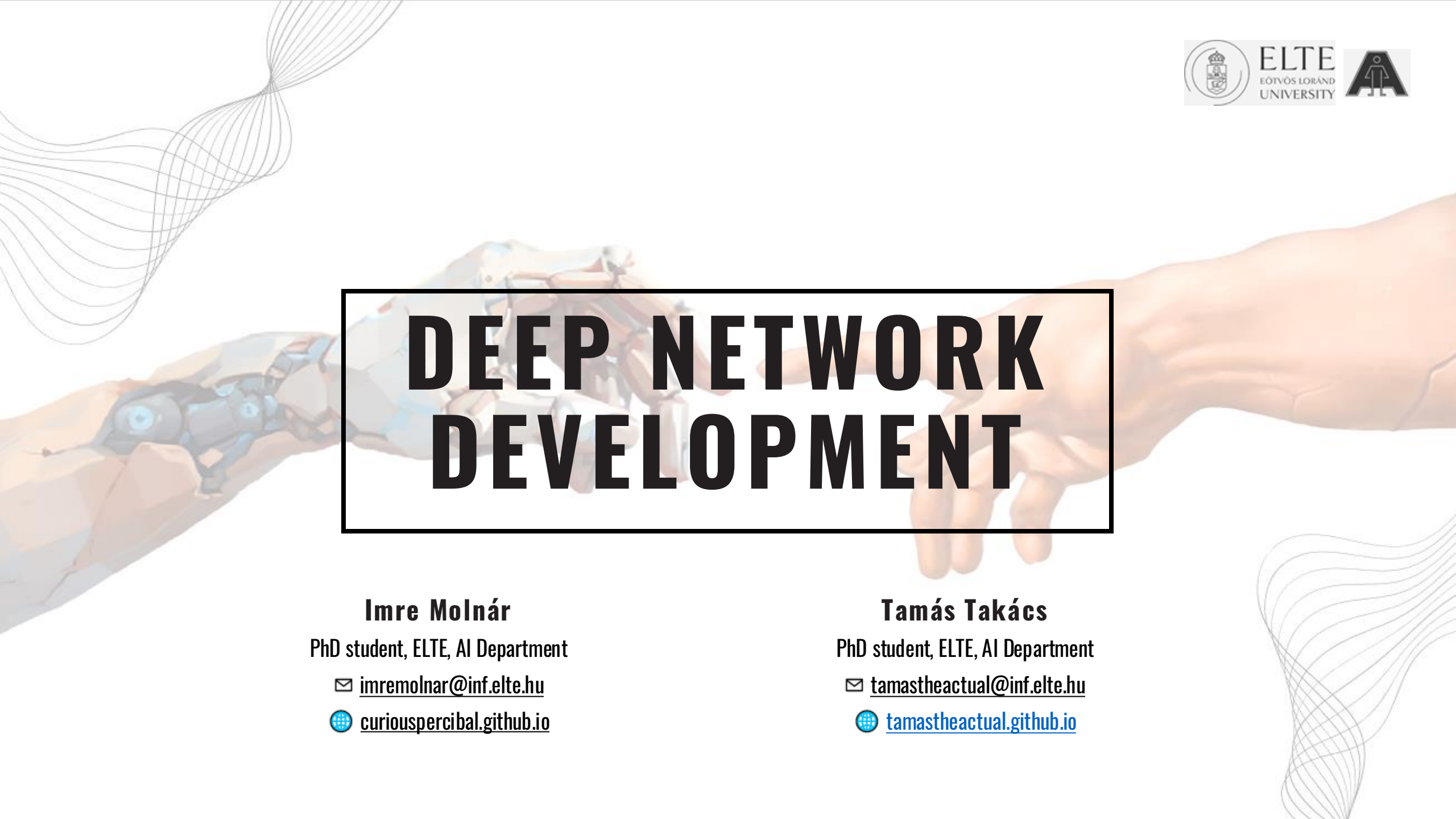


DEEP NETWORK DEVELOPMENT

Imre Molnár

PhD student, ELTE, AI Department

imremolnar@inf.elte.hu

curiouspercibal.github.io

Tamás Takács

PhD student, ELTE, AI Department

tamastheactual@inf.elte.hu

tamastheactual.github.io

Deep Network Development

Lecture 11.

