Advancements in Achieving Accurate Metric Depth from Monocular Depth Estimation

2025 Winter Seminar

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

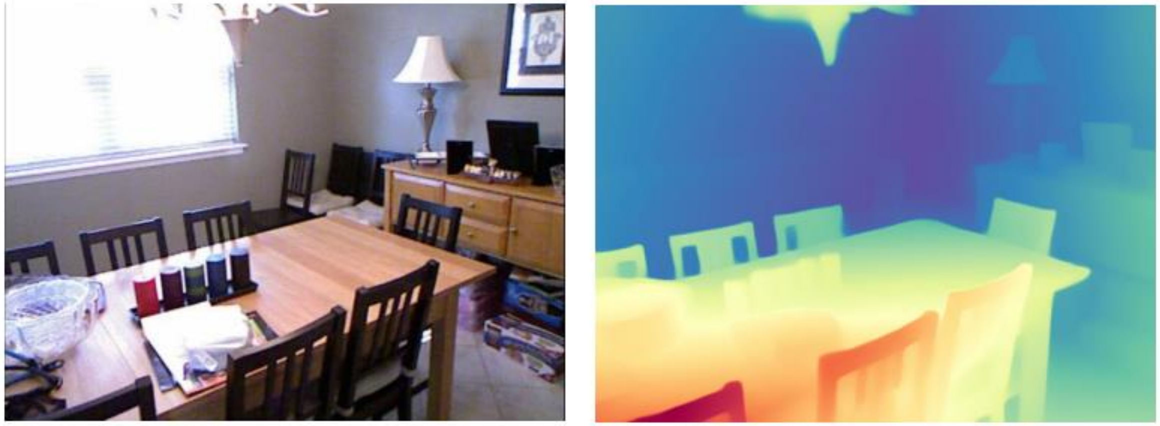
Presented By
Matti Zinke

Contents

- Background
- Depth Pro: Sharp Monocular Metric Depth in Less Than a Second (arXiv 2024)
- RSA: Resolving Scale Ambiguities in Monocular Depth Estimators through Language Descriptions (NeurIPS 2024)

Background

- What is Monocular Depth Estimation?
  - Monocular depth estimation aims to transform a photographic image into a depth map, i.e., evaluate a range value for every pixel
  - Task arises whenever the 3D structure of scene is needed, and no direct range or stereo measurements are available
    - Used for 3D reconstruction, autonomous driving etc.
  - Projecting 3D world to 2D image is geometrically ill-posed problem, solvable by prior knowledge of scene

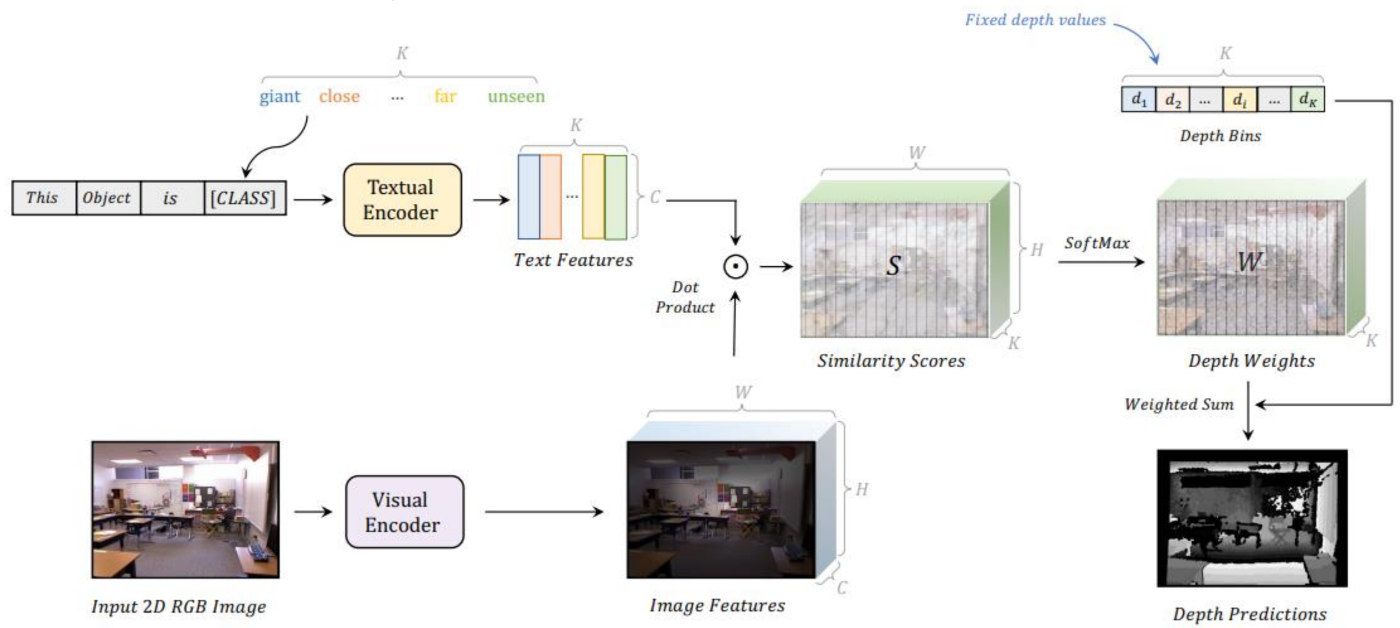


Background

- What is Metric Depth?
  - Shows accurate depth to any point given in an image
  - These days most models produce inverse relative depth, following the work of MiDaS
    - Foundation models are trained on many different datasets
    - Used to gain great zero-shot accuracy
      - Not every used dataset features necessary metadata for accurate metric depth
      - Unable to produce accurate metric depth, therefore normalize on scale from 0-255
    - Metric depth necessary for most downstream tasks though
  - Different ways to earn metric depth
    - Previous works focused on fine-tuning a MDE model with a certain dataset to earn metric data for this environment
    - Other works try to guess global scale and shift and apply them to all images

