OpenARK — Tackling AR Challenges via Open-Source Development Kit

Allen Y. Yang

with
Luisa Caldas, Mohammad Keshavarzi, Woojin Ko,
Joe Menke

Berkeley
UNIVERSITY OF CALIFORNIA

Organizers

OpenARK Team

Faculty

- Luisa Caldas
- Shankar Sastry

Students

- Oladapo Afolabi
- Adam Chang
- Alex Yu
- Bill Zhou

GitHub: https://github.com/augcog/OpenARK

Berkeley
UNIVERSITY OF CALIFORNIA

Berkeley
UNIVERSITY OF CALIFORNIA

Slides available: vivecenter.berkeley.edu

FHL VIVE CENTER
FOR ENHANCED REALITY

Mission

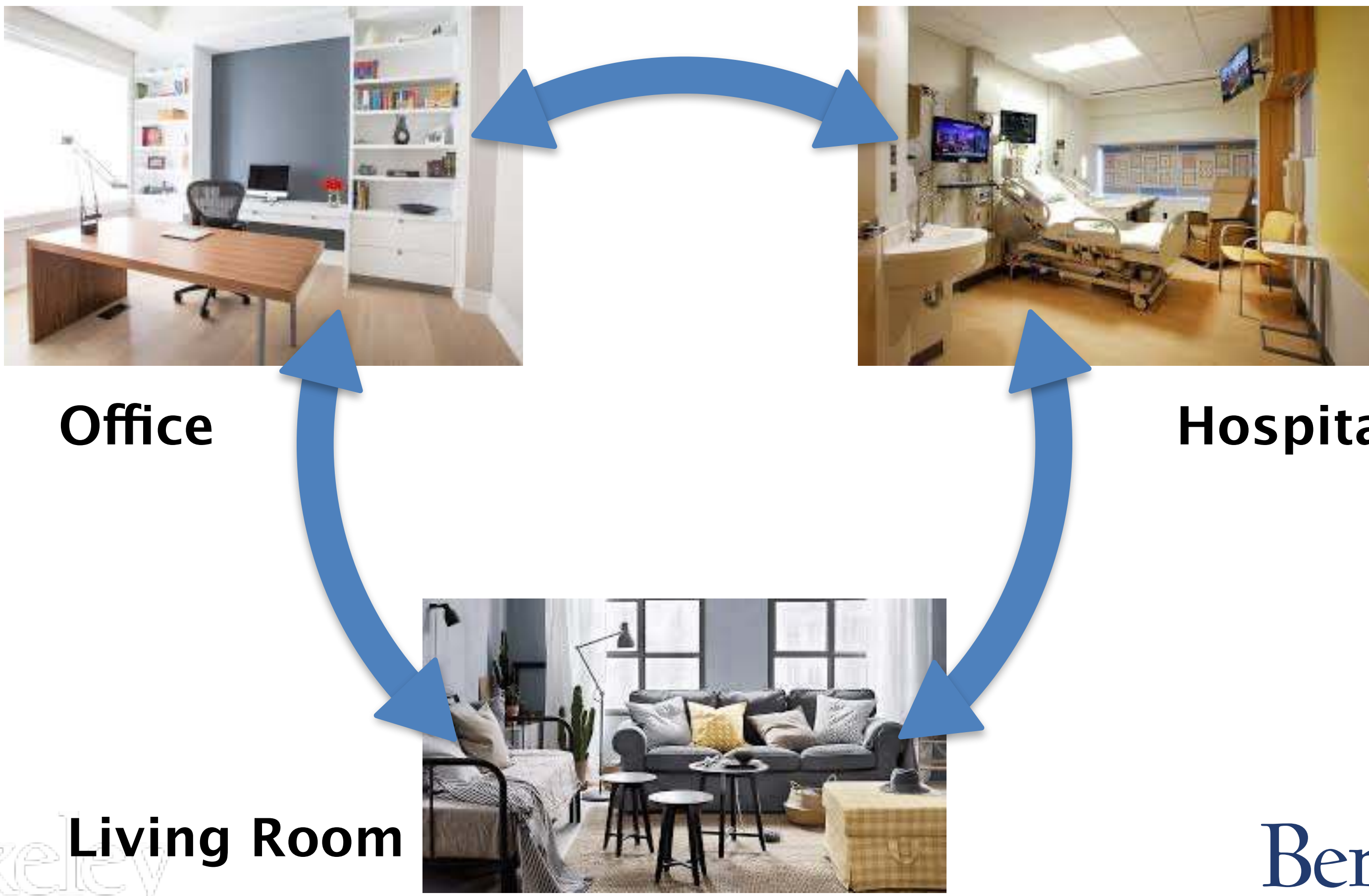
The main goals of the center are to sponsor critical fundamental research and high-impact applications in the emerging fields of Virtual Reality (VR), Augmented Reality (AR), and Artificial Intelligence (AI), and at the same time serve as the central hub to facilitate the deployment of disruptive VR, AR, and AI technologies across the Berkeley campus for cross-disciplinary research and education.

We aim to achieve these goals by offering seed grant to our faculty, supervising and facilitating student research activities, and fostering external industry partnerships with other stakeholders.

Princess Leia's Hologram: First asynchronous telepresence

Multi-User Interaction Demo: Microsoft Holoportation

Multi-User AR Applications: Model, Share, Manipulate



3D Modeling for Personal AR Products

- Infer 3D using minimal number of images
- Accurate dense reconstruction for virtual augmentation
- Complete without holes
- Sharable and editable space models for multi-users
- Augmentation of human users as realistic avatars
- **Privacy and standards

Solutions for Future AR Experience

1. Modeling Background Layout
2. Modeling Foreground Objects

3. Modeling User Avatars
4. Optimization & Sharing in Mutual Space

Outline of the OpenARK Tutorial

- Session I: Contexture 3D Scene and Avatar Modeling
- Session II: 3D Reconstruction, SLAM, and Gesture Recognition
- Session III: Maximization and Manipulation of Contextual Mutual Space Models

Traditional 3D Vision —
“Building Rome on a (cloudless) day”