Deep Learning Tools For Computer Vision part 1

Budapest, 02nd May 2025

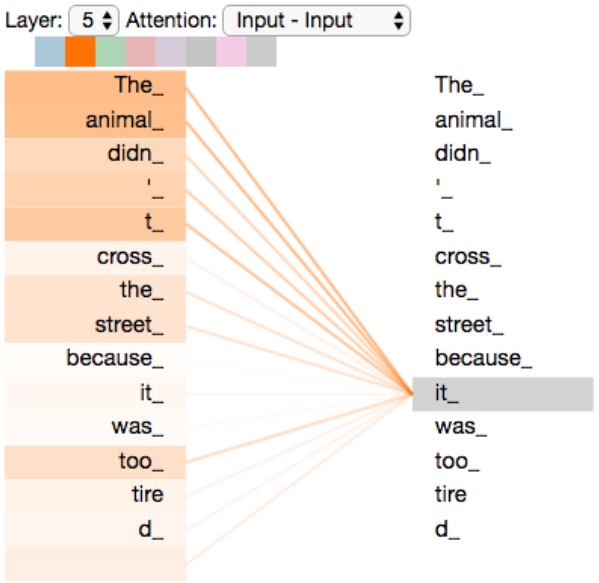
1 Annotation Tools

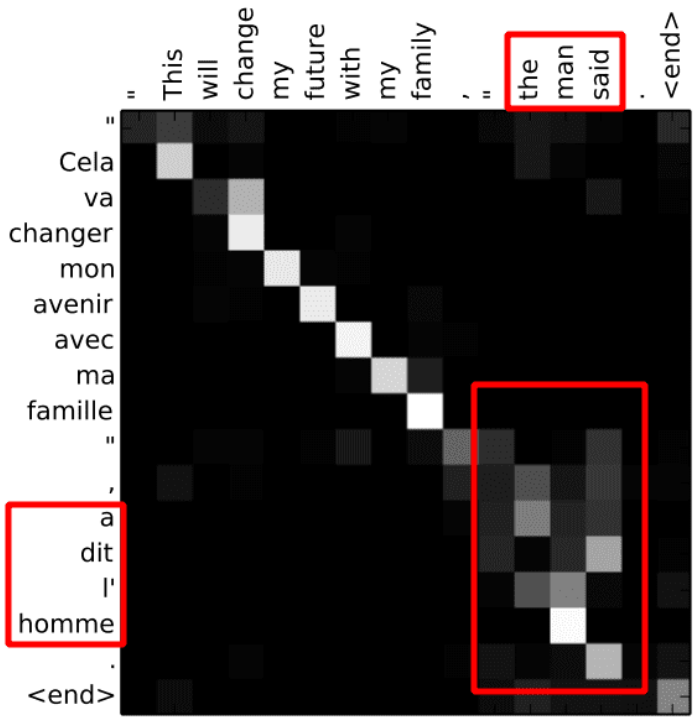
2 Depth Estimation

