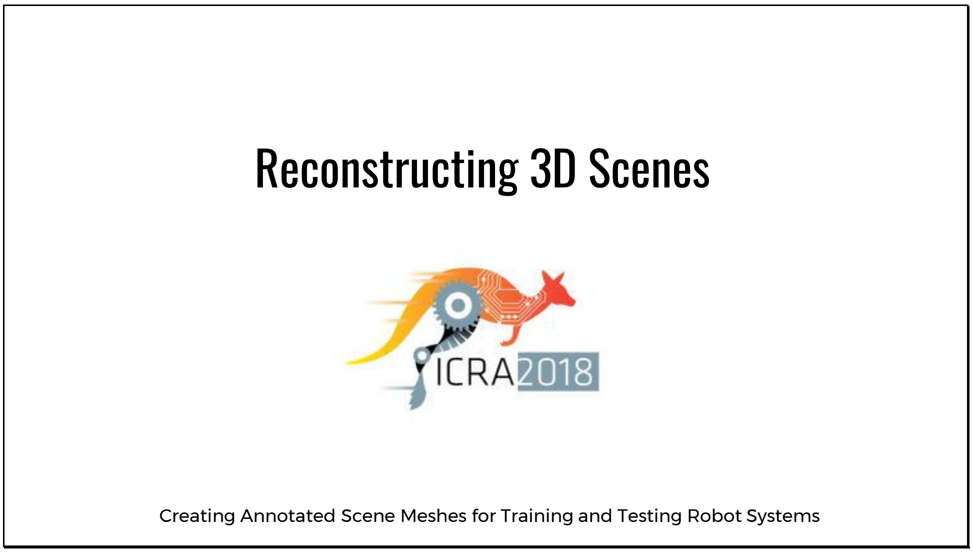
Slide 1

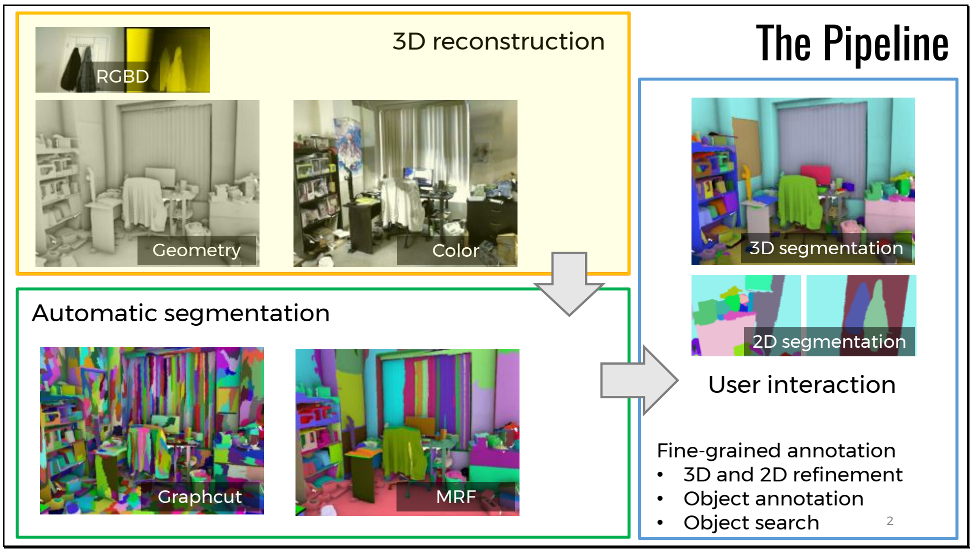


Reconstructing 3D Scenes

Creating Annotated Scene Meshes for Training and Testing Robot Systems

Hi everyone, thank you for coming this presentation!

Slide 2



The Pipeline

3D reconstruction

Automatic segmentation

User interaction

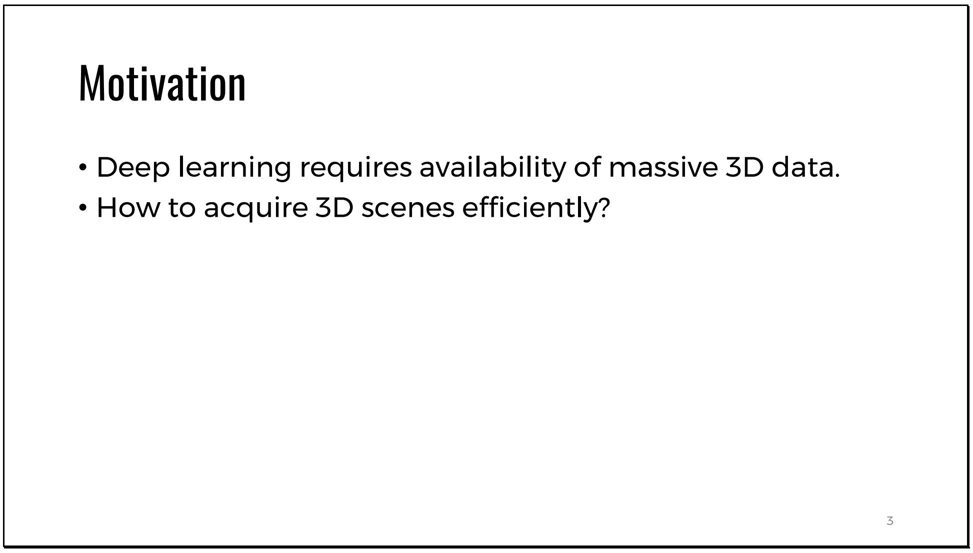
Fine-grained annotation

- 3D and 2D refinement
- Object annotation
- Object search

So here we will start by exploring the state-of-the-art techniques in 3D reconstruction. Here, we focus on scanning devices for consumers at the cost of a few hundred dollars.

3D reconstruction is an important step in the entire pipeline because we know that, building a dataset is always costly, therefore, we would like to start with high-quality reconstruction.

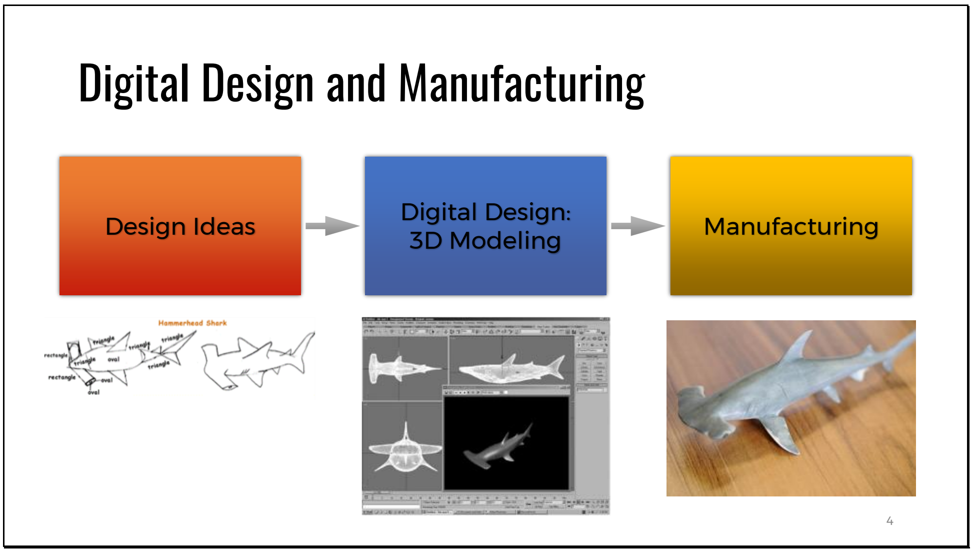
Slide 3



Motivation

- Deep learning requires availability of massive 3D data.
- How to acquire 3D scenes efficiently?

