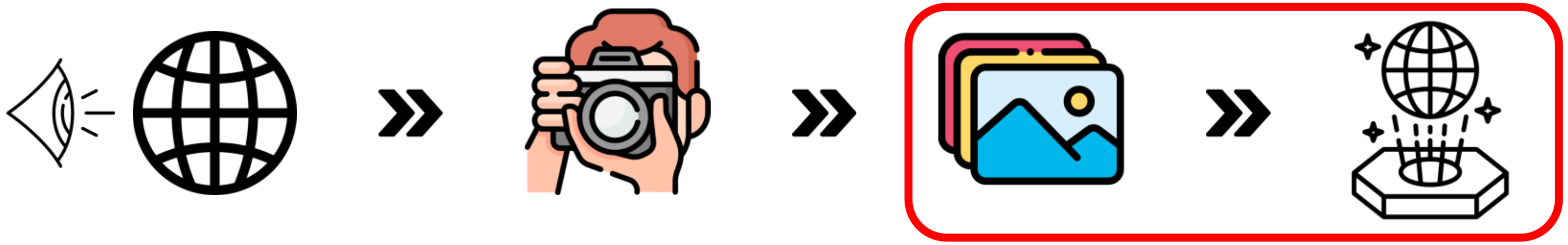
Chair of Computer Graphics and Visualization

Accurate Depth Extraction from 3DGS Models

Evelyn Regina Sidarta, 23.04.2025

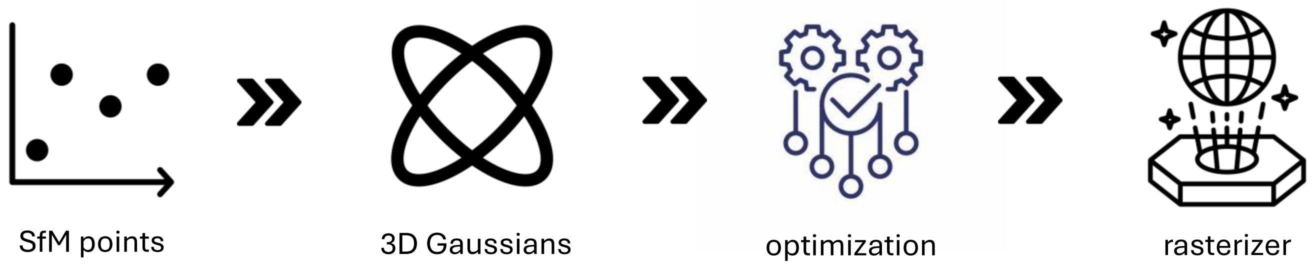
Motivation – 3D Reconstruction



- Visualization and reconstruction of 3D scenes – used to aid multiple fields, e.g. medical imaging, aerospace, aviation
- NeRF (2020): accurate 3D reconstruction from collection of 2D images using neural networks

Intensive resource consumption, failure to achieve real-time display

Motivation – 3D Gaussian Splatting

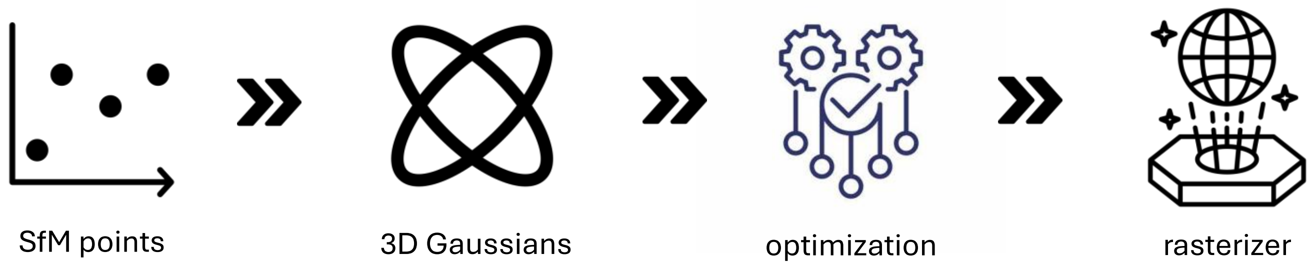


- 3DGS (2023): 3D Gaussians instead of traditional meshes to reconstruct scene.

  Idea: skip conversion into surface or line primitives, directly „splat“ Gaussians to paint a scenery.

  Optimization: adjust distribution of Gaussians, minimize resource consumption in bland areas photogrammetry vs. 3DGS

Motivation – 3D Gaussian Splatting



- Limitations: “fuzzy” representation, hard to determine geometry of the scene.

How to create accurate geometric representation?

How to identify depth accurately?

Contribution

Review mechanisms of current available state-of-art methods for depth estimation:

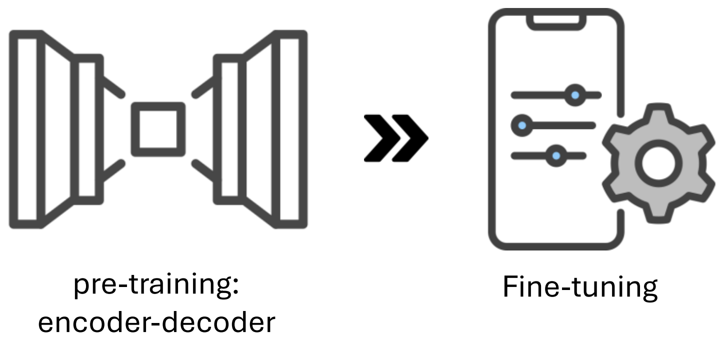
MDE models vs. Gaussian Splatting derivatives

1

Provide statistical and visual comparison of select MDE and Gaussian Splatting models

