Algorithms for Single-View Depth Image Estimation

Fangchang Ma (AeroAstro & LIDS)
PhD Thesis Defense
May 10, 2019

Thesis committee: Sertac Karaman, John Leonard, Nicholas Roy
External Readers: David Rosen, Michael Boulet
Algorithms for Single-View Depth Image Estimation

- Motivation
- Applications of Single-View Depth Estimation
- Algorithmic Challenges
- Contributions
Algorithms for Single-View Depth Image Estimation

• Motivation
• Applications of Single-View Depth Estimation
• Algorithmic Challenges
• Contributions
3D Reconstruction and Autonomous Navigation
Stanford Cart (Moravec, 1979)
3D Reconstruction and Autonomous Navigation

Stanford Cart (Moravec, 1979)

DARPA Urban Challenge (Leonard et al 2008)
3D Reconstruction and Autonomous Navigation

Stanford Cart (Moravec, 1979)

DARPA Urban Challenge (Leonard et al 2008)

Indoor Flight (Richter, Bry, Roy 2013)
3D Reconstruction and Autonomous Navigation

Stanford Cart (Moravec, 1979)

DARPA Urban Challenge (Leonard et al 2008)

Indoor Flight (Richter, Bry, Roy 2013)

Navigation in forests (Skydio 2018)
3D Reconstruction and Augmented Reality
3D Reconstruction and Augmented Reality

Apple FaceID and “Animoji”
3D Reconstruction and Augmented Reality

Apple FaceID and “Animoji”  
Snapchat “Landmarkers”
Depth Sensing Techniques

- Stereo Cameras
- Structured-Light Sensors
- LiDARs
Stereo Cameras

- Low precision at texture-less regions [2].
- Long baseline needed for high accuracy [3].

1. Pillai et al, ICRA, 2016
2. Seitz et al, CVPR, 2006
Structured-light Sensors

- Resolution goes down with ambient light, given limited power budget [1]

• LiDARs: sparse measurements (4% of image pixels) [1][2]

1. Uhrig, Jonas et al, IEEE 3DV, 2017
2. Magden et al, Nature Communications, 2018
Common Challenge: Resolution

- Difficulty in achieving high resolution that is also dense
Single-View Depth Image Estimation
Algorithms for Single-View Depth Image Estimation

- Motivation
- Applications of Single-View Depth Estimation
- Algorithmic Challenges
- Contributions
Application 1: Sensor Enhancement / Miniaturization

Kinect [1]

[Image of Kinect sensor data]

Velodyne LiDAR [2]

[Image of LiDAR sensor data]

2. Ma, Cavalheiro, Karaman, ICRA, 2019
Application 2: Sparse Map Densification

• State-of-the-art, real-time SLAM algorithms are mostly (semi) feature-based, resulting in a sparse map representation

PTAM [1]  

LSD-SLAM [2]

• Depth completion as a downstream, post-processing step for sparse SLAM algorithms, creating a dense map representation

2. Engel et al, ECCV, 2014
Algorithms for Single-View Depth Image Estimation

- Motivation
- Applications of Single-View Depth Estimation
- Algorithmic Challenges
- Contributions
Challenges in Depth Completion

• Extreme sparse input (5% or below)
• Biased spatial sampling (e.g., at the bottom)
• Cross-modality fusion (RGB + Depth) [1]-[3]
• Lack of dense ground truth annotations for depth data [1][2]
• Lack of performance guarantees

1. Uhrig, Jonas et al., IEEE 3DV, 2017
Algorithms for Single-View Depth Image Estimation

- Motivation
- Applications of Single-View Depth Estimation
- Algorithmic Challenges
- Contributions
Model-Based Approach

1. An algorithm for efficient completion of piecewise-planar surfaces

2. A deep regression network and a self-supervised training framework

3. A generative-network inversion algorithm for perfect reconstruction

Data-Driven Approach
Model-Based Approach

1. An algorithm for efficient completion of piecewise-planar surfaces

2. A deep regression network and a self-supervised training framework

3. A generative-network inversion algorithm for perfect reconstruction

Data-Driven Approach
Contribution 1: Piecewise-planar surface reconstruction

Fangchang Ma, Luca Carlone, Ulas Ayaz, Sertac Karaman. “Sparse sensing for resource-constrained depth reconstruction”. IROS’16

Piecewise-Planar Assumption

• Planar Assumption: a relatively structured environment can be well approximated by a small number of planar surfaces

• Observation: 2nd derivative of planar surfaces is sparse

• Implication: 2nd derivative of a structured environment is approximately sparse (sparsity of 2nd derivative is a measure of scene complexity)
Problem Formulation

Goal: find the simplest depth image (with the sparsest 2nd derivative) that is aligned with our measurements

\[
\min_x \|\Delta x\|_0, \quad \text{subject to } y = Ax
\]

Convex Relaxation (Linear Programming):

\[
\min_x \|\Delta x\|_1, \quad \text{subject to } y = Ax
\]

Noisy Measurements (Linear Programming):

\[
\min_x \|\Delta x\|_1, \quad \text{subject to } \|y - Ax\|_\infty < \epsilon
\]
Theoretical Results

Theorem 17 (3D Sign Consistency ⇒ Optimality). Let $Z \in \mathbb{R}^{r \times c}$ be a 3D profile, feasible for problem (L1$\Delta$). Assume the sample set $\mathcal{M}$ is a grid sample set. Then $Z$ is in the set of minimizers of (L1$\Delta$) if it is 3D sign consistent.

