Synthetic Data and Graphics Techniques in Robotics

Prabindh Sundareson, 2022

National Symposium on Trends and Developments in

Intelligent Systems & Robotics (TD-ISR'22)

PES University Bangalore

Agenda

- Role/Definition of Data
- Types of Data – Random, Time, 2D, 3D
- Graphics techniques – Applications in Intelligent systems design
- Part 1 – Synthetic data and 3D processing in Navigation
- Part 2 – Synthetic data and 2D objects in Vision processing
- Challenges
- Conclusion

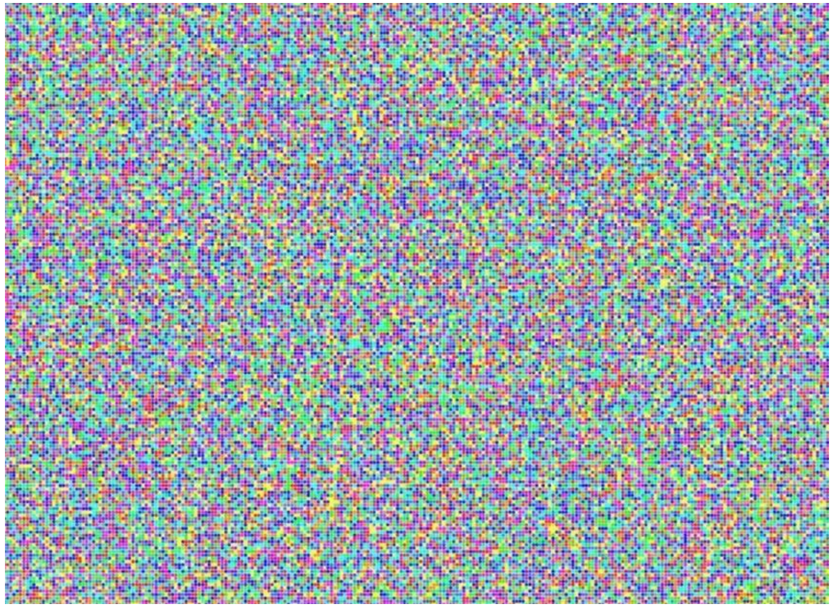
Role of Data in Intelligent systems

- Function (task) Learning
  - Fit a given set of data points to a function (generator)
- Feature Understanding
  - How relevant is a feature, correlations across features
- Forecasting
  - Probability of an event occurring, given some data
- And so on …

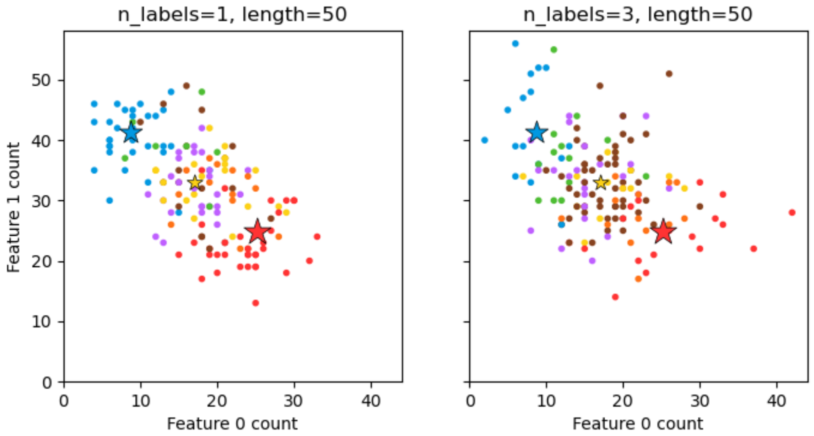
In Machine Learning, training with more data has led to better results

What is Data ?

- Simplest – “Random number”
  - Also, time series, etc



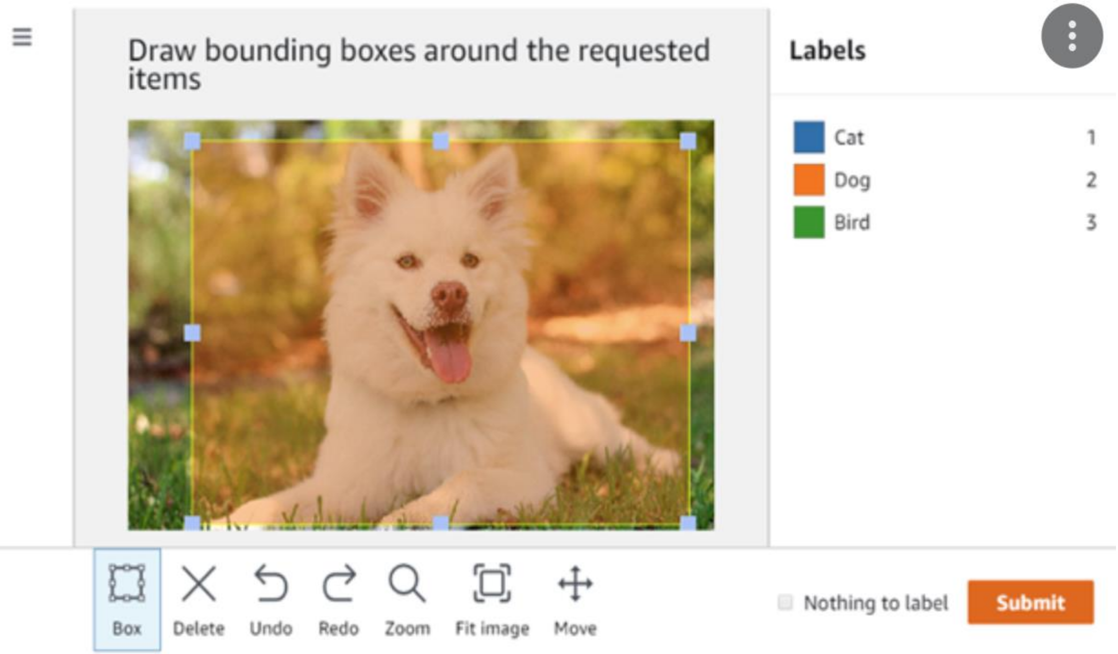
- import sklearn.datasets as dt
  RANDOM_SEED = np.random.randint(2**10)
- dt.make_multilabel_classification



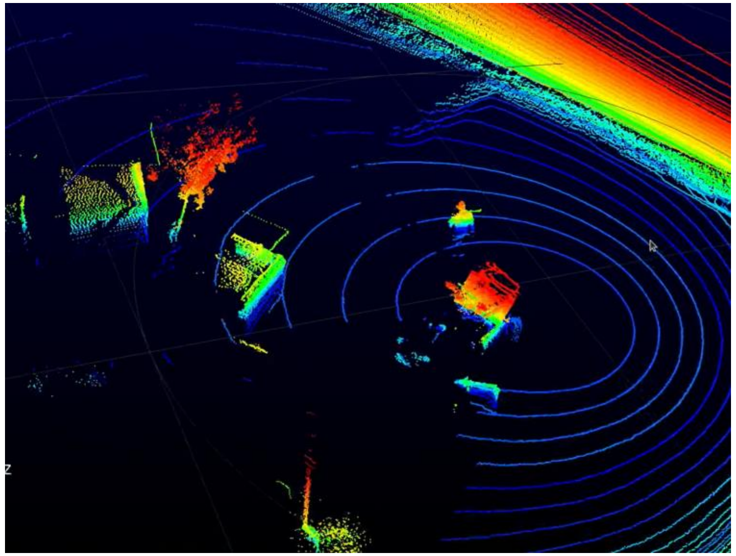
What is Data (continued)

- Onto 2D data – What processing is involved ?
  - Annotated images for image classification
  - Labelled (bounding-box) images for object detection
  - Segmented (pixel-wise) images for object segmentation

- These activities require effort !
  - Currently being solved with human effort



What is Data (Continued)



- Onto 3D data – What type of data ?
  - LiDAR point-clouds
  - Orientation data
  - 3D Meshes ..

- As dimensions increase, the data, annotations /ground-truth generation gets more complex

Synthetic Data =

Data generated with the aid of algorithms

Why synthetic data – In summary

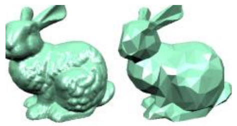
- Cost and speed of acquisition and annotation (ground-truth – boxing/labelling, segmentation .....)
- Privacy issues
- Consequences are virtual
  - Ex Robots in Medical systems
- Ability to configure – coarse to fine features
  - Ex, In object detection
    - ball vs non-ball (coarse), ball vs what-type-of-ball (fine features)

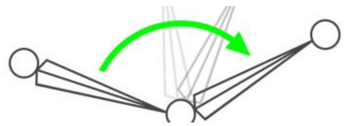
Synthetic Data Vault
https://sdv.dev

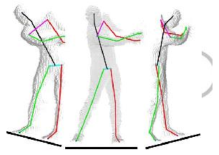
What if, we could solve both problems –

Data generation, and Annotation, together ?

