An Overview of Computer Vision & Computational Photography

CSE 591: Sep 19, 2018
What is Computer Vision?

Computational Systems to make sense of the physical world by looking at images and videos
INTRODUCTION

Recognize Objects
INTRODUCTION

Classify Scene
INTRODUCTION

Motion / Action
INTRODUCTION

Identify Materials

Foliage

Ceramic
Surface Properties

Wet?

Slippery?
INTRODUCTION

Computer Vision

- Develop Algorithms that extract a description of the world from images

Computational Photography

- Think of modified cameras and acquisition setups that make this extraction easier
INTRODUCTION

Broad Overview of (many a) Vision Algorithm
INTRODUCTION

Broad Overview of (many a) Vision Algorithm

1. Understand the Image Formation Model: Scene to Image

\[ I = \mathcal{F}(S) \]
Broad Overview of (many a) Vision Algorithm

1. Understand the Image Formation Model: Scene to Image

\[ I = \mathcal{F}(S) \]

2. Invert the Model: Gives us Multiple Physically Feasible Solutions
INTRODUCTION

Broad Overview of (many a) Vision Algorithm

1. Understand the Image Formation Model: Scene to Image

\[ I = \mathcal{F}(S) \]

2. Invert the Model: Gives us Multiple Physically Feasible Solutions

\[ \{S\} = \mathcal{F}^{-1}(I) \]

3. Learn What Natural Scenes Look Like: Use to select likely scene among those that are feasible
Broad Overview of (many a) Vision Algorithm

1. Understand the Image Formation Model: Scene to Image

\[ I = \mathcal{F}(S) \]

4. Computational Photography: Modify the Image Formation Model to make measurements more informative

\[ \{S\} = \mathcal{F}^{-1}(I) \]

2. Invert the Model: Gives us Multiple Physically Feasible Solutions

3. Learn What Natural Scenes Look Like: Use to select likely scene among those that are feasible
**Broad Overview of (many a) Vision Algorithm**

1. Understand the Image Formation Model: Scene to Image

   \[ I = \mathcal{F}(S) \]

2. Invert the Model: Gives us Multiple Physically Feasible Solutions

   \[ \{S\} = \mathcal{F}^{-1}(I) \]

3. Learn What Natural Scenes Look Like: Use to select likely scene among those that are feasible

4. Computational Photography: Modify the Image Formation Model to make measurements more informative
INTRODUCTION

Broad Overview of (many a) Vision Algorithm

1. Understand the Image Formation Model: Scene to Image

\[ I = \mathcal{F}(S) \]

2. Invert the Model: Gives us Multiple Physically Feasible Solutions

\[ \{S\} = \mathcal{F}^{-1}(I) \]

3. Learn What Natural Scenes Look Like: Use to select likely scene among those that are feasible

4. Computational Photography: Modify the Image Formation Model to make measurements more informative
REAL-WORLD IMPACT

ICCP 2018 Sponsors

Gold
- Qualcomm
- Intel

Silver
- Adobe
- Oculus
- Google

Bronze
- Samsung
- Omron
- Snap Inc.
- NEC Laboratories America
- Amazon
- Quasar MR
WHAT DOES VISION RESEARCH LOOK LIKE?

As a Grad Student working on a problem, you will have to:

- Understand the physics, geometry, optics, etc. of the setup.
- Understand to what degree the image formation process is invertible, characterize the ambiguity.
- Figure out how to the statistics of natural images could resolve this ambiguity.
- Use this to choose a model / architecture.
- Figure out how to train / learn parameters of this model.
- Develop an algorithm to use this model for actual inference.
- Make sure this is efficient and practical.
Photometric Stereo

Use the fact that intensity depends on relative angle between surface normal and light source.
Photometric Stereo

Use the fact that intensity depends on relative angle between surface normal and light source.

\[ \rho \] Albedo (Surface Color)
Photometric Stereo

Use the fact that intensity depends on relative angle between surface normal and light source.