Slide 4

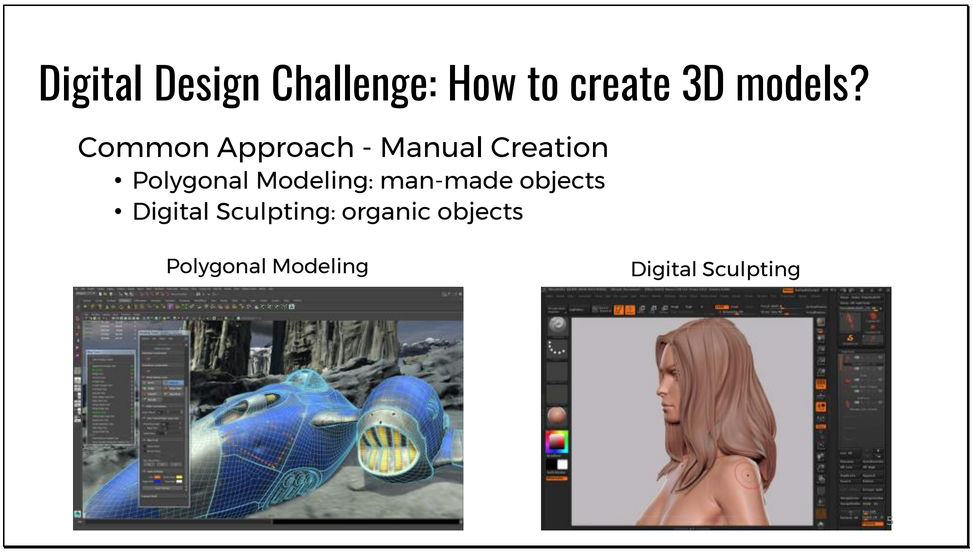


Digital Design and Manufacturing

Design Ideas → Digital Design: 3D Modeling → Manufacturing

Before I go into details.
Let’s first take a quick look of a brief pipeline of the Digital Design and Manufacturing process.
Usually, we have some design ideas, and then we make use of digital design tool, such as 3D modeling software to create the 3D objects. With the 3D objects then we can manufacture them.

Slide 5



Digital Design Challenge: How to create 3D models?

Common Approach - Manual Creation

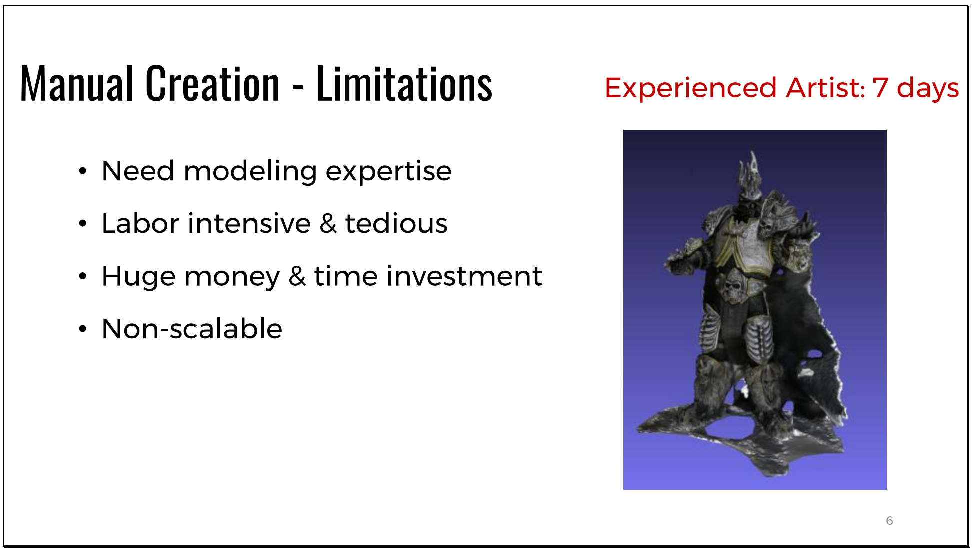
- Polygonal Modeling: man-made objects
- Digital Sculpting: organic objects

Polygonal Modeling

Digital Sculpting

The first problem is how to create the 3D objects? A common approach is by manual creation by modeling artist. For example, polygonal modeling for man-made objects, digital sculpting for organic objects.

Slide 6



Manual Creation - Limitations

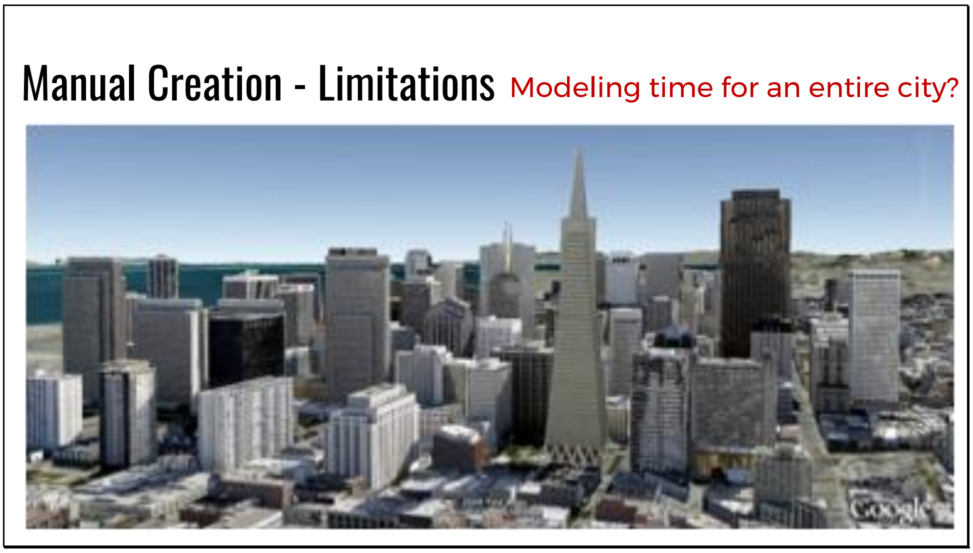
Experienced Artist: 7 days

- Need modeling expertise
- Labor intensive & tedious
- Huge money & time investment
- Non-scalable

6

But there are lot of limitations in doing this. First, you need to have the expertise. However, even if you are an expert, it will still take you time for the modeling task. For example, it takes 7 days for an experienced artist to model this complex character. Which means it cost money and time.