Data in 2D and 3D – Topics in Robotics and Graphics

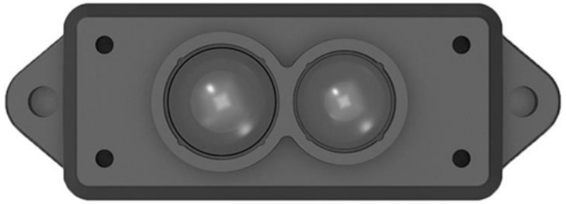




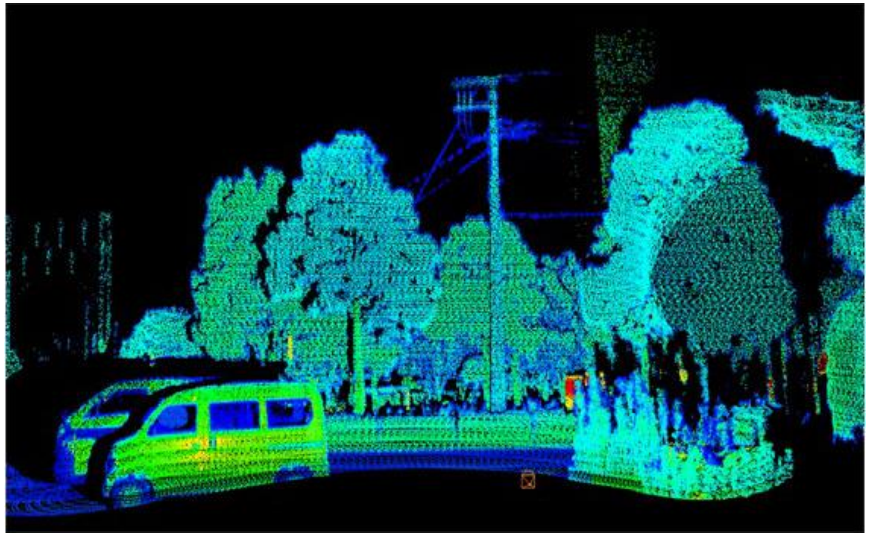


Topic\Field	Robotics field	Graphics field
Efficient 3D processing	3D points - Localisation and mapping in Robotics	Synthetic data/ conversions/generation in rendering 3D meshes
Kinematics	Movement control in Robotics	Animation in Graphics Rendering
Pose	Pose estimation for orientation detection in Robotics	Animation transfer to arbitrary 3D meshes

Part 1 - Robotics Navigation and 3D processing



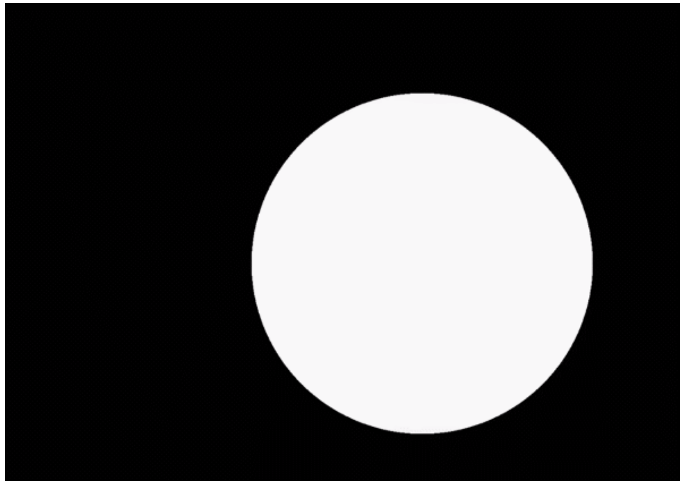
- How do intelligent systems navigate an unknown area ?
  - On-board sensors, on-board analysis
  - Satellite connectivity, external inputs
  - Sensors on UAV/drones, combined inputs
- Onboard sensors – Radar, LiDAR, ToF, Stereo cam, depth, IMU ..
  - Obtain 3D map of the environment
- Flow
  - Sensor → Data preprocessing (Ground plane, align)
- What can we do with a Point-cloud ?
  - SLAM (Simultaneous Localisation and Mapping)
    - Registration, Loop closure etc ..
- What else ?



https://www.ncos.co.jp/en/products/3dpoint.html

What “else” can be done with a Point-Cloud ?

- Learn “Geometry” via Signed Distance Field

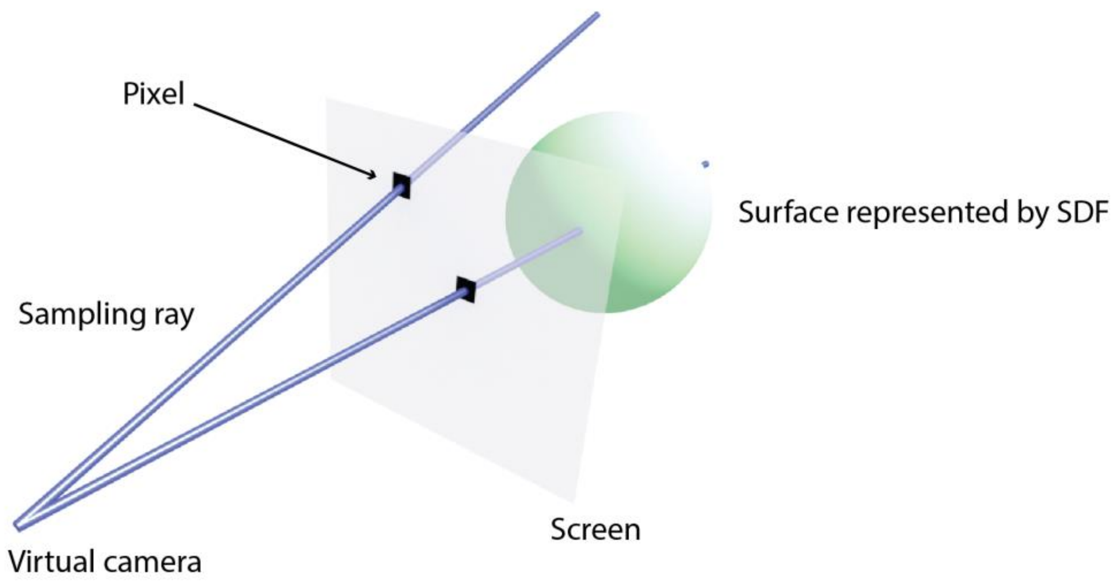


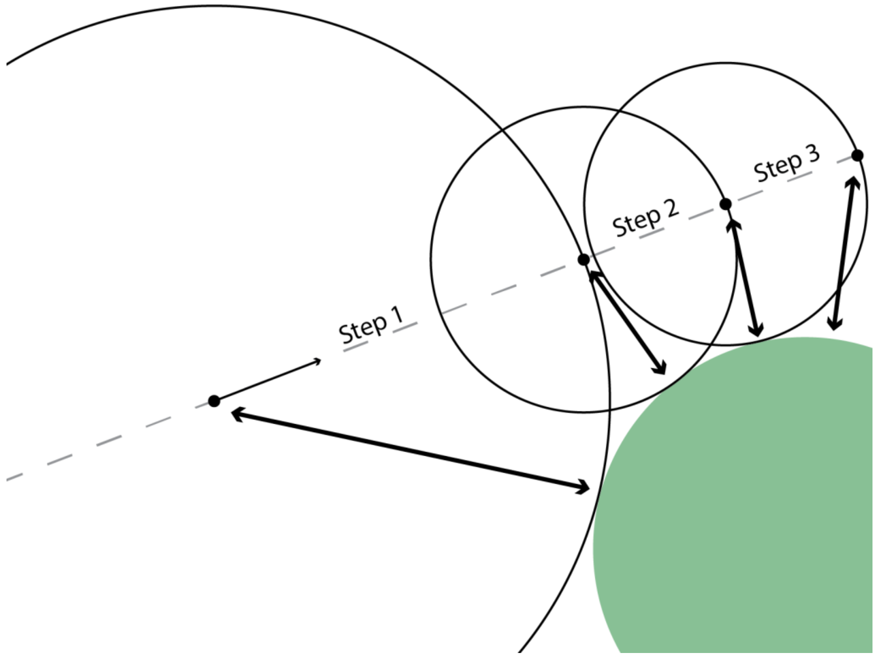
https://www.youtube.com/watch?v=XuSnLbB1j6E

“A Volumetric Method for Building Complex Models from Range Images”, Curless and Levoy, 1996