2

Model Overview – MDE: ZoeDepth



- Pioneering MDE model: combines Relative Depth Estimation and Metric Depth Estimation
- better generalization ability, high accuracy, zero-shot capability

Model Overview – MDE: DepthAnything v1

- **idea #1**: use data augmentation tools to develop more complex scenarios as training materials
- **idea #2:** use pre-trained encoders to ensure model inherits rich semantics
- overall better depth estimation in broader scenarios

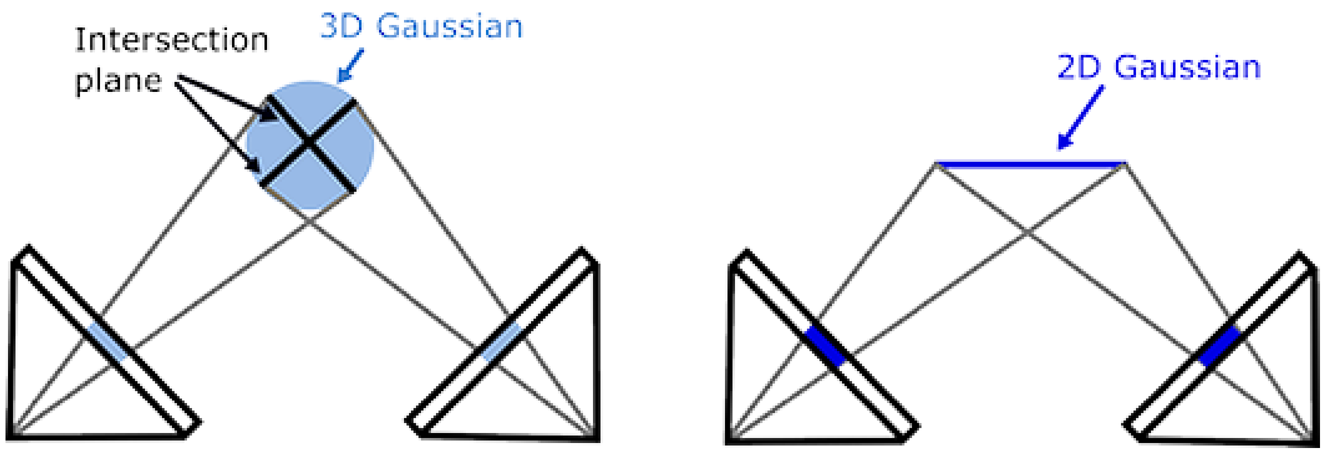
»

Model Overview – MDE: DepthAnything v2

- **improvement #1**: synthetic data instead of labeled real images
- **improvement #2:** improve capacity of teacher model
- **improvement #3:** generate large amount of pseudo-labeled real images for training to enhance generality
- Even more detailed depth estimation

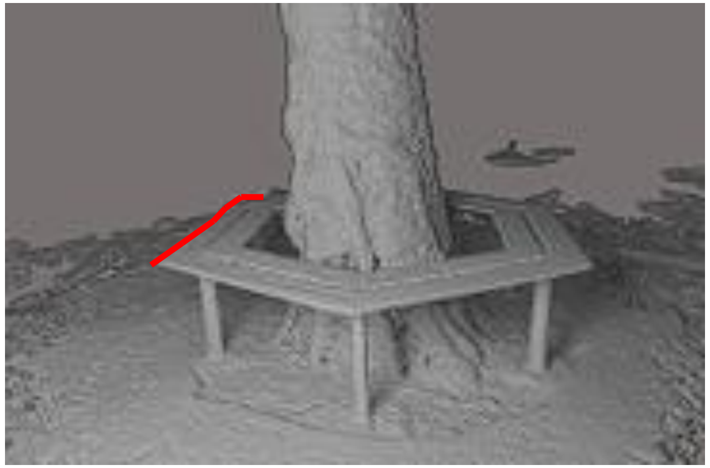
Model Overview – GS: 2DGS

- **main goal:** multi-view accurate surface representation
  - 3DGS: lack of accurate surface representation due to lack of explicit representation
- **main idea:** use 2D Gaussians instead of 3D Gaussians