Surface Orientation

\( \hat{n} \)

\( \rho \) Albedo (Surface Color)
Photometric Stereo

Use the fact that intensity depends on relative angle between surface normal and light source.
Photometric Stereo

Use the fact that intensity depends on relative angle between surface normal and light source.

\[ I = \rho \cos \theta \]
**Photometric Stereo**

Use the fact that intensity depends on relative angle between surface normal and light source.

\[
I = \rho \cos \theta = \rho \langle \hat{n}, \ell \rangle
\]
Photometric Stereo

Use the fact that intensity depends on relative angle between surface normal and light source.

\[ I = \rho \cos \theta = \rho \langle \hat{n}, \ell \rangle \]
Photometric Stereo

Use the fact that intensity depends on relative angle between surface normal and light source.

\[ I = \rho \cos \theta = \rho \langle \hat{n}, \ell \rangle \]

One Observation: Three Unknowns
Photometric Stereo

Use the fact that intensity depends on relative angle between surface normal and light source.

\[ I = \rho \cos \theta = \rho \langle \hat{n}, \ell \rangle \]

One Observation: Three Unknowns
Take multiple images with different lighting
Photometric Stereo
CASE STUDY: SHAPE FROM CONTROLLED LIGHTING

Photometric Stereo

Great, but requires you to take multiple images. What if the object is moving?
CASE STUDY: SHAPE FROM CONTROLLED LIGHTING

RGB Photometric Stereo

Take three shots in one: use an RGB Camera

\[ I = \rho < \hat{n}, \ell > \]
RGB Photometric Stereo

Take three shots in one: use an RGB Camera

\[
I_R = \rho_R \langle \hat{n}, \ell_R \rangle \\
I_G = \rho_G \langle \hat{n}, \ell_G \rangle \\
I_B = \rho_B \langle \hat{n}, \ell_B \rangle
\]
CASE STUDY: SHAPE FROM CONTROLLED LIGHTING

RGB Photometric Stereo

Take three shots in one: use an RGB Camera

But now we have extra unknowns for surface color

\[
\begin{align*}
I_R &= \rho_R \langle \hat{n}, \ell_R \rangle \\
I_G &= \rho_G \langle \hat{n}, \ell_G \rangle \\
I_B &= \rho_B \langle \hat{n}, \ell_B \rangle
\end{align*}
\]

\(\hat{n}\) Albedo (Surface Color)
RGB Photometric Stereo

Take three shots in one: use an RGB Camera

But now we have extra unknowns for surface color

Solution 1: Measure albedo separately (assuming it's constant)

\[ I_R = \rho_R \langle \hat{n}, \ell_R \rangle \]
\[ I_G = \rho_G \langle \hat{n}, \ell_G \rangle \]
\[ I_B = \rho_B \langle \hat{n}, \ell_B \rangle \]
RGB Photometric Stereo

Take three shots in one: use an RGB Camera

But now we have extra unknowns for surface color

Solution 1: Measure albedo separately (assuming it's constant)

Solution 2: Paint the object, so that albedo is known and constant

\[
\begin{align*}
I_R &= \rho_R < \hat{n}, \ell_R > \\
I_G &= \rho_G < \hat{n}, \ell_G > \\
I_B &= \rho_B < \hat{n}, \ell_B >
\end{align*}
\]
Single-image RGB Photometric Stereo With Spatially-varying Albedo

Ayan Chakrabarti  
TTI-Chicago

Kalyan Sunkavalli  
Adobe Research

Abstract

We present a single-shot system to recover surface geometry of objects with spatially-varying albedos, from images captured under a calibrated RGB photometric stereo setup—with three light directions multiplexed across different color channels in the observed RGB image. Since the problem is ill-posed point-wise, we assume that the albedo map can be modeled as piece-wise constant with a restricted number of distinct albedo values. We show that under ideal conditions, the shape of a non-degenerate local constant albedo surface patch can theoretically be recovered exactly. Moreover, we present a practical and efficient algorithm that uses this model to robustly recover shape from real images. Our method first reasons about shape locally in a dense set of patches in the observed image, producing shape distributions for every patch. These local distributions are then combined to produce a single consistent surface normal map. We demonstrate the efficacy of the approach through experiments on both synthetic renderings as well as real captured images.