Background

- How do CLIP-based depth models work?
  - Most works function by using so called depth bins
    - Contain a set depth value for a certain type of scene
      - If the input class from the text prompt aligns with the given bin, it is set to that depth value
    - Either human-set or can also be learned on their own
    - Due to this still has many restrictions and does not perform as well as other MDE methods



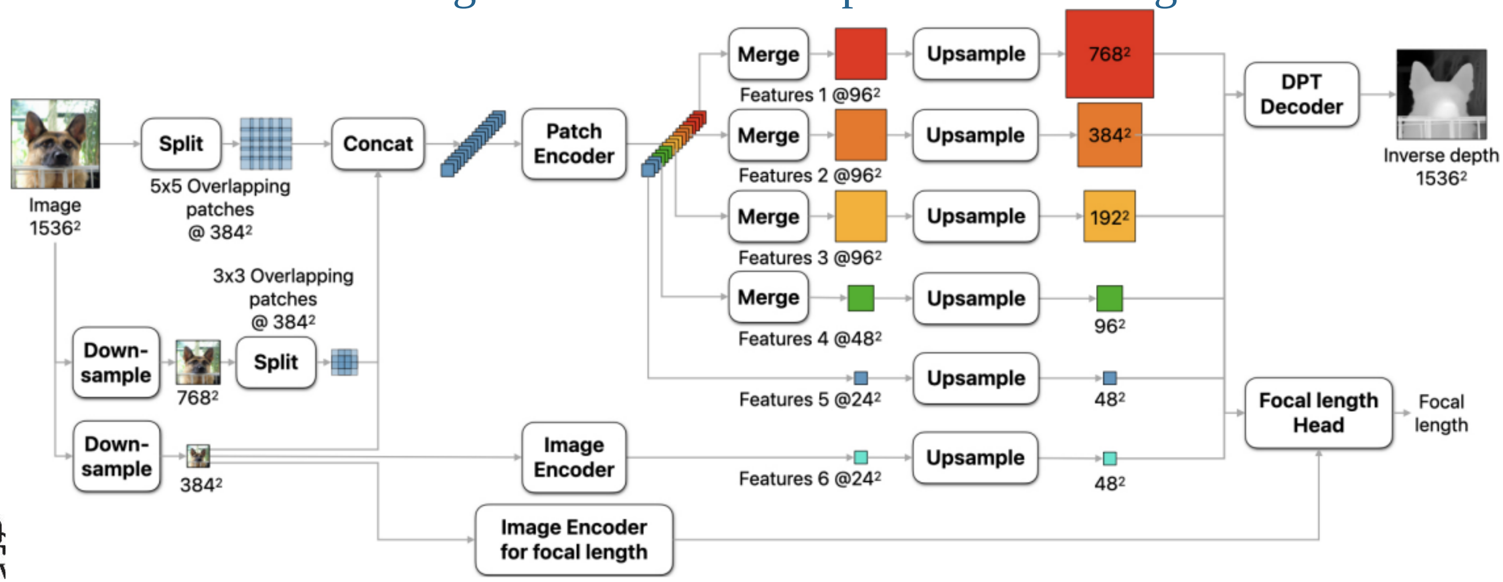
- Depth Pro: Sharp Monocular Metric Depth in Less Than a Second (arXiv 2024)

Introduction

- Foundation model for zero-shot metric monocular depth estimation
- Motivation
  - Depth estimators should work zero-shot on any image
    - Should not be restricted to certain domain
    - Should ideally produce metric depth maps for broad applicability
    - Metric depth should be accessible without meta data like camera intrinsics
  - Depth estimators should operate at high resolution and produce fine-grained depth maps
  - Should have low latency
    - High-resolution images should still be processable
- Fast metric prediction with absolute scale and high boundary tracing
  - Produces 2.25-megapixel depth map in 0.3 seconds

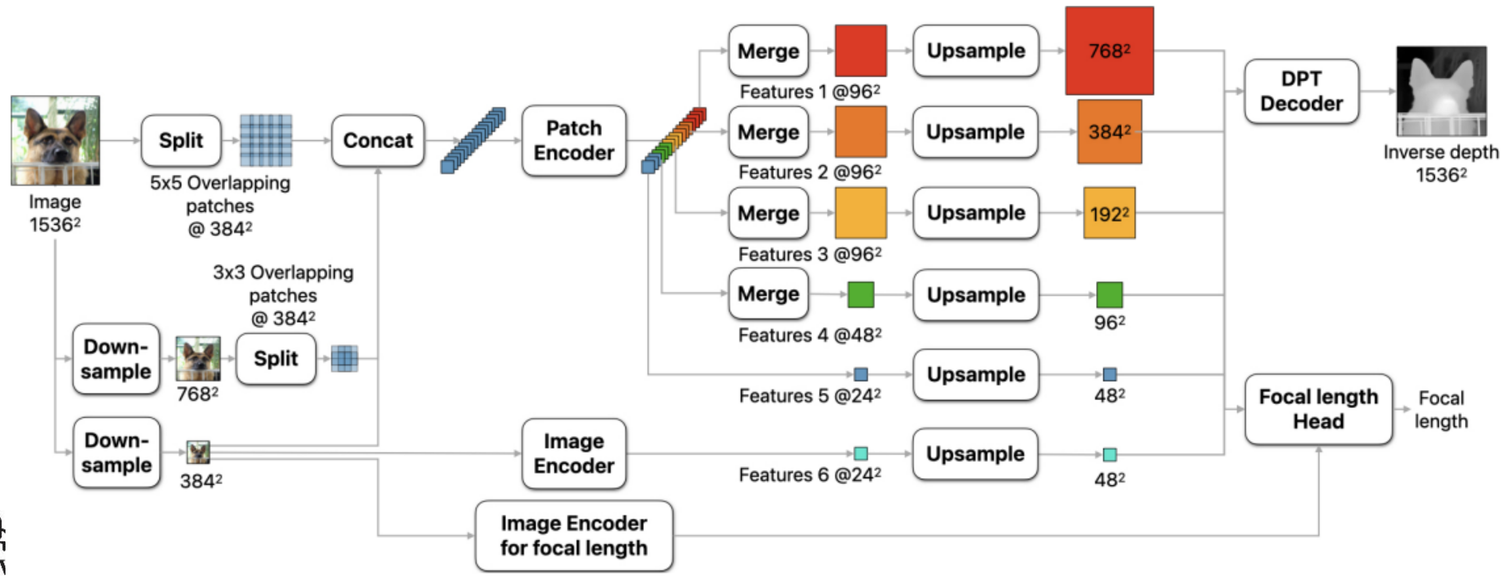
Method

- Network architecture
  - Key idea is to apply ViT on patches at multiple scales
    - Results get fused into single high-resolution dense prediction
  - Employs two ViT encoders for predicting depth
    - Patch encoder
    - Image encoder
  - Decoder resembles DPT
  - Separate ViT and focal length head encoder to predict focal length



