3-D Depth Reconstruction from a Single Still Image

International Journal of Computer Vision (IJCV), Aug 2007

Ashutosh Saxena, Sung H. Chung, Andrew Y. Ng.

April 17, 2013

Presenter: Yiying Li

Outline

- Motivation
- Previous Attempts
- Methods and Evaluation
  - Feature Extraction
- Evaluation
- Replication
- Difficulties
- Importance

Motivation

- Recovering depth is an important application in scene understanding, robotics, and 3-d reconstruction.

- Previous work have been extracting depth using stereopsis, or structure from motion.

- Humans use certain monocular cues to indicate depth:
  - Texture variation
  - Gradients
  - Defocus
  - Color and haze

Previous Attempts

- Reconstruction for known fixed objects (faces, hands), (Nagai et al. 2002)
- Using uniform color and Lambertian surfaces (Many many papers).
- Fourier spectrum to compute mean depth (Torralba and Oliva 2002).
- Supervised learning for 1-D distance for specific obstacles (Michels et al. 2005).
- Fixed sky, ground, vertical regions on the image (Hoeim et al. 2005). No real depth map.

Feature Extraction

- Monocular cues include: texture variations/gradients, light, haze, defocus, occulsion, know object sizes, and many others.

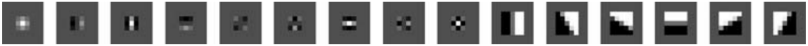
- If we only look at these features at a local scale we miss out on global properties of the image.

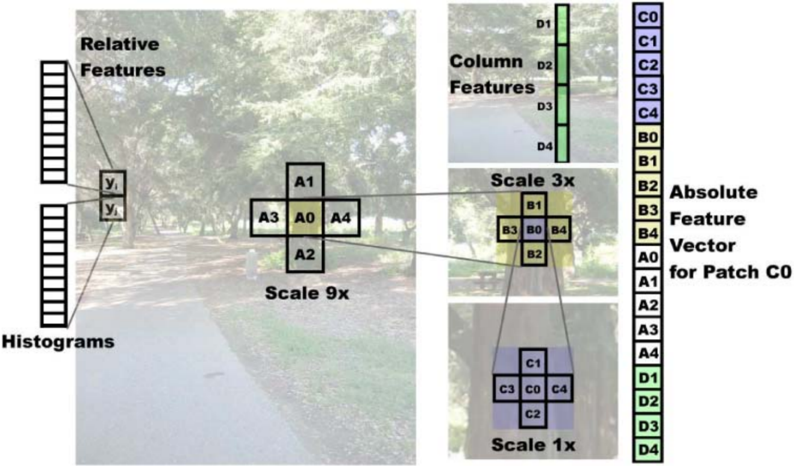
- Convert to YCbCr color space for separation of intensity and color.

Feature Extraction

- Absolute Features:
  - Texture information is in the intensity channel.
  - Law's mask and six rotated edge filters are convolved with intensity.
  - Haze is usually in low frequency of color channels.
  - Color channels are convolved with a local averaging filter.
  - 17 features: 9 Law's, 6 edge, 2 color.
  - The author also chose to square each filter output as well leaving us with 34 dimensions.
- Relative Features:
  - 10-bin histogram of each of the 17 filter outputs.
  - 170 dimensions.

Features





Relative
Features
Histograms
y
y
A1
A3
A0
A4
A2
Scale 9x
Column
Features
D1
D2
D3
D4
Scale 3x
B1
B3
B0
B4
B2
C1
C3
C0
C4
C2
Scale 1x
C0
C1
C2
C3
C4
B0
B1
B2
B3
B4
A0
A1
A2
A3
A4
D1
D2
D3
D4
Absolute
Feature
Vector
for Patch C0