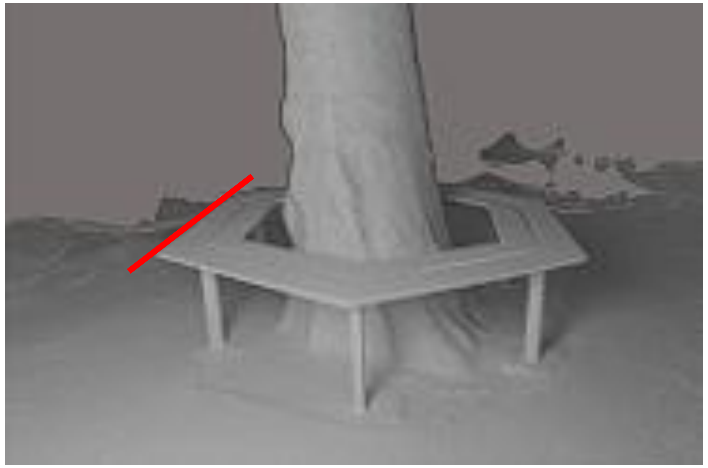


Model Overview – GS: 2DGS

Ground Truth



3DGS

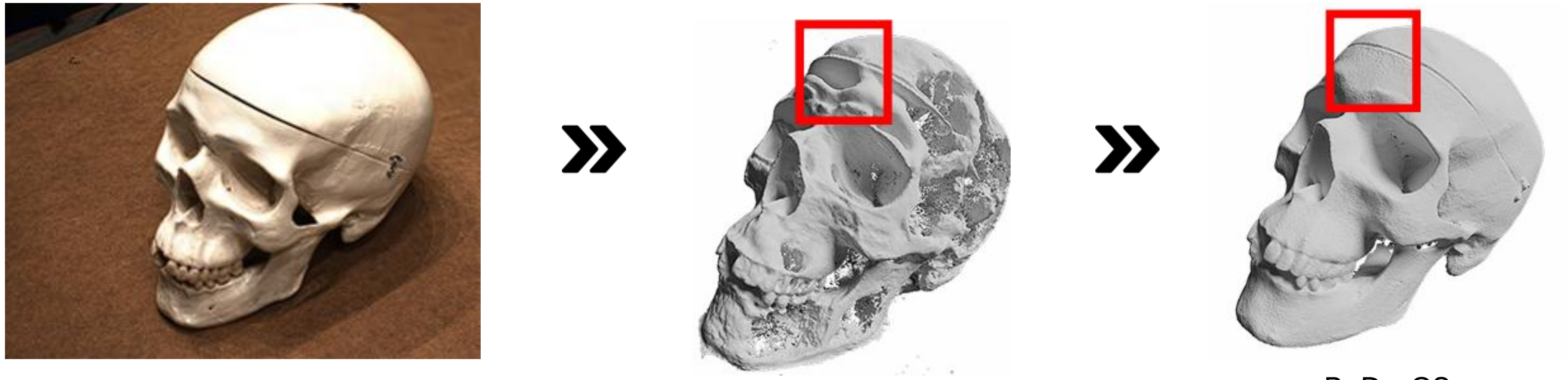


2DGS

notable problem: oversmoothing leads to loss of details

Model Overview – GS: RaDe-GS

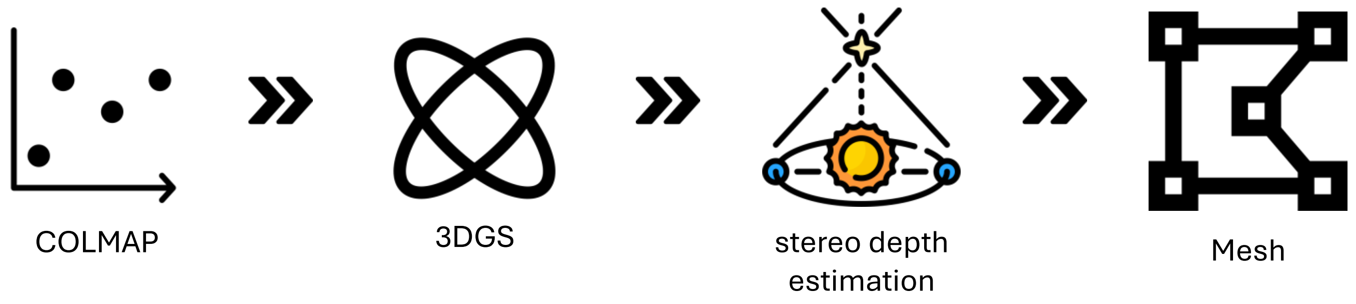
- **main goal**: improve geometric reconstruction details
- **main idea:** use rasterized approach to render depth map and surface normal maps of 3D Gaussians
- Closed-form solution in calculating intersection between light ray and splats
- Similar efficiency, more detailed results and clear geometry



3DGS

RaDe-GS

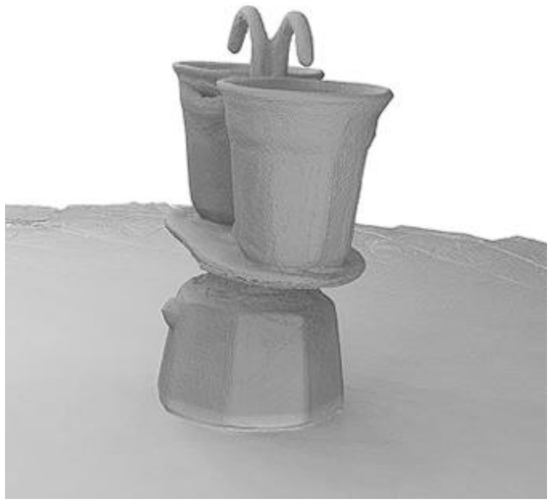
Model Overview – GS: GS2Mesh



- **main idea:** extract depth and geometry through pre-trained stereo-matching model
- After 3DGS step: generate stereo-calibrated images from the training image inputs and apply stereo matching algorithm for depth (DLNR model).
- Mesh extraction using TSDF and marching cubes algorithm

Model Overview – GS: GS2Mesh

»