Proposition 18 (3D Recovery Error - noiseless samples). Let $Z^\circ \in \mathbb{R}^{r \times c}$ be the ground truth profile generating noiseless measurements (4). Let $\mathcal{M}$ be a grid sampling set and assume $Z^\circ$ to be 3D sign consistent with respect to $\mathcal{M}$. Moreover, let $Z \in \mathbb{R}^{r \times c}$ and $\tilde{Z} \in \mathbb{R}^{r \times c}$ be the point-wise lower and upper bound of the row-wise envelope, built as in Fig. 5(b) by considering each row of the 3D depth profile as a 2D profile. Then, $Z^\circ$ is an optimal solution of (L1$\Delta$), and any other optimal solution $Z^*$ of (L1$\Delta$) satisfies:

$$|Z_{i,j}^\circ - Z_{i,j}^*| \leq \max(|Z_{i,j} - Z_{i,j}^*|, |\tilde{Z}_{i,j} - Z_{i,j}^*|)$$  \hspace{1cm} (16)
Experimental Results: comparison against interpolation
Contribution 1: An algorithm for efficient completion of 3D surfaces

1. Planar Assumption: 2nd-derivative of depth images are typically sparse

2. Formulation: linear programming

\[
\begin{align*}
\min_x & \|\Delta x\|_1, \\
\text{subject to} & \|y - Ax\|_\infty < \epsilon
\end{align*}
\]

3. Theoretical Results: conditions for exact recovery and error bounds

4. Experimental Results: highest accuracy and fastest runtime among optimization-based methods, and outperforms deep-learning methods under certain conditions

Pros:
- High accuracy with planar, indoor environments
- Low computational complexity
- Performance guarantees (with uniform sampling)
- No tuning/learning required

Cons:
- Does not generalize to more complicated, outdoor scenes
- Does not generalize to biased sampling patterns.
Model-Based Approach

1. An algorithm for efficient completion of piecewise-planar surfaces

2. A deep regression network and a self-supervised training framework

3. A generative-network inversion algorithm for perfect reconstruction

Data-Driven Approach
Contribution 2: Deep Regression Network and Self-Supervised Learning

- Supervised Training
- Self-supervised Learning
Contribution 2: Deep Regression Network and Self-Supervised Learning

- Supervised Training
- Self-supervised Learning

Input: RGB + sd
Output: dense depth
Depth completion as a deep regression problem

- Direct encoding: use 0s to represent no-measurement
- Early-fusion strategy: concatenate RGB and sparse Depth at input level
- Network Architecture: standard convolutional neural network
- Train end-to-end using ground truth depth
Results on NYU Dataset

- RGB only: RMS=51cm
- RGB + 20 measurements: RMS=35cm
- RGB + 50 measurements: RMS=28cm
- RGB + 200 measurements: RMS=23cm
Experiment: RMSE=0.814m (ranked 1st on KITTI in 2018).
Scaling of Accuracy vs. Samples

![Graph showing the scaling of accuracy vs. samples for RGB and RGBd with sparse depth. The graph plots REL (Root Mean Square Error) on the y-axis against the number of depth samples on the x-axis. The RGB and RGBd lines show a decrease in error as the number of samples increases, with RGBd generally having a lower error for a given number of samples. The graph includes legends for RGBd, sparse depth, and RGB.]
Application to Sparse Point Clouds
Application to Sparse Point Clouds
Contribution 2: Deep Regression Network and Self-Supervised Learning

- Supervised Training
- Self-supervised Learning

Input: RGB + sd
Output: dense depth
Self-supervision: enforce temporal photometric consistency
Self-supervision: enforce temporal photometric consistency
Self-supervision: enforce temporal photometric consistency
Self-supervision: enforce temporal photometric consistency
Self-supervision: enforce temporal photometric consistency

Estimate pose from LiDAR and RGB

Real RGB₁

Real RGB₂
Self-supervision: enforce temporal photometric consistency

Estimate pose from LiDAR and RGB

Inverse warping using both depth and pose
Self-supervision: enforce temporal photometric consistency

Estimate pose from LiDAR and RGB

Inverse warping using both depth and pose
Self-supervision: enforce temporal photometric consistency

Estimate pose from LiDAR and RGB

Inverse warping using both depth and pose

Penalize photometric differences
Self-supervision: temporal photometric consistency

Supervised training requires ground truth depth labels, which are hard to acquire in practice.

