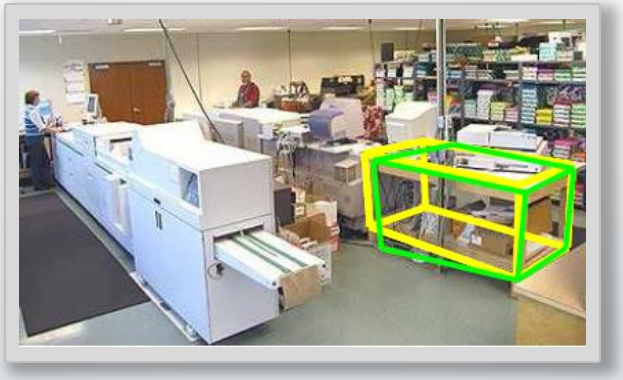




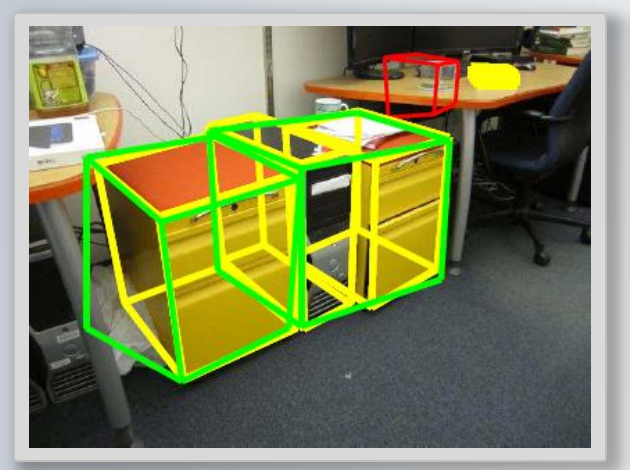
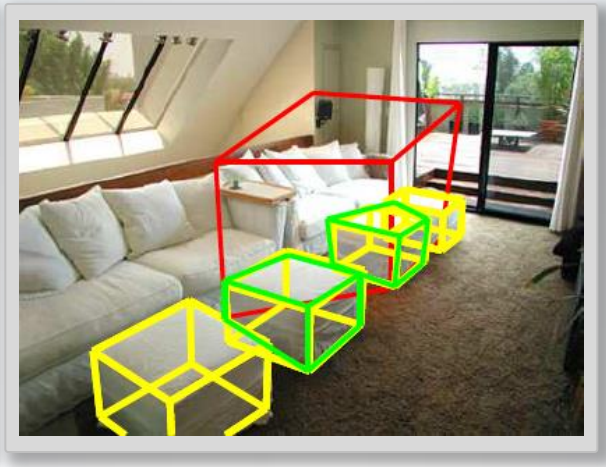