In this paper, we show that efficient and high-quality surface recovery from a single image is possible, when using a calibrated lighting environment that is specifically chosen to be directly informative about shape. Specifically, we use the RGB (or color) photometric stereo (RGB-PS) setup [3, 11, 14], where an object is illuminated by three monochromatic directional light sources, such that each of the red, green, and blue channels in the observed image is “lit” from a different direction. For natural lighting, directional diversity in color has been shown to be informative towards shape [10]. But the benefits of this lighting setup for shape recovery can be better understood by interpreting it as one that multiplexes the multiple images of classical PS into the different color channels of a single image.
Proposition 1. Given noiseless observed intensities \( v(p) \) at a set of locations \( p \in \Omega \) on a diffuse surface patch known to have constant albedo, i.e., \( \kappa(p) = \kappa_\Omega, \forall p \in \Omega \), the true surface normals \( \{ \hat{n}(p) : p \in \Omega \} \) and common albedo \( \kappa_\Omega \) are uniquely determined, if:

1. All intensities \( v(p) \) are strictly positive.
2. The true surface is non-degenerate in the sense that the set \( \{ \hat{n}(p)^T \hat{n}(p) : p \in \Omega \} \), of outer-products of the true normal vectors, span the space \( \text{Sym}_3 \) of all \( 3 \times 3 \) symmetric matrices.

Proof: Given \( \kappa_\Omega \) and \( \hat{n}(p) \) as the true patch albedo and normals, let \( \kappa'_\Omega, \hat{n}'(p) \) be a second solution pair that also explains the observed intensities \( v(p) \) in the patch \( \Omega \). Since the observed intensities are strictly positive, this implies that the albedos \( \kappa_\Omega, \kappa'_\Omega \) are strictly positive as well, and further that no point is in shadow under any of the lights, i.e. \( L^T \hat{n}(p), L^T \hat{n}'(p) > 0, \forall p \in \Omega \). Then, since \( L^T \) is invertible, we can write

\[
\text{diag}[\kappa_\Omega] L^T \hat{n}(p) = \text{diag}[\kappa'_\Omega] L^T \hat{n}'(p) \\
\Rightarrow \hat{n}'(p) = A \hat{n}(p), \forall p \in \Omega,
\]

(2)

where we define the matrix \( A = L^{-T} R L^T \), with \( R = \text{diag}[\kappa'_\Omega]^{-1} \text{diag}[\kappa_\Omega] \) being a diagonal matrix whose entries

Analyze ambiguities, and show that for “most” local regions, if the albedo is constant inside the region, we can recover its shape and albedo uniquely.
Proposition 1. Given noiseless observed intensities \( v(p) \) at a set of locations \( p \in \Omega \) on a diffuse surface patch known to have constant albedo, i.e., \( \kappa(p) = \kappa_\Omega, \forall p \in \Omega \), the true surface normals \( \{\hat{n}(p) : p \in \Omega\} \) and common albedo \( \kappa_\Omega \) are uniquely determined, if:

1. All intensities \( v(p) \) are strictly positive.
2. The true surface is non-degenerate in the sense that the set \( \{\hat{n}(p)\hat{n}(p)^T : p \in \Omega\} \), of outer-products of the true surface normals, is non-degenerate, which means that the matrix \( L \) is invertible, i.e., \( \text{det}(L) \neq 0 \).

Proof: We prove that if the surface normals, \( \hat{n}(p) \), are non-degenerate, then the observed intensity is uniquely determined. Indeed, suppose \( L \) is invertible. Then, the equation \( v = L\kappa \) can be solved for \( \kappa \) uniquely, and consequently, \( \kappa_\Omega \) can be recovered.

Analyze ambiguities, and show that for "most" local regions, if the albedo is constant inside the region, we can recover its shape and albedo uniquely.

But we don't know which patches are constant albedo, and which have boundaries in them.
**Proposition 1.** Given noiseless observed intensities \( v(p) \) at a set of locations \( p \in \Omega \) on a diffuse surface patch known to have constant albedo, i.e., \( \kappa(p) = \kappa_\Omega, \forall p \in \Omega \), the true surface normals \( \{\hat{n}(p) : p \in \Omega\} \) and common albedo \( \kappa_\Omega \) are uniquely determined, if:

1. All intensities \( v(p) \) are strictly positive.
2. The true surface is non-degenerate in the sense that the set \( \{\hat{n}(p)\hat{n}(p)^T : p \in \Omega\} \), of outer-products of the true normals, is non-degenerate. More precisely, any subset of \( \Omega \) has a non-zero determinant, i.e. \( L^T L \) is invertible.