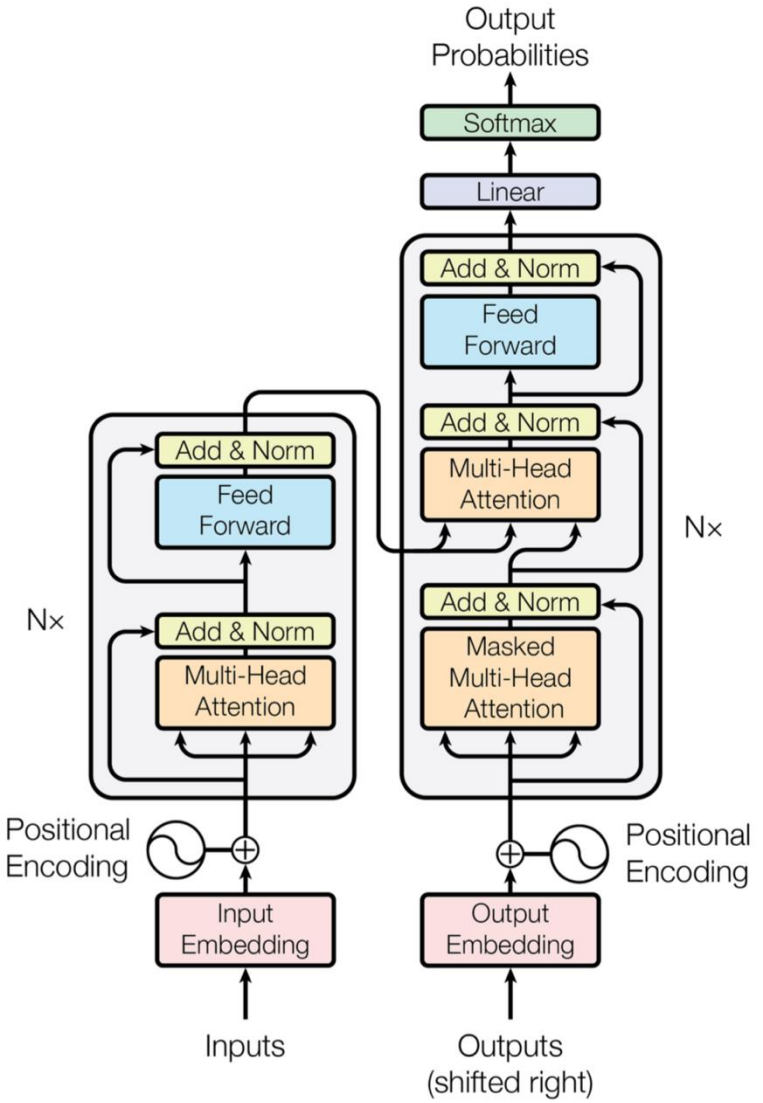
3 Optical Flow

Previously on Lecture 10

- Attention, Self-Attention, Multi-Head Attention
- Transformers

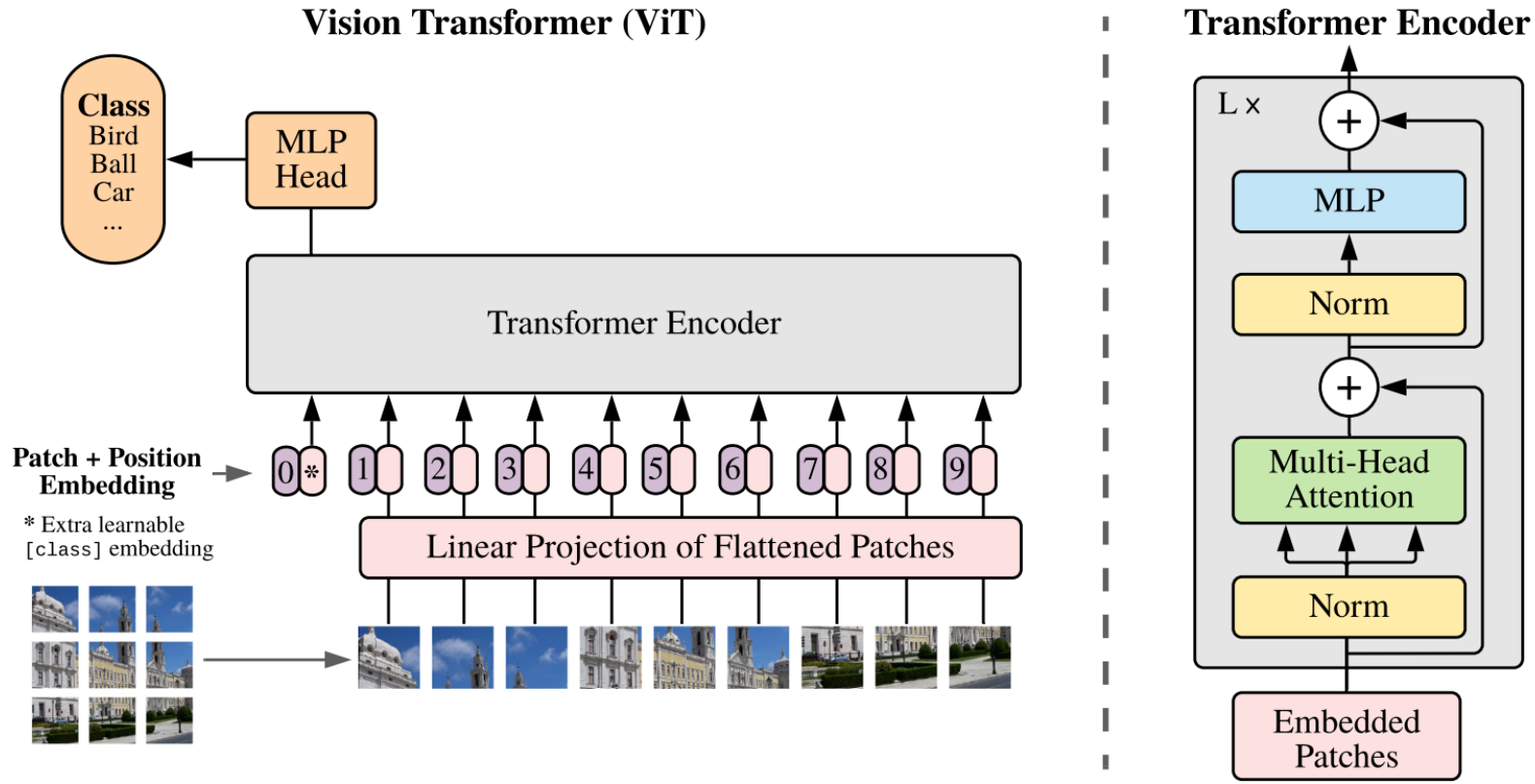