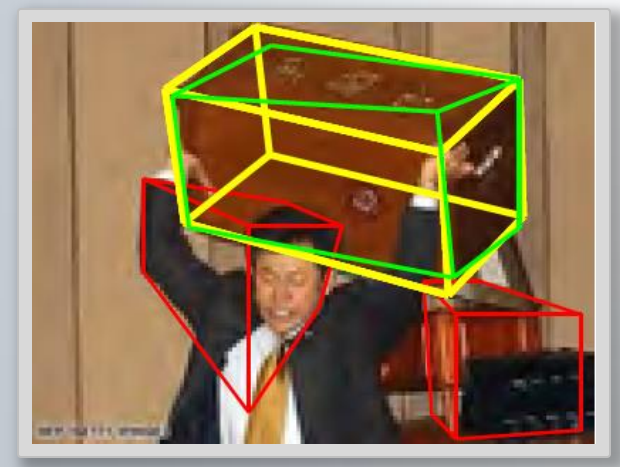
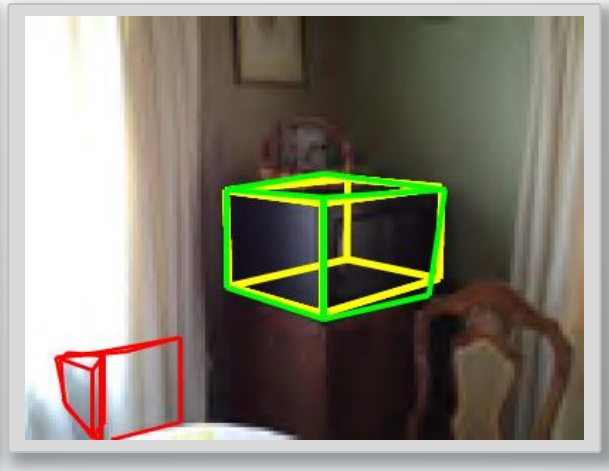
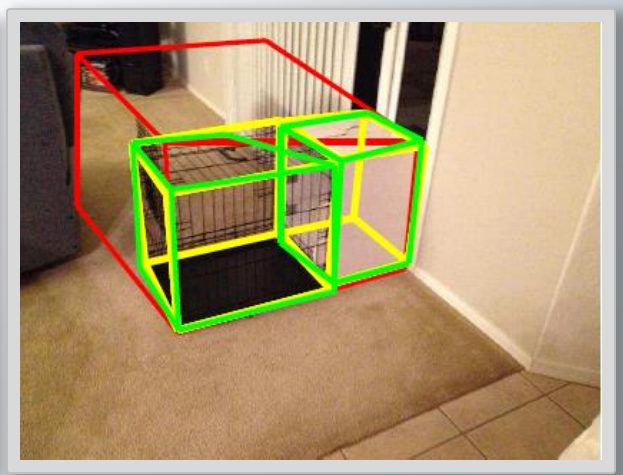