**Proof:** Using this proposition, one could analyze ambiguities, and show that for "most" local regions, if the albedo is constant inside the region, we can recover its shape and albedo uniquely.

But we don't know which patches are constant albedo, and which have boundaries in them.

Also, uniqueness holds in idealized conditions. In reality, we'll have noise, 'non diffuse' reflection, ....
CASE STUDY: SHAPE FROM CONTROLLED LIGHTING

RGB Photometric Stereo Setup

Inference Algorithm

Build Albedo Histogram of scores from all patches & find peaks

Global Albedo Set

Local Distributions for Each Patch Set of candidate shapes and scores, one for each albedo in global set.

Final Estimate of Object Shape

Dense set of Overlapping Patches over Observed Image

Score

Globalization

Surface Normal $\hat{n}$

Surface Albedo $[\kappa_R, \kappa_G, \kappa_B]$

$v_B = \kappa_B \frac{I_B}{\hat{n}}$
$v_G = \kappa_G \frac{I_G}{\hat{n}}$
$v_R = \kappa_R \frac{I_R}{\hat{n}}$
CASE STUDY: SHAPE FROM CONTROLLED LIGHTING
CASE STUDY: MONOCULAR DEPTH ESTIMATION

Depth from a Single Image
CASE STUDY: MONOCULAR DEPTH ESTIMATION

Depth from a Single Image

- No explicit geometric / optical cues.
- Must learn to map familiar patterns to depth.

  Shading.
  Contours & Boundaries.
  Foreshortening of regular patterns.
  Scale of familiar objects.
CASE STUDY: MONOCULAR DEPTH ESTIMATION

$Z(n)$
CASE STUDY: MONOCULAR DEPTH ESTIMATION

\[ Z(n) \] Lots of numbers
(200k for a 500x400 image)
CASE STUDY: MONOCULAR DEPTH ESTIMATION

\[ Z(n) \]

Estimate each \( Z(n) \) independently.
CASE STUDY: MONOCULAR DEPTH ESTIMATION

\[ Z(n) \]

Estimate each \( Z(n) \) independently.
CASE STUDY: MONOCULAR DEPTH ESTIMATION

Estimate each $Z(n)$ independently and *locally*. 

$Z(n)$
Estimate each $Z(n)$ independently and locally.
CASE STUDY: MONOCULAR DEPTH ESTIMATION

Estimate each $Z(n)$ independently and \textit{locally}.

Problem: Local information may be ambiguous.
Estimate each $Z(n)$ independently and \textit{locally}.

Problem: Local information may be ambiguous.
CASE STUDY: MONOCULAR DEPTH ESTIMATION

Problem: Local information may be ambiguous.

Estimate each \(Z(n)\) independently and *locally*.

Scene Maps have Structure
CASE STUDY: MONOCULAR DEPTH ESTIMATION

Scene Maps have Structure
CASE STUDY: MONOCULAR DEPTH ESTIMATION

\[ Z(n) \]

Build an algorithm that effectively extracts and exploits this structure

Scene Maps have Structure
Local Estimation: Mid-level Representation

Output Map

\[ Z(n) \]

Input
CASE STUDY: MONOCULAR DEPTH ESTIMATION

Local Estimation: Mid-level Representation

Output Map

$Z(n)$
Local Estimation: Mid-level Representation

\[ Z(n) \]

Derivative Filters
- Different Scales
- Different Orders
- Different Orientations

\[ \{w_i(n) = (k_i \ast Z)(n)\} \forall i, n \]
Local Estimation: Mid-level Representation

\[ Z(n) = 1/d(n) \]

\[ \{w_i(n) = (k_i \ast Z)(n)\} \forall i, n \]

Derivative Filters
- Different Scales
- Different Orders
- Different Orientations
Local Estimation: Mid-level Representation