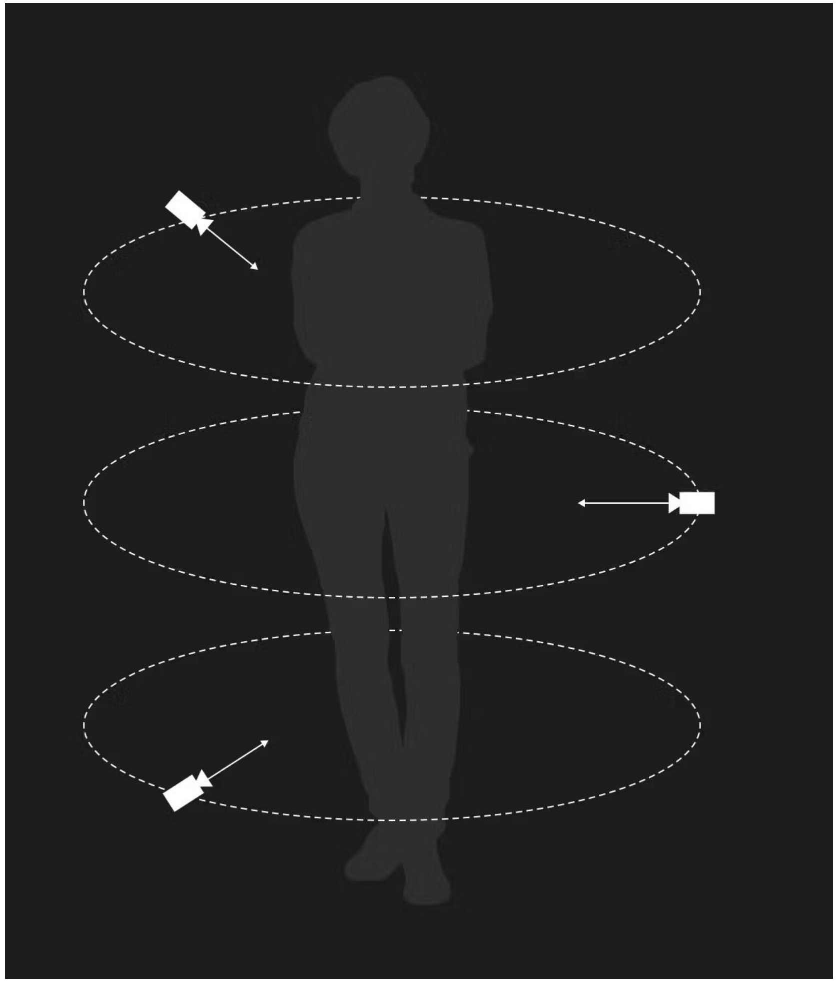


- **Limitations:** retains weakness of 3DGS (noisy results), TSDF is inefficient for larger inputs (exponential amount of points to check), inefficient rendering

Methodology – Dataset Creation

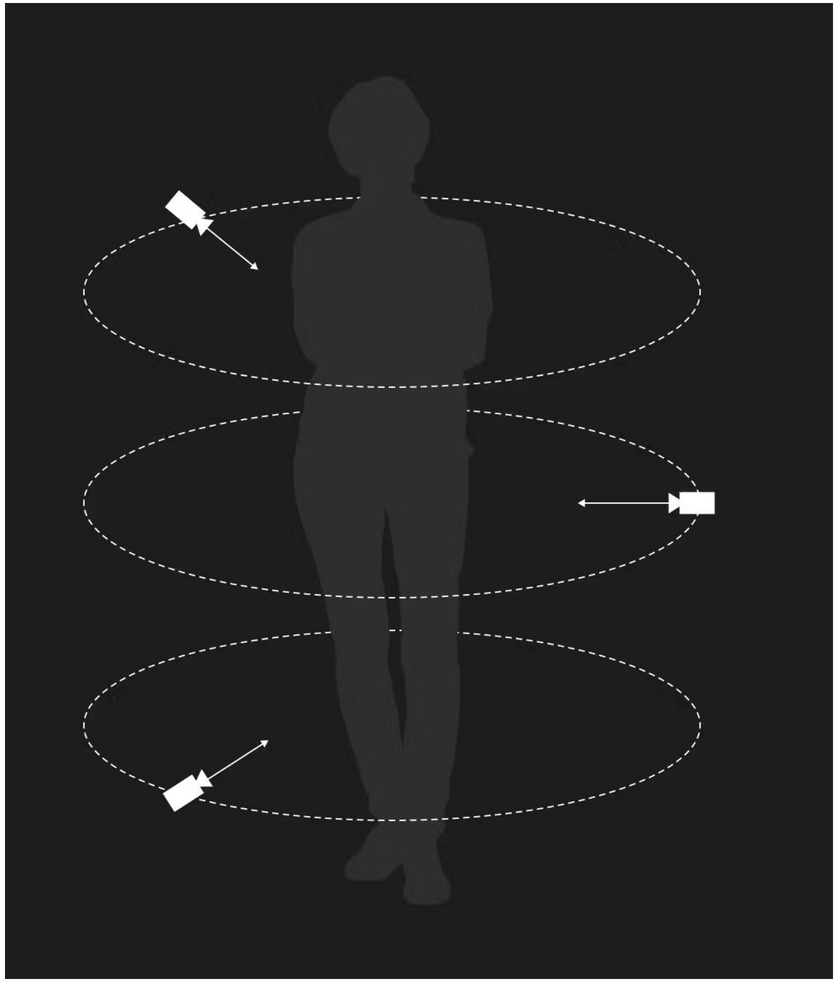
- **model**: chocolate bunny
- **preprocessing**: adding lighting and creating script for input creation

Methodology – Dataset Creation



- **Extraction of Synthetic Depth Data** (GT data) and **input dataset creation** from model
- Systematically capture model from multiple angles with Python script.
- **output:** set of 120 images (6 different height values, 20 images per height value) complete with camera poses and depth data

Methodology – Dataset Creation



code adaption:

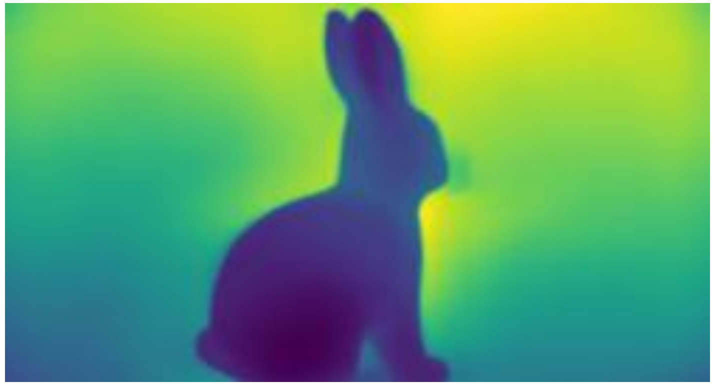
https://github.com/evelynsidarta/gaussian-splatting-thesis