Slide 7

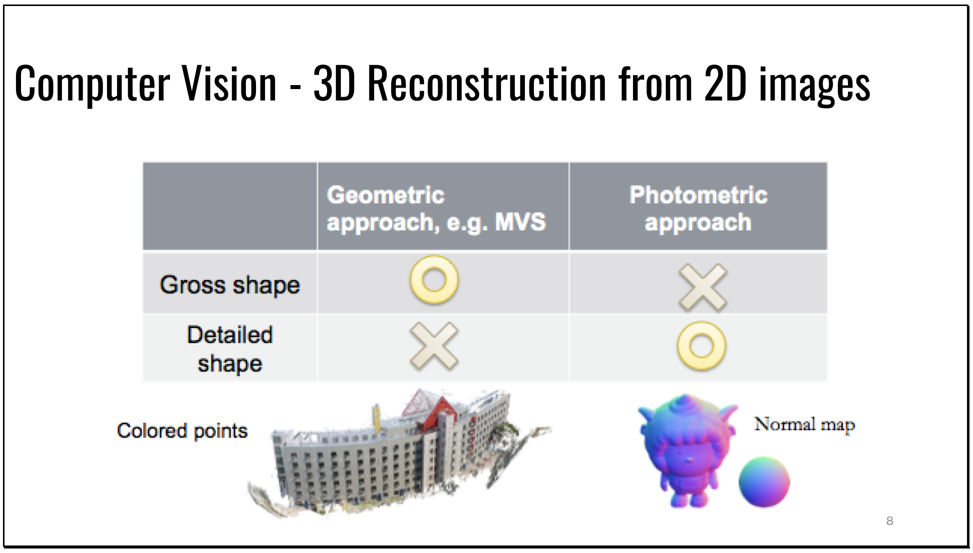


Manual Creation - Limitations

Modeling time for an entire city?

Further more, it is not really scalable, what if I ask you to model the whole city?

Slide 8



Computer Vision - 3D Reconstruction from 2D images

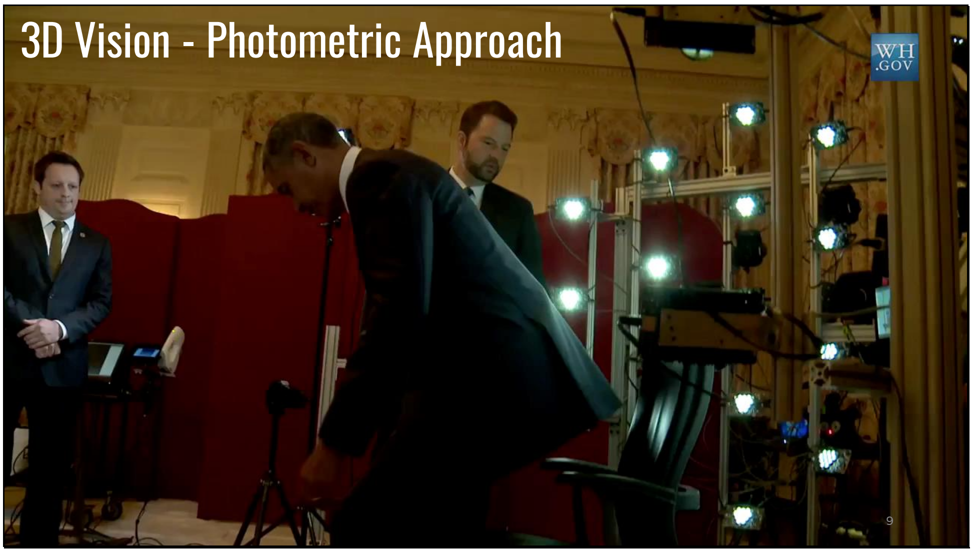
	Geometric approach, e.g. MVS	Photometric approach
Gross shape	○	✗
Detailed shape	✗	○

Colored points
Normal map

So an alternative is to explore photogrammetry. From recorded photographs, we try to obtain the 3D information of the scene such as surface coordinates, surface orientation, size, etc.

In computer vision, this is often referred to as 3D reconstruction, and overall we can categorize common techniques into two categories:

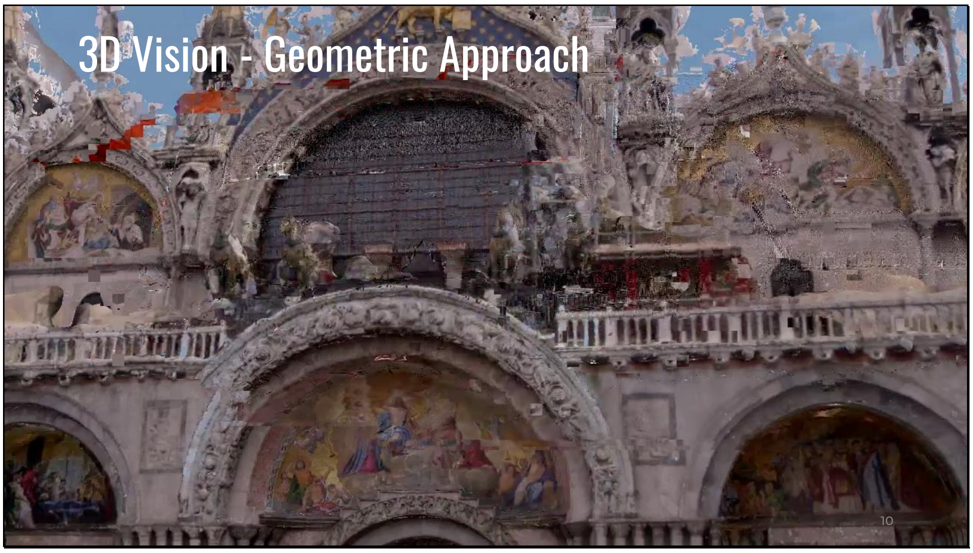
Slide 9



3D Vision - Photometric Approach

The key idea is to illuminate the target from different lighting directions. Notice the change of lighting on the President's face from 0:15.

Slide 10

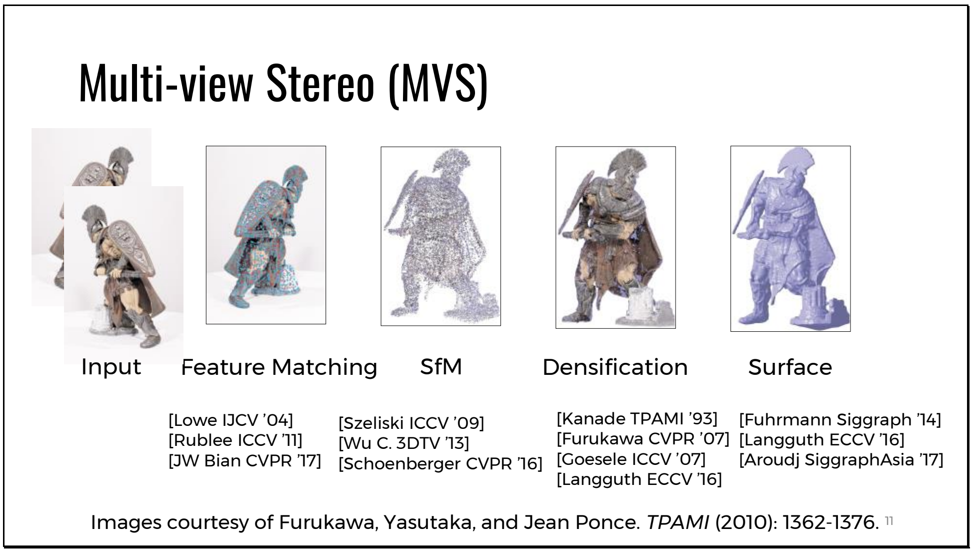


3D Vision - Geometric Approach

Large-scale Multi-view Stereo, Seitz et al., University of Washington, 2013

Another approach is multi-view stereo, which requires taking photos from different angles. This is the preferred method for outdoor scene reconstruction, since it can cover a large area and produce a rough geometry of the scene compared to the photometric approach. So we will focus more on this approach in later section.