“DemoScene” in popular art sub-culture

Implicit Representations in SDF





Steps
1. Raymarching step by step
2. Surface hit ? → Angle of ray, normals

<u>SDF of circle (r=1)</u>

$f(x,y,z)$= sqrt(x2+y2+z2)−1

Why

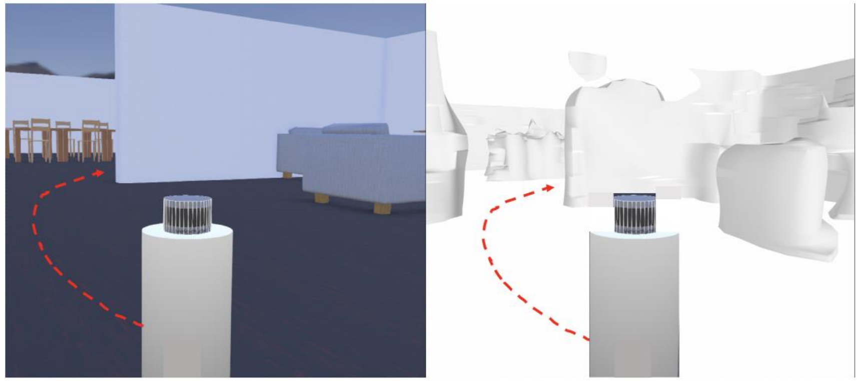
Why SDF ? Some commonalities

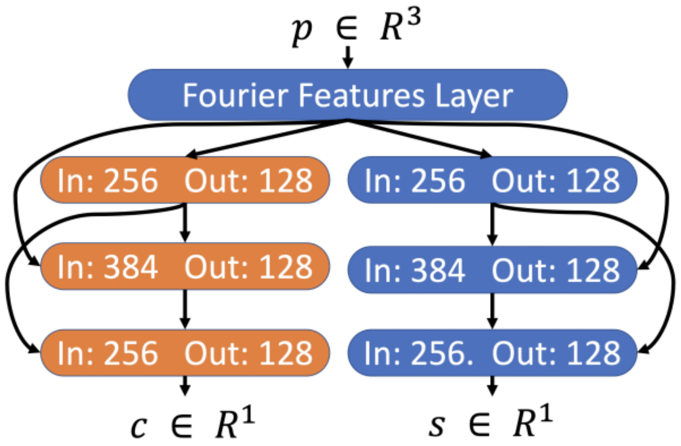
- Concepts of
  - “distance” - Similar in 3D rendering and Robotics navigation
  - “operations” – union, intersection etc (ex the Snail)
- Robotics – Distance provided by a sensor, ex LiDAR, or learnt from RGB

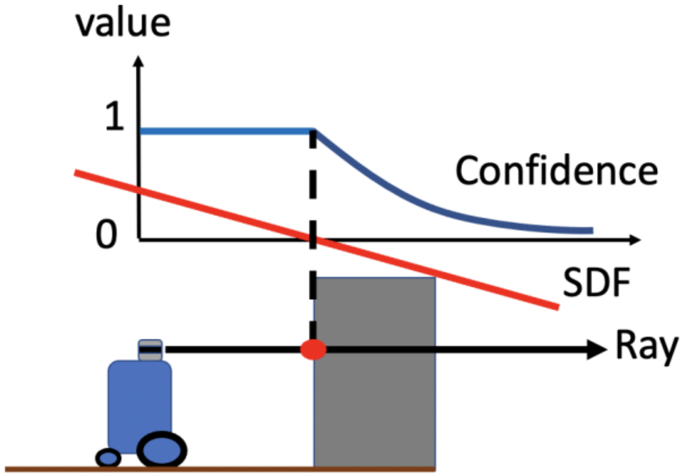
- Application to Robotics - “Signed Distance Fields: A Natural Representation for Both Mapping and Planning” – Oleynikova et al, 2016
- How do we generate the SDF !!

*https://www.youtube.com/watch?v=bMhR-UYlKqU - 3D Surface Reconstruction Using a Generalized Distance Function, 2016, Microsoft Research
* Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision – Niemeyer et al, CVPR 2020
* Neural Geometric Level of Detail: Real-time Rendering with Implicit 3D Shapes – Towaki et al 2021
* DeepSDF - Learning Continuous Signed Distance Functions for Shape Representation, Jeong Park et al, 2019
* On the Effectiveness of Weight-Encoded Neural Implicit 3D Shapes – Thomas Davies et al, ICML 2021 (https://github.com/u2ni/ICML2021)
* Neural-Pull, Ma et al, ICML 2021 - https://github.com/mabaorui/NeuralPull-Pytorch , learns directly from 2D-images, does not require ground truth SDF

(a) SDFs from LiDAR (using neural networks)



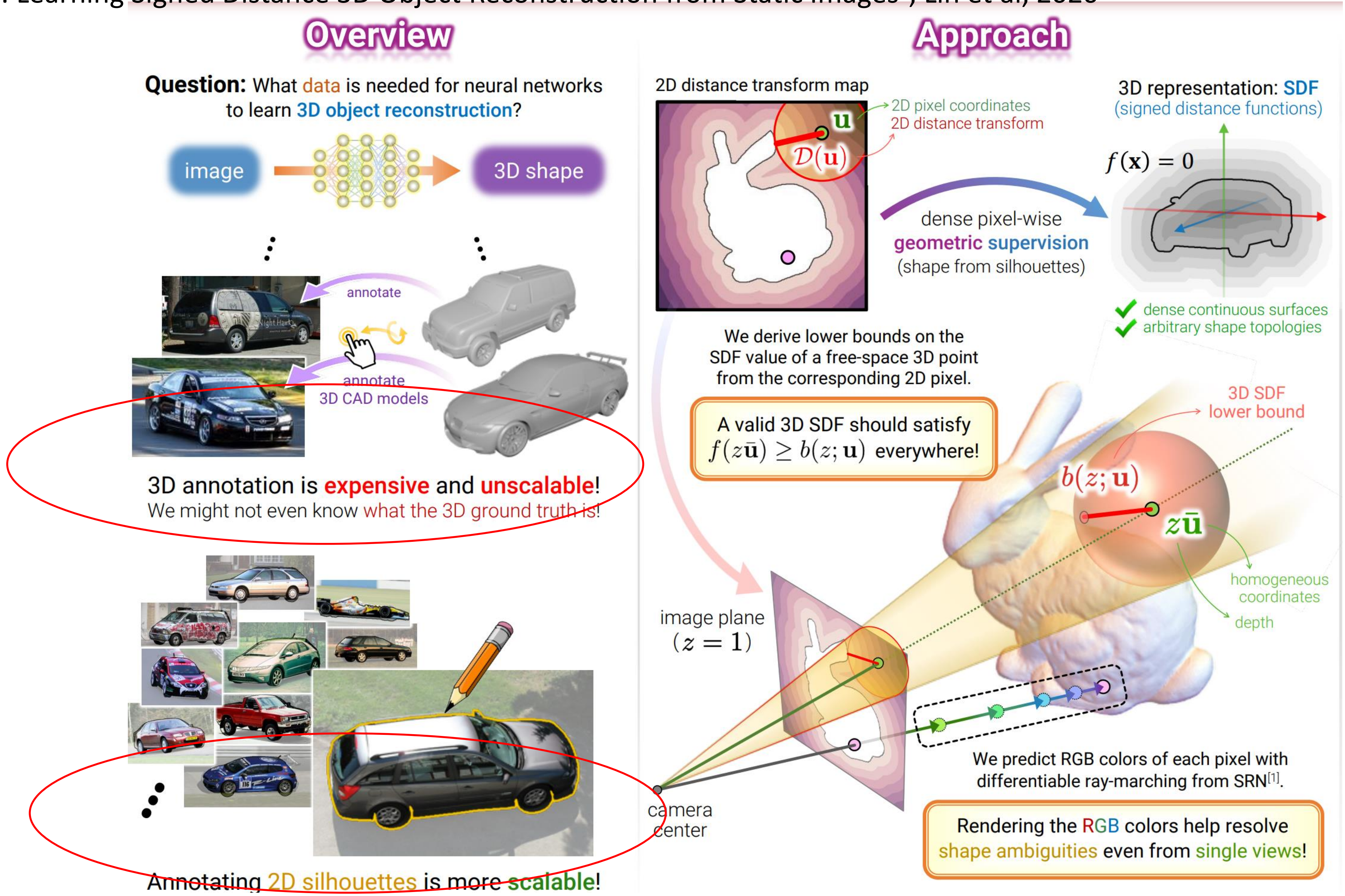




- Learn SDFs directly from LiDAR point clouds
- "Learning Deep SDF Maps Online for Robot Navigation and Exploration" – Camps et al, 2022
  - Neural network to regress SDF (per-frame) from LiDAR data
  - Hausdorff distance between successive SDFs used to determine mapping changes
  - Optimization steps applied for generating navigation stages

(b) SDFs from RGB images/Silhoutte (using neural networks)

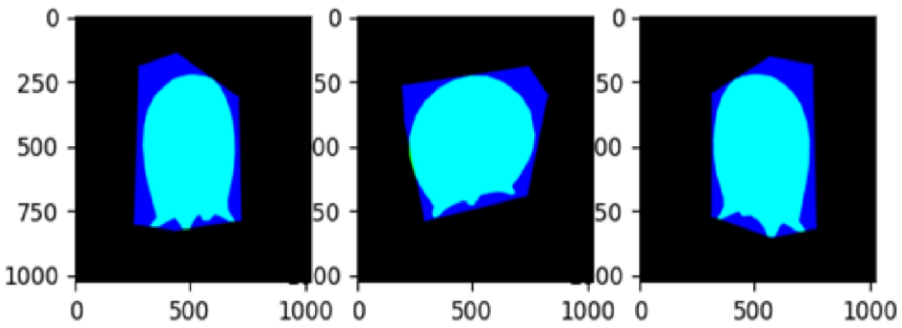
“**SDF-SRN**: Learning Signed Distance 3D Object Reconstruction from Static Images”, Lin et al, 2020