![Diagram](image-url)
Experiment 2. Self-Supervised. RMSE=1.30m
Contribution 2: A deep regression network and a self-supervised training framework

A. Fusion of RGB and sparse depth images

B. A Self-supervised framework that trains the network using raw LiDAR and RGB images by enforcing temporal photometric consistency

C. Result: Ranked 1st on KITTI among published work

Pros:
- High accuracy (on average)
- Can deal with biased sampling
- Capable of utilizing the RGB image

Cons:
- Requires large amount of data (labeled or unlabelled)
- Implicitly assume similar distributions of train/test data
- Once trained, only adapt to a specific sampling pattern
- No performance guarantees
Model-Based Approach

1. An algorithm for efficient completion of piecewise-planar surfaces

2. A deep regression network and a self-supervised training framework

3. A generative-network inversion algorithm for perfect reconstruction

Data-Driven Approach
Assumption: image can be modeled by a convolutional generative neural network
Sub-sampling Process

Input $z$

Hidden Layers

Output $x=G(z)$

Samples $y=Ax$
Rephrasing the depth-completion/image-inpainting problems

Question: can you find $x$ (or equivalently, $z$), given only $y$?
Rephrasing the depth-completion/image-inpainting problems

If \( z \) is recovered, then we can reconstruct \( x \) as \( G(z) \) using a single forward pass.
The latent code $z$ can be computed efficiently using gradient descent

$$\hat{z} = \arg \min_z \| AG(z) - y \|^2$$
Algorithm: Two-Stage Reconstruction

Input: $A, G, y$

\[ \hat{z} = \arg\min_z \| AG(z) - y \|^2 \]

\[ \hat{x} = G(\hat{z}) \]

Output: $\hat{x}$
For a 2-layer network, the latent code $z$ can be recovered from the undersampled measurements $y$ using gradient descents (with high probability) by minimizing the empirical loss function.
Experimental Results (Multi-Layer Networks)

Undersampled Measurements

Reconstructed Images

Ground Truth
Contribution 3: A generative-network inversion algorithm for depth completion and image inpainting

A. Depth completion (and image inpainting) reformulated as a neural network inversion problem

B. Theoretical results: the latent code $z$ can be computed using gradient descent with performance guarantee

C. Experimental results: the latent code and reconstruction are perfect with sufficient measurements

Pros:
- Perfect reconstruction with high probability
- Adapt to different sampling patterns
- Performance guarantees

Cons:
- Strong assumption on given G network
- Slower than deep-learning-based approaches
- Implicit requirements on sampling
Summary

Model-Based

Contribution 1: An algorithm for efficient completion of 3D surfaces

Pros:
• High accuracy with planar, indoor environments
• Low computational complexity
• Performance guarantees
• No tuning/learning required

Cons:
• Does not generalize to more complicated, outdoor scenes
• Computation not highly parallelizable

Contributions 2: A generative-network inversion algorithm

Pros:
• Perfect reconstruction
• Adaptable to different sampling patterns
• Performance guarantees

Cons:
• Strong assumption on given G network
• Slower than deep-learning-based approaches
• Implicit requirements on sampling

Contributions 3: A generative-network inversion algorithm

Pros:
• High accuracy (on average)
• Can deal with biased sampling
• Capable of utilizing the RGB image

Cons:
• Requires large amount of data (labeled or unlabelled)
• Implicitly assumes similar distributions of train/test data
• Once trained, only adapts to a specific sampling pattern
• No performance guarantees

Data-Driven
Summary of Research Contributions and Publications

Contributions:
1. An algorithm for efficient completion of piecewise-planar surfaces [1][2]
2. A self-supervised training framework and a deep regression network for depth completion [3][4][5]

Related Publications:
1. [IROS’16] Sparse Sensing for Resource-Constrained Depth Reconstruction
2. [IJRR’19] Sparse Depth Sensing for Resource-Constrained Robots
3. [ICRA’18] Sparse-to-Dense: Depth Prediction from Sparse Depth Samples and a Single Image
4. [ICRA’19] Self-supervised Sparse-to-Dense: Self-supervised Depth Completion from LiDAR and Monocular Camera
6. [NIPS’18] Invertibility of Convolutional Generative Networks from Partial Measurements
Model-Based

Pros:
- High accuracy with planar, indoor environments
- Low computational complexity
- Performance guarantees
- No tuning/learning required

Cons:
- Does not generalize to more complicated, outdoor scenes
- Computation not highly parallelizable

Data-Driven

Contribution 1: An algorithm for efficient completion of 3D surfaces

Pros:
- High accuracy with planar, indoor environments
- Low computational complexity
- Performance guarantees
- No tuning/learning required

Cons:
- Does not generalize to more complicated, outdoor scenes
- Computation not highly parallelizable

Contribution 2: deep regression network and self-supervised training

Pros:
- High accuracy (on average)
- Can deal with biased sampling
- Capable of utilizing the RGB image

Cons:
- Requires large amount of data (labeled or unlabelled)
- Implicitly assume similar distributions of train/test data
- Once trained, only adapt to a specific sampling pattern
- No performance guarantees

Contribution 3: A generative-network inversion algorithm

Pros:
- Perfect reconstruction
- Adapt to different sampling patterns
- Performance guarantees

Cons:
- Strong assumption on given G network
- Slower than deep-learning-based approaches
- Implicit requirements on sampling