Perspective Camera

\[ d(x, y) \Rightarrow (dx, dy, d) \]

World 3D Co-ordinates

Plane Equation

\[
\frac{1}{d} = \alpha x + \beta y + \gamma
\]

Zeroth Derivative = Absolute depth
First Derivative = Surface orientation
Second Derivative = 0: Planar
Curvature, contours.
CASE STUDY: MONOCULAR DEPTH ESTIMATION

Local Estimation: Mid-level Representation

\[ Z(n) = \frac{1}{d(n)} \]

\[ \{w_i(n) = (i \ast Z)(n)\} \forall i, n \]
Local Estimation: Mid-level Representation

Depth Map

\[ Z(n) = 1/d(n) \]

\( \{w_i(n)\} \) Derivatives of Depth

Input Image
CASE STUDY: MONOCULAR DEPTH ESTIMATION

Local Estimation: Mid-level Representation

\[ Z(n) = \frac{1}{d(n)} \]

Depth Map

\( \{w_i(n)\} \) Derivatives of Depth

Convolutional Neural Network

Input Image
Local Estimation: Mid-level Representation

CNN → \{ W_i \}
Derivatives
Local Estimation: Mid-level Representation

Distribution over Each Derivative

CNN
Local Estimation: Mid-level Representation

Distribution over Each Derivative
CASE STUDY: MONOCULAR DEPTH ESTIMATION

Local Estimation: Mid-level Representation

\[ p(w_i(n)) = \sum_{j=1}^{M} \tilde{p}_i^j(n) \frac{1}{\sqrt{2\pi}\sigma_i} \exp \left( -\frac{|w_i(n) - c_i^j|^2}{2\sigma_i^2} \right) \]

Convnet learns to predict mixing probabilities

Fixed prior to network training, using k-means on GT depth maps.
CASE STUDY: MONOCULAR DEPTH ESTIMATION

CNN

\{p(w_i(n))\}
CASE STUDY: MONOCULAR DEPTH ESTIMATION

CNN

Local window around n

\{p(w_i(n))\}
CNN

CASE STUDY: MONOCULAR DEPTH ESTIMATION

Local window around n

Scene Features

$\{p(w_i(n))\}$
Training

Ground Truth Depth

Input Image
CASE STUDY: MONOCULAR DEPTH ESTIMATION

Training

$\bar{w}_i(n)$

$\{p(w_i(n))\}$

CNN

Ground Truth Depth

Input Image
Training

\[ \hat{q}_i^j(n) \propto \exp \left( -\frac{|\bar{w}_i(n) - c_i^j|^2}{2\sigma_i^2} \right) \]

\[ L = -\sum_{i,n} \sum_{j=1}^{M} \hat{q}_i^j(n) \left( \log \hat{p}_i^j(n) - \log \hat{q}_i^j(n) \right) \]

KL-Divergence
Training

Ground Truth

Ground Truth Depth

Input Image
CASE STUDY: MONOCULAR DEPTH ESTIMATION

Training

$\mathcal{E}(x)$

Ground Truth Depth

Input Image
CASE STUDY: MONOCULAR DEPTH ESTIMATION

Training

\[ f(x) \]

Ground Truth Depth

Input Image
CASE STUDY: MONOCULAR DEPTH ESTIMATION

Training

\[ \mathcal{L}(\theta) \]

 Ground Truth Depth

 Input Image

\[ > \]

\[ > \]
Training

Ground Truth Depth

Input Image
Globalization
Globalization
Globalization

CNN \rightarrow \{p(w_i(n))\}
Globalization

CNN $\{p(w_i(n))\}$ $Z(n)$
Globalization
CASE STUDY: MONOCULAR DEPTH ESTIMATION

Globalization

\[ \{ p(w_i(n)) \} \]

\[ Z(n) \]
Globalization

\[ Z = \arg \max_Z \sum_{i,n} \log p_{i,n} \left( (Z * k_i)(n) \right) \]