Learning Model

- Represented using a multi-scale hierarchal Markov Random Field (MRF).
- Two different models that only different in inference.
- Gaussian Model:

$$P_G(d|X;\theta,\sigma) = \frac{1}{Z_G}\exp\left(-\sum_{i=1}^{M}\frac{(d_i(1)-x_i^T\theta_r)^2}{2\sigma_{1r}^2} - \sum_{s=1}^{3}\sum_{i=1}^{M}\sum_{j\in N_s(i)}\frac{(d_i(s)-d_j(s))^2}{2\sigma_{2rs}^2}\right)$$

- Laplacian Model:

$$P_L(d|X;\theta,\lambda) = \frac{1}{Z_L}\exp\left(-\sum_{i=1}^{M}\frac{(d_i(1)-x_i^T\theta_r)^2}{\lambda_{1r}} - \sum_{s=1}^{3}\sum_{i=1}^{M}\sum_{j\in N_s(i)}\frac{|d_i(s)-d_j(s)|}{\lambda_{2rs}}\right)$$

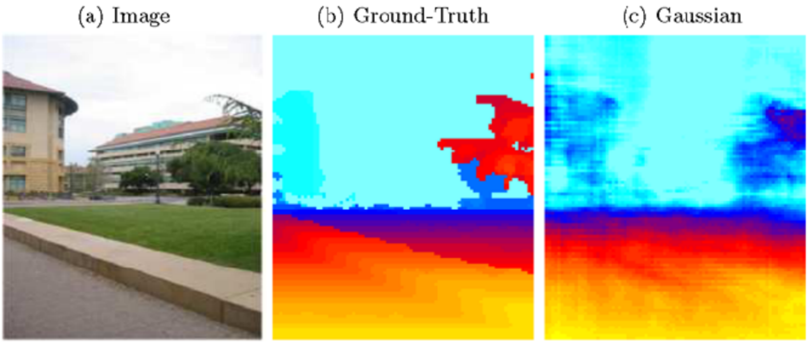
Evaluation

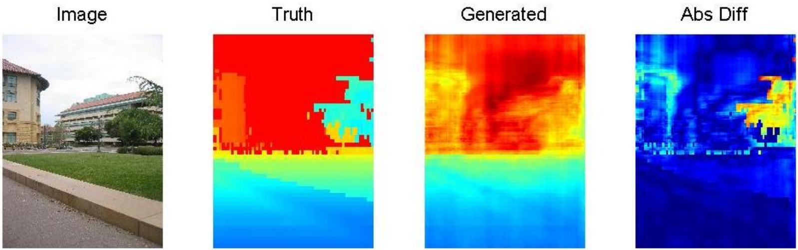
- Ground truth depth maps are captured by a 3-D laser scanner.
- Depth maps's are synced to the same FoV of the camera.
- Various different environments, trees, buildings, etc. . .
- 400 training pairs.
- 134 testing pairs.

Replication Results

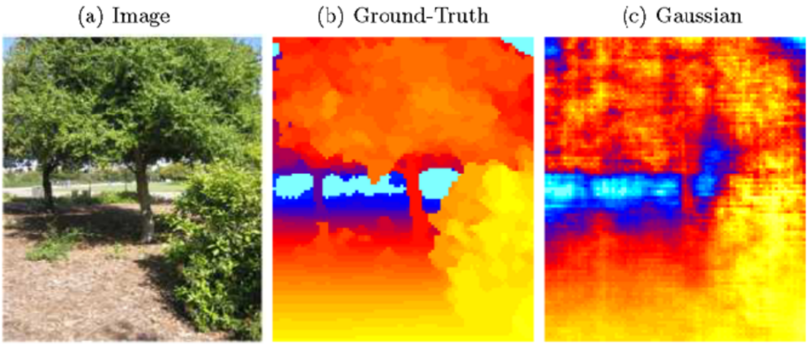
- Replicated using a 8x8 pixel patch in the first scale.
- Author's mean log 10 error for Gaussian Model: 0.133
- Replicated mean log 10 error for Gaussian Model: 0.150
- Replicate mean error: 1.4155 meters