* Jan-Michael Frahm, et al. 2010

Pros and Cons of SfM

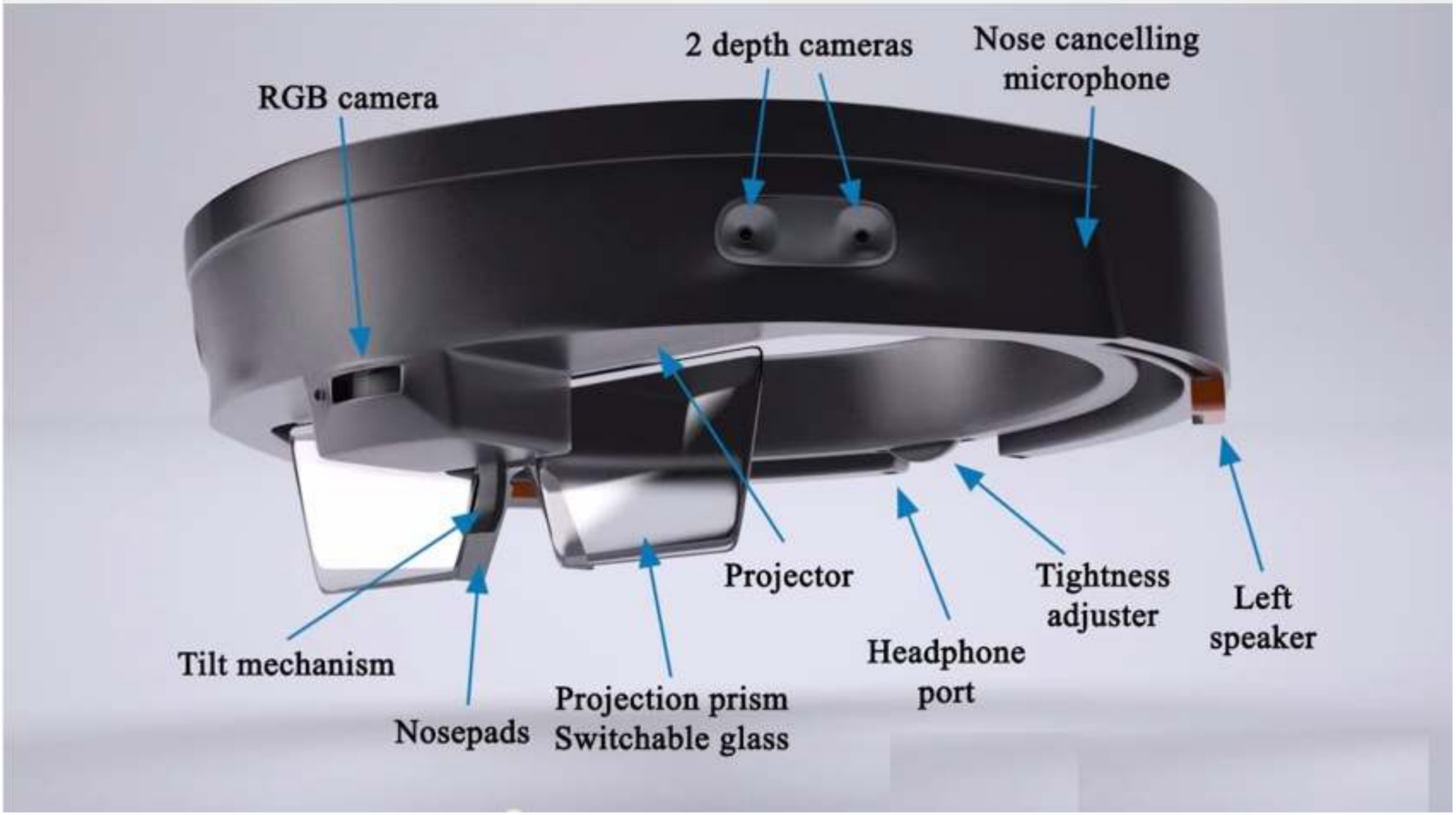
Pros

- Widely available hardware
- Unlimited range
- Uniform noise model (Gaussian)
- Retain surface texture

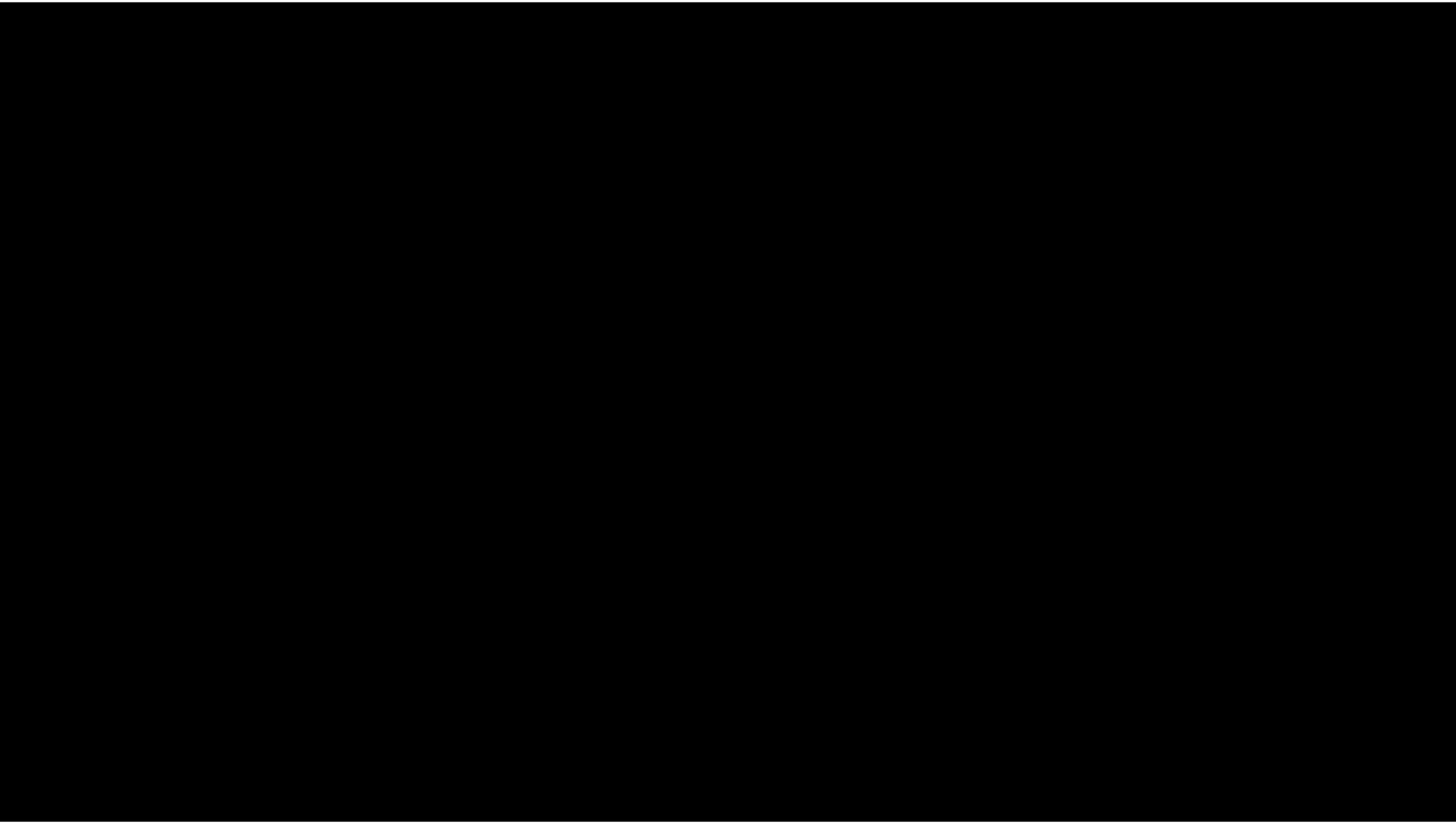
Cons

- Computationally intensive to recover 3D depth
- Doesn’t work in dark
- Doesn’t work when lack of texture
- Lead to only sparse geometry vs. dense 3D map