Detection Result



Multiple Instances

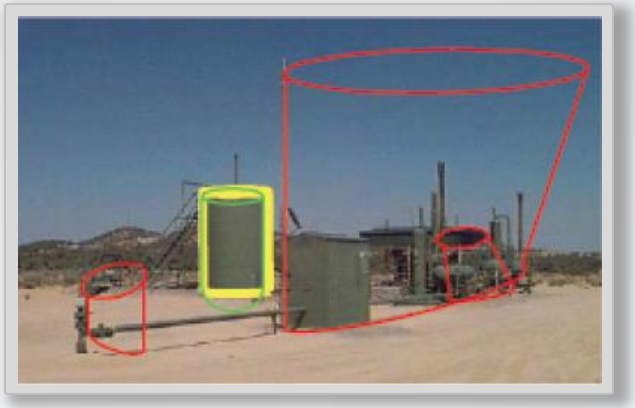
2D vs 3D

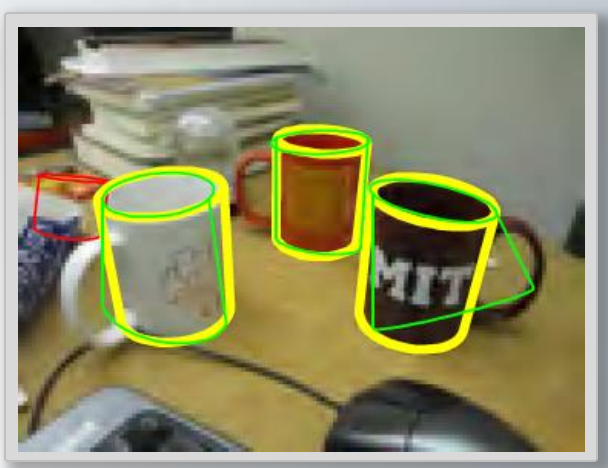
Ground truth

2D Detector

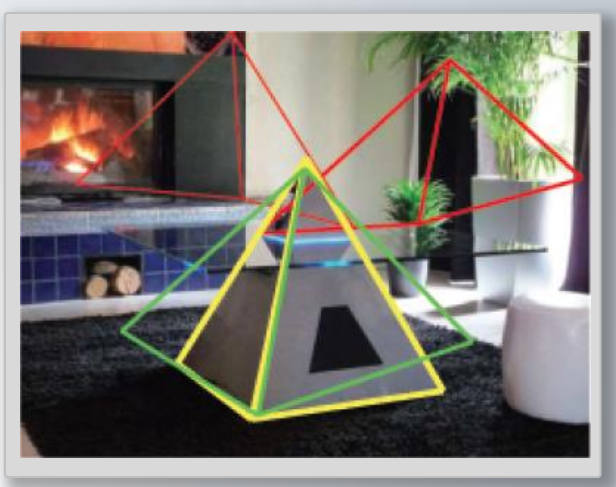
3D Detector