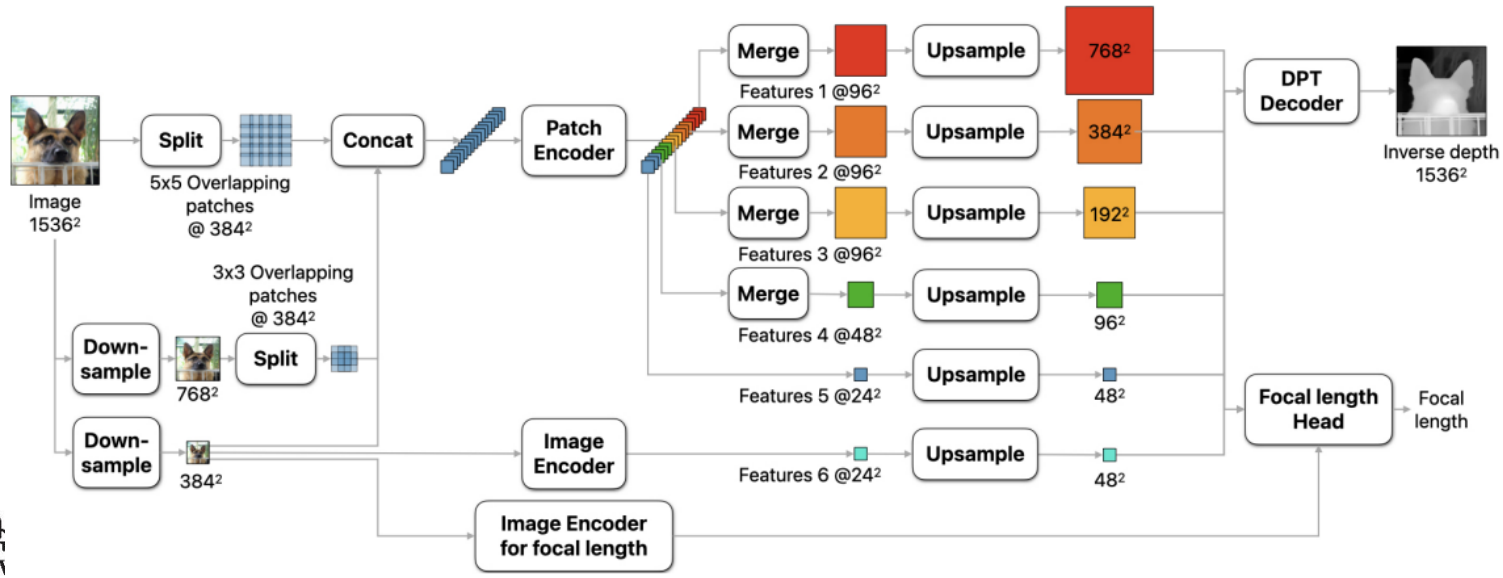
Method

- Depth prediction network
  - Patch encoder
    - Applied on patches which were extracted at multiple scales
    - Allows learning scale-invariant representations, as weights are shared across scales
  - Image encoder
    - Anchors patch predictions in global context
    - Applied to whole input image
    - Downsampled to base input resolution of chosen encoder backbone (here 384x384)



