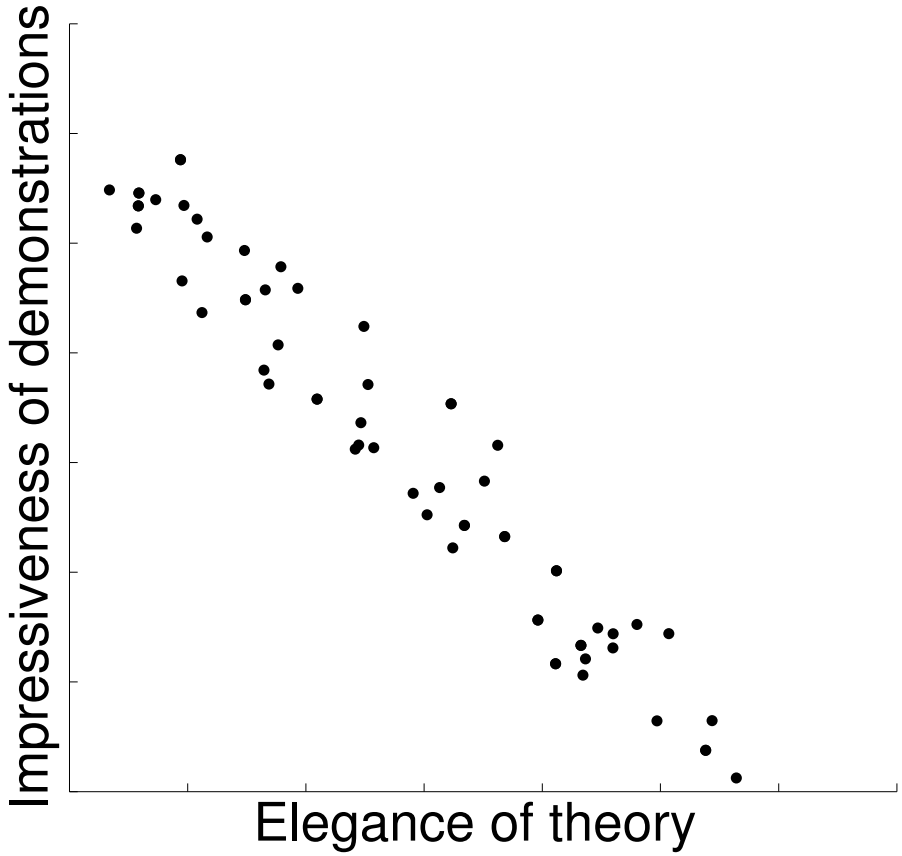
Learning 3-D Scene Structure from a Single Still Image, Ashutosh Saxena, Min Sun and Andrew Y. Ng 2007

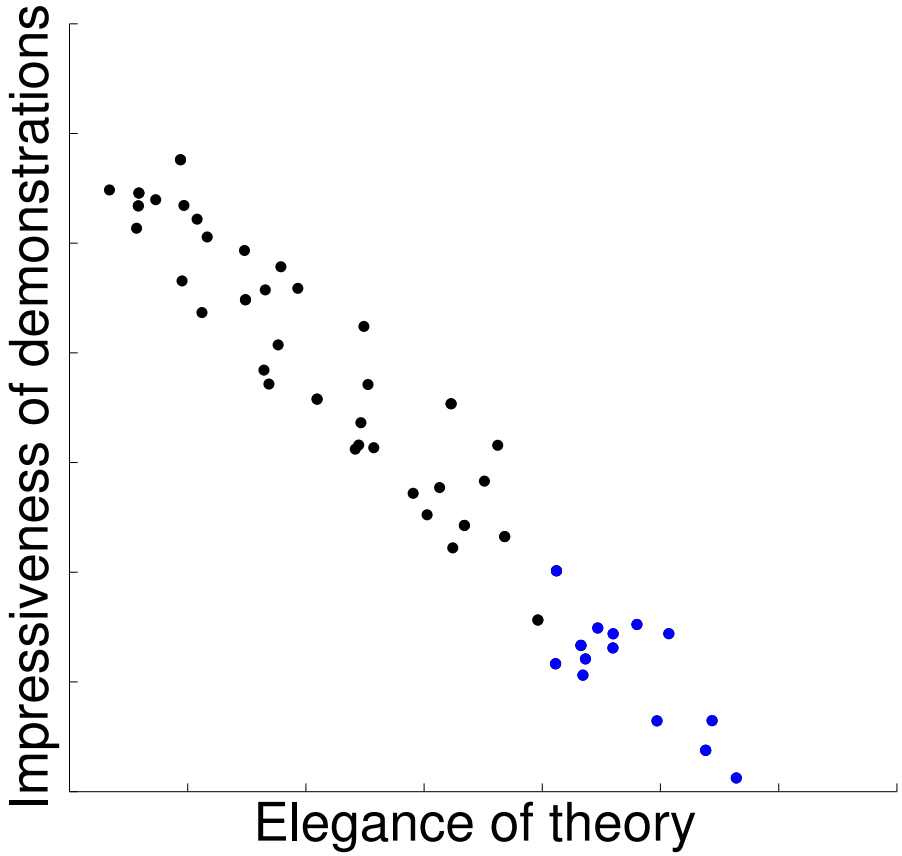
Rich Turner (turner@gatsby.ucl.ac.uk)