Kaolin

Tool - Differentiable Rendering for training

- “Differentiable rendering can be used to optimize the underlying 3D properties, like geometry and lighting, by backpropagating gradients from the loss in the image space”
  - “Learning to Estimate 3D Object Pose from Synthetic Data”, Zakharov, 2020
- Tools:
  - Kaolin - A Pytorch Library for Accelerating 3D Deep Learning Research
    - Ex – Fit a 3D bounding box around a 2D image
      - [Example Kaolin link]
    - Ex – Classification using point-cloud directly
      - https://colab.research.google.com/drive/1DoBlEt0G



Part 2 - Synthetic Datasets from 3D Games

- Precise Synthetic Image and LiDAR (PreSIL) Dataset for Autonomous Vehicle Perception – Hurl et al, 2019
  - From Grand Theft Auto V (game)
  - By scanning the “depth” buffer
  - 50k+ HD images with
    - full resolution depth information,
    - semantic segmentation (images),
    - point-wise segmentation (point clouds), ground point labels (point clouds), and detailed annotations for all vehicles and people
  - Improvement of up to 5% average precision on the KITTI 3D Object Detection
- Others - https://paralleldomain.com/

Don.riccobene

Synthetic 2D Datasets by Augmentation

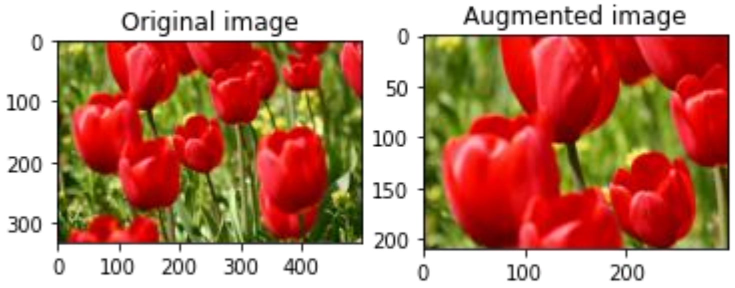
- Augmentation = Add data by applying operations on existing images
- Augmentation can improve accuracy / robustness on Vision tasks

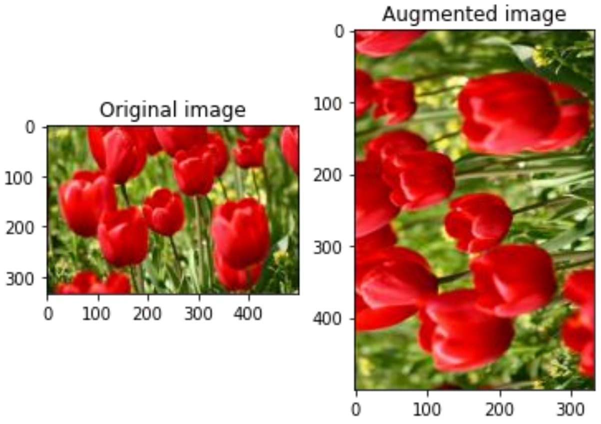
Metrics	Statistics	without Augmentation	with Augmentation One-Shot	Two Shot
Accuracy (%)	Mean	78.62	99.07	99.34
	Min	63.59	97.35	98.87
	Max	91.16	99.7	99.80
	STD	7.89	0.72	0.34
Precision (%)	Mean	81.06	98.85	99.35
	Min	72.61	95.45	98.66
	Max	90.31	99.77	99.66
	STD	6.29	1.40	0.33
Recall (%)	Mean	91.85	99.71	99.74
	Min	78.28	99.09	99.43
	Max	99.32	100	100
	STD	7.57	0.36	0.25
F1-Score (%)	Mean	85.91	99.28	99.54
	Min	75.34	97.67	99.21
	Max	94.03	99.77	99.83
	STD	5.28	0.66	0.23

Robertas Damaševičius et al, 2021

(a) 2D Image Augmentation in Frameworks

Examples from TensorFlow





https://www.tensorflow.org/api_docs/python/tf/image

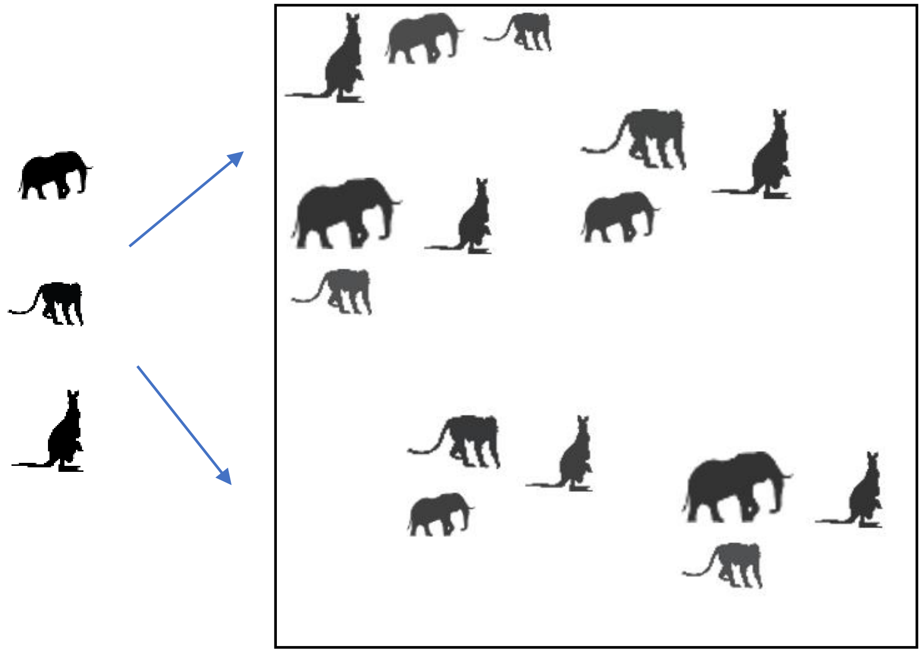
PyTorch

random_crop = torchvision.transforms.**RandomCrop**(size)
fivecrop = torchvision.transforms.**FiveCrop**(size)
horizontal_flip = torchvision.transforms.**RandomHorizontalFlip**()
vertical_flip = torchvision.transforms.**RandomVerticalFlip**()
random_persp = torchvision.transforms.**RandomPerspective**()
random_jitter = torchvision.transforms.**ColorJitter**()
normalize = torchvision.transforms.**Normalize**([0, 0, 0], [1, 1, 1])

https://pytorch.org/vision/main/transforms.html

Tools in industry:
Roboflow,
CVEDIA (Synthetic data) …

Optimal 2D augmentation for object detection



Augmented + multi-class combined-canvas image

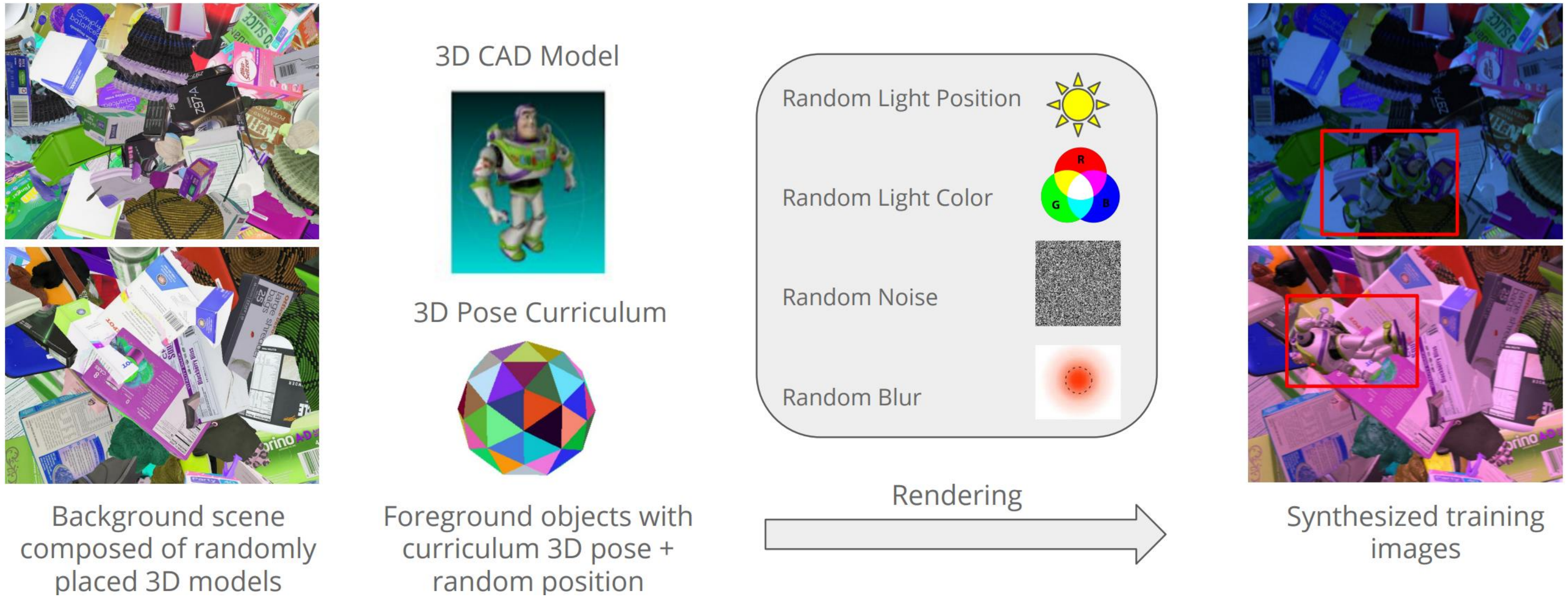
1. Small original set of n-class representative images = input
2. Perform augmentation on original n-class images at runtime
3. 2D rectangle packing into larger canvas at runtime (dynamically)
4. Multiple augmentation-packing steps results in combined canvas images
5. Multiple-classes in one image reduces storage requirements significantly