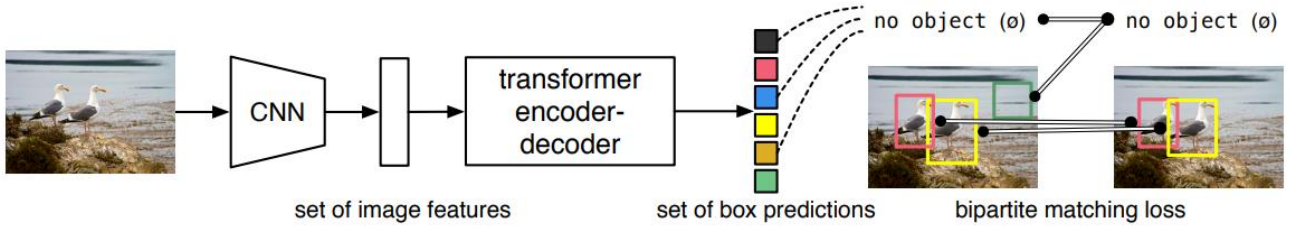
Previously on Lecture 10

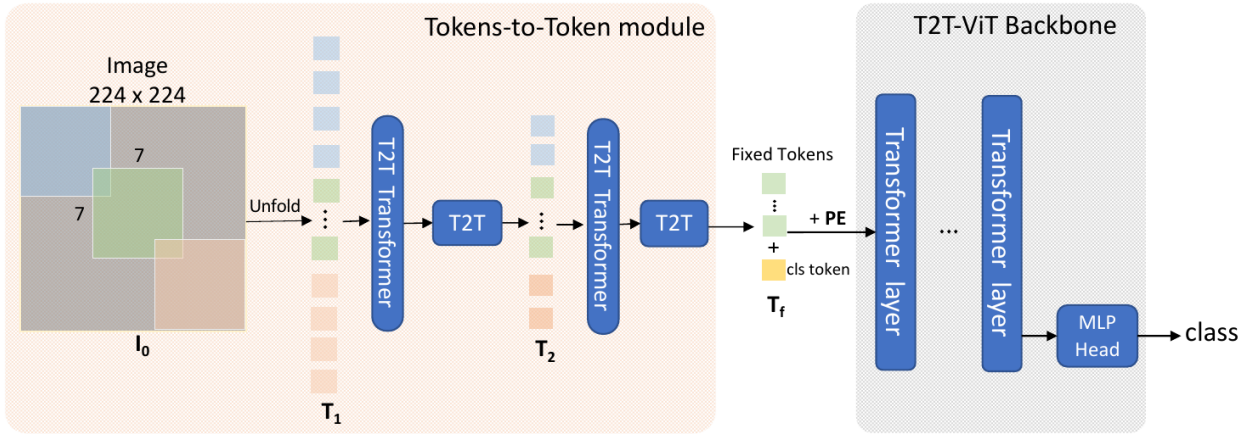
- Vision Transformer

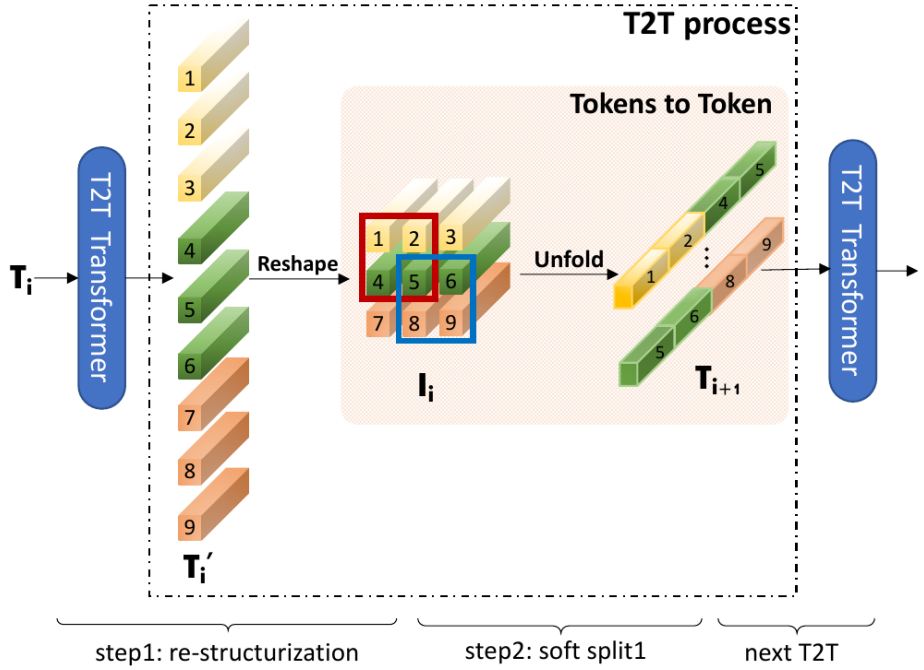