Slide 11



Multi-view Stereo (MVS)

Input Feature Matching SfM Densification Surface

[Lowe IJCV '04]
[Rublee ICCV '11]
[JW Bian CVPR '17]

[Szeliski ICCV '09]
[Wu C. 3DTV '13]
[Schoenberger CVPR '16]

[Kanade TPAMI '93]
[Furukawa CVPR '07]
[Goesele ICCV '07]
[Langguth ECCV '16]

[Fuhrmann Siggraph '14]
[Langguth ECCV '16]
[Aroudj SiggraphAsia '17]

Images courtesy of Furukawa, Yasutaka, and Jean Ponce. *TPAMI* (2010): 1362-1376.

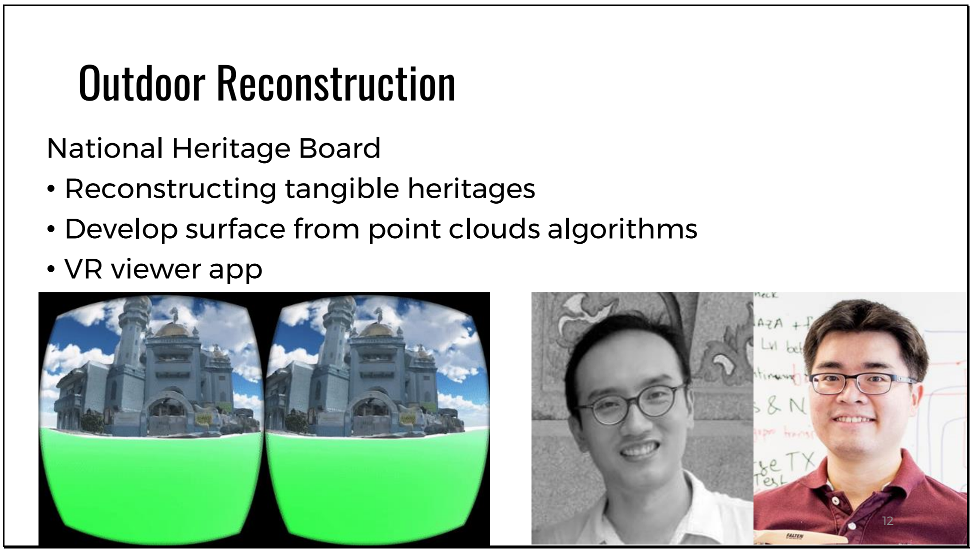
A typical multi-view stereo reconstruction pipeline. Different applications may use different implementations of each of the main blocks, but the overall approach is:

- Collect images
- Find sparse correspondences between image pairs (feature matching)
- Compute camera parameters for each image (SfM)
- Reconstruct the 3D geometry of the scene from the set of images and corresponding camera parameters (Densification + Surface)
- Optionally reconstruct the materials of the scene.

In the recent years, more research has been put to improve each stage in this pipeline.

I also want to highlight that the Structure-from-Motion problem is very similar to the visual SLAM problem in robotics. However, visual SLAM is often performed in real-time with a camera, while the Structure-from-Motion problem is more general, which requires no fixed camera information and focusing more on quality of reconstruction.

Slide 12

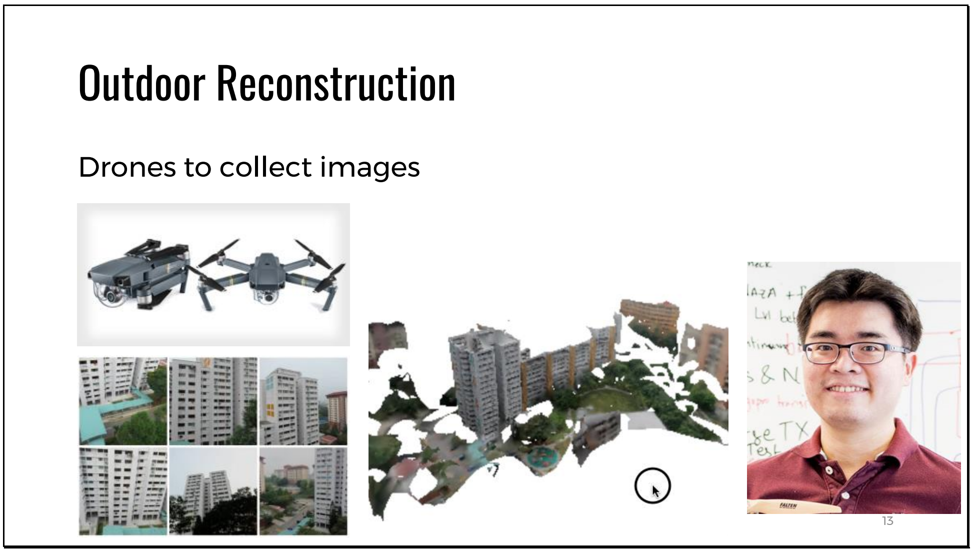


Outdoor Reconstruction

National Heritage Board

- Reconstructing tangible heritages
- Develop surface from point clouds algorithms
- VR viewer app

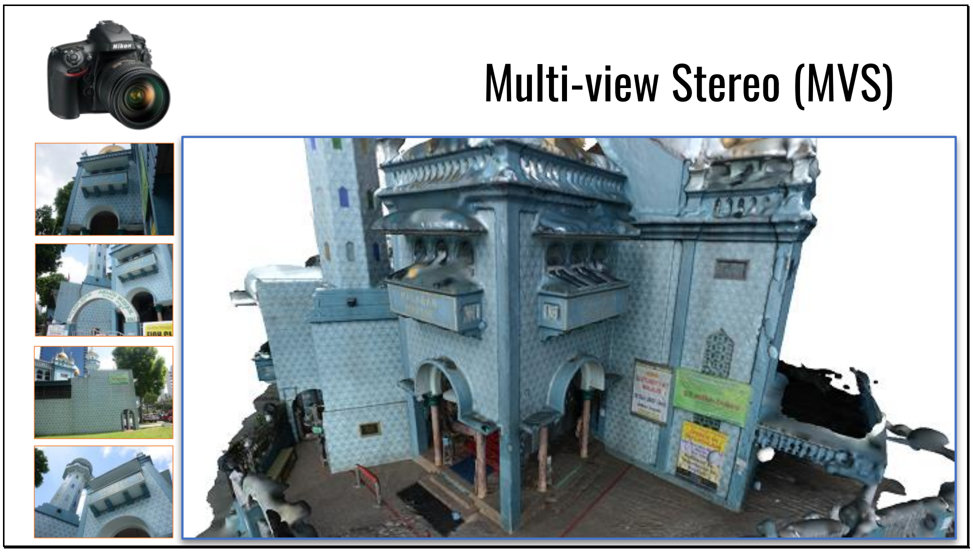
Slide 13



Outdoor Reconstruction

Drones to collect images

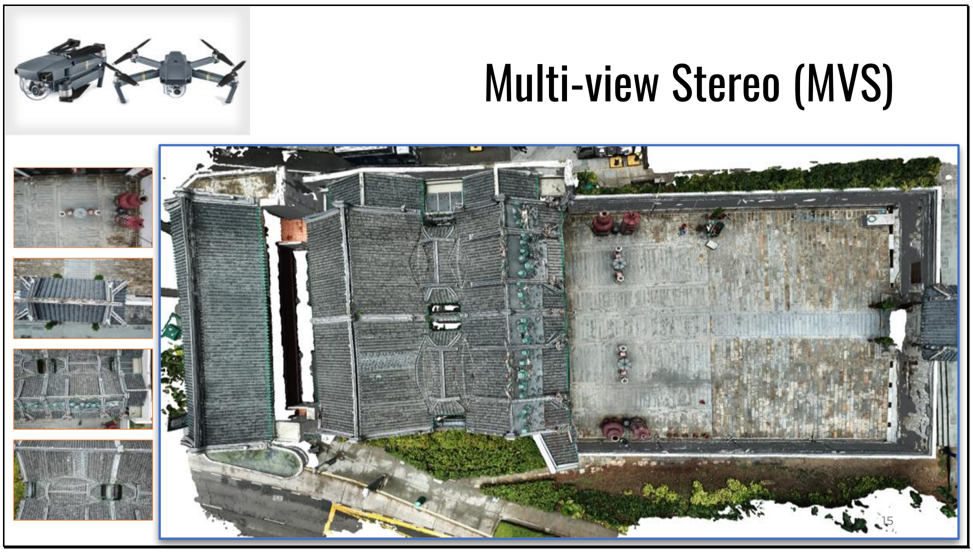
Slide 14



Multi-view Stereo (MVS)