Example multi-object augmented output (Pascal VOC format)


```
<annotation>
  <object>
    <difficult>0</difficult>
    <name>1</name>
    <bndbox>
      <xmin>5</xmin>
      <ymin>5</ymin>
      <xmax>55</xmax>
      <ymax>37</ymax>
    </bndbox>
  </object>
  <object>
    <difficult>0</difficult>
    <name>0</name>
    <bndbox>
      <xmin>70</xmin>
      <ymin>5</ymin>
      <xmax>120</xmax>
      <ymax>34</ymax>
    </bndbox>
  </object>
  <object>
    <difficult>0</difficult>
    <name>2</name>
    <bndbox>
      <xmin>5</xmin>
      <ymin>49</ymin>
      <xmax>30</xmax>
      <ymax>77</ymax>
    </bndbox>
  </object>
  <filename>"C:\Users\psundareson\Downl
  <size>
    <width>416</width>
    <height>416</height>
    <depth>3</depth>
  </size>
</annotation>
```

(b) Synthetic Data for Recognition – from 3D

- "An Annotation Saved is an Annotation Earned: Using Fully Synthetic Training for Object Instance Detection." –
  - Hinterstoisser et al, Google AI, 2019 (Generate 2D images from 3D)

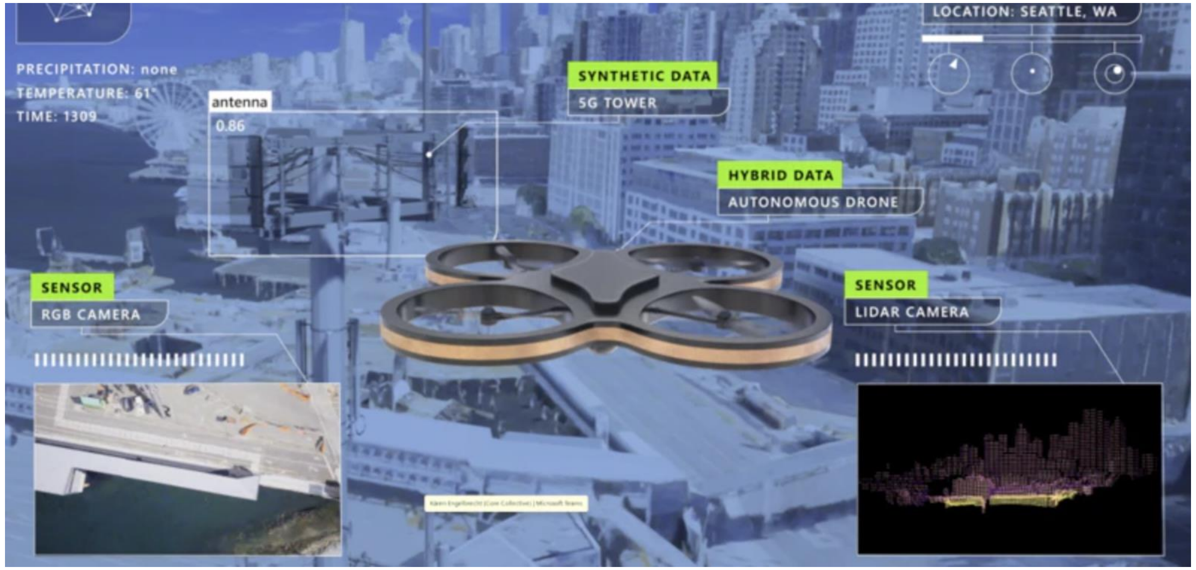


For more realistic data – Start from BIM (Building Information Models)

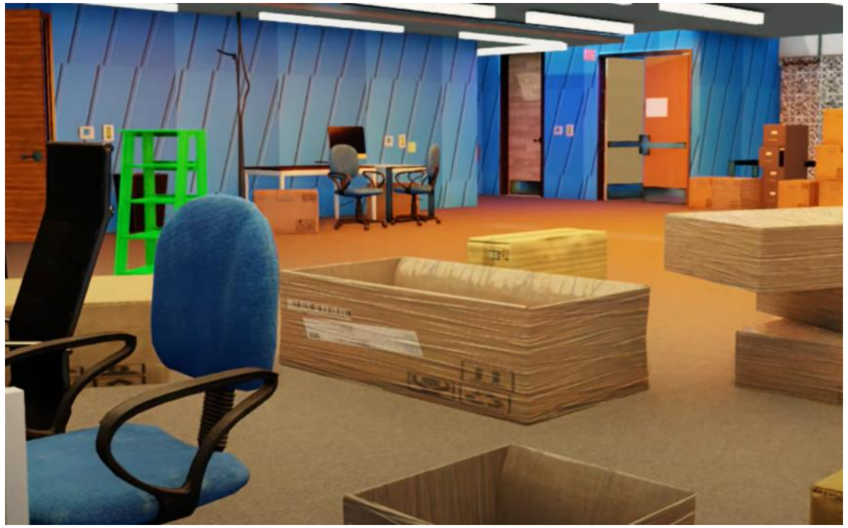
Synthetic Data – for Drone Flight Controls

- AirSim works as a flight simulator for drones
- Built as Unreal Engine plugin
- Allows drone producers to train their machines to work in risky and dangerous places.
- Roadmap to Autonomous driving

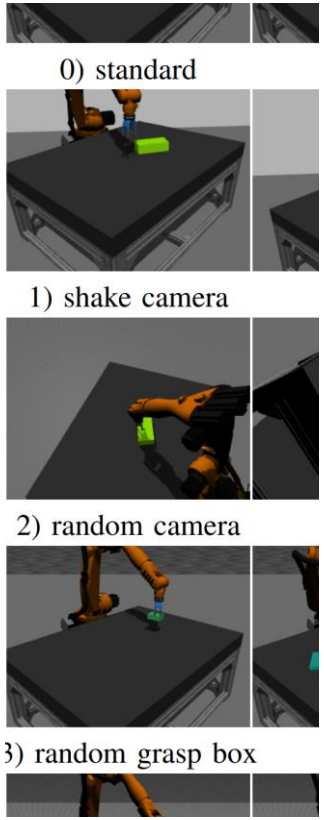
- "Aerial Informatics and Robotics Platform", Microsoft Research, https://www.microsoft.com/en-us/research/project/aerial-informatics-robotics-platform/



Challenges in Synthetic Data (Object detection)



- Appearance gap –
  - Pixel differences, lighting, shadows, textures, …
- Content gap –
  - Missing/ new objects



- Domain Randomization – from Synthetic data to real world data
  - https://www.youtube.com/watch?v=O6KEKI3abl0&t=11s
  - "Evaluation of Domain Randomization Techniques for Transfer Learning" – Grun et al, 2019
  - Brings results close to real-world datasets

Future

Future of Synthetic Data

- Definite improvements in performance in current use-cases
- Much of the data today is encoded for the “man-in-the-middle”
  - Ex, Traffic signs
- Possible to generate and train on synthetic data
  - That may be more efficient for Robotics
    - Ex, glyphs that encode much more data than human-readable-signs
  - More efficient for storage (or even no-storage, if metadata is stored)

- **Synthetic Data could become the “real” data for Intelligent systems of the future**