From CNN
CASE STUDY: MONOCULAR DEPTH ESTIMATION

Globalization

\[
Z = \arg \max_Z \sum_{i,n} \log p_{i,n} \((Z \ast k_i)(n)\)
\]

\[
Z = \arg \min_Z \min_{\{w_i(n)\}} \left[ \sum_{i,n} \log p_{i,n} \(w_i(n)\) \right]
\]

Auxiliary Vars
for Derivatives
Globalization

\[
Z = \arg \max_Z \sum_{i,n} \log p_{i,n} ((Z \ast k_i)(n))
\]

\[
Z = \arg \min_Z \min_{\{w_i(n)\}} \left[ \sum_{i,n} \log p_{i,n} (w_i(n)) \right] + \frac{\beta}{2} \left[ \sum_{i,n} |w_i(n) - (Z \ast k_i)(n)|^2 \right]
\]

Auxiliary Vars for Derivatives
CASE STUDY: MONOCULAR DEPTH ESTIMATION

Globalization

\[ Z = \mathop{\arg\max}_{Z} \sum_{i,n} \log p_{i,n} ((Z \ast k_i)(n)) \]

\[ Z = \mathop{\arg\min}_{Z} \min_{\{w_i(n)\}} - \left[ \sum_{i,n} \log p_{i,n} (w_i(n)) \right] + \frac{\beta}{2} \left[ \sum_{i,n} |w_i(n) - (Z \ast k_i)(n)|^2 \right] \]

Auxiliary Vars for Derivatives

Equivalent as \( \beta \to \infty \)
Globalization

\[ Z = \arg \min_Z \min_{\{w_i(n)\}} \left[ \sum_{i,n} \log p_{i,n}(w_i(n)) \right] + \frac{\beta}{2} \left[ \sum_{i,n} |w_i(n) - (Z \ast k_i)(n)|^2 \right] \]
Globalization

\[
Z = \arg \min_Z \min_{\{w_i(n)\}} \left[ \sum_{i,n} \log p_{i,n}(w_i(n)) \right] + \frac{\beta}{2} \left[ \sum_{i,n} |w_i(n) - (Z * k_i)(n)|^2 \right]
\]

Alternatingly minimize $Z$ and $\{w_i(n)\}$, keeping the other constant.
CASE STUDY: MONOCULAR DEPTH ESTIMATION

Globalization

Fix w, minimize wrt Z

Efficient least-squares in the Fourier-domain.

\[ Z = \arg \min_Z \min_{\{w_i(n)\}} \left[ \sum_{i,n} \log p_{i,n}(w_i(n)) \right] + \frac{\beta}{2} \left[ \sum_{i,n} |w_i(n) - (Z * k_i)(n)|^2 \right] \]
Globalization

Fix $Z$, minimize wrt $w$

Independent for each $w_i(n)$

\[
Z = \arg \min_Z \min_{\{w_i(n)\}} \left[ \sum_{i,n} \log p_{i,n}(w_i(n)) \right] + \frac{\beta}{2} \left[ \sum_{i,n} |w_i(n) - (Z * k_i)(n)|^2 \right]
\]
Experimental Results
Experimental Results

NYUv2 Depth Benchmark
- Ground truth data from Kinect.
- 56,000 training pairs, 100 validation.
- 654 Test scenes.
Experimental Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (within threshold)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>80.6%</td>
</tr>
<tr>
<td>Eigen 2015</td>
<td>76.9%</td>
</tr>
<tr>
<td>Liu 2015</td>
<td>61.4%</td>
</tr>
<tr>
<td>Eigen 2014</td>
<td>61.4%</td>
</tr>
<tr>
<td>Baig 2016</td>
<td>61.0%</td>
</tr>
<tr>
<td>Wang 2015</td>
<td>60.5%</td>
</tr>
</tbody>
</table>
Experimental Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>80.6%</td>
</tr>
<tr>
<td>Eigen 2015</td>
<td>76.9%</td>
</tr>
<tr>
<td>Liu 2015</td>
<td>61.4%</td>
</tr>
<tr>
<td>Eigen 2014</td>
<td>61.4%</td>
</tr>
<tr>
<td>Baig 2016</td>
<td>61.0%</td>
</tr>
<tr>
<td>Wang 2015</td>
<td>60.5%</td>
</tr>
</tbody>
</table>