Here is one of the reconstruction where data is captured by ourselves. In this example, we use only a DSLR camera and capture eye-level images by letting the operator walk about the building.

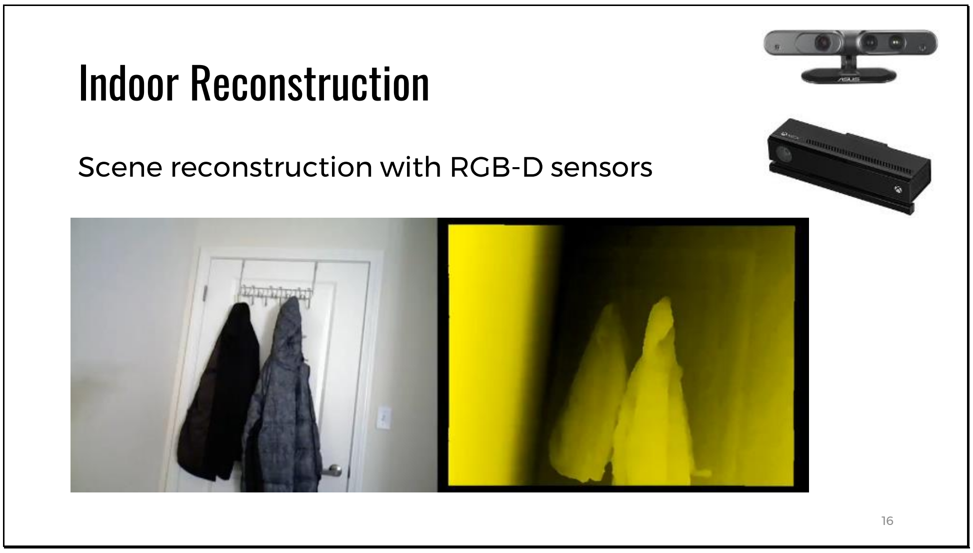
Slide 15



Multi-view Stereo (MVS)

Here is a reconstruction with data captured by a drone.

Slide 16



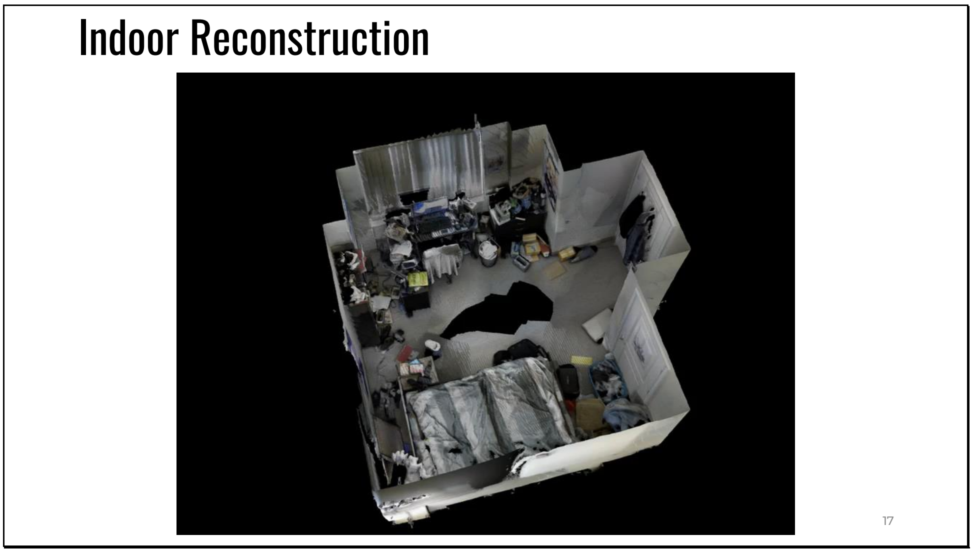
Indoor Reconstruction

Scene reconstruction with RGB-D sensors

16

Our input is a color and depth video of a scene captured by a consumer-grade RGBD camera. To record a video, we connect the camera to a laptop and move the camera slowly in the scene. This example shows a color and depth video captured by an Asus Xtion depth camera.

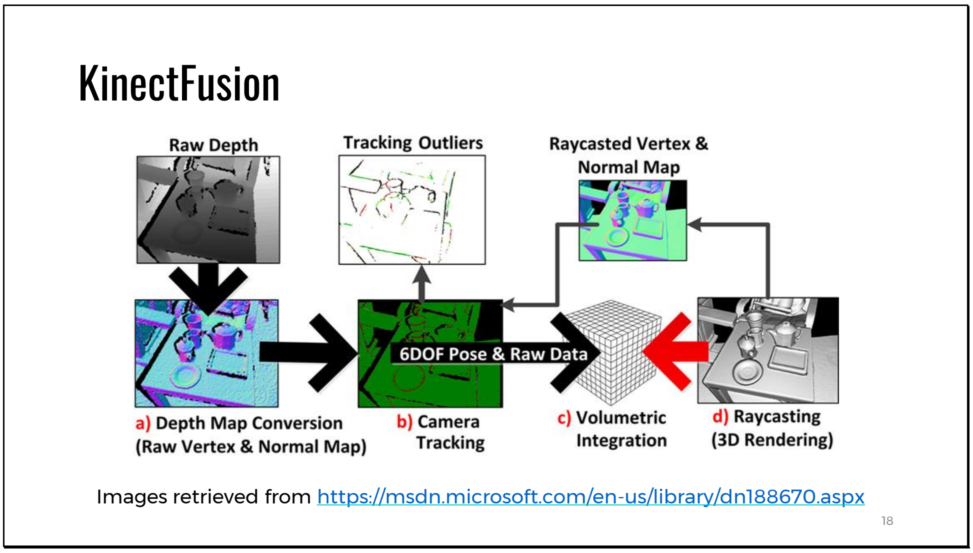
Slide 17



Indoor Reconstruction

We then reconstruct the scene from the input video. We represent the scene as a triangular mesh.
The texture of the mesh is computed by taking the median value of all color pixels that correspond to each vertex.

Slide 18



KinectFusion

Raw Depth

Tracking Outliers

Raycasted Vertex & Normal Map

6DOF Pose & Raw Data

a) Depth Map Conversion (Raw Vertex & Normal Map)

b) Camera Tracking

c) Volumetric Integration

d) Raycasting (3D Rendering)

Images retrieved from https://msdn.microsoft.com/en-us/library/dn188670.aspx

After the release of MS Kinect

There have been many prior works about RGBD reconstruction. One of the most well-known work is probably KinectFusion. It takes as input a sequence of depth images (recorded from a depth sensor such as Kinect, Asus Xtion, or Structure Sensor, etc.) and perform real-time camera tracking and depth frame alignment to produce very high-quality meshes.