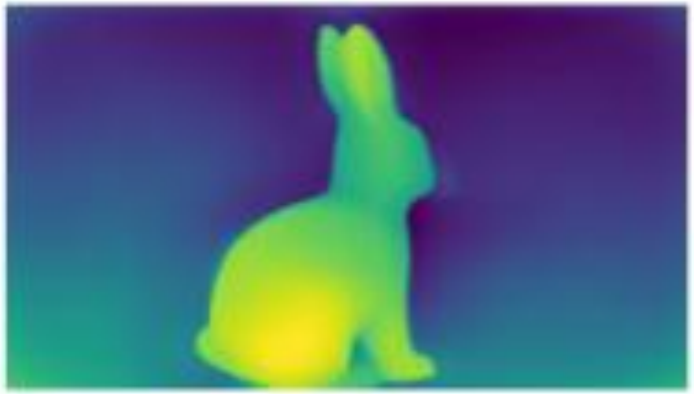
training:

10,000 iterations each, depth extraction with the help of 2DGS rendering script

Methodology – Postprocessing

- All depth map outputs converted into **inverted depth map** for better numerical stability: infinite background = 0





- **normalization** using min-max scaling: range [0, 1]

Methodology – Evaluation Metrics

- Absolute Relative Error (REL) $\downarrow$: absolute difference between output and GT compared to GT value
- Scale Invariant Logarithmic Loss (SiLog) $\downarrow$: minimize impact of scaling, relative error considered in logarithmic space
- Threshold Accuracy $\uparrow$ $(\delta_1, \delta_2, \delta_3)$: % of pixels that lie within a certain value of the GT:
  - δ_1: 25%
  - δ_2: 56.25%
  - δ_3: 95.31%

Methodology – Evaluation Metrics

- Root Mean Squared Error (RMSE) ↓ : absolute difference between GT and predicted values, highlights discrepancies
- Root Mean Squared Logarithmic Error (RMSLE) ↓ : logarithmic difference instead of absolute difference, higher discrepancies more proportionately scaled
- Logarithmic Error ↓
- Relative Square Error (RSE) ↓ : magnify error values

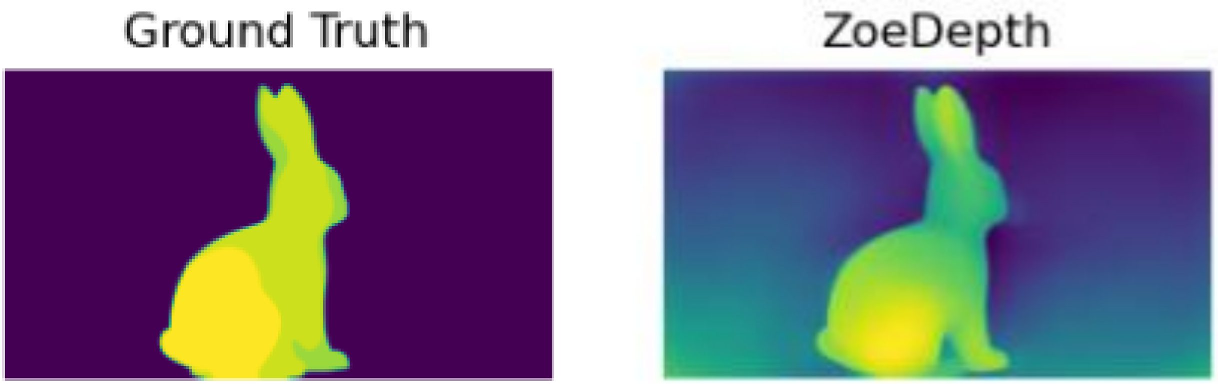
Evaluation – Statistical Analysis Results

Methods	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	REL $\downarrow$	RSE $\downarrow$	$\log_{10} \downarrow$	RMSE $\downarrow$	RMSLE $\downarrow$	SiLog $\downarrow$
3DGS	0.851	0.851	0.852	0.139	0.111	0.140	0.311	0.841	77.96
2DGS	0.855	0.855	0.855	0.131	0.107	0.139	0.311	0.838	77.50
RaDe	**0.856**	0.856	0.856	**0.128**	0.106	0.138	0.310	0.837	77.34
GS2M	0.853	0.853	0.853	0.155	0.124	0.138	0.309	0.833	77.04
DA1-s	0.807	0.903	**0.957**	0.143	**0.027**	**0.048**	0.097	0.252	24.71
DA1-b	0.828	**0.907**	0.948	0.154	0.032	0.049	**0.095**	**0.251**	**24.25**
DA1-l	0.725	0.831	0.898	0.278	0.061	0.081	0.118	0.359	33.88
DA2-s	0.561	0.688	0.788	0.502	0.094	0.140	0.127	0.503	43.32
DA2-b	0.583	0.710	0.811	0.443	0.079	0.129	0.127	0.472	41.95
DA2-l	0.613	0.745	0.850	0.354	0.058	0.110	0.124	0.419	38.47
ZoeD	0.472	0.620	0.742	0.679	0.163	0.179	0.159	0.581	47.51

Evaluation – Observations

- MDE performs better than GS models – simpler task
- ! DepthAnything v1 models perform better than v2 counterparts (see visual analysis later)
- GS models have high $\delta_1, \delta_2, \delta_3$ scores – estimation lie very close to the actual values
- RaDe-GS: best overall performance between the GS derivatives (also overall highest δ_1), consistent with findings of RaDe-GS author.
- However: most likely due to **no background bleeding**
- Overall performance very close since model is simple.