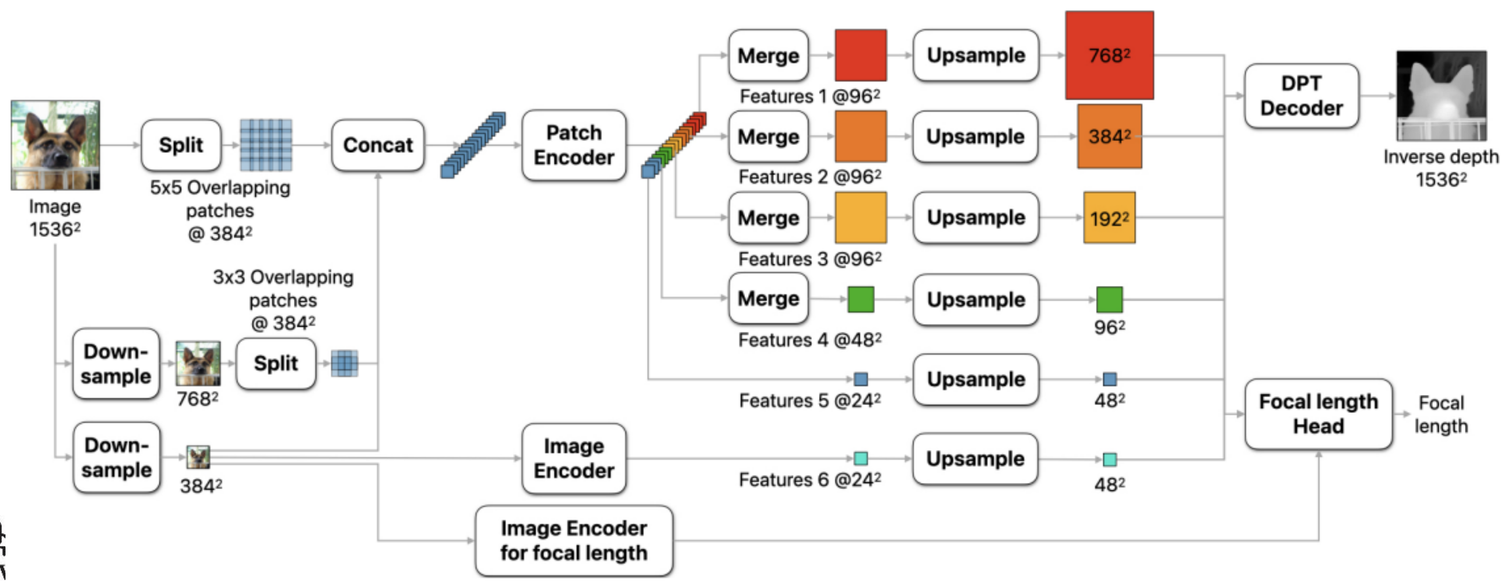
Method

- Depth prediction network
  - Whole network operates at 1536x1536 resolution
    - Guarantees sufficient receptive field and constant runtimes for any image
  - After downsampling to 1536x1536, image is split into patches of 384x384
    - Patches overlap to avoid seams
    - Yields 25 and 9 patches respectively
    - Patches extracted from all scales (35 in total) then concatenated along batch dimension and fed into patch encoder



Method

- Depth prediction network
  - Yields feature tensor at resolution 24x24 per input patch (features 3-6)
  - At finest scale, further intermediate features are extracted to capture finer details (1-2)
    - Yield another 50 feature patches
  - Feature patches then get merged into maps
    - Maps get fed into DPT decoder
  - Patch based approach also has advantage of allowing parallelization



Method

- Training objectives
  - For each input image I, network f, predicts canonical inverse depth image $C = f(I)$
    - Canonical inverse depth prioritizes areas close to camera over farther areas or whole scene
    - $\hat{C}$ describes ground-truth canonical inverse depth
  - Obtain dense metric depth map D_m by scaling horizontal field of view
    - Represented by focal length f_{px} and width w: $D_m = \frac{f_{px}}{wC}$
  - For training on metric datasets, the mean absolute error (L_{MAE}) per pixel i is used
    - $L_{MAE}(\hat{C}, C) = \frac{1}{N} \sum_1^N \left|\hat{C}_i - C_i\right|$
      - Pixels with error in top 20% per image get discarded for all real-world datasets
      - Chosen for robustness in handling potentially corrupted real-word ground truth

Method

- Training objectives
  - For non-metric datasets, normalize predictions and GT via mean absolute deviation from median
    - Further compute errors on first and second derivatives of inverse depth maps
    - Multi-scale derivative loss over M scales as
    - $L_{*,p,M}(C,\hat{C}) = \frac{1}{M}\sum_j^M \frac{1}{N_j}\sum_i^{N_j}\left|\nabla_* C_i^j - \nabla_* \hat{C}_i^j\right|^p$
      - ∇_* indicates spatial derivative operator *, such as Laplace (L) or Scharr (S) and p the error norm
      - Scales j computed by blurring and downsampling inverse depth maps

Method

- Training curriculum
  - Based on three observations
    - Training on large mix of real-world data and synthetic datasets improves generalization
    - Synthetic datasets provide pixel-accurate, high-quality ground truths
      - Real-world datasets often contain missing areas or mismatched depth
    - Predictions get sharper over course of training
  - Two-stage training curriculum follows these observations
    - In first stage aim to learn robust features that allow network to generalize across domains
      - Train on mix of all labeled data
      - Minimize L_{MAE} on metric datasets and its normalized version of non-metric ones
      - To steer network towards sharp boundaries, supervise gradients of predictions
      - Done naively can hinder and optimization
        - ✓Apply scale-and-shift invariant loss on gradients of only the synthetic data

Method

- Training curriculum
  - Two-stage training curriculum follows these observations
    - Second stage designed to sharpen boundaries and reveal fine details in depth maps
      - To minimize effect of inaccurate GT, train in this stage only on synthetic data
      - Opposed to real data, synthetic data provides high-quality pixel-accurate GTs
      - Minimize L_{MAE} again and supplement it with selection of first- and second-order derivates
- Focal length estimation
  - Predict horizontal angular field-of-view from separate ViT image encoder
    - Small convolutional head ingests frozen features from depth estimation network and task-specific features
    - Uses L_2 training loss
    - Gets trained after depth estimation training
  - Focal length training is separated
    - Has benefits, as avoids necessity of balancing depth and focal length training objectives
    - Also allows training of focal length head on focal length supervision datasets

Method

- Evaluation metrics for sharp boundaries
  - Common MDE benchmarks rarely take boundary sharpness into account
  - Propose set of metrics specifically for the evaluation of depth boundaries
  - Key observation: can leverage existing high-quality annotations for matting, saliency or segmentation as GT for depth boundaries
    - Treat annotations for these tasks as binary maps
    - Define foreground/background relationship between object and environment
      - Only consider pixel around edges in binary maps
    - Use pairwise depth ratio of neighboring pixels to define foreground/background relationship
    - Occluding contour c_d derived from depth map d as
      - $c_d(i,j) = \left[\frac{d(j)}{d(i)} > \left(1 + \frac{t}{100}\right)\right]$
        - ✓ i, j are locations of two neighboring pixels
        - ✓ Indicates presence of occluding contour between pixels i and j if depth differs more than t%

Method

- Evaluation metrics for sharp boundaries
  - Key observation: can leverage existing high-quality annotations for matting, saliency or segmentation as GT for depth boundaries
    - Can now compute precision P and recall R for all neighboring pixel
      - $P(t)=\frac{\sum_{i,j\in N(i)} c_d(i,j)\land c_{\hat{d}}(i,j)}{\sum_{i,j\in N(i)} c_d(i,j)}$ and $R(t)=\frac{\sum_{i,j\in N(i)} c_d(i,j)\land c_{\hat{d}}(i,j)}{\sum_{i,j\in N(i)} c_{\hat{d}}(i,j)}$
      - Precision and recall are scale-invariant, for experiment report F1 score
      - Performed weighted averaging of F1 values with thresholds from $t_{min}=5$ and $t_{max}=25$
  - Does not require any manual edge annotation
    - Can use pixelwise ground truth available in synthetic datasets
  - Given binary mask b over image, define presence of c_b between pixel i,j as
    - $c_b(i,j)=b(i)\land\neg b(j)$
    - Can compute recall by replacing occluding contours from depth maps with those from binary maps