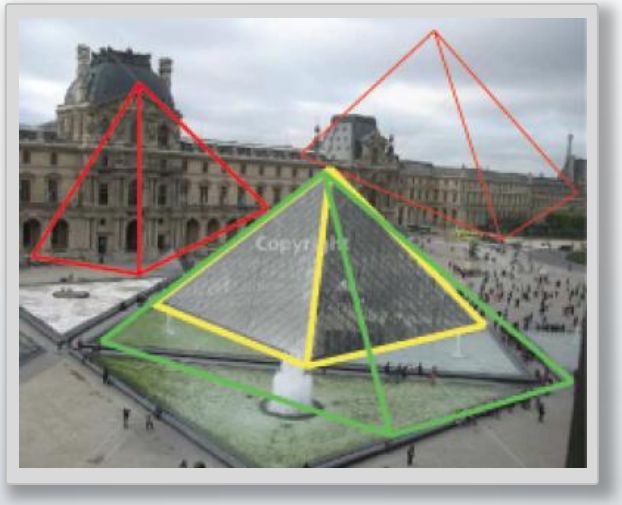
Cylinder Detection

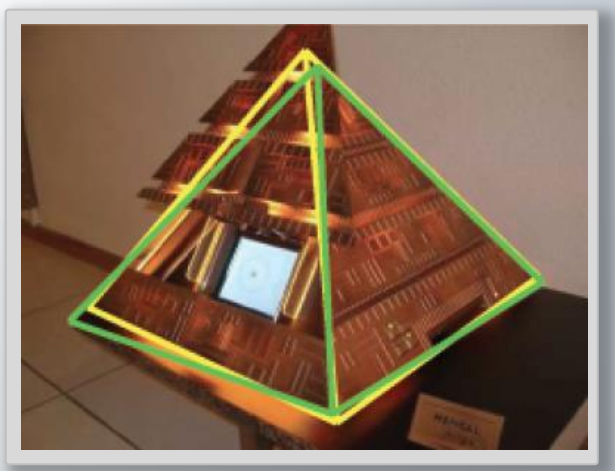




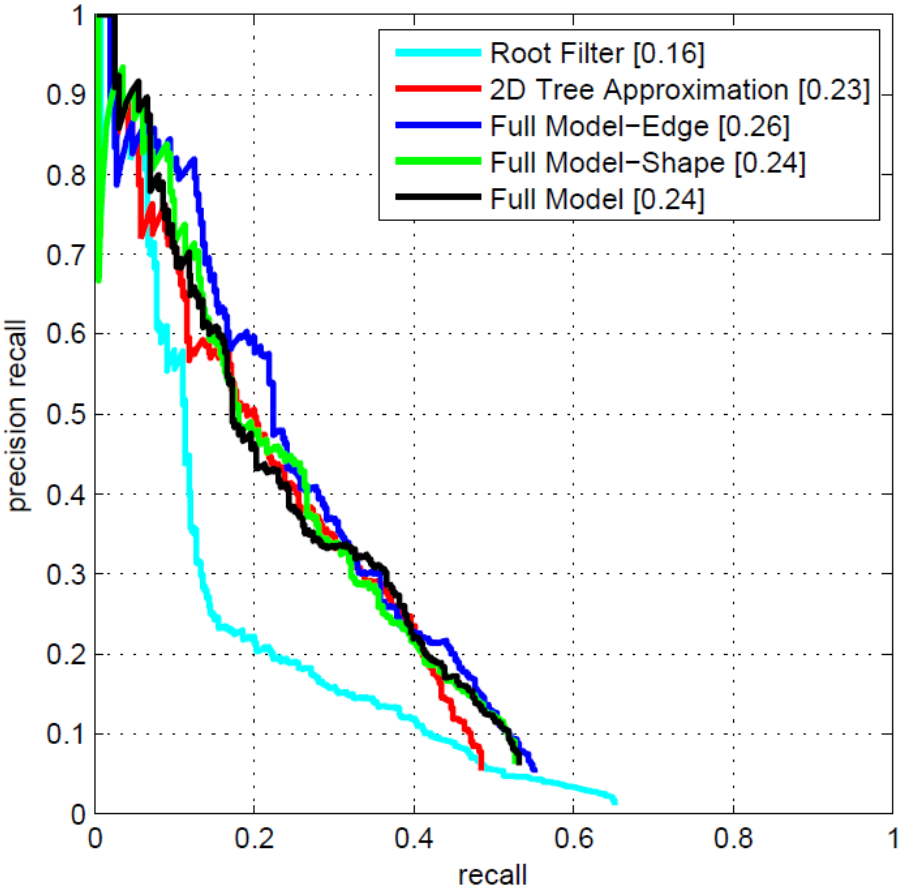
Pyramid Detection

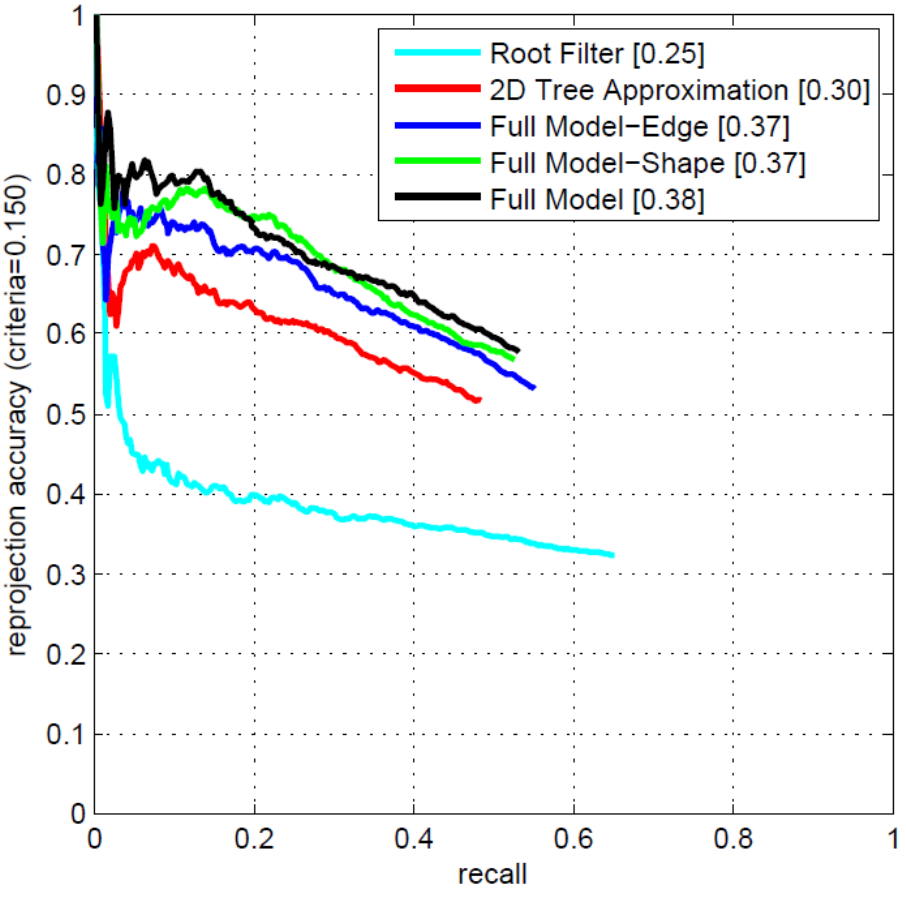






Quantitative Results





Thanks