Anatomy of an AR platform

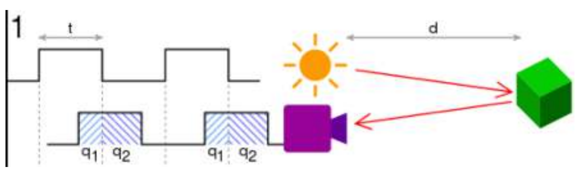


Using Depth Camera to Scan Spaces

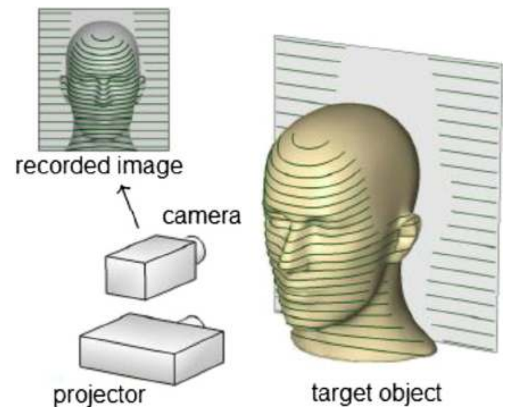


Depth Cameras

Depth from Single Camera

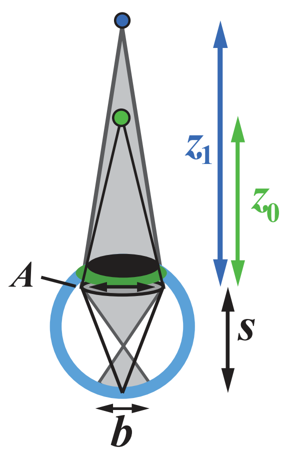


Time of Flight



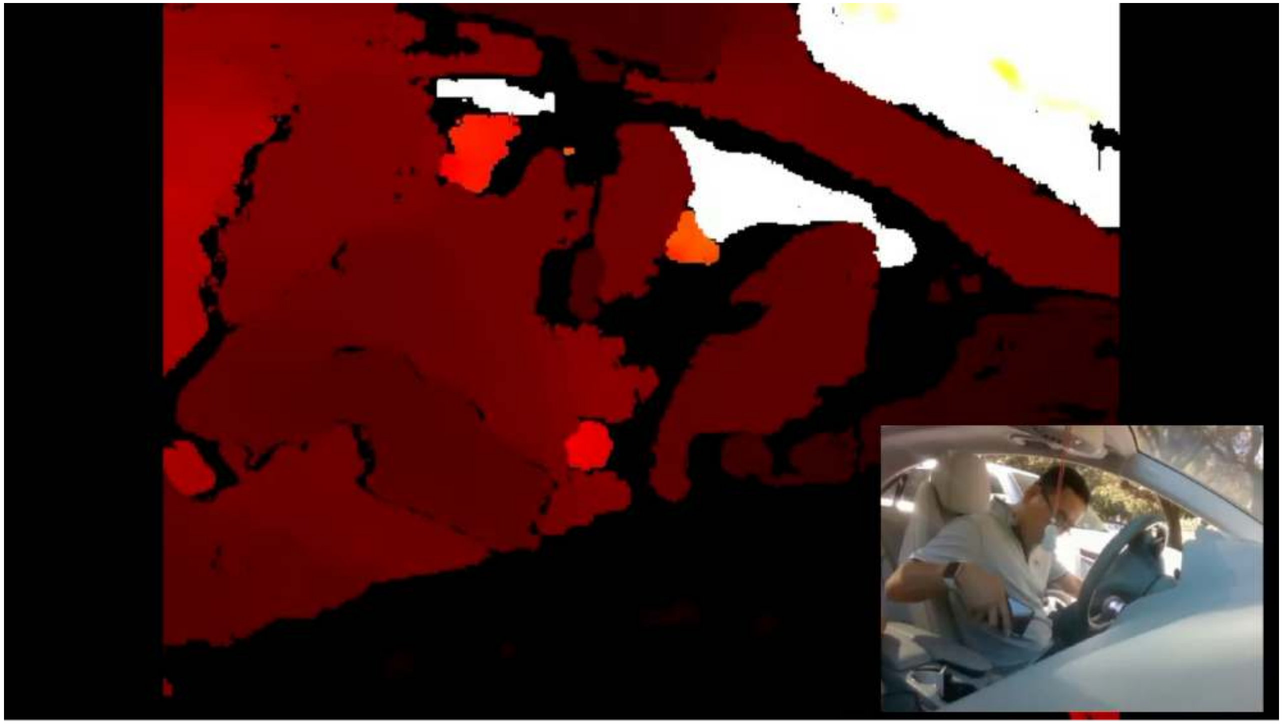
Structured Light

Blur:



Depth from Defocus

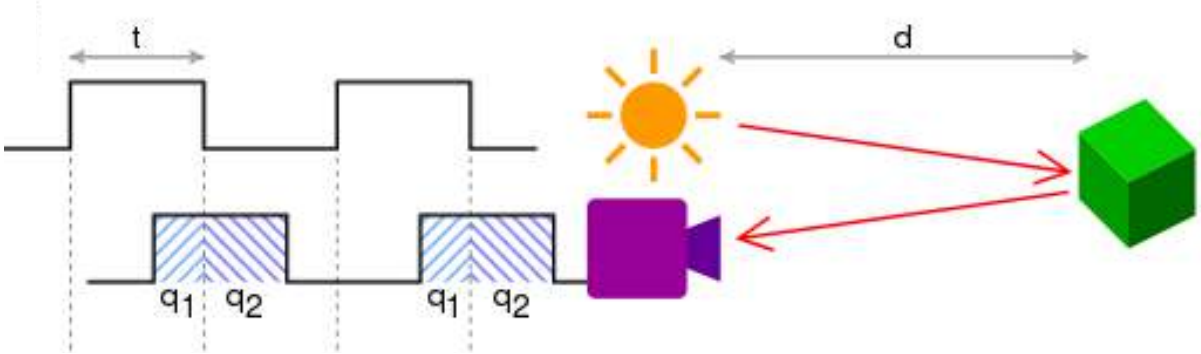
Depth from Stereo vs Time of Flight



Depth from Stereo vs Time of Flight

Basic ToF Principles

$$t_D = 2 \cdot \frac{D}{c}$$

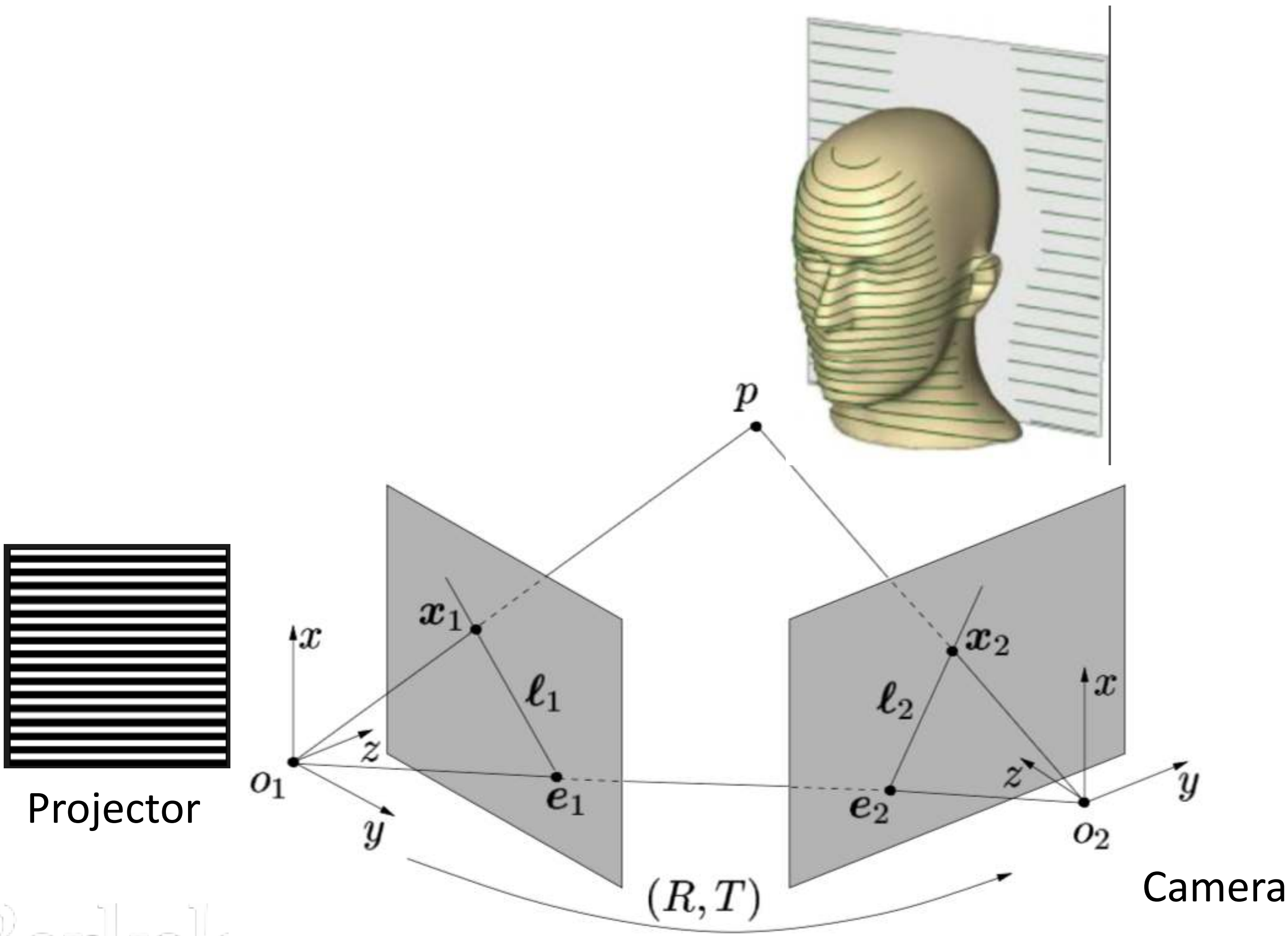


$$d = \frac{c\,t}{2} \frac{q2}{q1 + q2}$$

Limitation: t_D must be smaller than t

https://en.wikipedia.org/wiki/Time-of-flight_camera

Spatial Structured Light Approach



Disparity from calibrated patterns

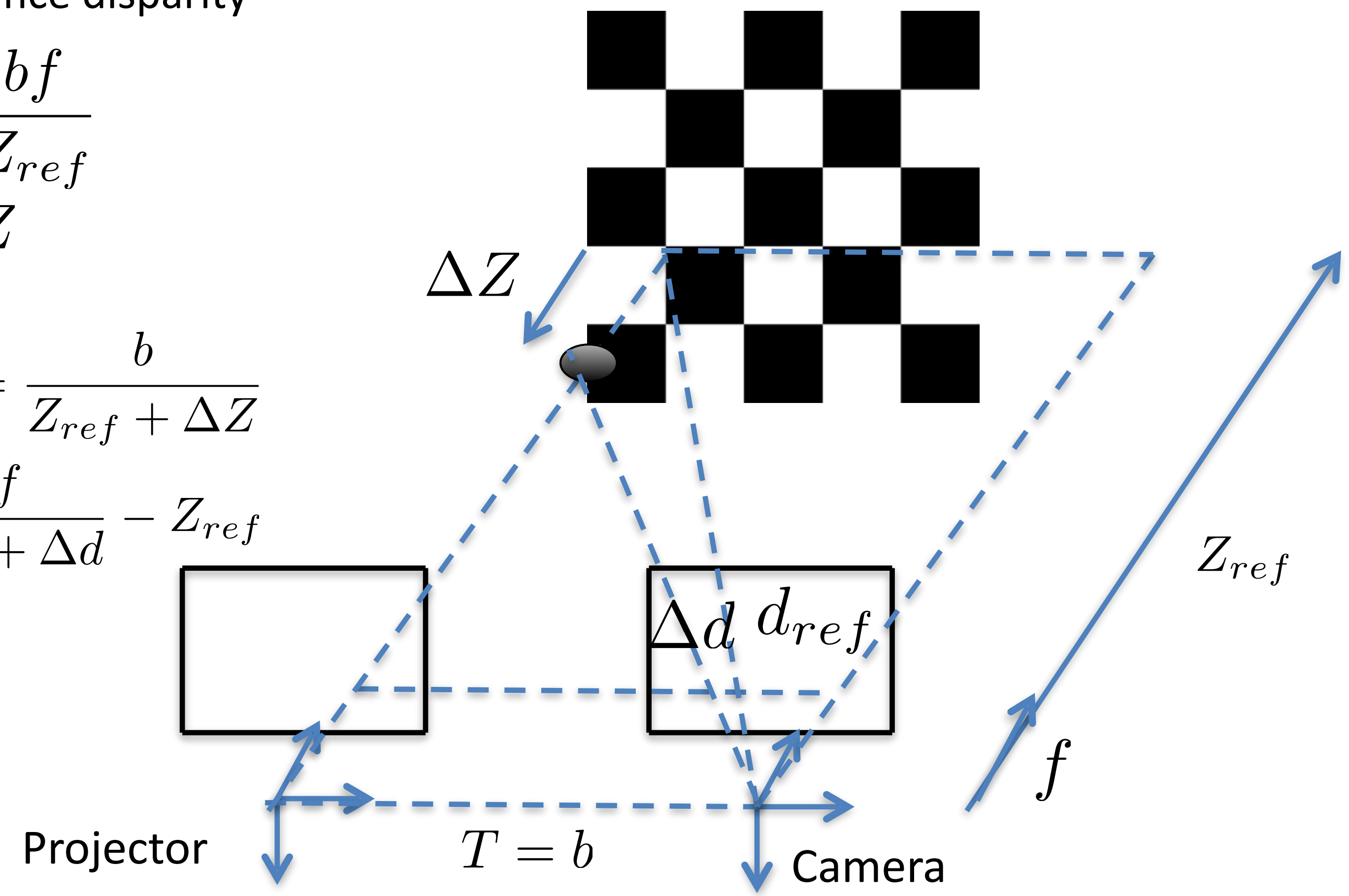
- Assume a reference disparity

$$d_{ref} = \frac{bf}{Z_{ref}}$$

- Compute ΔZ

$$\frac{d_{ref} + \Delta d}{f} = \frac{b}{Z_{ref} + \Delta Z}$$

$$\Delta Z = \frac{bf}{d_{ref} + \Delta d} - Z_{ref}$$



Pros and Cons of (most) Depth Cameras

Pros

- Computationally simpler (using light reflection and look-up tables)
- Work in the dark
- Work on texture-less surfaces
- Full dense 3D map for AR/VR

Cons

- Light emitter may consumer more power
- Challenge with sunlight
- Uneven noise model in depth
- Cost more to manufacture emitter and sensor

Available RGB-D Databases

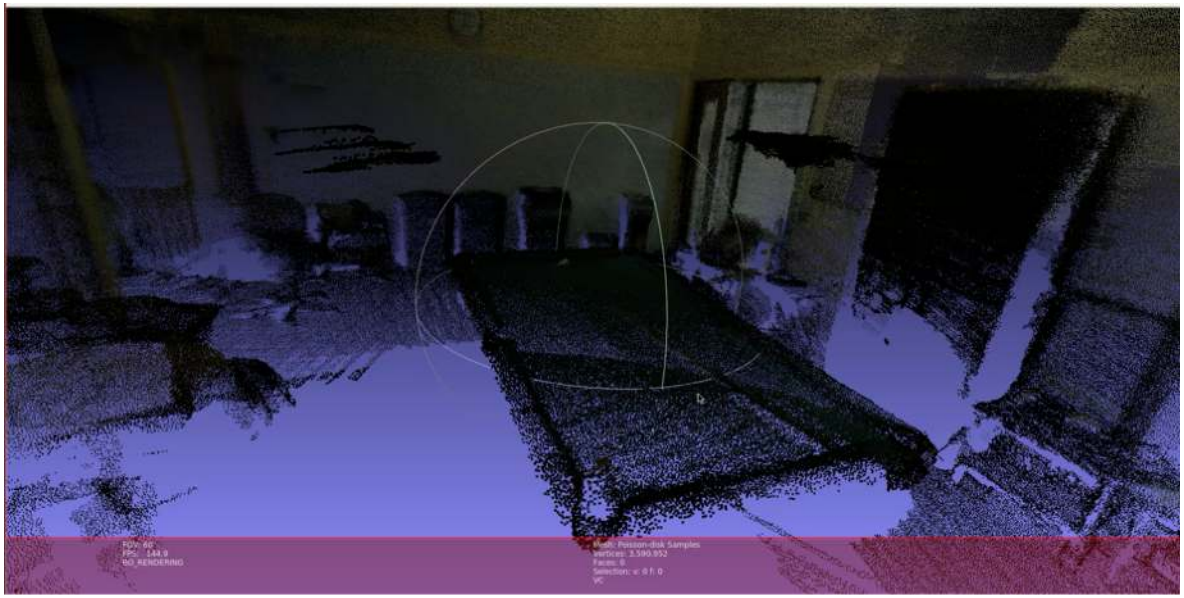
- Indoor LIDAR-RGBD Scan: 5 models
- Matterport 3D: 90 scenes
- ScanNet (Structure): 707 spaces
- Gibson (Matterport): 572 buildings
- ShapeNet (CAD): 51,300 models
- PanoContext (panoramic): 700 Panoramas

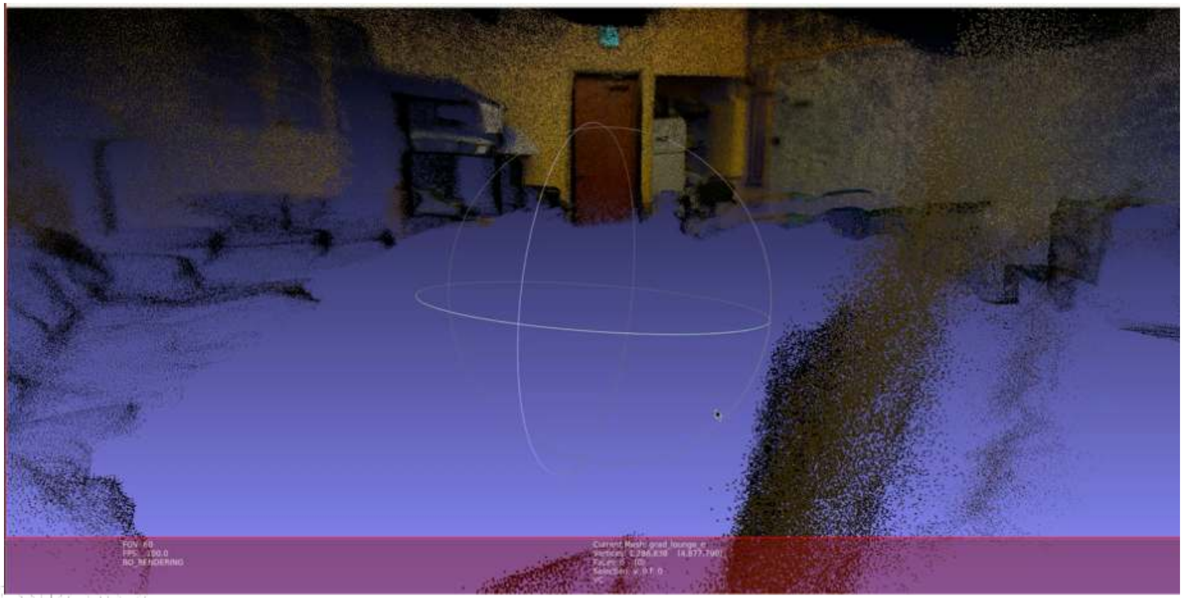
Berkeley ATLAS
— Multi-resolution database for geometric super resolution