Experiment

- Quantitative results
  - Zero-shot metric depth accuracy (δ_1, higher is better)

Method	Booster	ETH3D	Middlebury	NuScenes	Sintel	Sun-RGBD	Avg. Rank↓
DepthAnything (Yang et al., 2024a)	52.3	9.3	39.3	35.4	6.9	85.0	5.7
DepthAnything v2 (Yang et al., 2024b)	59.5	36.3	37.2	17.7	5.9	72.4	5.8
Metric3D (Yin et al., 2023)	4.7	34.2	13.6	64.4	17.3	16.9	5.8
Metric3D v2 (Hu et al., 2024)	39.4	87.7	29.9	82.6	38.3	75.6	3.7
PatchFusion (Li et al., 2024a)	22.6	51.8	49.9	20.4	14.0	53.6	5.2
UniDepth (Piccinelli et al., 2024)	27.6	25.3	31.9	**83.6**	16.5	**95.8**	4.2
ZeroDepth (Guizilini et al., 2023)	OOM	OOM	46.5	64.3	12.9	OOM	4.6
ZoeDepth (Bhat et al., 2023)	21.6	34.2	53.8	28.1	7.8	85.7	5.3
Depth Pro (Ours)	46.6	41.5	**60.5**	49.1	**40.0**	89.0	**2.5**

  - Zero-shot boundary accuracy (F1 score and recall, higher is better)

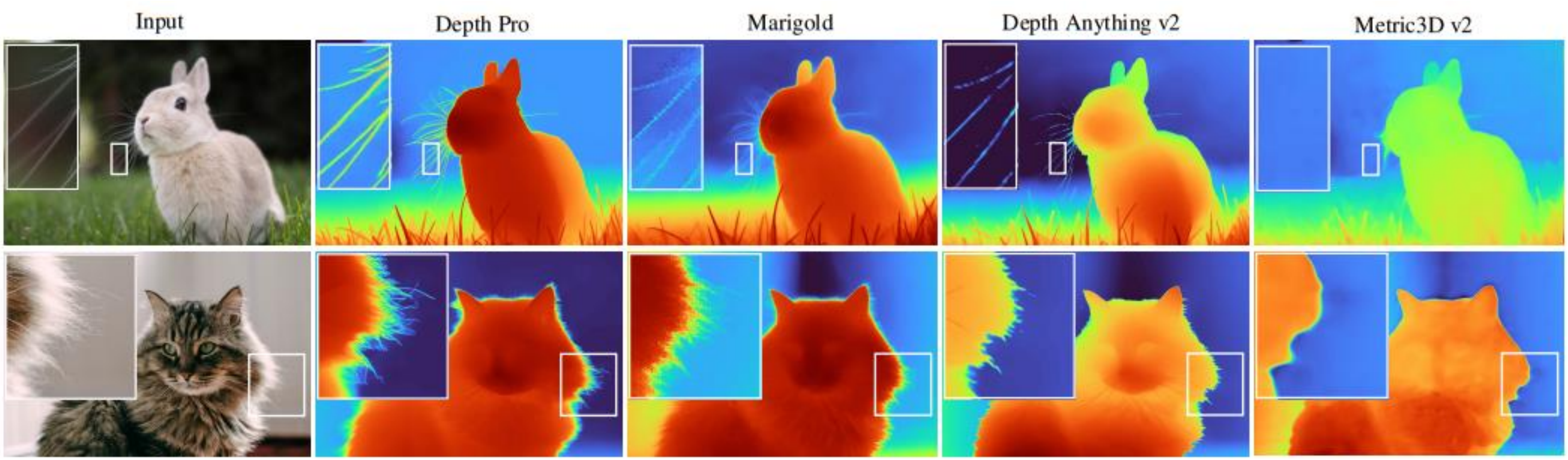
	Method	Sintel F1↑	Spring F1↑	iBims F1↑	AM R↑	P3M R↑	DIS R↑
Absolute	DPT (Ranftl et al., 2021)	0.181	0.029	0.113	0.055	0.075	0.018
	Metric3D (Yin et al., 2023)	0.037	0.000	0.055	0.003	0.003	0.001
	Metric3D v2 (Hu et al., 2024)	0.321	0.024	0.096	0.024	0.013	0.006
	ZoeDepth (Bhat et al., 2023)	0.027	0.001	0.035	0.008	0.004	0.002
	PatchFusion (Li et al., 2024a)	0.312	0.032	0.134	0.061	0.109	0.068
	UniDepth (Piccinelli et al., 2024)	0.316	0.000	0.039	0.001	0.003	0.000
Rel.	DepthAnything (Yang et al., 2024a)	0.261	0.045	0.127	0.058	0.094	0.023
	DepthAnything v2 (Yang et al., 2024b)	0.228	0.056	0.111	0.107	0.131	0.056
	Marigold (Ke et al., 2024)	0.068	0.032	0.149	0.064	0.101	0.049
	Depth Pro (Ours)	**0.409**	**0.079**	**0.176**	**0.173**	**0.168**	**0.077**

Experiment

- Quantitative results
  - Comparison on focal length estimation ($\delta_{25\%}$, $\delta_{50\%}$, higher is better)

	DDDP		FiveK		PPR10K		RAISE		SPAQ		ZOOM	
	$\delta_{25\%}$	$\delta_{50\%}$	$\delta_{25\%}$	$\delta_{50\%}$	$\delta_{25\%}$	$\delta_{50\%}$	$\delta_{25\%}$	$\delta_{50\%}$	$\delta_{25\%}$	$\delta_{50\%}$	$\delta_{25\%}$	$\delta_{50\%}$
UniDepth (Piccinelli et al., 2024)	6.8	40.3	24.8	56.2	13.8	44.2	35.4	74.8	44.2	77.4	20.4	45.4
SPEC (Kocabas et al., 2021)	14.6	46.3	30.2	56.6	34.6	67.0	49.2	78.6	50.0	82.2	23.2	43.6
im2pcl (Baradad & Torralba, 2020)	7.3	29.6	28.0	60.0	24.2	61.4	51.8	75.2	26.6	55.0	22.4	42.8
Depth Pro (Ours)	**66.9**	**85.8**	**74.2**	**92.4**	**64.6**	**88.8**	**84.2**	**96.4**	**68.4**	**85.2**	**69.8**	**91.6**