To facilitate both tracking and alignment, they use a volume to represent the world.

Each time a new depth frame arrives, we assume that its camera pose varies very little compared to the current pose, so we can perform an ICP to align this depth frame to the current volume (see step b).

After alignment, the depth frame is accumulated into the volume (step c), refining the current mesh (imagine that each depth frame is noisy, and when more depth frames are accumulated, the surface gets nicer and nicer).

Finally, we can render the volume to screen (step d).

KinectFusion is a powerful tool to capture indoor scenes and it outputs very high surface quality with just a consumer depth sensor.

One notable drawback of KinectFusion is that the camera tracking is very simple, and thus might fail for situations like a part of the scene is not well aligned when revisited (loop closure), or it lacks a global agreement when aligning all depth frames. Several works attempt to address this issue.

Slide 19

SIGGRAPH Talks 2011

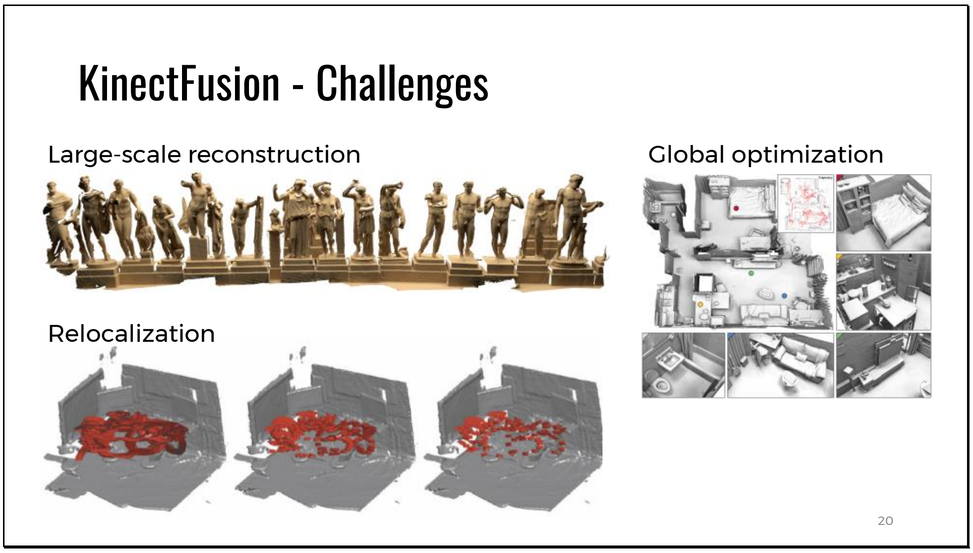
KinectFusion:

Real-Time Dynamic 3D Surface Reconstruction and Interaction

**Shahram Izadi 1, Richard Newcombe 2, David Kim 1,3, Otmar Hilliges 1,
David Molyneaux 1,4, Pushmeet Kohli 1, Jamie Shotton 1,
Steve Hodges 1, Dustin Freeman 5, Andrew Davison 2, Andrew Fitzgibbon 1**

1 Microsoft Research Cambridge 2 Imperial College London
3 Newcastle University 4 Lancaster University
5 University of Toronto

Slide 20



KinectFusion - Challenges

Large-scale reconstruction

Relocalization

Global optimization

Slide 21

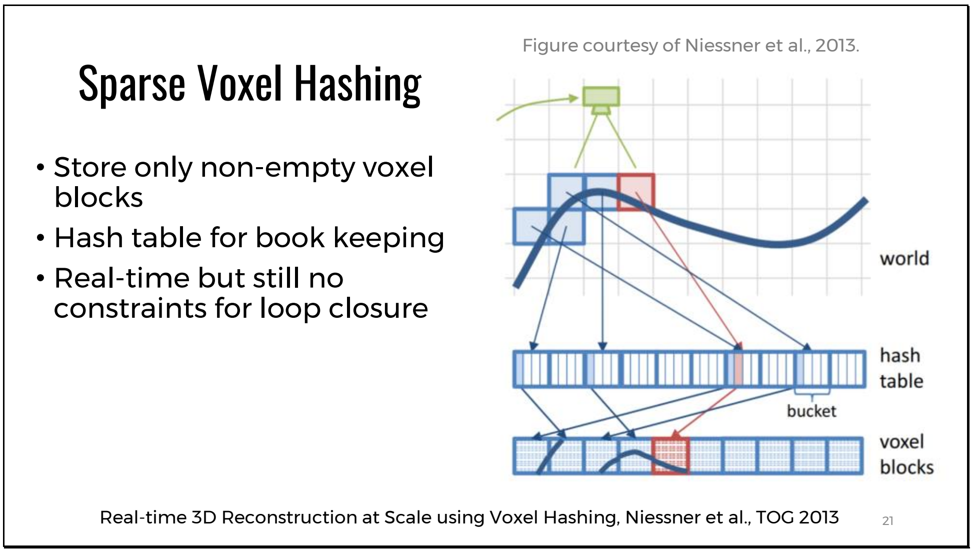


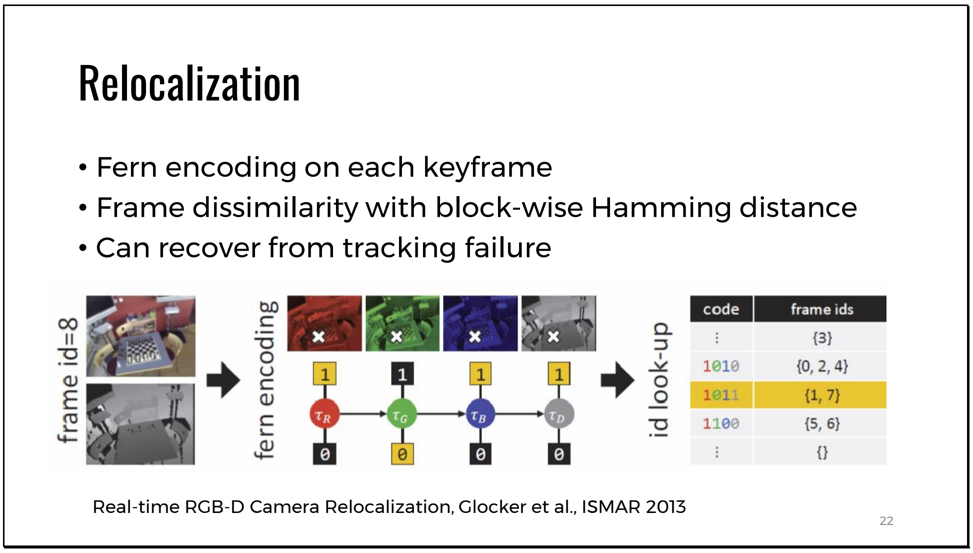
Figure courtesy of Niessner et al., 2013.

