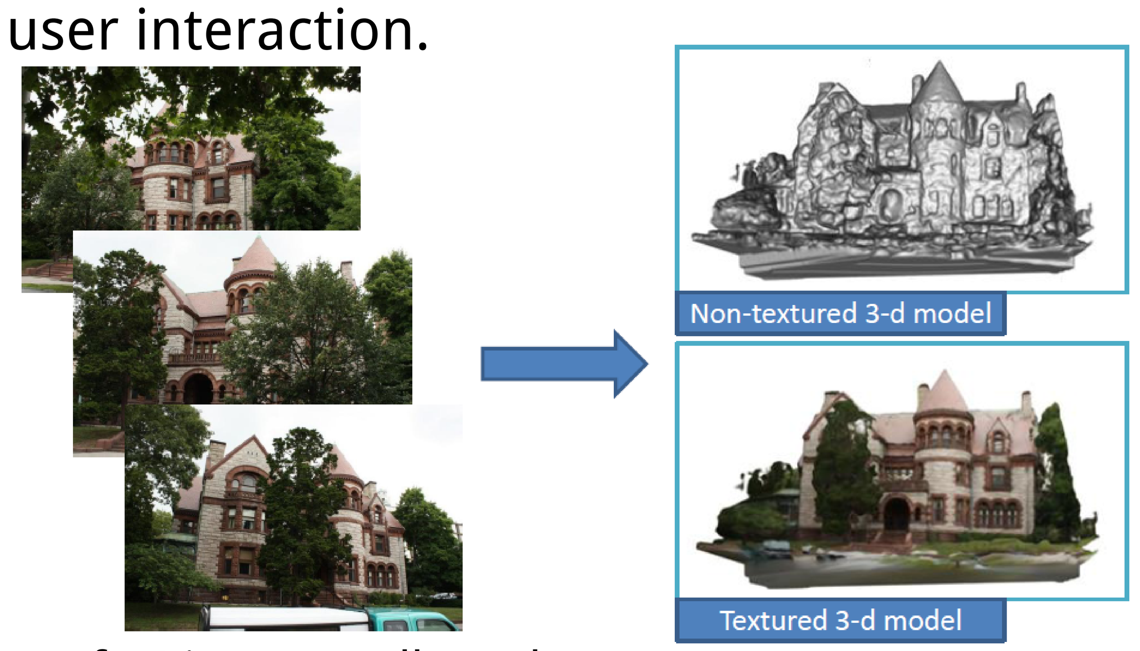
A Complete Statistical Inverse Ray Tracing Approach to Multi-View Stereo

Shubao Liu — GE Global Research — liius@ge.com

David B. Cooper — Brown University — cooper@lems.brown.edu

Chanllenges in Multi-view Stereo

* Accurate and photo-realistic 3D reconstruction.
* Works for both indoor and outdoor scenes without or with minimal user interaction.



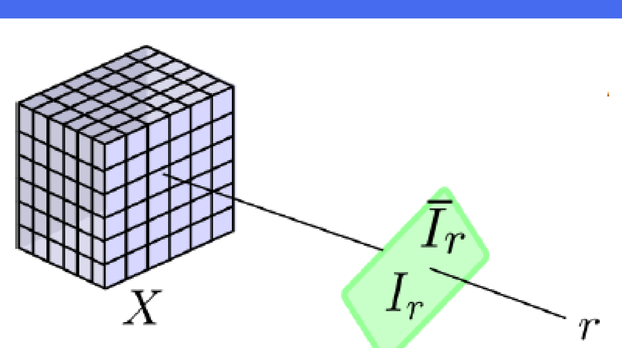
A collection of 14 images collected with a hand-held camera

Automatic photo-realistic 3D reconstruction

Key Ideas

* Solve the above challenges by modeling the multi-view stereo problem with a **statistical inverse ray tracing** approach, based on our previous work in CVPR'10.
* Optimally estimate the scene geometry and apperance **jointly** with a message-passing algorithm.