Previously on Lecture 10

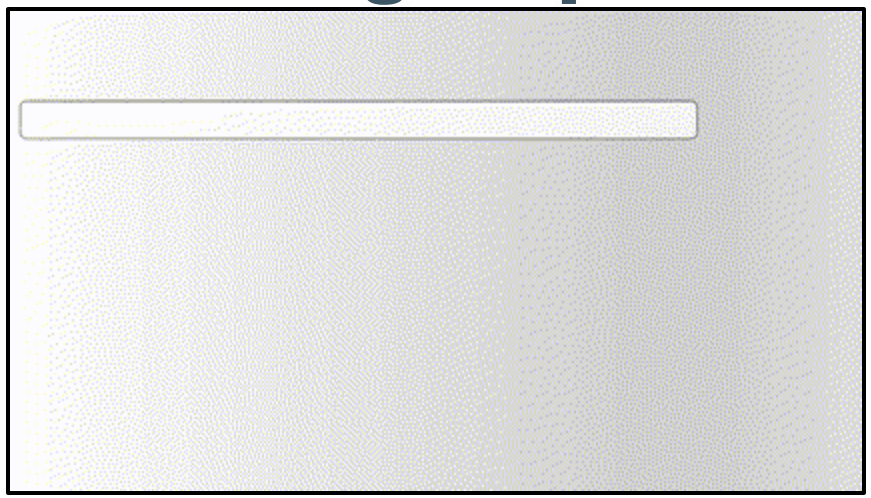
- Token-to-Token ViT
- SOTA Vision Transformers (DETR, SAM, etc.)





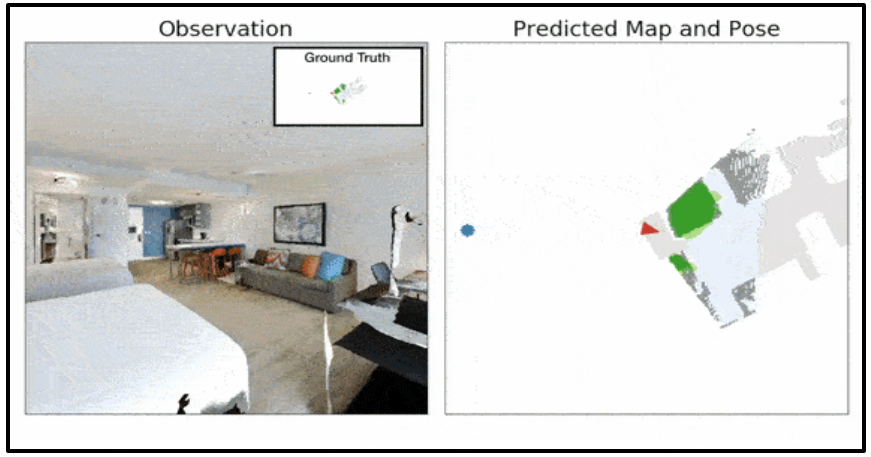