- Cross-reference ground-truth LIDAR data with consumer-grade depth camera data (RealSense) for one-to-one correspondences
  - PX-80 LIDAR (construction grade)
  - RealSense D435i (consumer grade)
  - PointGrey stereo cameras

- Technical challenges
  - Multi-sensor synchronization (Arduino)
  - Multi-sensor calibration (manually)
  - Multi-sensor SLAM (OpenARK)
  - 3D labeling (Python CV toolkit)

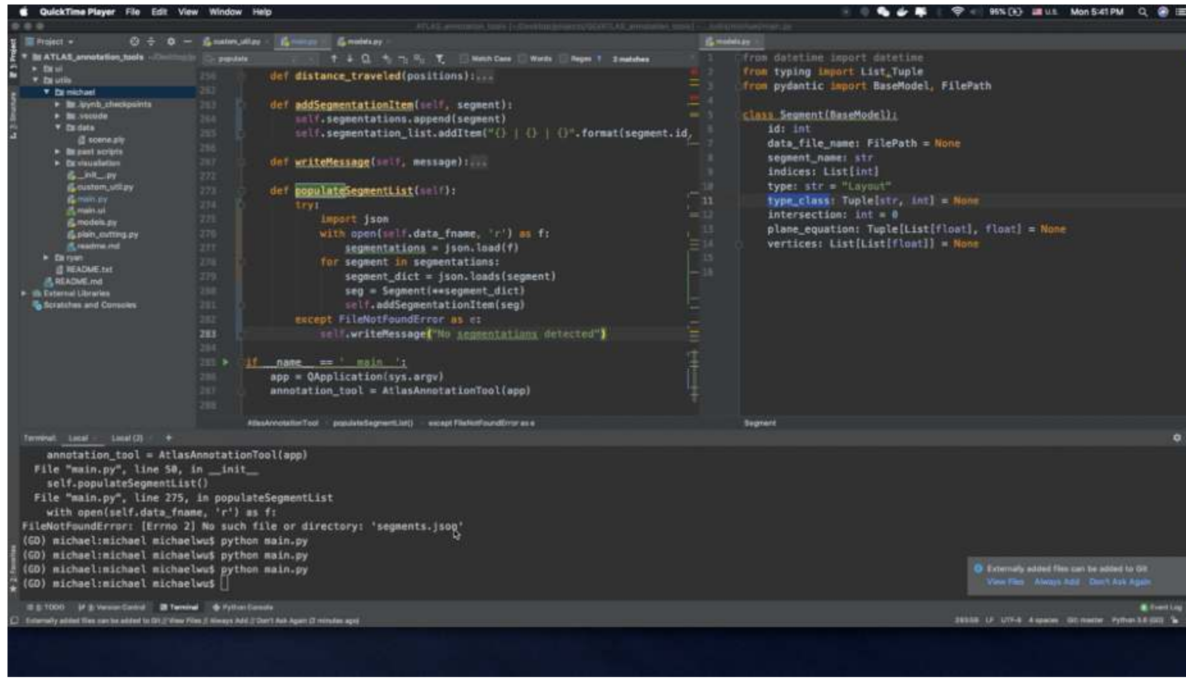
Berkeley
UNIVERSITY OF CALIFORNIA


FOV: 60
FPS: 144.9
BO_RENDERING
Mesh: Poisson-disk Samples
Vertices: 3,590,952
Faces: 0
Selection: v: 0 f: 0
VC


FOV: 60
BO_RENDERING
Current Mesh: grad_lounge_e
VC

A CV-assisted 3D labeling system

- Background room surfaces are first separated by clicking 3 points to define a surface
- Foreground objects are automatically bounded. We then label their categories
- Establish correspondence between LIDAR, RealSense, and images



Bottleneck in 3D Objects via Point Cloud

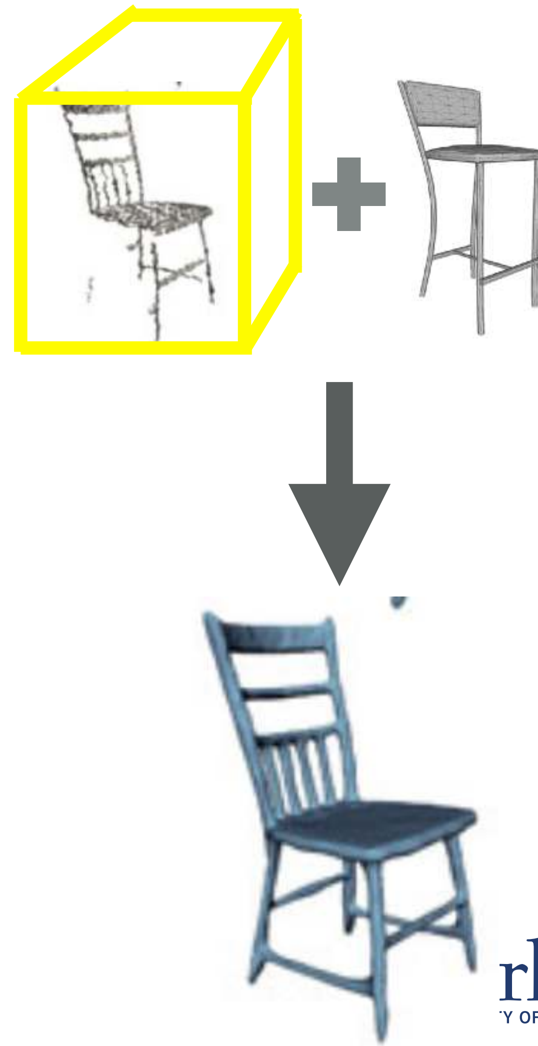
- Usually obtain an **incomplete** model of objects from RGB-D sensors.

- Due to noise, occlusions, or material properties

- **Task: Complete the 3D objects for accurate virtual augmentation**

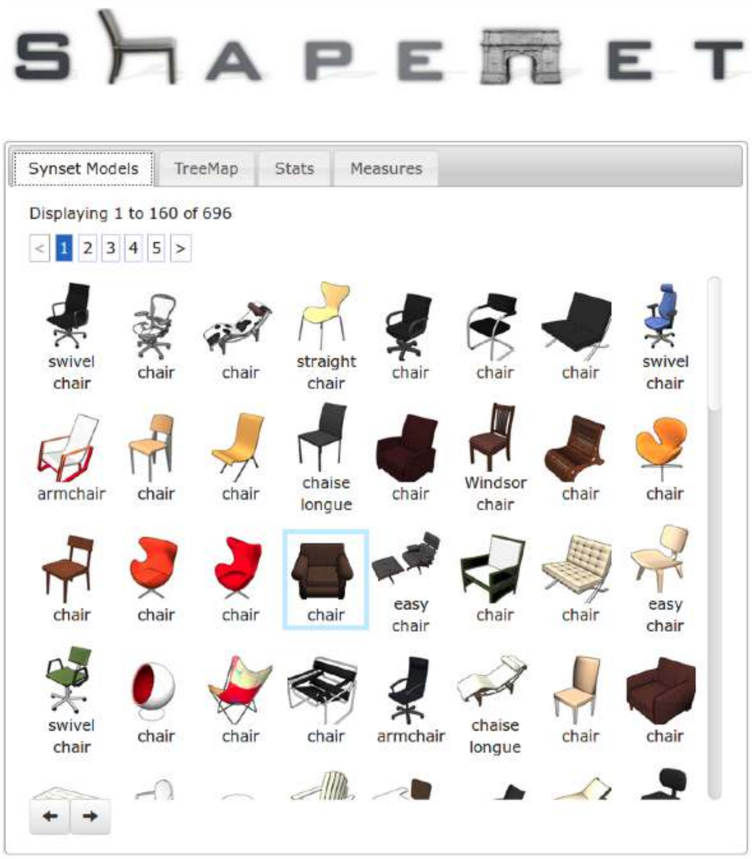
Approach: Deformable CAD Models

- Propose to use **deformable** CAD model
- Match observation (data) while being geometrically complete
- Also solve for optimal scale and rigid body transformation in addition