Conclusion

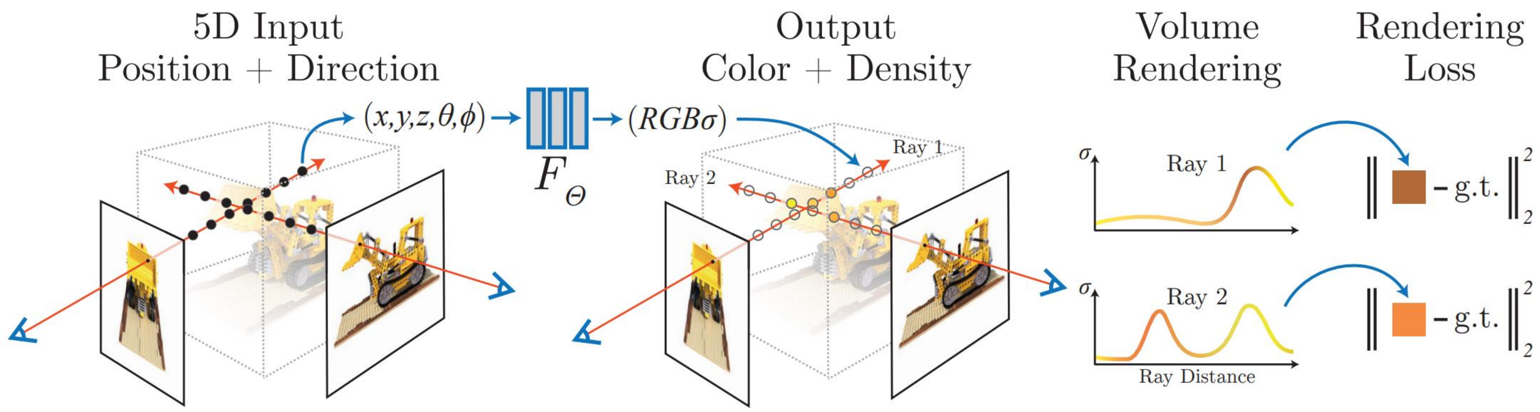
- Synthetic Data is valuable, not just for augmentation purpose, but also increases the overall robustness of recognition systems
- New techniques like online-SDF learning (using differentiable rendering) can increase the capabilities of robotics systems to learn from surroundings
- Possibility of machine learning without intermediate human involvement or “interpretation” step

Backup - Resources for Vision/Compute/Data

- OpenCV
- VPI Vision Programming Interface (VPI)
  - https://developer.nvidia.com/embedded/vpi
- Robotics Simulators – Gazebo, Webots, Isaac (ROS adaptation)
- SDF Datasets - FAMOUS, ABC
- 3D Model datasets – ModelNet10, 40
- SDFs introduction
  - https://www.youtube.com/watch?v=8--5LwHRhjk
  - https://jasmcole.com/2019/10/03/signed-distance-fields/
  - http://jamie-wong.com/2016/07/15/ray-marching-signed-distance-functions/
  - http://graphics.stanford.edu/courses/cs468-10-fall/LectureSlides/04_Surface_Reconstruction.pdf
- Raymarching – Procedural generation of images
  - Introduction - https://www.youtube.com/watch?v=PGtv-dBi2wE
  - https://iquilezles.org/articles/raymarchingdf/
  - https://www.shadertoy.com/view/WsSBzh
  - https://www.shadertoy.com/view/3lsSzf
- "Neural RGB-D Surface Reconstruction" – Azinovic et al, CVPR 2022

Neural Radiance Fields

- Radiance - a differential opacity controlling how much radiance is accumulated by a ray passing through (x, y, z)
- Utilises Differentiable volume rendering, ray marching
- “NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis”, MildenHall et al, ECCV 2020 (https://github.com/bmild/nerf)
  - Represent a continuous scene as a 5D vector-valued function whose input is a 3D location x = (x, y, z) and 2D viewing direction (θ, ϕ), and whose output is an emitted color c = (r, g, b) and volume density σ



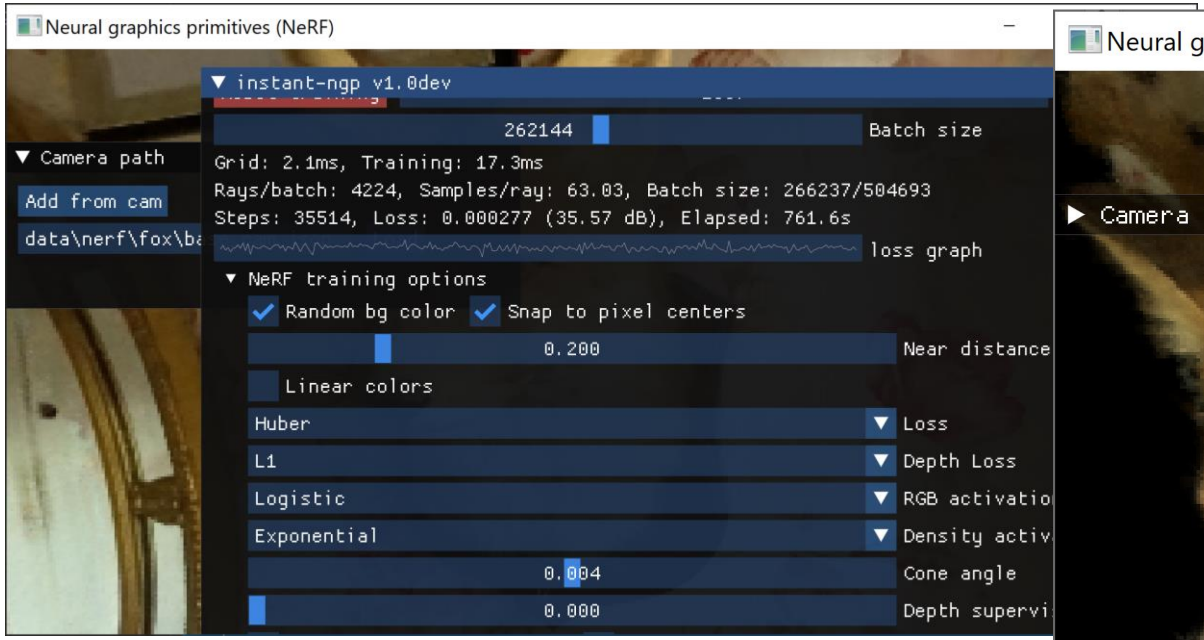
Instant NeRF - network

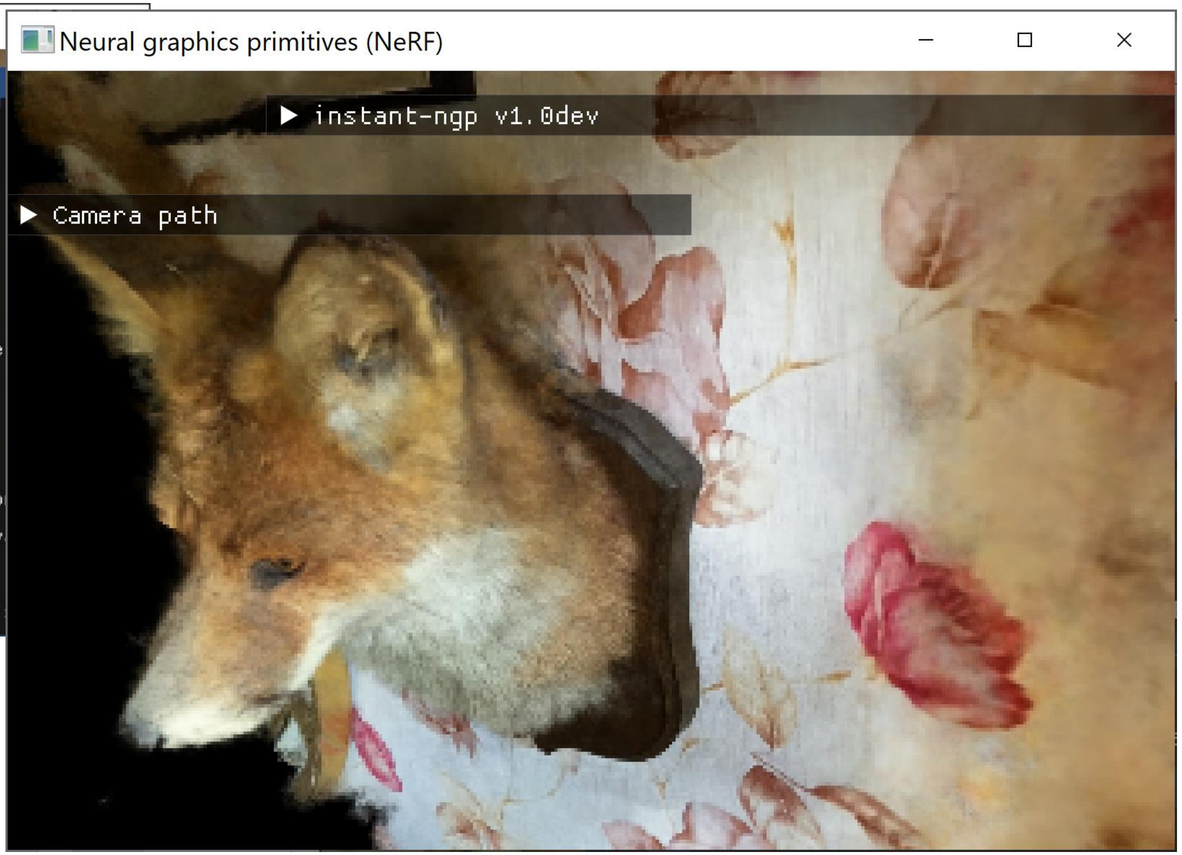
- Network configuration ?
- Uses Huber Loss
- Requires camera position

- NeRF networks with joint optimization for camera position ie operating only on image frames also emerging

Instant NeRF training

- 2 seconds !