Tools Utilizing Deep Learning



GLIGEN (Grounded Language-to-Image Generation)

Gaussian Splatting



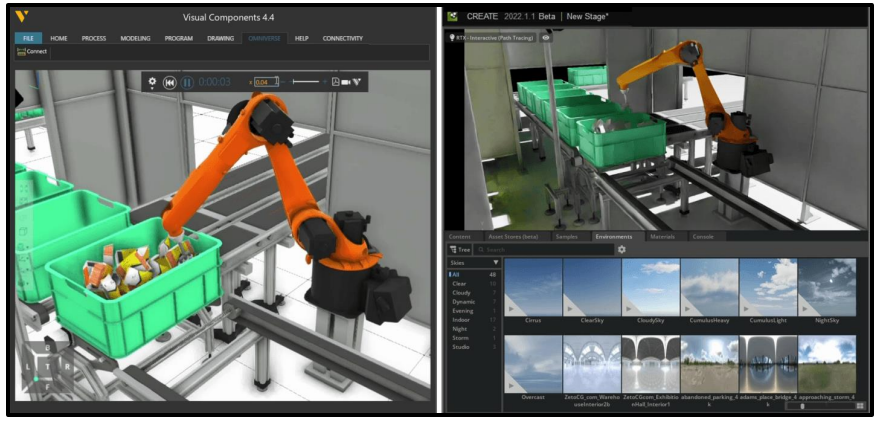
Neural SLAM (Simultaneous Localization and Mapping)