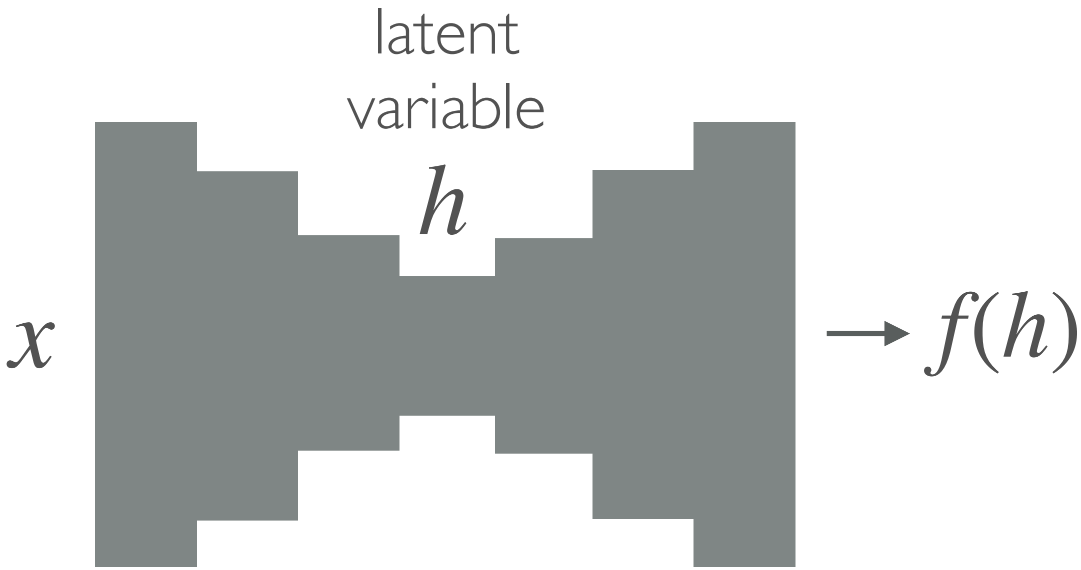
ShapeNet CAD Models

- ShapeNet [Chang et al. arXiv 2015] is a richly annotated large scale dataset of 3D shapes.
- Models are normalized to unit cube, so need to be scaled and rigidly transformed.
- Provides annotations for:
  - upright, front direction
  - parts information
  - Symmetry etc.



Berkeley
UNIVERSITY OF CALIFORNIA

3D Shape Completion



$$\min_h |f(h) - x|$$

- Auto-encoder network to estimate latent representation of the deformed CAD model space, which minimizes approximation error

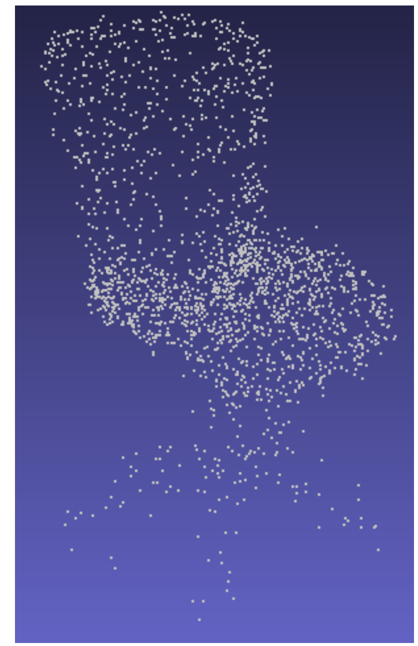
* Achlioptas, Panos, et al. "Learning representations and generative models for 3D point clouds." arXiv preprint arXiv:1707.02392, 2017.

Approach I: Auto-Encoder (AE)

- Estimation error created via point clouds

Original CAD

Object Point Cloud



Deformed CAD

AE Representation for 3D Point Clouds

- Point Cloud: an object is sampled by N 3D points

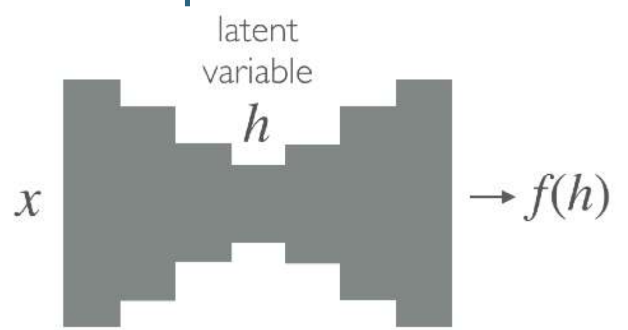
  S is an N-by-3 matrix

- Challenges with point clouds
  1. Point Cloud sample are ambiguous and not unique
  2. A set of 3D points are not ordered (compared to images and videos)

$$d_{CH}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2.$$

Auto-Encoder (AE) Objective

- A deep-learning architecture that learns to reproduce its input with most informative representation



- h is called the latent code for representing a family of input data

- Goal: Minimize reconstruction error

$$\mathcal{L}(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|^2 = \|\mathbf{x} - \sigma'(\mathbf{W}'(\sigma(\mathbf{W}\mathbf{x} + \mathbf{b})) + \mathbf{b}')\|^2$$

Applications of AE on Point Cloud

- Changing appearance

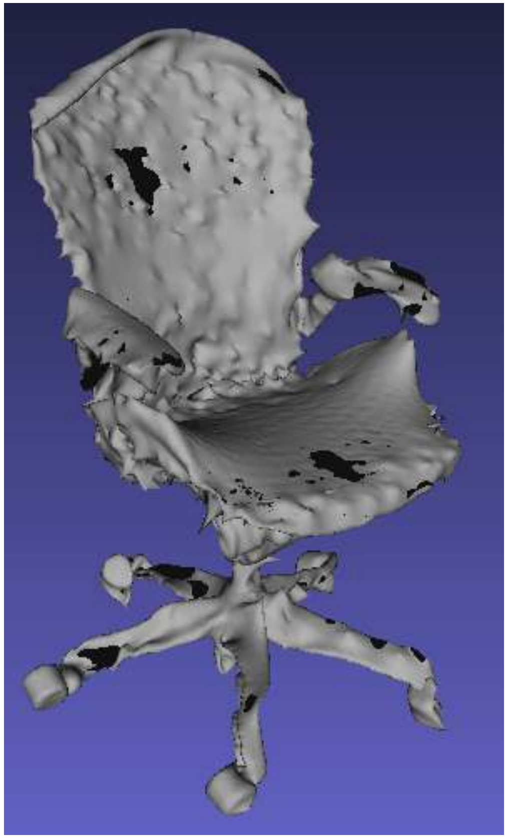
- Point-cloud completion

- Classification based on h (when trained on all categories)

	A	B	C	D	E	ours EMD	ours CD
MN10	79.8	79.9	-	80.5	91.0	**95.4**	**95.4**
MN40	68.2	75.5	74.4	75.5	83.3	84.0	**84.5**

* Achlioptas, Panos, et al. "Learning representations and generative models for 3D point clouds." arXiv preprint arXiv:1707.02392, 2017.