COLMAP steps

1. Feature detection and extraction
2. Feature matching and geometric verification
3. Structure and motion reconstruction
    - up vector was [-0.90487011 0.10419316 0.41273947]
    - computing center of attention...
    - [-12.27771833   1.27148092 -0.09314177]
    - avg camera distance from origin 12.701290776903985
    - 55 frames
    - writing transforms.json

Adoption of feature-encoding layers

- As applicable to “Coordinate networks”
- “Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains” – Tancik et al, 2020
  - Concept of Neural Tangent Kernels (NTK)

- Faster convergence
- Examples
- https://colab.research.google.com/github/ndahlquist/pytorch-fourier-feature-networks/blob/master/demo.ipynb

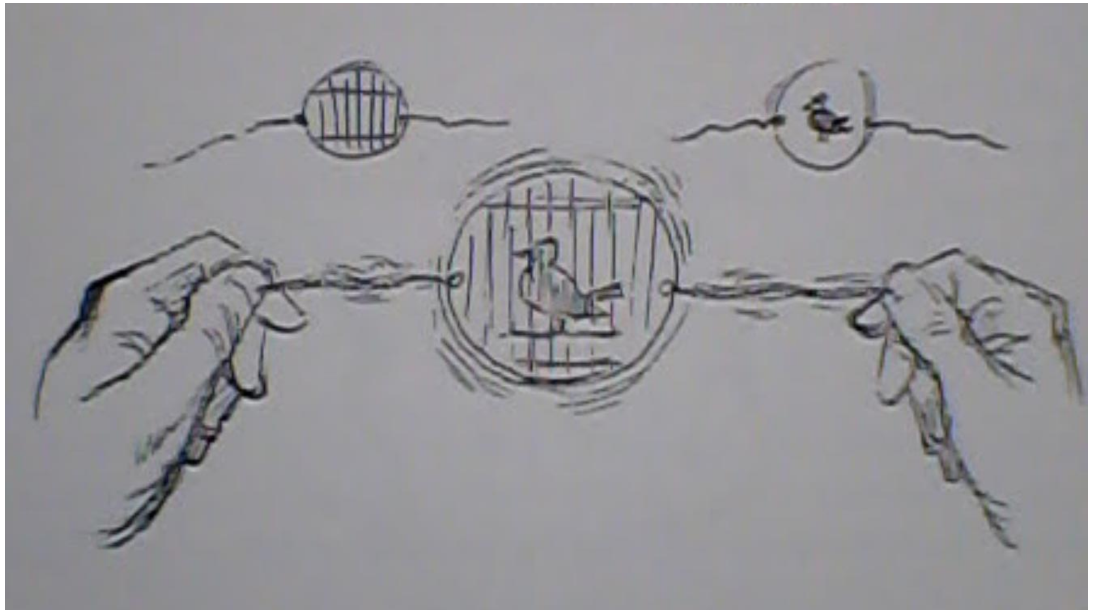
Differentiable Volume Rendering

- Makes a mesh “learnable” from the sensor data
  - LiDAR data
- Eikonal Equation

  $||\nabla f||=1$

  Learns a SDF

Animation - Thaumatrope



Core Techniques 3 - Inverse Kinematics

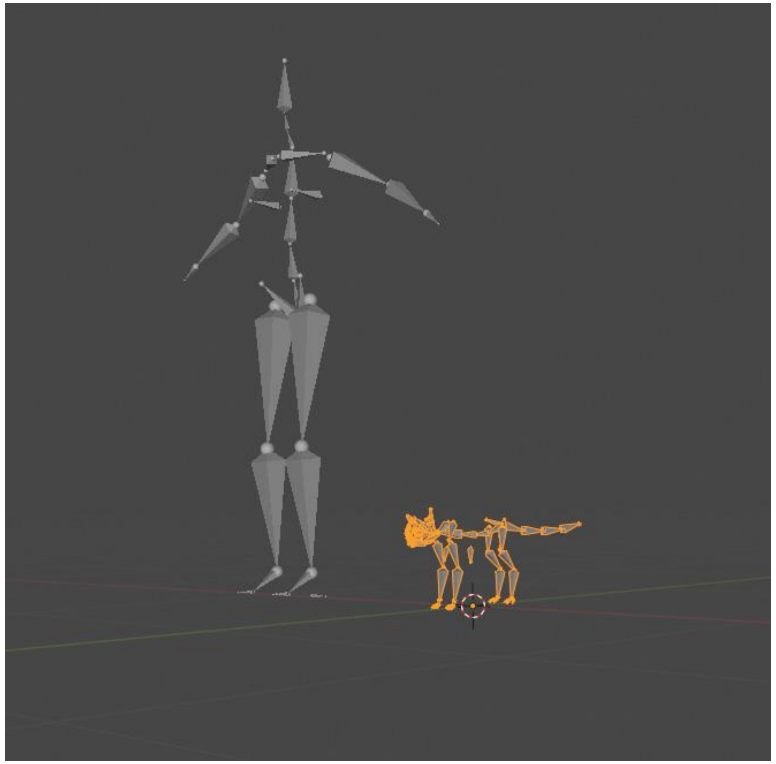
- Find valid orientations for intermediate joints, given an end position
- Gradient Descent (cyclic coordinate descent(CCD))

- Deep Reinforcement Learning (DDPG) approaches

Image courtesy IIITH

Armature and Bones

Blender – Armature (Skeleton) controls the character movement (rigging)



Physics Simulators

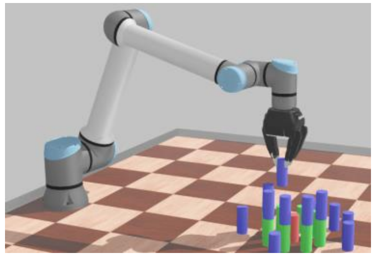


TABLE 1. Feature comparison between popular robotics simulators.

Simulator	RGBD + LiDAR	Force Sensor	Linear + Cable Acutator	Multi-Body Import	Soft-Body Contacts	DEM Simulation	Fluid Mechanics	Headless Mode	ROS Support	HITL	Teleoperation	Realistic Rendering	Inverse Kinematics
Airsim	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓, unreal	✗
CARLA	✓	✗	✗	✗	✗	✗	✗	✓	✓	✗	✓	✓, unreal	✗
CoppeliaSim	✓	✓	Linear only	✓	✗	✓	✗	✓	✓	✓	✓	✗	✓
Gazebo	✓	✓	Linear only	✓	✗	Through Fluidix	Through Fluidix	✓	✓	✓	✓	✗	✓
MuJoCo	✓	✓	✓	✓	✓	✓	Limited	✓	✗	HAPTIX only	HAPTIX only	✗	✗
PyBullet	✓	✓	Linear only	✓	✓	✓	✗	✓	✗	✗	✓	✗	✓
SOFA	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓, Unity	✗
UWSim	RGBD only	✓	✗	✓	✗	✗	✗	✓	✓	✓	✓	✓, custom	✗
Chrono	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗	✓	✓, offline	✓
Webots	✓	✓	linear	✓	✗	✗	Limited	✓	✓	✗	✓	✗	✗

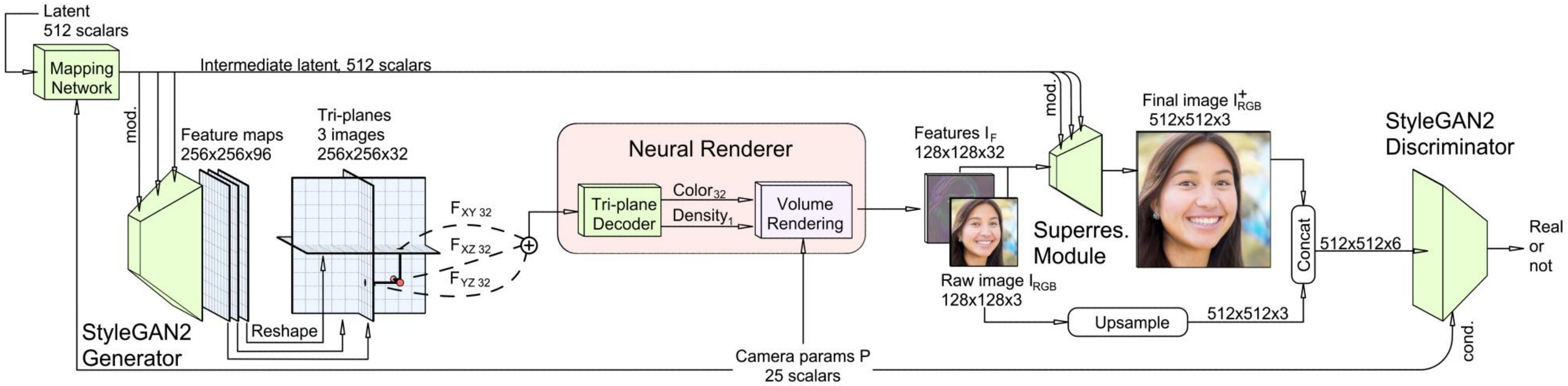
A Review of Physics Simulators for Robotic Applications – Collins et al, 2021