Statistical Inverse Ray Tracing Framework



A 3-d discrete volume is composed of voxels.

$$X = \{x_i, i = 0, 1, \cdots, n\}$$

Each voxel has two properties: binary occupancy and RGB color

$$x_i = (x_i^o, x_i^c)$$

$$x_i^o = \begin{cases} 0, & \text{if the voxel is empty} \\ 1, & \text{if the voxel is solid} \end{cases}$$

- Analysis by Synthesis: for each pixel
  - Given a 3-d volume X
  - Rendered pixel value from volumetric ray tracing: $\bar{I}_r = \Phi_r(X)$
  - Observed pixel value: I_r
  - We want to find a 3-d model X^* that minimizes the difference between the rendered pixel value and the observed pixel value:

$$X^* = \arg\min_X \sum_{r \in R} ||\bar{I}_r - I_r||^2$$

(R is the set of all the pixel rays from all the observation images)

Ray Markov Random Field

$$E(X) = \sum_{r\in R} E_r(X_r) + w_p^o \sum_{\langle i,j\rangle \in \mathcal{N}} E_p^o(x_i^o, x_j^o) + w_p^c \sum_{\langle i,j\rangle \in \mathcal{N}} E_p^c(x_i^c, x_j^c) + w_u^o \sum_{k\in\Omega} E_u^o(x_k^o)$$

$$E_r(X_r) = ||I_r - \Phi_r(X_r)||^2$$

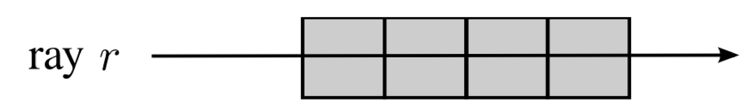
$$E_u^o(x_k^o) = \begin{cases} 1, & x_k^o = 0 \\ 0, & x_k^o = 1 \end{cases}$$

$$E_p^o(x_i^o, x_j^o) = \begin{cases} 0, & x_i^o = x_j^o \\ 1, & x_i^o \neq x_j^o \end{cases}$$ Ising model

$$E_p^c(x_i^c, x_j^c) = ||x_i^c - x_j^c||^2$$ Gaussian model

Estimate voxel occupancies, given voxel colors ⇄ Estimate voxel colors, given voxel occupancies

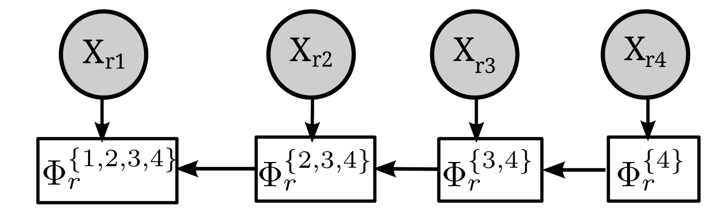
Estimate Voxel Occupancies, Given Colors



occupancy	x_{r1}^o	x_{r2}^o	x_{r3}^o	x_{r4}^o
color	x_{r1}^c	x_{r2}^c	x_{r3}^c	x_{r4}^c

$$\Phi_r(X_r) = \begin{cases} x_{r1}^c, & X_r^o = 1, \times, \times, \times \\ x_{r2}^c, & X_r^o = 0, 1, \times, \times \\ x_{r3}^c, & X_r^o = 0, 0, 1, \times \\ x_{r4}^c, & X_r^o = 0, 0, 0, 1 \\ x_{rg}^c, & X_r^o = 0, 0, 0, 0 \end{cases}$$

⇔



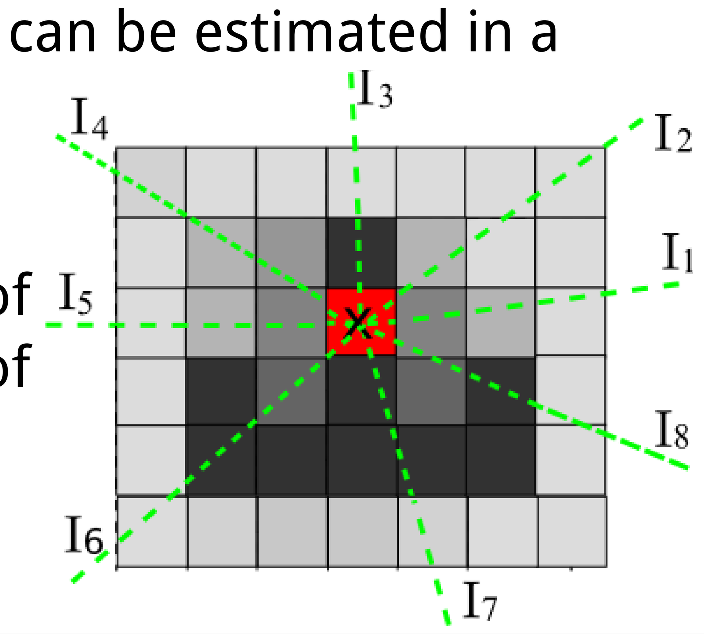
$$\Phi_r(x_{r1}, x_{r2}, x_{r3}, x_{r4}) = \psi(x_{r1}, \Phi_r(x_{r2}, x_{r3}, x_{r4}))$$
$$\Phi_r(x_{r2}, x_{r3}, x_{r4}) = \psi(x_{r2}, \Phi_r(x_{r3}, x_{r4}))$$
$$\Phi_r(x_{r3}, x_{r4}) = \psi(x_{r3}, \Phi_r(x_{r4}))$$
$$\Phi_r(x_{r4}) = \psi(x_{r4}, x_{rg}^c))$$
$$\psi(x_{ri}, z_{i+1}) = \begin{cases} x_{ri}^c, x_{ri}^o = 1 \\ z_{i+1}, x_{ri}^o = 0 \end{cases}$$