Sparse Voxel Hashing

- Store only non-empty voxel blocks
- Hash table for book keeping
- Real-time but still no constraints for loop closure

Real-time 3D Reconstruction at Scale using Voxel Hashing, Niessner et al., TOG 2013

Slide 22



Relocalization

- Fern encoding on each keyframe
- Frame dissimilarity with block-wise Hamming distance
- Can recover from tracking failure

code	frame ids
⋮	{3}
1010	{0, 2, 4}
1011	{1, 7}
1100	{5, 6}
⋮	{}

Real-time RGB-D Camera Relocalization, Glocker et al., ISMAR 2013

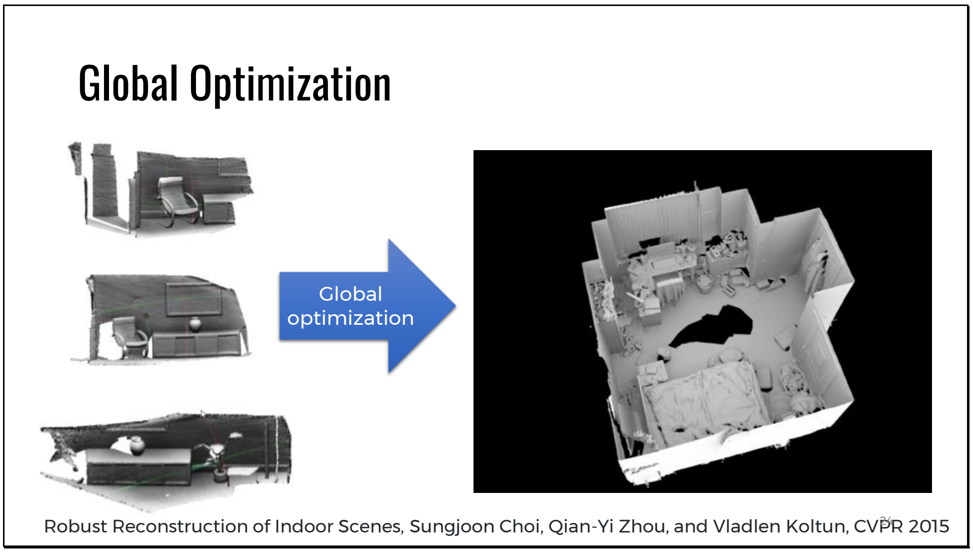
Slide 23

Relocalization

Our method is based on
compact encoding with randomized ferns...

Real-time RGB-D Camera Relocalization, Glocker et al., ISMAR 2013

Slide 24



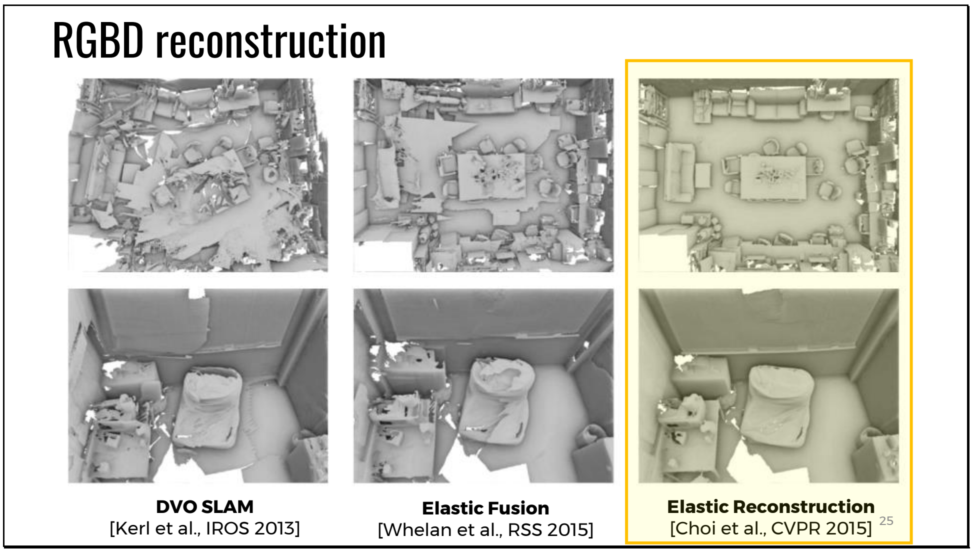
Global Optimization

Robust Reconstruction of Indoor Scenes, Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun, CVPR 2015

We leverage the technique proposed by Choi et al. in CVPR 2015 for global registration.

We run 3D reconstruction to align small geometry fragments (sub-scene) from KinectFusion. This global registration produces complete scene meshes.

Slide 25



RGBD reconstruction

DVO SLAM
[Kerl et al., IROS 2013]

Elastic Fusion
[Whelan et al., RSS 2015]

Elastic Reconstruction
[Choi et al., CVPR 2015]

25

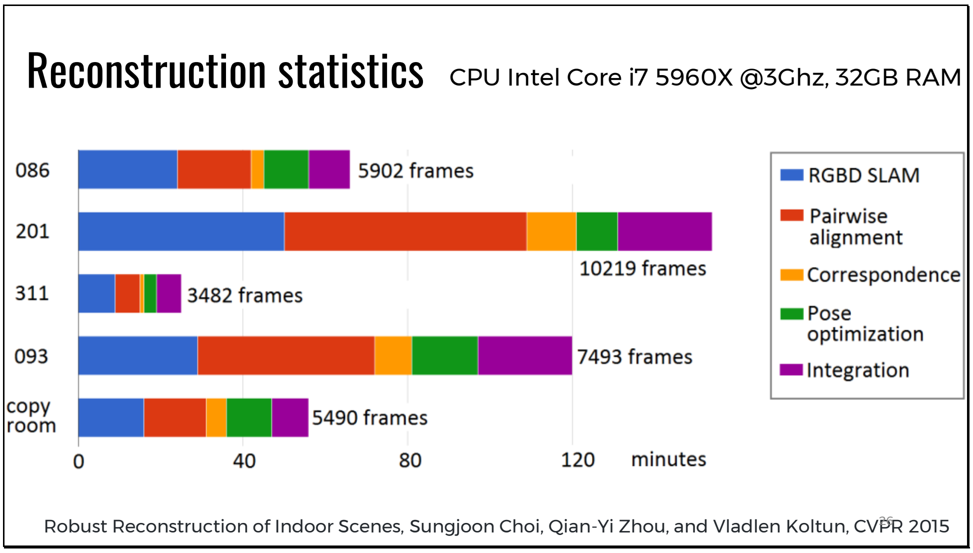
There have been many prior works about RGBD reconstruction. We focus on those that have implementation publicly available.

We tested DVO SLAM by Kerl et al, Elastic Fusion by Whelan et al, and Elastic Reconstruction by Choi et al with our scenes.

While these methods work comparably for simple scenes, for complex scenes DVO SLAM and Elastic Fusion performs poorly.
The pairwise alignment and global optimization in Elastic Reconstruction makes it robust to a wider range of scenes, and therefore, it is our method of choice for reconstruction.

The disadvantage of this approach is that non-rigid registration tends to take much longer time, and the implementation is not stable enough for use. To reconstruct our dataset, we use rigid registration in Elastic Reconstruction code. This provides high-quality geometry alignment.