Intelligence for Autonomous Agents

- Ex Minecraft
  - Survival
  - Harvest
  - Combat
- Training with 750,000 Minecraft YouTube videos, scraping from Minecraft Wiki, ...
- Tasks
  - Programmatic tasks
  - Creative tasks.
- https://github.com/MineDojo/MineDojo

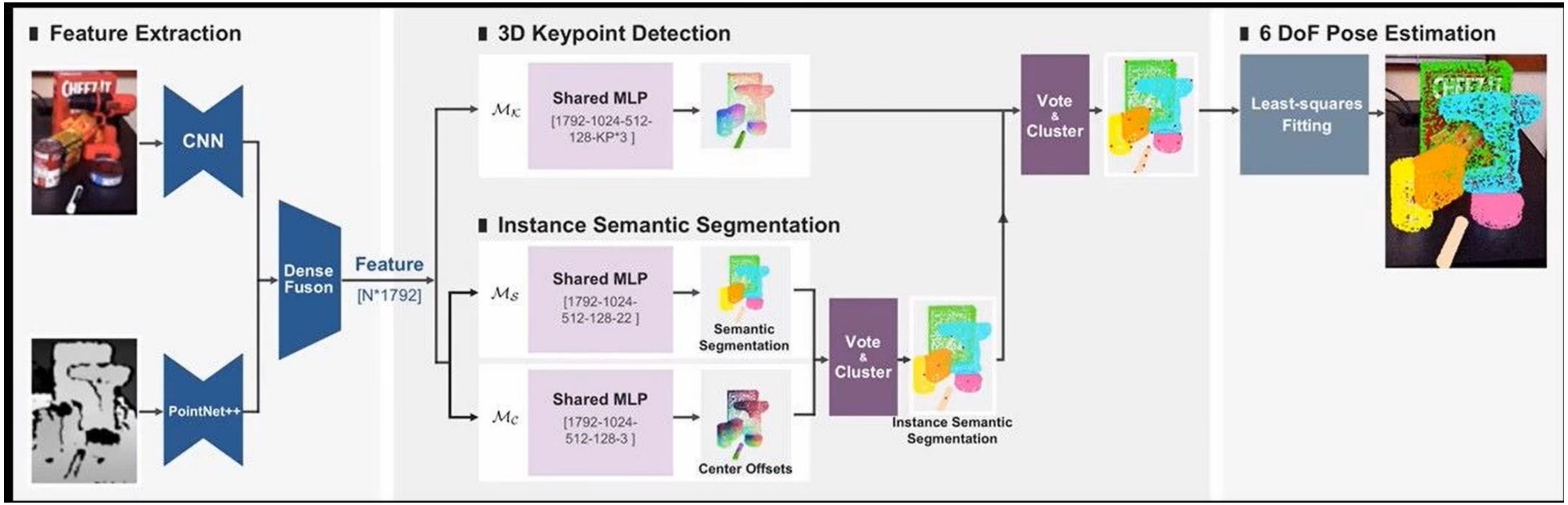
2D-3D

- EG3D: Efficient Geometry-aware 3D Generative Adversarial Networks - *Chan et al, 2022*
- Unsupervised generation of high-quality multi-view-consistent images and 3D shapes using only collections of single-view 2D photographs

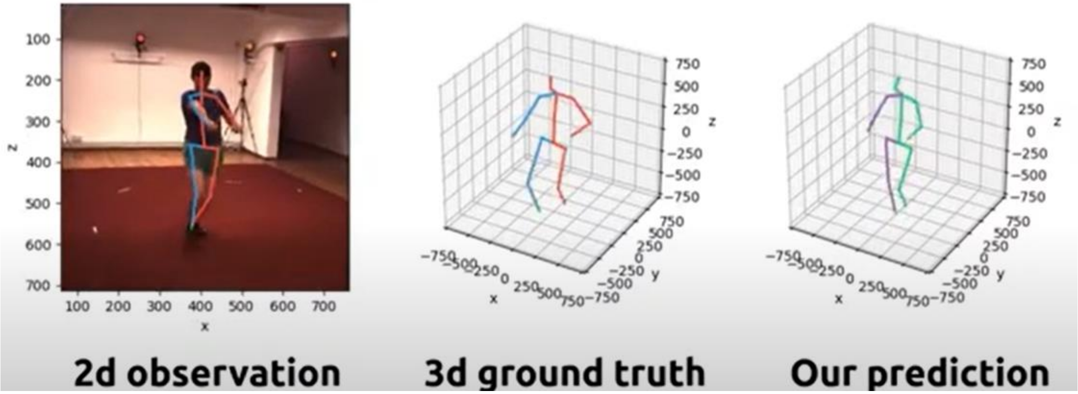


Topic 2 - Pose

- Important for gripping in Robotics
- PVN3D - 3D keypoint estimator from 2D images
- Trained via Synthetic data, won 2nd in OCTOC challenge



Pose in Graphics Animation

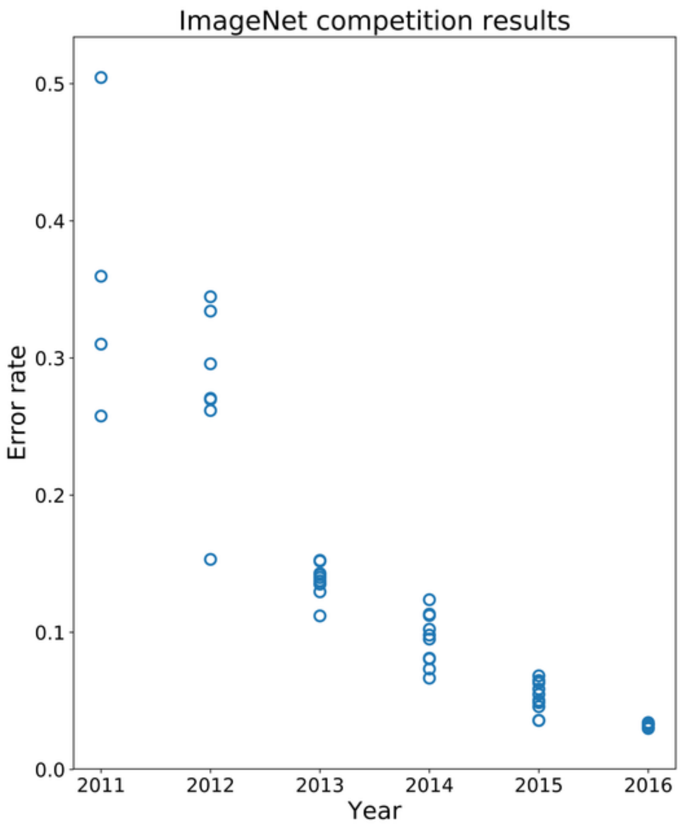


- 2D video to 3D avatars
  - Translate pose in 2D to any 3D object
- Steps
  - Body-pose estimation (2D)
    - Similar techniques as in Robotics, applied to human body landmarks
  - 2D to 3D mapping
  - Map to synthetic object’s keypoints
    - Mapping can be innovative and depends on the use-case

SDF from point clouds (Traditional)

- (CSCI 621: Digital Geometry Processing), 2019
- Construct SDF from point samples - Distance to points is not enough
  - • Need inside/outside information
  - • Requires normal vectors
    - • Examine local neighborhood for each point
      - • Compute best approximating tangent plane , covariance, MST tree construction,..
- Implicit Reconstruction - Estimate signed distance function (SDF) • Extract Zero isosurface by Marching Cubes • Approximation of input points • Result is closed two-manifold surface

Does data change ?



- Early stage research vs productization challenges
- Conditions for data
  - Sufficiency and relevancy
- Data needs to be relevant to the task
- Some data = static datasets
- Production goals demand new data, continuous training

Types of Synthetic Data

- Time-series (Movement, Distance, Sensory, ..)
- Visual data
  - Augmented Natural Data
    - Rotated, Resized, color ranges, Blurred, Lit, …
  - Synthetic Data
    - Generated by simulations (ex Robotic movements)
    - Generated by Graphics API (ex Synthetic face/body, animation)
  - Combined
    - Generated by human in a virtual environment
      - ex Flight simulators

https://varjo.com

Synthetic data generation

- "Act against Multiplication" – 1404 Law, outlawing "synthetic creation" of gold and silver, Britain
- Today, we can realistically generate any data (almost)
    - Sensor Data (Gazebo, Webots.. simulators)
    - Visual Data (Unity, UE, Omniverse, ..)
    - Columnar Data (numpy, SDV, Gretel ..)
- Synthetic != Fake, has to match the use-case

DALL-E-2

Why

LiDAR vs Depth

- LiDAR more accurate, but object segregation difficult
- A fusion of depth, RGB required for good results in both navigation and detection use-cases