The above **recursive chain structure** of the volumetric ray tracing function can be explored by **dynamic programming** to derive a **linear computational complexity** algorithm to estimate voxel occupancies, based on belief propagation.

Estimate Voxel Colors, Given occupancies

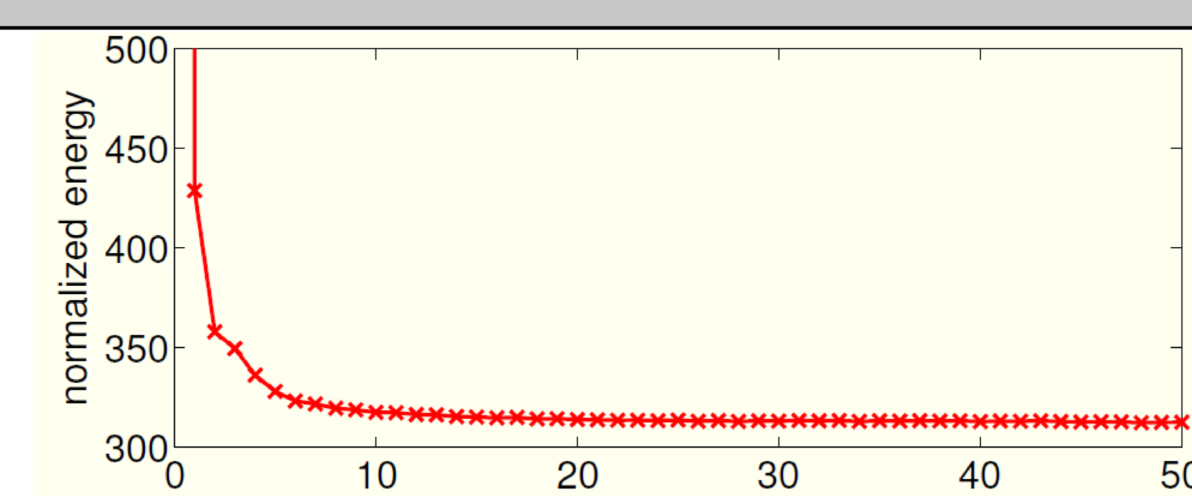
Color is different from occupancy:
1. # of color states (256^3=16M) >> # of occupancy states (2) for each voxel; So the estimation method for occupancy is not directly applicable to color estimation.
2. On the other hand, the voxel colors can be estimated in a closed form, given voxel occupancies.

The key is to estimate the **visibility** of the a ray, i.e., the **joint probability** of the voxel occupancies along the ray.



The dynamic programing accelerated belief propagation algorithm discussed above provides an estimation of the **joint probability**, which is used to estimate voxel colors.