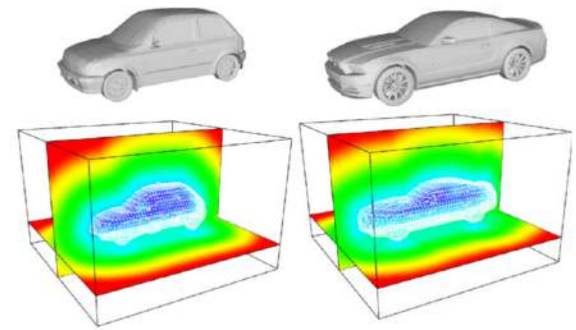
Drawback from AE Representation in Dense Shape Completion

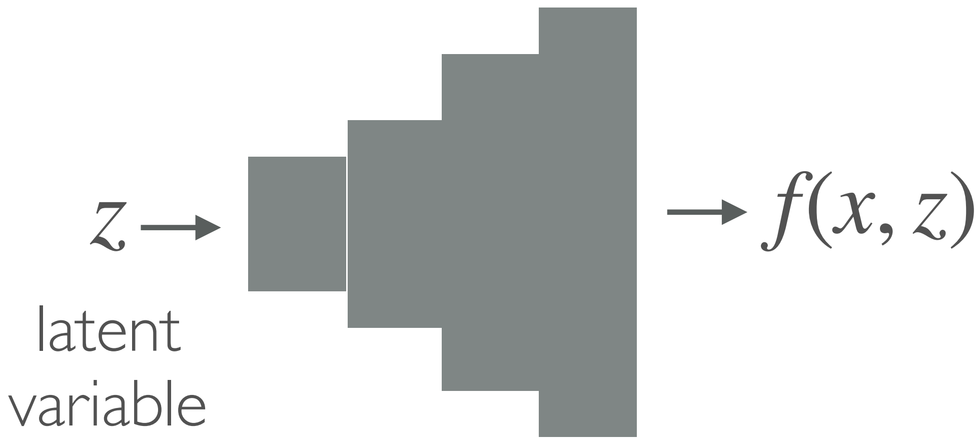


- Resulting point cloud minimizes the estimation error
- However, deformed mesh model may not be contextually plausible or visually appealing
- Generation of new shape is only related to the decoder part, but not the encoder

Approach II

- Auto-Decoder Network
- Continuous signed distance function





$$\min_{z} \sum_{x} |f(x, z) - y|$$

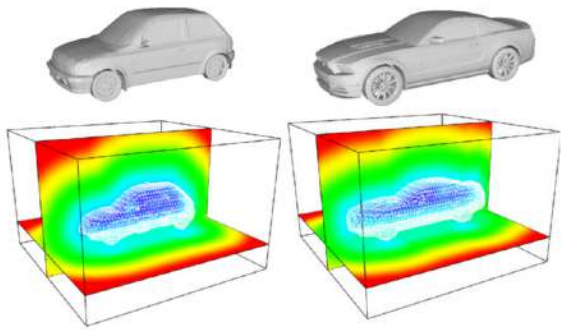
[2] Park, Jeong Joon, et al. “DeepSDF” CVPR, 2019.

Signed Distance Function

- SDF with respect to a set Ω

$$f(x) = \begin{cases} -d(x, \partial\Omega) & \text{if } x \in \Omega \\ d(x, \partial\Omega) & \text{if } x \in \Omega^c \end{cases}$$

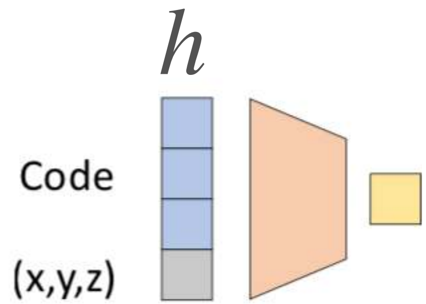
where $\partial\Omega$ is the boundary of the set

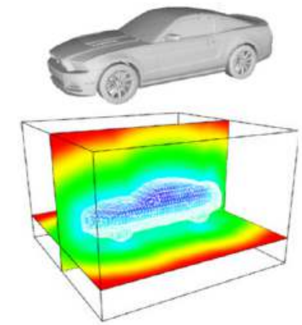


- SDF is a continuous function
- Magnitude of $\nabla f(x)$ is always unit (1)
- On the boundary, $\nabla f(x)$ is equal to the normal vector

Auto-Decoder (AD) Network

- Auto-Decoder Network





- How to inference optimal code
  - Training: assume each code $\{z_1, \ldots, z_N\}$ corresponds to one shape

$$\underset{\theta, \{\boldsymbol{z}_i\}_{i=1}^{N}}{\arg\min} \sum_{i=1}^{N} \left(\sum_{j=1}^{K} \mathcal{L}(f_\theta(\boldsymbol{z}_i, \boldsymbol{x}_j), s_j) + \frac{1}{\sigma^2} ||\boldsymbol{z}_i||_2^2 \right).$$

  - Testing:

$$\hat{\boldsymbol{z}} = \underset{\boldsymbol{z}}{\arg\min} \sum_{(\boldsymbol{x}_j, \boldsymbol{s}_j) \in X} \mathcal{L}(f_\theta(\boldsymbol{z}, \boldsymbol{x}_j), s_j) + \frac{1}{\sigma^2} ||\boldsymbol{z}||_2^2 .$$

[2] Park, Jeong Joon, et al. “DeepSDF” CVPR, 2019.

Berkeley
UNIVERSITY OF CALIFORNIA

Shape Completion

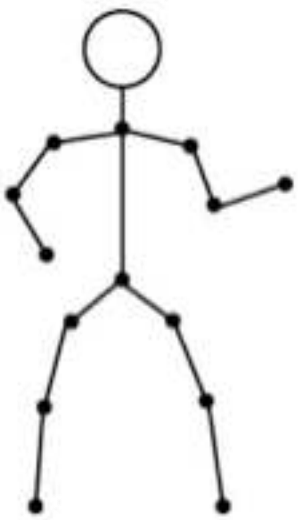
- Shape Completion Problem under AD Network

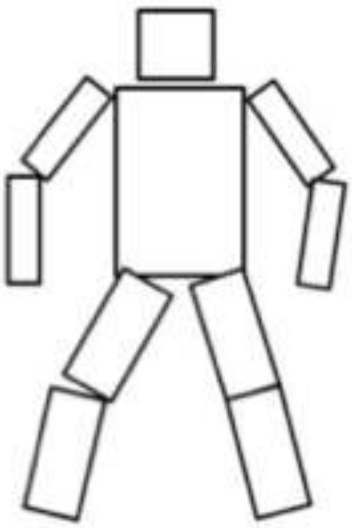
Finding the optimal z^* given trained shape parameters θ and partial observations $\{x_1, \ldots, x_K\}$

- The network can approximate any number of points, un-ordered.
- (x, y, z) can be any 3D point, so AD encodes continuous SDF.
- No encoder part is needed, therefore the main motivation to ignore during training.

Results with Improvements (ongoing)

3D Avatar Modeling


Skeletal


Fully Articulated


Deformable

3D Avatar Modeling

1. Photo-realistic video games
2. Social media
3. Telepresence
4. Human simulations

Deformable

Berkeley
UNIVERSITY OF CALIFORNIA

Existing Literature: Single-Camera Avatar Modeling

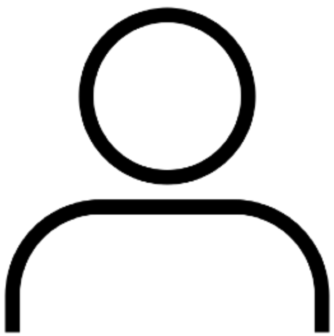
- **Template-based: Using body skeleton/silhouette**
  - Vlasic et al., 2008
  - Taylor et al., 2012
  - OpenPose: Zhe Cao, et al, 2018

- **Model-based: SMPL, SCAPE,** …
  - BodyFusion: T. Yu, et al., 2017
  - DoubleFusion: Tao Yu, et al., 2018
  - Delta: Federica Bogo, et al., 2015

- **Free form: Static vs Dynamic**
  - KinectFusion: ISMAR 2011
  - DynamicFusion: CVPR 2015

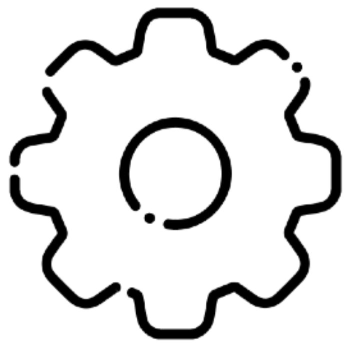
Berkeley UNIVERSITY OF CALIFORNIA

Our Approach



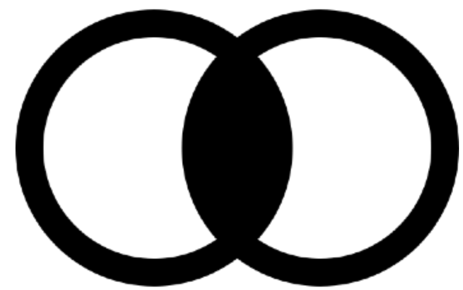
Low Dimensional Model

Increasing robustness and speed by working in a low-dimensional space.