- Qualitative results



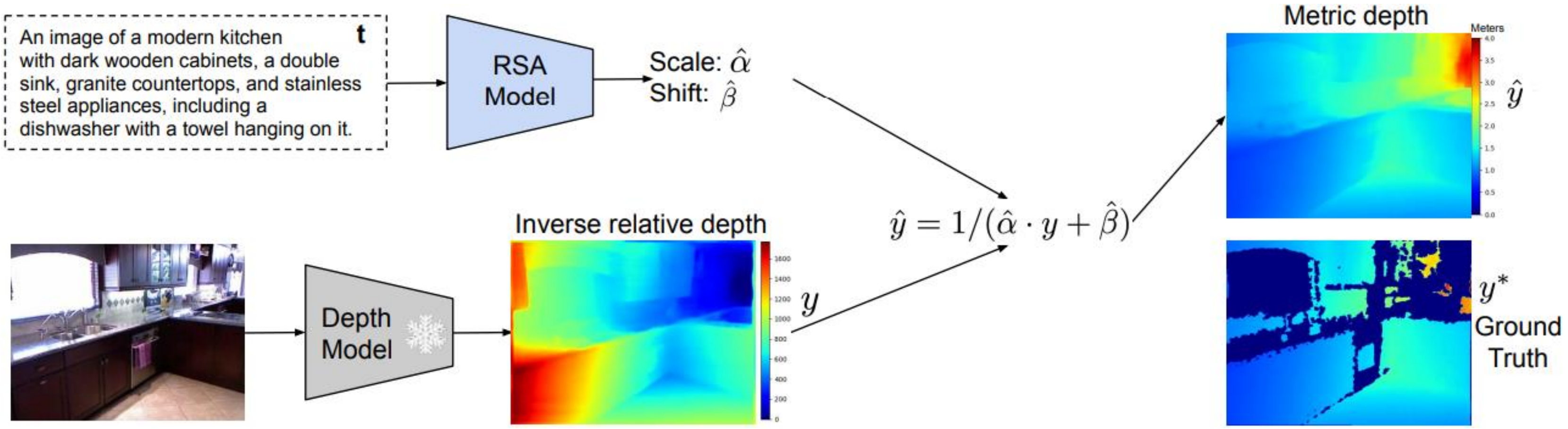
- RSA: Resolving Scale Ambiguities in Monocular Depth Estimators through Language Descriptions (NeurIPS 2024)

Introduction

- First method for metric-scale monocular depth estimation with language
- Recovers metric-scaled depth maps through linear transformation
  - Based on observation, that certain objects (cars, trees, street signs) are typically found or associated with certain types of scenes (e.g. outdoor)
- Takes as input a text caption describing objects present in a scene and outputs parameters of linear transformation
  - Parameters can then be applied globally to a relative depth map to yield a metric-scaled prediction
- Model can be trained on multiple datasets to be used in zero-shot settings

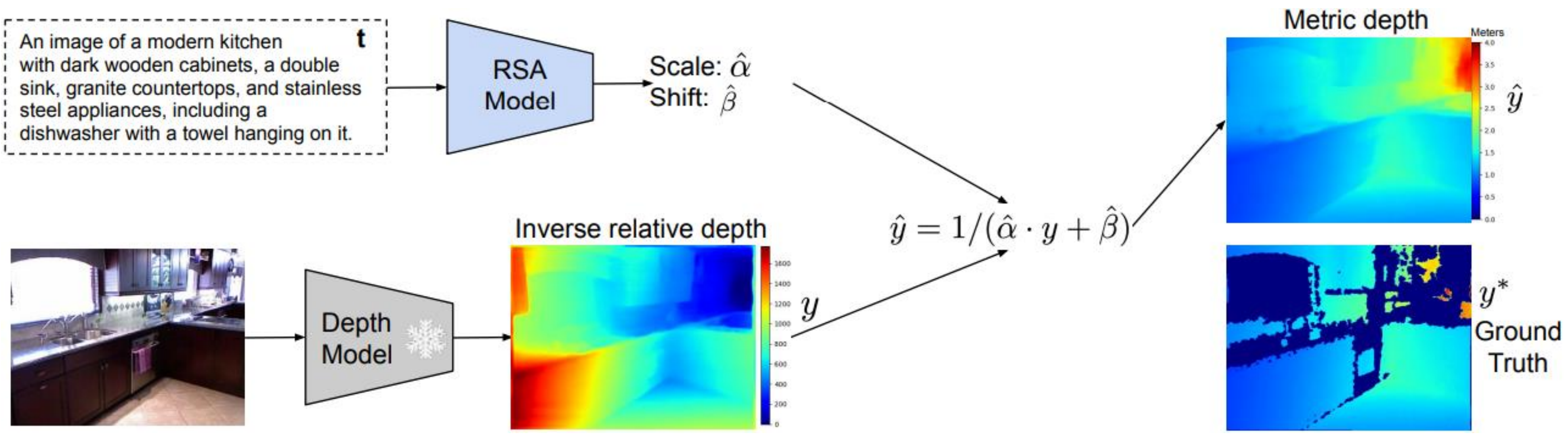
Method

- Consider dataset $D = \{I^{(n)}, t^{(n)}, y^{*(n)}\}_{n=1}^{N}$ with N samples synchronized RGB images
  - I denotes an image, y^* the ground truth depth map and t a text description of the image
  - Assume access to pretrained monocular depth estimation model h_θ to learn parameters to predict transformation between relative and metric depth
  - Given an image, a MDE predicts inverse relative depth $y \coloneqq h_\theta(\cdot)$
    - Consider global linear transformation through use of language description pretraining to recover metric-scale