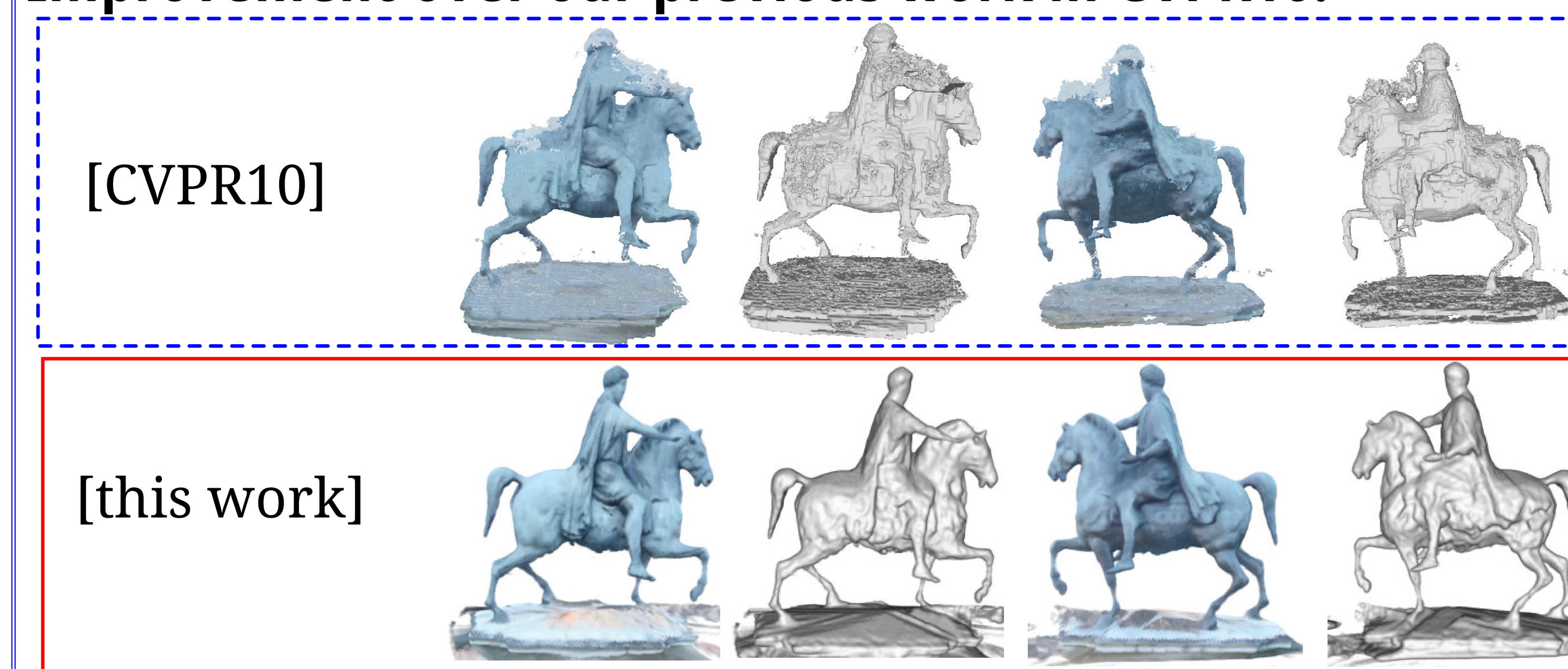
Convergence behavior of the alternating estimation



The **computation and memory cost** of the proposed algorithm is **linear to the number of voxels**. Better implementations based on **sparse** volumetric representations will be explored in the fututure to further reduce the cost in order to reconstruct large scale scenes.

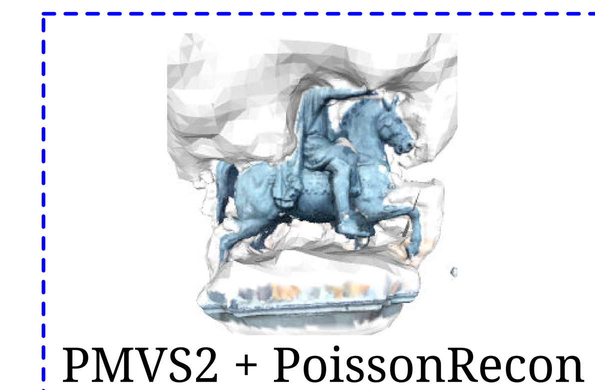
Experimental Results

Improvement over our previous work in CVPR10:

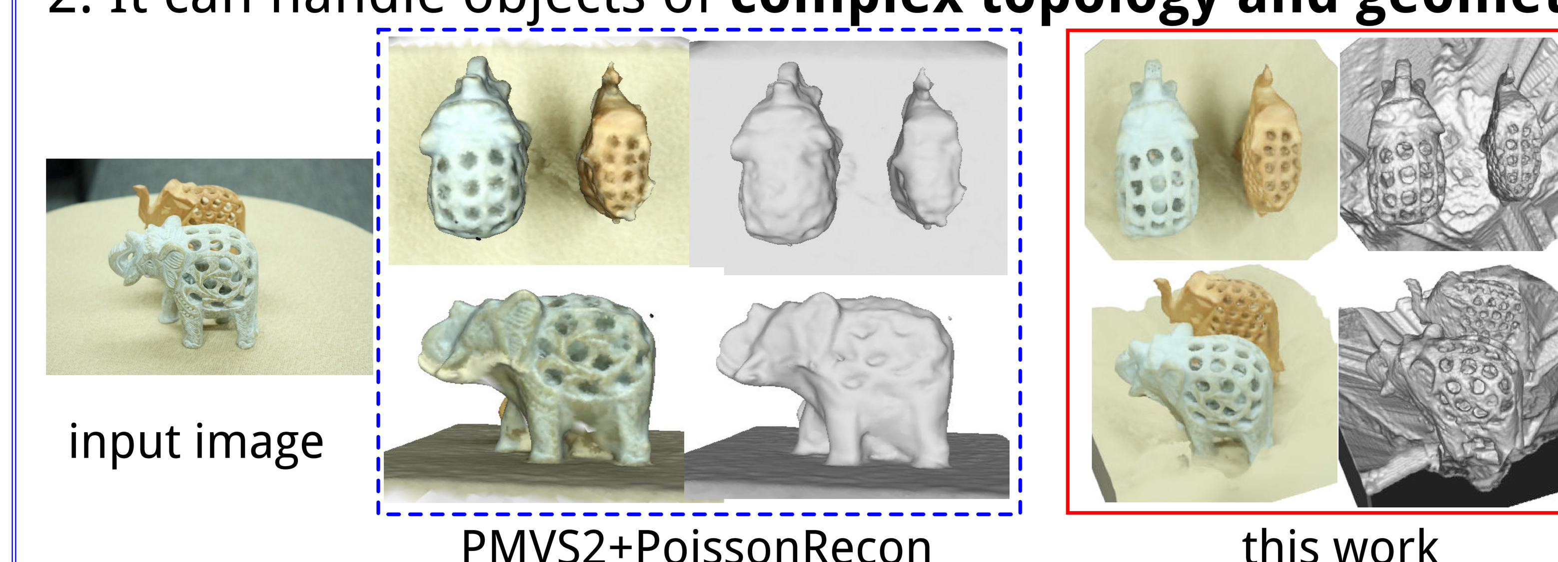


Merits of the proposed approaches:

1. **Free space constraint** is implicitly built into the formulation.

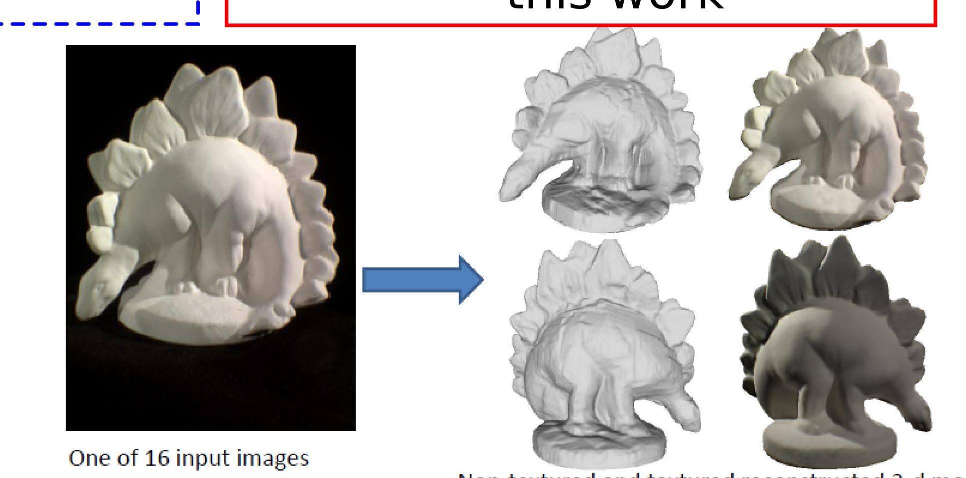


2. It can handle objects of **complex topology and geometry**.



3. The proposed approach generates **more complete** reconstructions than most of the other approaches.

4. The proposed algorithm genrates **accurate geometry**.



One of 16 input images

Non-textured and textured reconstructed 3-d models

Algorithm	Accuracy (mm)	Completeness (%)
PMVS2 (16 input images)	0.37	99.2
IRAY (our work) (16)	0.63	96.7
Voxel world model (40)	2.61	91.4

Acknowledgement

This work is supported by National Science Foundation grant IIS-0808718.