Gatsby Unit, 06/11/2007

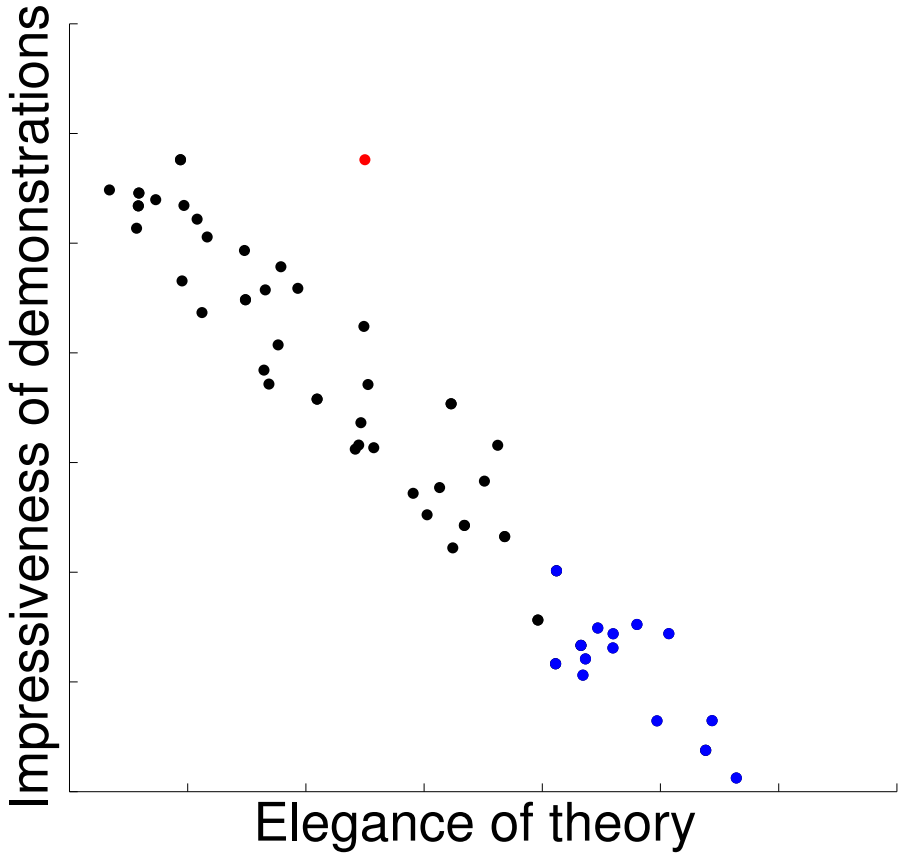
Introduction



Bayesians

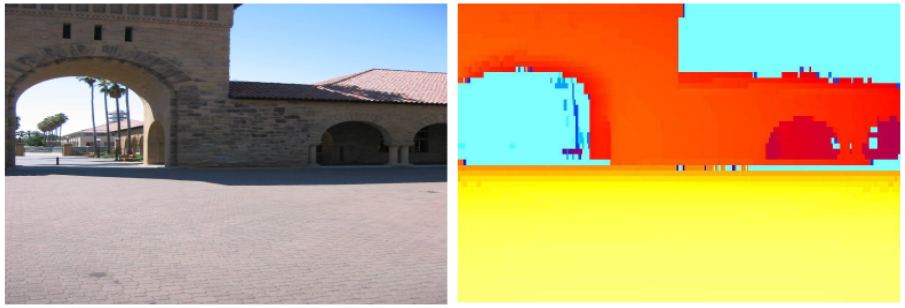


This paper - daring and impressive



Setup - Supervised Learning

- **Goal**: infer depth of each point of a query image
- **Training data**: images and depth maps (3D laser range finder)



Basic Approach:

- Describe the images by a set of oriented planes
- Infer the boundaries and the orientations of these planes

Priors

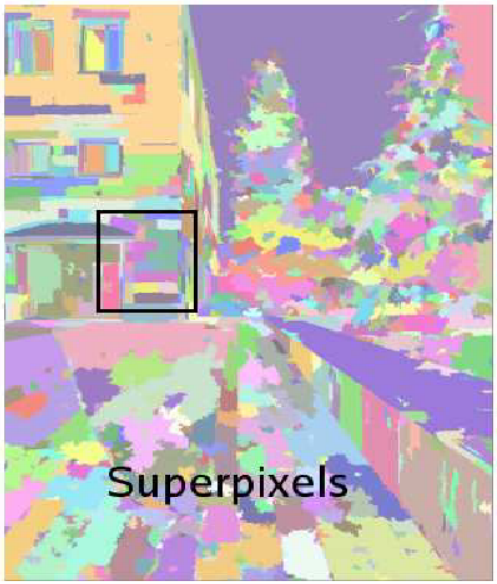
Types of **prior information** they try and inject:

- **texture variations** and **gradients**
- **colour** (blue things at the top of the image tend to be far away)
- **haze** and **defocus**

Integrate these **local cues** over space to get a realistic depth map

Step 1: Infer the plane boundaries

- Find small homogeneous (texture & colour) regions
- These **Superpixels** are assumed **locally planar**



Step 2: Infer the connectivity of the planes

- Neighbouring superpixels can be either
  - **connected** (e.g. if they are part of the same object)
  - **disconnected** (e.g. when they are parts of different occluding objects)
- Train a logistic classifier to decide ($y_{ij} = 1/0 \rightarrow$ connected/not)