Evaluation – Visual Analysis



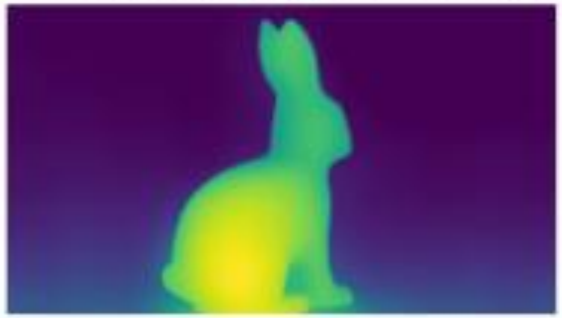
- Overall smooth shape
- Trying to create a bunny, not flat like the original chocolate bunny model
- Background bleeding – weak statistical performance due to this

Evaluation – Visual Analysis

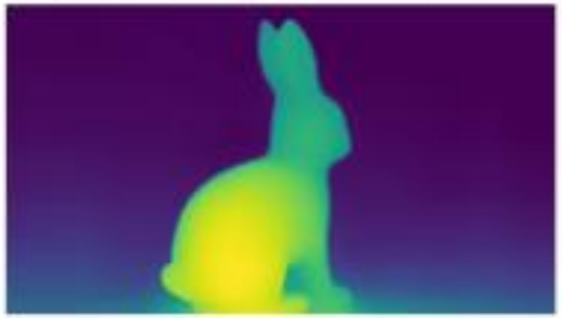
Ground Truth

- Large model more detailed (see ear splits)
- Small and base models do well on statistical analysis due to "flatter" look

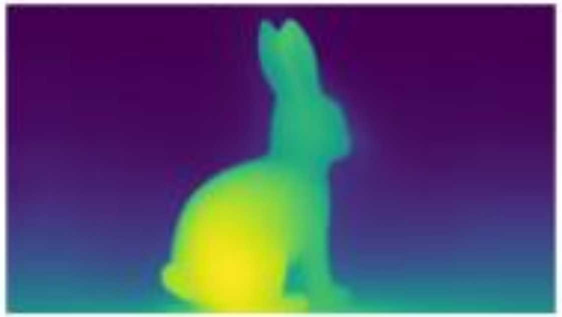
Depth Anything v1 Small



Depth Anything v1 Base



Depth Anything v1 Large

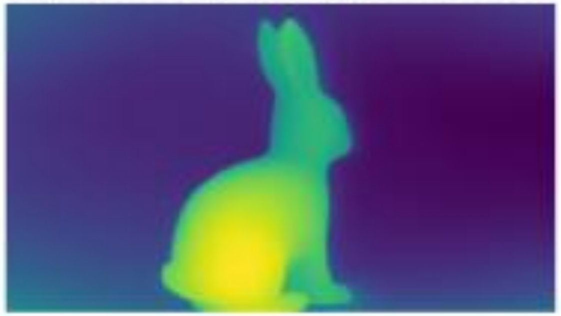


Evaluation – Visual Analysis

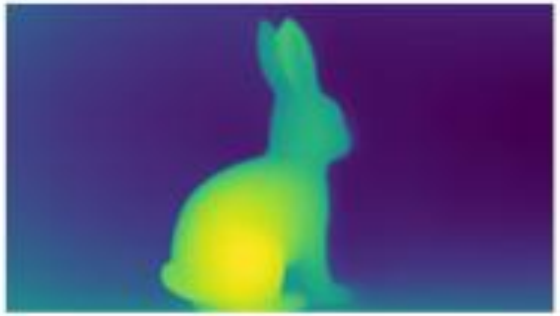
Ground Truth

- Overall a lot sharper than v1
- Ear splits and leg splits apparent even in small model
- However: oversmoothing