- **Pair-wise MRFs**
- **Direct Regression**
### Experimental Results

<table>
<thead>
<tr>
<th>Method</th>
<th>RMSE (lin.)</th>
<th>RMSE (log)</th>
<th>Abs Rel.</th>
<th>Sqr Rel.</th>
<th>δ &lt; 1.25</th>
<th>δ &lt; 1.25^2</th>
<th>δ &lt; 1.25^3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>0.620</td>
<td>0.205</td>
<td>0.149</td>
<td>0.118</td>
<td>80.6%</td>
<td>95.8%</td>
<td>98.7%</td>
</tr>
<tr>
<td>Eigen 2015</td>
<td>0.641</td>
<td>0.214</td>
<td>0.158</td>
<td>0.121</td>
<td>76.9%</td>
<td>95.0%</td>
<td>98.8%</td>
</tr>
<tr>
<td>Wang 2015</td>
<td>0.745</td>
<td>0.262</td>
<td>0.220</td>
<td>0.210</td>
<td>60.5%</td>
<td>89.0%</td>
<td>97.0%</td>
</tr>
<tr>
<td>Baig 2016</td>
<td>0.802</td>
<td>-</td>
<td>0.241</td>
<td>-</td>
<td>61.0%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Eigen 2014</td>
<td>0.877</td>
<td>0.283</td>
<td>0.214</td>
<td>0.204</td>
<td>61.4%</td>
<td>88.8%</td>
<td>97.2%</td>
</tr>
<tr>
<td>Liu 2015</td>
<td>0.824</td>
<td>-</td>
<td>0.230</td>
<td>-</td>
<td>61.4%</td>
<td>88.3%</td>
<td>97.1%</td>
</tr>
<tr>
<td>Zoran 2015</td>
<td>1.22</td>
<td>0.43</td>
<td>0.41</td>
<td>0.57</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Pair-wise MRFs**

- 80.6%
- 76.9%
Experimental Results

Input Image

Ground Truth Depth
Experimental Results

Input Image

Ground Truth Depth

Proposed Method
Experimental Results

Input Image

Ground Truth Depth

Proposed Method

Eigen 2015 (VGG)
CASE STUDY: MONOCULAR DEPTH ESTIMATION

Experimental Results

Input Image

Ground Truth Depth

Proposed Method

Eigen 2015 (VGG)
Experimental Results

Input Image

Ground Truth Depth

Proposed Method

Eigen 2015 (VGG)
Experimental Results

Input Image

Ground Truth Depth

Proposed Method

Eigen 2015 (VGG)
Beyond the Benchmark

\[ Z = \arg \max_z \sum_{i, n} \log p_{i,n} ((Z \ast k_i)(n)) \]
CASE STUDY: MONOCULAR DEPTH ESTIMATION

Beyond the Benchmark

\[ Z = \arg \max_z \sum_{i,n} \log p_{i,n} ((Z \ast k_i)(n)) \]

Rich, distributional, interpretable
Beyond the Benchmark

\[ Z = \arg \max_Z \sum_{i,n} \log p_{i,n} ((Z \ast k_i)(n)) \]
CASE STUDY: MONOCULAR DEPTH ESTIMATION

Beyond the Benchmark

\[ Z = \arg \max_Z \sum_{i,n} \log p_{i,n} ((Z \ast k_i)(n)) \]

Other cues.
User input.
Noisy / sparse depth measurements.
Beyond the Benchmark

\[ Z = \arg \max_Z \sum_{i,n} \log p_{i,n} ((Z \ast k_i)(n)) \]

Other cues.
User input.
Noisy / sparse depth measurements.

Common substrate for local estimates from different cues.
Beyond the Benchmark

\[ Z = \arg \max_Z \sum_{i,n} \log p_{i,n} ((Z \ast k_i)(n)) \]
Beyond the Benchmark

\[ Z = \arg \max_Z \sum_{i,n} \log p_{i,n} ((Z \ast k_i)(n)) \]

\[ P(Z(n) < \delta) \]
DISCUSSION

- Flavor of what a research project looks like.
- Look at group website for papers describing some of our other recent work.

Questions?