Fast Solver

Towards high performance on low-compute devices.



Model Fusion

Combining 3D Depth + 2D RGB information for enhanced realism and robust tracking.

OpenARK Avatar Open-Source Library

Fusion of Multiple 3D Cues

$$\lambda_S E_S(\boldsymbol{\theta}, \boldsymbol{\beta}) + \lambda_J E_J(\boldsymbol{\theta}) + \lambda_P E_P(\boldsymbol{\theta})$$

ICP Error
Physical model

Joint Prior
Neural network model

Pose Prior
Probabilistic model based on human knowledge

Iterative Closest Point (ICP) Error

Sum squared distance from observed body to modeled body

Nearest neighbor on SMPL model to observed point

$$E_S(\boldsymbol{\theta}, \boldsymbol{\beta}) = \sum_i \|\mathbf{p_i} - S(\mathbf{p_i}; \boldsymbol{\theta}, \boldsymbol{\beta})\|^2$$

Observed Point

Joint Prior

Sum squared distance from CNN joint positions to model joint positions

Joint Position on SMPL Model

$$E_J(\boldsymbol{\theta}) = \sum_i \left\| \hat{\mathbf{J}}_\mathbf{i} - \mathbf{J}_\mathbf{i}(\boldsymbol{\theta}) \right\|^2$$

Joint Position Predicted by Neural Net

Pose Prior

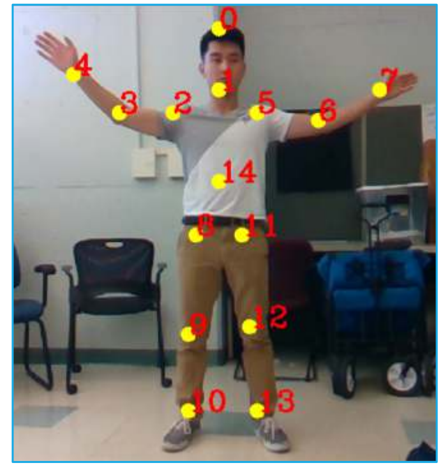
Likelihood of modeled pose being real human pose

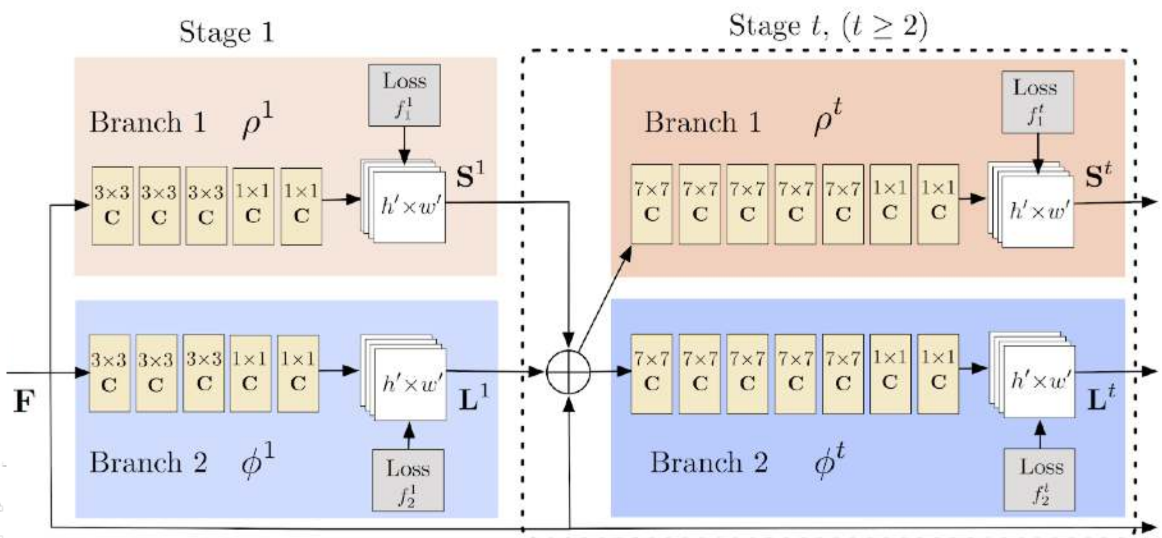
PDF Parameterized by Joint Pose

$$E_P(\boldsymbol{\theta}) = \min\left(-\log c_i \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_{\boldsymbol{\theta},i}, \boldsymbol{\Sigma}_{\boldsymbol{\theta},i})\right)$$

Weights Trained from CMU MoCap Dataset

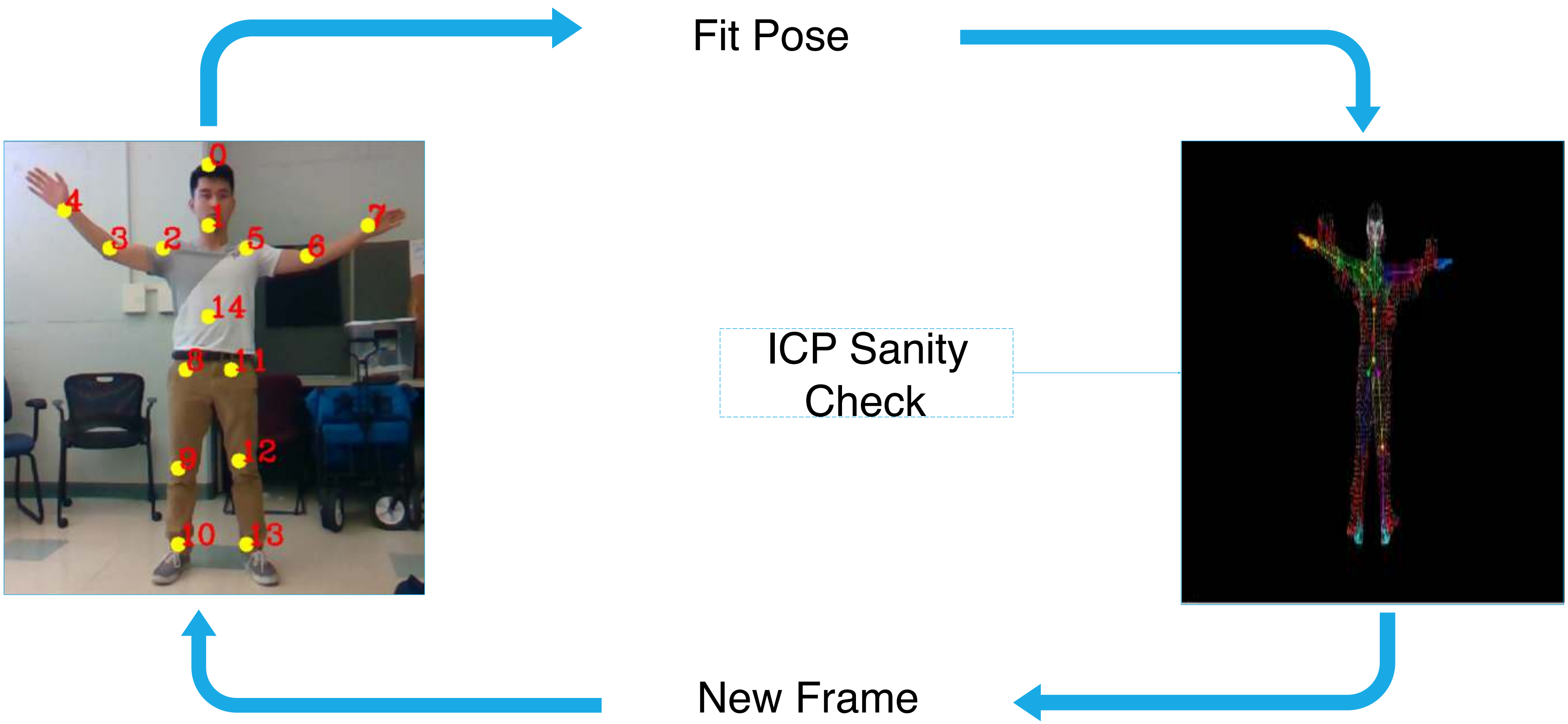
OpenPose as skeleton anchor





* Zhe Cao, et al. OpenPose: Realtime multi-person 2D pose estimation using part affinity fields, 2018

Basic Process



GitHub: https://github.com/augcog/OpenARK