Method

- Given a text description t, RSA predicts pair of scalars denoting scale and shift parameters of transformation, described as
  - $(\hat{\alpha}, \hat{\beta}) = g_{\psi}(t)$
  - $\hat{\alpha}$ describes the guessed scale and $\hat{\beta}$ the guessed shift
- Metric depth can now be obtained by
  - $\hat{y} = 1/(\hat{\alpha} \cdot y + \hat{\beta})$



Method

- RSA model
  - Employs pretrained CLIP text encoder as feature extractor to infer scale and shift
    - Text encoder frozen within RSA
  - First encodes text descriptions into text embeddings and then feed into 5-layer shared MLP to project them into k = 256 hidden dimensions
    - Followed by two separate sets of 5-layers
      - One serves as output head $\psi_{\hat{\alpha}}$ for scale parameter $\hat{\alpha}$
      - Other serve as output head $\psi_{\hat{\beta}}$ for shift parameter $\hat{\beta}$
  - Optimizing RSA involves minimizing supervised loss with respect to ψ
    - $\psi^* = \arg\min_{\psi} \sum_{n=1}^{N} \frac{1}{|M^{(n)}|} \sum_{x\in\Omega} M^{(n)}(x) |\hat{y}^{(n)}(x) - y^{*(n)}(x)|$
      - $\hat{y}^{(n)} = 1/(\hat{\alpha}^{(n)} \cdot y^{(n)} + \hat{\beta}^{(n)})$ is predicted metric-scale depth aligned from relative depth
      - $x \in \Omega$ denotes an image coordinate
      - M denotes binary mask indicating valid coordinates in ground truth depth

Method

- Text prompt design
  - Require text descriptions to be paired with each image
  - Create these descriptions themselves by using different models
    - First considered structured text, using a certain template
      - Use MaskDINO to extract significant objects and background in the image
      - For an input image, segmentation model returns set of B object and background instances $\{n^{(i)}, c^{(i)}\}_{i=1}^{B}$, where $c^{(i)}$ denotes class of object and $n^{(i)}$ the count of this object
      - Then instances structured to caption: "An image with $n^{(1)}\ c^{(1)}, n^{(2)}, c^{(2)}, \ldots n^{(B)}, c^{(B)}$."
      - Shuffle order of instances to produce five different of these prompts
    - Then consider natural text, which does not follow a certain template
      - Use two visual question-answering models for this
      - Each model provides 5 prompts each
    - During training, in each iteration a random of these in total 15 prompts gets chosen

Experiments

- Quantitative Results

Models	Scaling	Dataset	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$	Abs Rel $\downarrow$	$\log_{10} \downarrow$	RMSE $\downarrow$
ZoeDepth	Image	NYUv2	0.951	0.994	0.999	0.077	0.033	0.282
DistDepth	DA	NYUv2	0.706	0.934	-	0.289	-	1.077
DistDepth	DA,Median	NYUv2	0.791	0.942	0.985	0.158	-	0.548
ZeroDepth	DA	-	0.901	0.961	-	0.100	-	0.380
ZeroDepth	DA,Median	-	0.926	0.986	-	0.081	-	0.338
DPT	Median	NYUv2	0.736	0.919	0.981	0.181	0.073	0.912
	Linear Fit	NYUv2	0.926	0.991	0.999	0.094	0.040	0.332
	Global	NYUv2	0.904	0.988	**0.998**	0.109	0.045	0.357
	Image	NYUv2	0.914	**0.990**	**0.998**	**0.097**	**0.042**	0.350
	Image	NYUv2,KITTI	0.911	0.989	**0.998**	0.098	0.043	0.355
	Image	NYUv2,KITTI,VOID	0.903	0.985	0.997	0.100	0.045	0.367
	RSA (Ours)	NYUv2	**0.916**	**0.990**	**0.998**	**0.097**	**0.042**	**0.347**
	RSA (Ours)	NYUv2,KITTI	0.913	0.988	**0.998**	0.099	**0.042**	0.352
	RSA (Ours)	NYUv2,KITTI,VOID	0.912	0.989	**0.998**	0.099	0.043	0.355
MiDas	Median	NYUv2	0.449	0.694	0.850	0.411	0.151	2.010
	Linear Fit	NYUv2	0.780	0.970	0.995	0.151	0.069	0.433
	Global	NYUv2	0.689	0.949	0.992	0.183	0.078	0.600
	Image	NYUv2	0.729	0.958	**0.994**	0.175	0.072	0.563
	Image	NYUv2,KITTI	0.724	0.952	0.992	0.173	0.074	0.579
	Image	NYUv2,KITTI,VOID	0.712	0.948	0.988	0.181	0.075	0.583
	RSA (Ours)	NYUv2	0.731	0.955	0.993	0.171	0.072	0.569
	RSA (Ours)	NYUv2,KITTI	**0.737**	**0.959**	0.993	**0.168**	**0.071**	**0.561**
	RSA (Ours)	NYUv2,KITTI,VOID	0.709	0.944	0.989	0.173	0.076	0.580
DepthAnything	Median	NYUv2	0.480	0.734	0.886	0.353	0.135	1.743
	Linear Fit	NYUv2	0.965	0.993	0.997	0.058	0.025	0.232
	Global	NYUv2	0.630	0.926	0.987	0.199	0.087	0.646
	Image	NYUv2	0.749	0.965	**0.997**	0.169	0.068	0.517
	Image	NYUv2,KITTI	0.710	0.947	0.992	0.181	0.075	0.574
	Image	NYUv2,KITTI,VOID	0.702	0.943	0.990	0.178	0.078	0.583
	RSA (Ours)	NYUv2	0.775	**0.975**	**0.997**	**0.147**	**0.065**	**0.484**
	RSA (Ours)	NYUv2,KITTI	**0.776**	0.974	0.996	0.148	**0.065**	0.498
	RSA (Ours)	NYUv2,KITTI,VOID	0.752	0.964	0.992	0.156	0.071	0.528