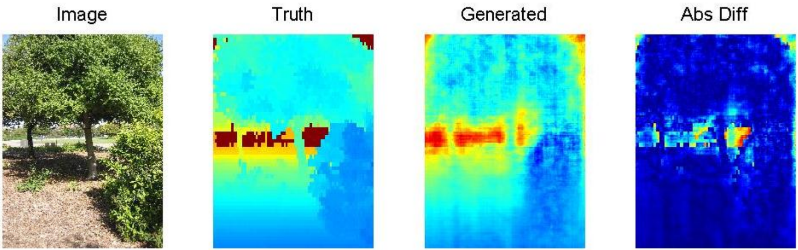
Compare Replication Results



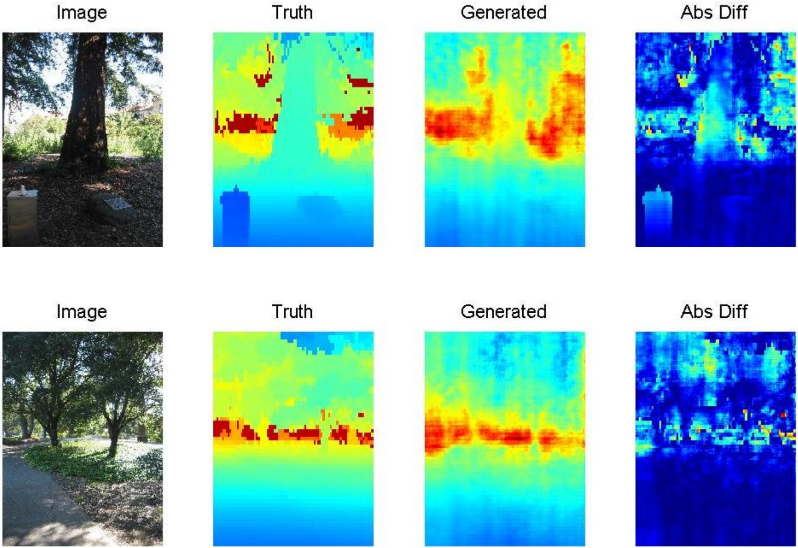


Compare Replication Results

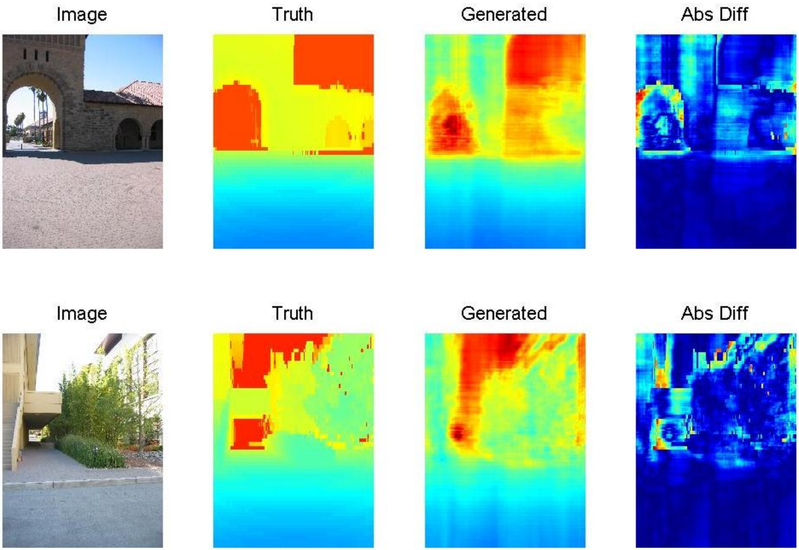




Good Results



Good Results



Bad Results

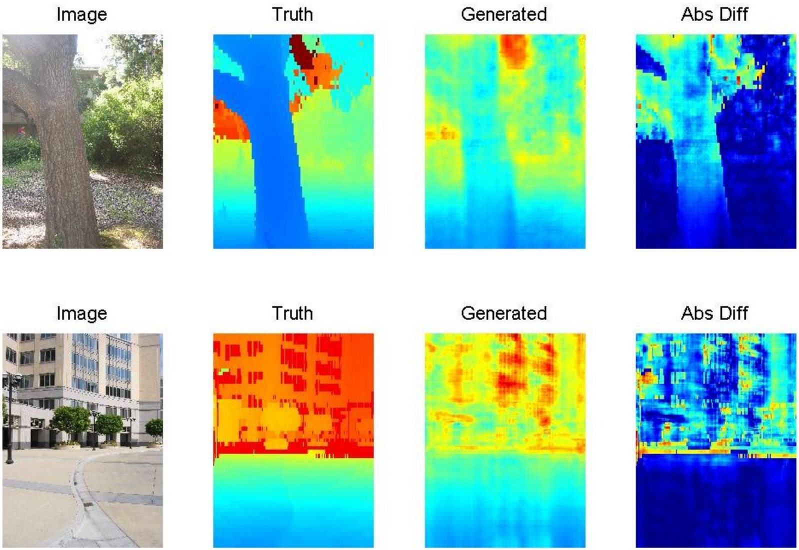


Image
Truth
Generated
Abs Diff
Image
Truth
Generated
Abs Diff

Bad Results

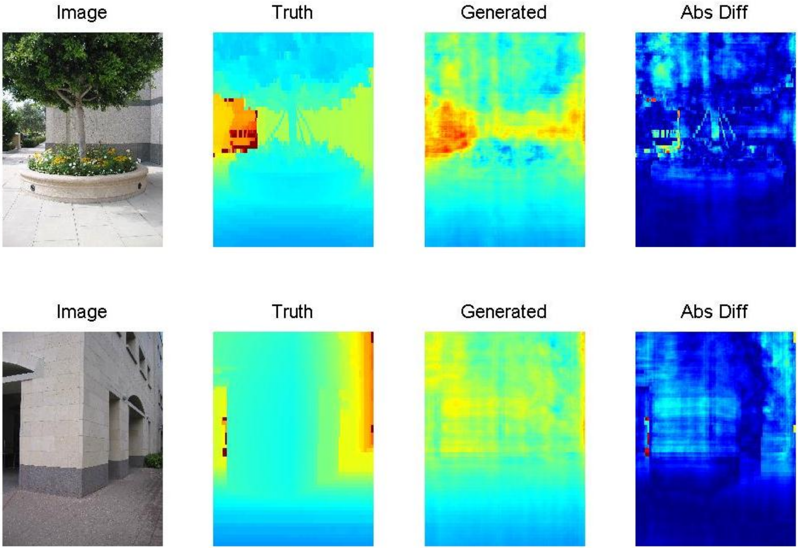


Image
Truth
Generated
Abs Diff
Image
Truth
Generated
Abs Diff

Difficulties

- Insurmountable difficulties:
  - 2nd term dropped from the model as the paper doesn't provide the MAP inference method.
  - Laplacian wasn't successfully implemented, as I couldn't find a correct way to solve the linear program in the MAP inference.
- Surmountable difficulties:
  - The cost to calculate the absolute features of images
  - 1704x2272 image had a feature "matrix" that cost about 280MB and without vectorization 300s to compute.
  - With some clever optimizations of pre-computing the require information, this was dropped to 30.
  - Can't load all image features to memory (100+ GB) for training.
  - Loading whole image features costs around 300s and around 400s for linear regression.
  - Save each image's features by rows reduces training time from (3 days to 1.3 days).

Importance and Future Gold

- The problem of generating depth is still an extremely interesting problem that humans seems to be much better at.
- The author takes a interesting approach to generated the right features that has the most information that has a regard to depth.
- The value of the author is more as a augmenting method inorder to generate accurate depth maps.
- The work produces a very general depth map that is extremely useful for simple inferences about the environment.
- It is interesting to see what other features detctors we can add to possible make this method more robust.
- It is an good example just how much information a standard image can hold.
- The method is potentially fast enough given GPU acceleration for real time robotics applications, to augment stereo depth estimation.