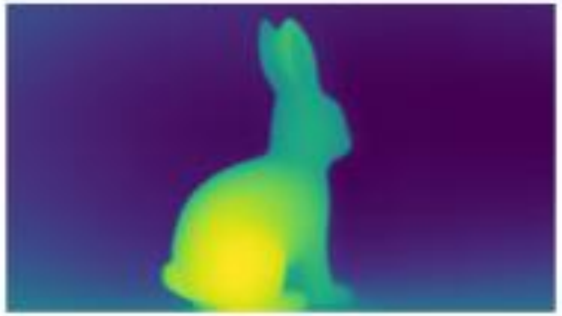
Depth Anything v2 Small



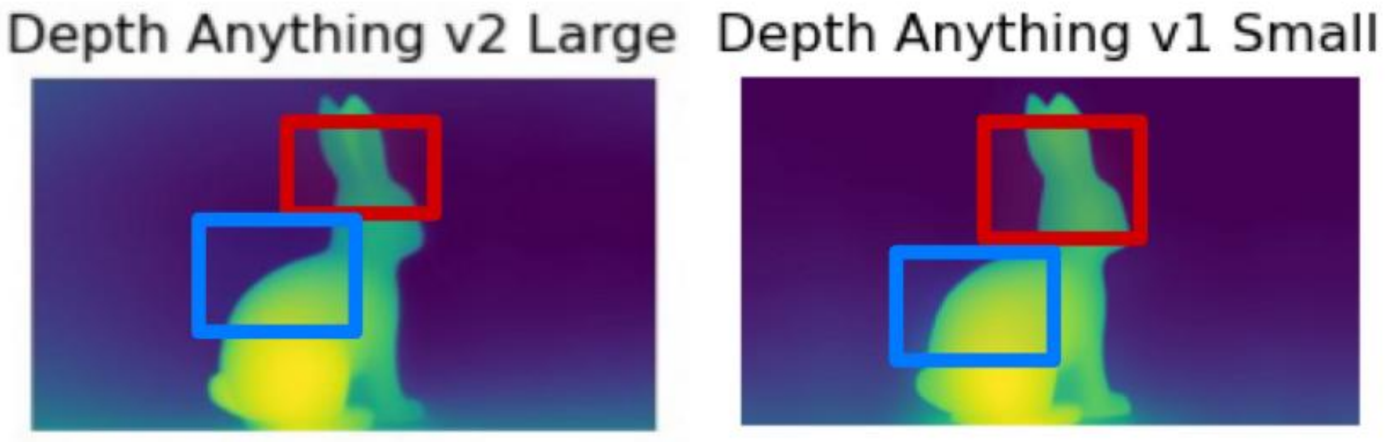
Depth Anything v2 Base



Depth Anything v2 Large

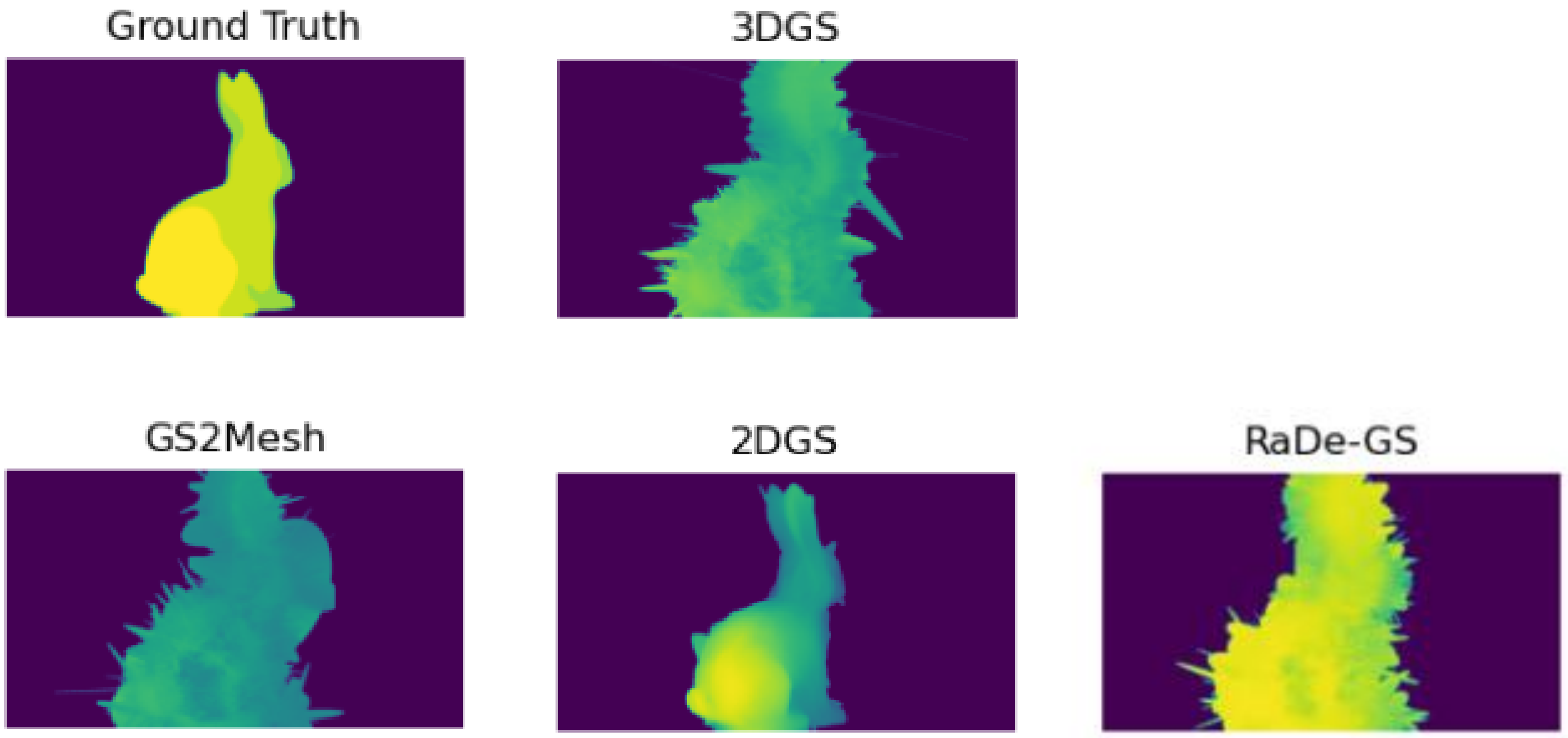


Evaluation – Visual Analysis

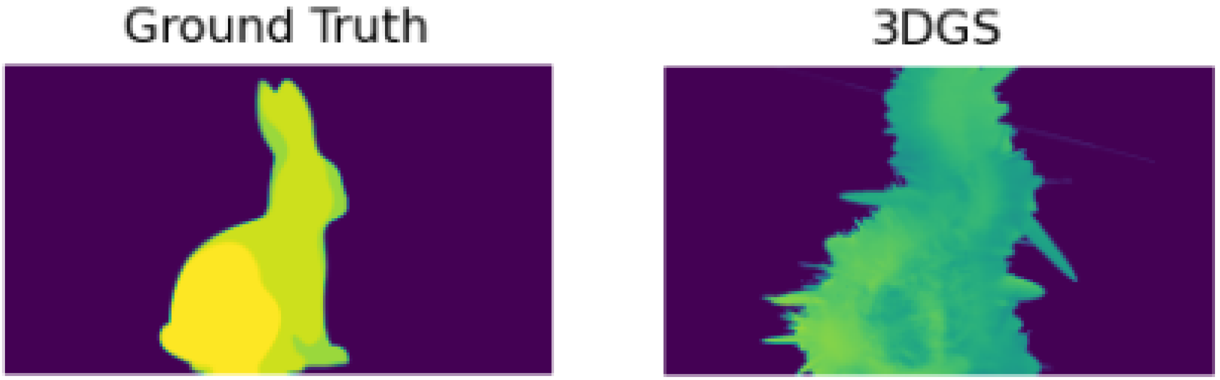


- v2 models are much sharper (see ear)
- v1 models are flatter in comparison – better statistical result since bunny is flat (see wider “yellow” area)

Evaluation – Visual Analysis

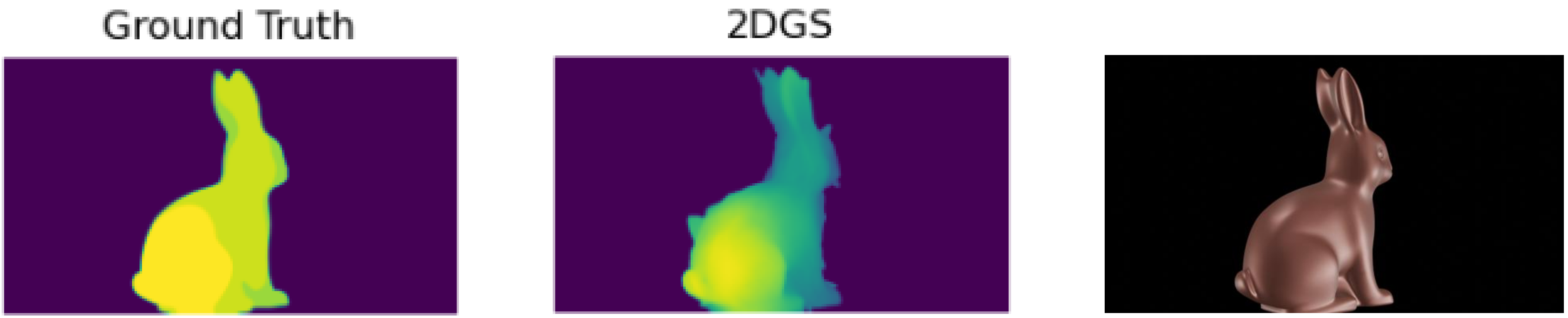


Evaluation – Visual Analysis



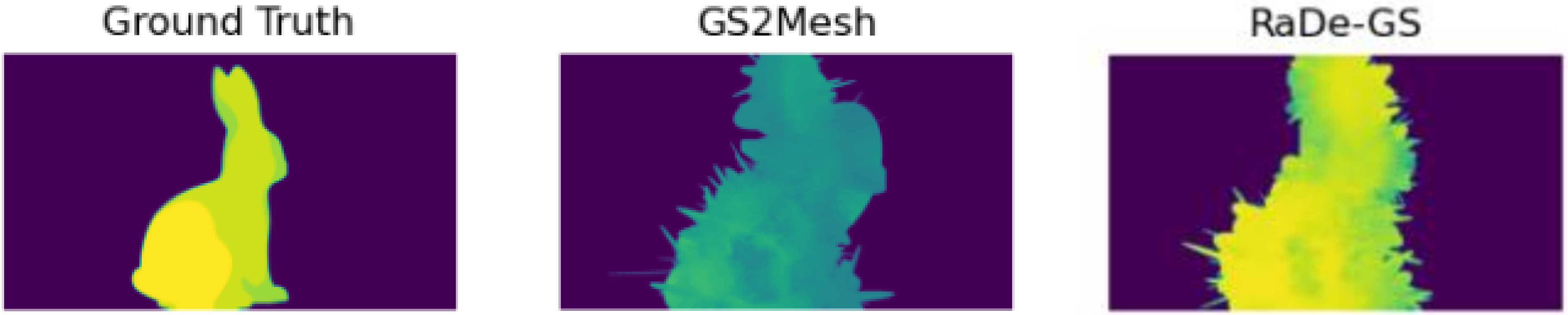
- GT image possess very basic form, very flat
- GS models much rougher: elliptical shape of the Gaussians
- 3DGS very rough, general shape clear, but details missing
- Orientation of the bunny unclear, some splats stick out at a weird angle

Evaluation – Visual Analysis



- 2DGS: overall best visuals
- Ear folds, legs, and tail distinguishable, orientation also obvious
- Smooth geometry not a problem since base GT model is not too detailed
- Oversmoothing – problem in statistical analysis when compared to RaDe-GS

Evaluation – Visual Analysis



- GS2Mesh – weird splats on some places – same as 3DGS
- Outline of bunny can be observed but details are entirely gone
- RaDe-GS – does not perform as well in visual analysis compared to in statistical analysis
- However: overall maintain flatter look – good score statistically

Limitations – Current State-of-Art Methods

- MDE models struggle from background bleeding, not used to dealing with unrealistic situations – poor statistical results even though performance is excellent visually
- GS models suffer from awkward shape of Gaussians: less smooth even though overall shape retained. Also more noisy due to density of Gaussians.
- Low opacity Gaussians still have depth values that needed to be considered for evaluation.

Suggestions – Further Improvements

- Using more variative models in the future
- Also incorporate more realistic data (not just synthetic models with purely silent backgrounds)
- Somehow consider transparency of the Gaussians when looking at the depth map