- Features $\mathbf{x}_{ij}$ derived by running different scene segmentation algorithms: if the two superpixels live in the same region $x_{ij}^{(k)} = 1$.

$$p(y_{ij} = 1 | \mathbf{x}_{ij}; \Phi) \quad = \quad \frac{1}{1 + \exp(-\Phi^T \mathbf{x}_{ij})}$$

Step 3: Infer 3D structure using a Markov Random Field

$$p(D|X, Y; \theta) = \frac{1}{Z} \exp(-E(D, X, Y, \theta))$$

- Hand craft the form of the **energy** to:

  1. Extract **local depth cues**
  2. Encourage pairs of pixels on a boundary between **connected** superpixels to have **similar depths**
  3. Encourage **connected** superpixels to be **co-planar**
  4. Encourage points either side of long straight lines to be **co-linear**

Example energy term: local term

- Derive **estimated pixel depth** linearly from features $\hat{d} = \theta^T \mathbf{x}$
- **Features**: local texture summaries, superpixel shape, location info
- Penalise **fractional depth error**: $(\hat{d} - d)/d$

$$E_{local} = \sum_{superpixels} \sum_{pixels} v|(\hat{d} - d)/d|$$

- Learn the parameters θ from the training images and depths $\max_\theta p(D|X, Y, \theta)$
- Infer the depths for new test images $\max_D p(D|X, Y, \theta)$