Slide 26



Reconstruction statistics CPU Intel Core i7 5960X @3Ghz, 32GB RAM

Robust Reconstruction of Indoor Scenes, Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun, CVPR 2015

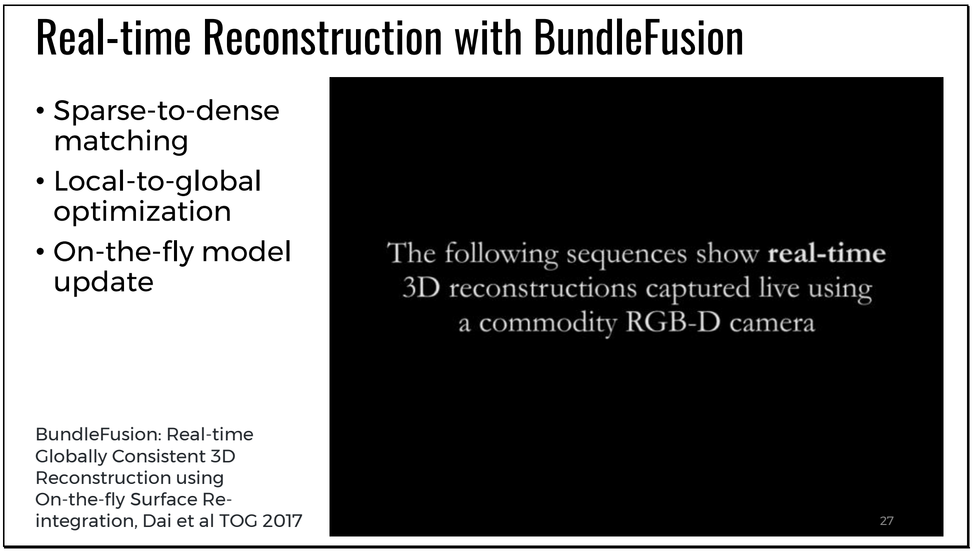
The performance of the reconstruction is shown in this plot.

Our scenes are reconstructed on a PC with an Intel Core i7 5960X with 16 cores at 3 GHz, and 32 GB of RAM.

Each scene takes more than an hour to finish on average, usually longer. We launch a few reconstructions at the same time to leverage parallelization.

While this approach is not the fastest, the high-quality geometry output is what we require to obtain high-quality segmentation.

Slide 27



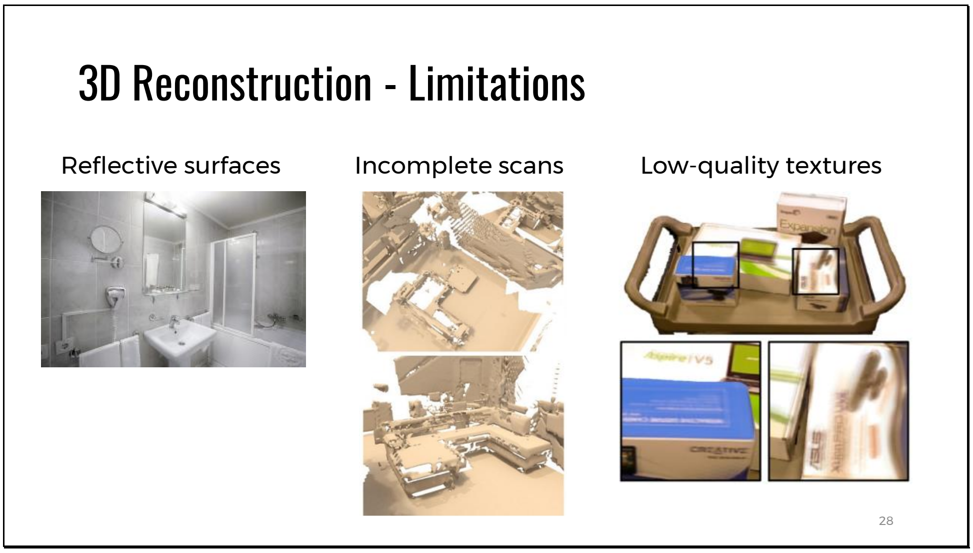
Real-time Reconstruction with BundleFusion

- Sparse-to-dense matching
- Local-to-global optimization
- On-the-fly model update

BundleFusion: Real-time Globally Consistent 3D Reconstruction using On-the-fly Surface Re-integration, Dai et al TOG 2017

The following sequences show **real-time** 3D reconstructions captured live using a commodity RGB-D camera

Slide 28



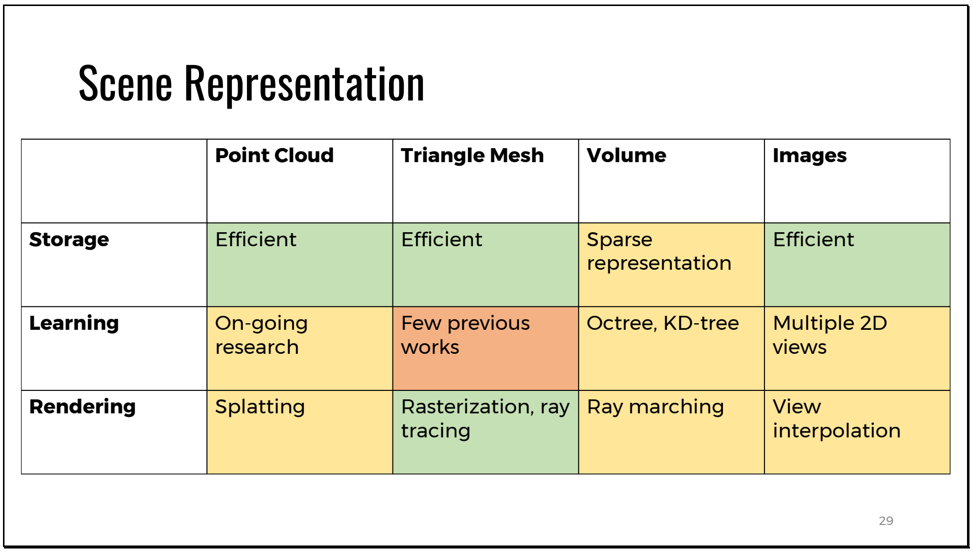
3D Reconstruction - Limitations

Reflective surfaces

Incomplete scans

Low-quality textures

Slide 29



Scene Representation

	Point Cloud	Triangle Mesh	Volume	Images
Storage	🟩 Efficient	🟩 Efficient	🟨 Sparse representation	🟩 Efficient
Learning	🟨 On-going research	🟥 Few previous works	🟨 Octree, KD-tree	🟨 Multiple 2D views
Rendering	🟨 Splatting	🟩 Rasterization, ray tracing	🟨 Ray marching	🟨 View interpolation

A very important design choice is scene representation. Unlike images that can be easily represented as a uniform grid, 3D scene has multiple options for representation. We provide a simple table of comparisons based on three requirements: storage, learning, and rendering.

Our color notation: good (green), applicable (yellow), more efforts required (red).

Volume needs more storage space in general since they capture the free space. This storage issue causes some limitations in learning, e.g., it is difficult to perform learning with high-resolution volumes. Sparse representation could solve the storage problem, but learning with neural network becomes no longer straightforward.

Among the representations, in the last and this year we have seen a few works about point cloud learning, which has shown some promises for robot applications as it balances between storage and learning and rendering. Another advantage is that point cloud data can be easily obtained from multiple sources such as RGB-D sensor, LiDAR, or multiple view geometry.

Another popular representation in computer graphics is triangle or quad mesh. Unfortunately, for learning this kind of representation is more challenging. There has been some previous works about object classification or segmentation with meshes, however, there has been few works that can perform learning with general 3D scene mesh representation. This could be a great research avenue in the near future.

Finally, rendering/visualization is not a big problem, but for high-quality rendering, triangle mesh has the most advantages as it can utilize existing rendering engines built for games or film production. Other representation, one might have to resort to custom built renderers.