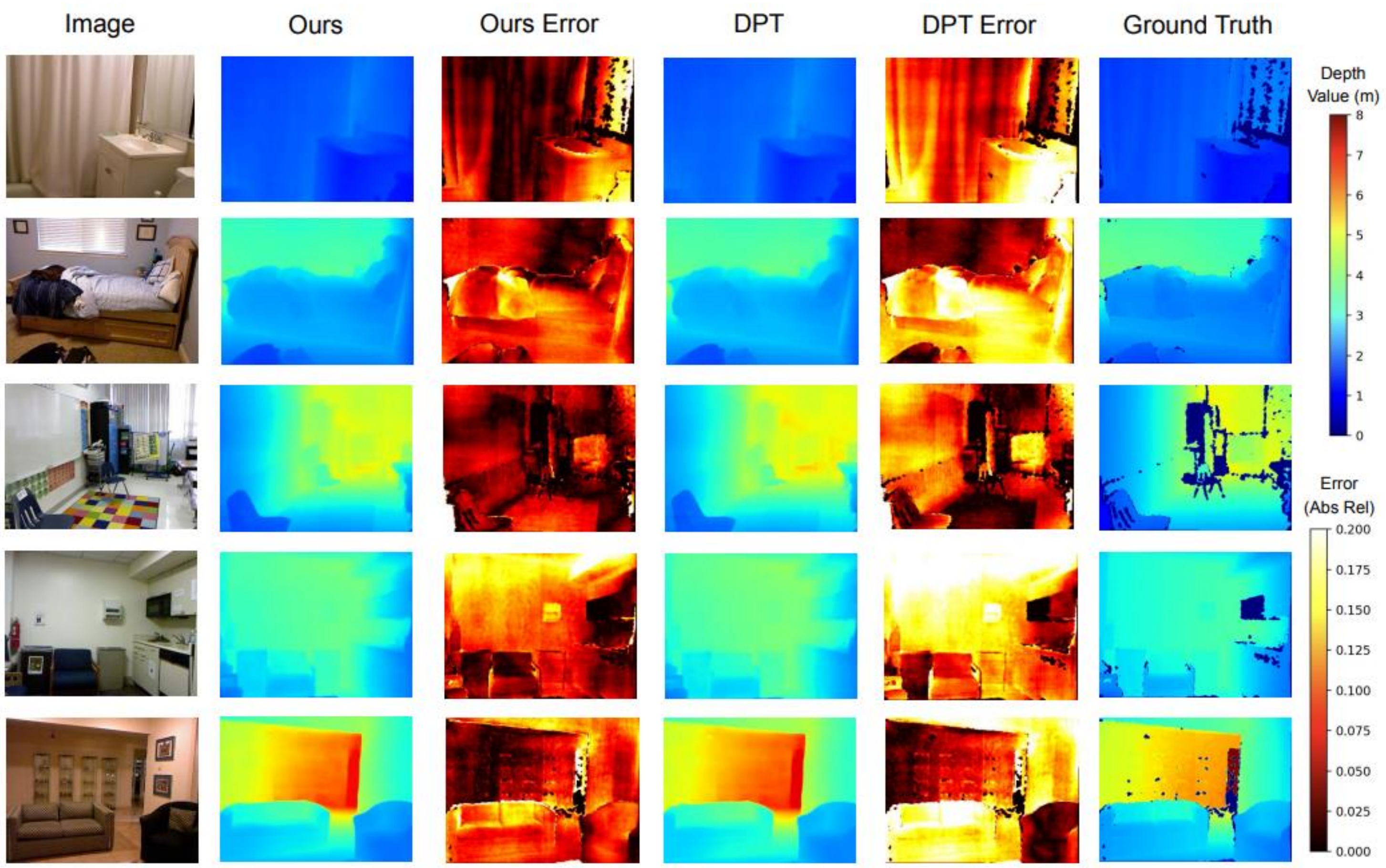
Experiments

- Quantitative Results

Models	Scaling	Dataset	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$	Abs Rel $\downarrow$	$\log_{10} \downarrow$	RMSE $\downarrow$
Adabins	-	NYUv2	0.771	0.944	0.983	0.159	0.068	0.476
DepthFormer	-	NYUv2	0.815	0.970	0.993	0.137	0.059	0.408
ZoeDepth-X	Image	NYUv2	0.857	-	-	0.124	-	0.363
ZoeDepth-M12	Image	NYUv2	0.864	-	-	0.119	-	0.346
ZoeDepth-M12	Image	NYUv2, KITTI	0.856	-	-	0.123	-	0.356
DPT	Linear Fit	SUN-RGBD	0.812	0.967	0.993	0.139	0.059	0.412
	Global	NYUv2	0.773	0.945	0.984	0.154	0.071	0.482
	Image	NYUv2,KITTI	0.778	**0.953**	0.984	0.153	0.068	0.478
	RSA (Ours)	NYUv2,KITTI	0.781	**0.953**	**0.986**	0.152	0.066	0.463
	RSA (Ours)	NYUv2,KITTI,VOID	**0.788**	**0.953**	**0.986**	**0.150**	**0.065**	**0.458**
MiDas	Linear Fit	SUN-RGBD	0.632	0.912	0.971	0.241	0.102	1.132
	Global	NYUv2	0.572	0.889	0.956	0.297	0.132	1.464
	Image	NYUv2,KITTI	0.594	0.895	0.962	0.275	0.125	1.374
	RSA (Ours)	NYUv2,KITTI	0.612	0.903	0.964	0.268	0.122	1.302
	RSA (Ours)	NYUv2,KITTI,VOID	**0.623**	**0.908**	**0.968**	**0.253**	**0.116**	**1.223**
DepthAnything	Linear Fit	SUN-RGBD	0.878	0.979	0.995	0.113	0.054	0.332
	Global	NYUv2	0.534	0.872	0.951	0.313	0.138	1.692
	Image	NYUv2,KITTI,VOID	0.588	0.892	0.963	0.279	0.126	1.392
	RSA (Ours)	NYUv2,KITTI	0.621	0.915	0.970	0.238	0.099	**1.024**
	RSA (Ours)	NYUv2,KITTI,VOID	**0.645**	**0.927**	**0.978**	**0.203**	**0.095**	1.137

Experiments

- Qualitative Results



Thank you!