OpenAI Sora

Tools Utilizing Deep Learning

VISUAL COMPONENTS

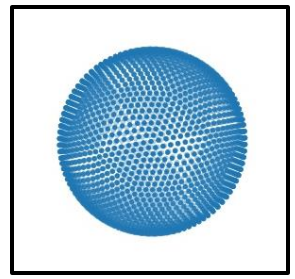
NVIDIA OMNIVERSE



3D scene and object generation using NVIDIA's OMNIVERSE

NeRF (Neural Radiance Fields)

Vid2Avatar



Meta's PyTorch3D

Meta's PyTorch3D is a Python framework designed for 3D object processing and deep learning on 3D data.

Deep Network Development

Lecture 11.

Deep Learning Tools For Computer Vision part 1

Budapest, 02nd May 2025

1 Annotation Tools

2 Depth Estimation

3 Optical Flow

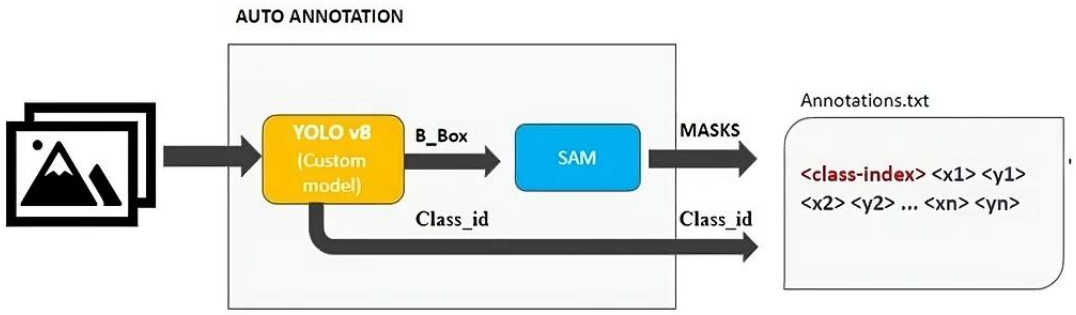
Automatic Annotation

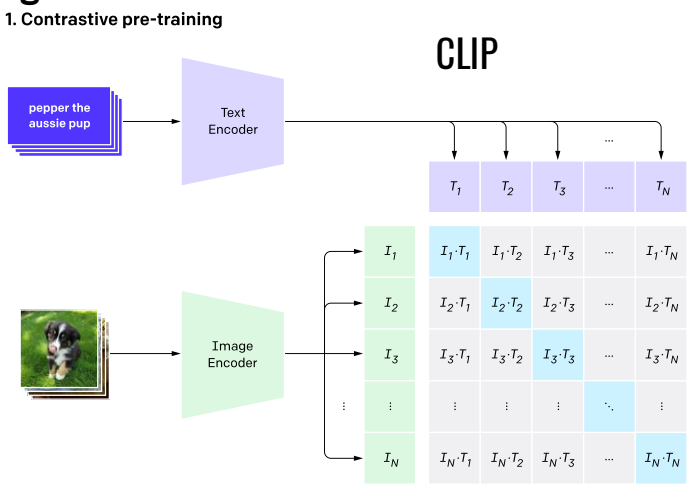
- Automatic annotation refers to the process where a computer system automatically assigns metadata, such as bounding boxes or masks, to digital images without human intervention.

- Typically, these systems leverage pre-trained vision models like **YOLO, SAM**, or **Mask R-CNN**, sometimes using combinations or ensembles of these models to enhance predictions on new data.

- Some solutions also offer adaptive options for detecting unusual objects. These options allow for real-time fine-tuning on a small subset of manually labeled data, enabling the model to perform automatic labeling on similar tasks afterward.

Simple model-assisted annotation pipeline



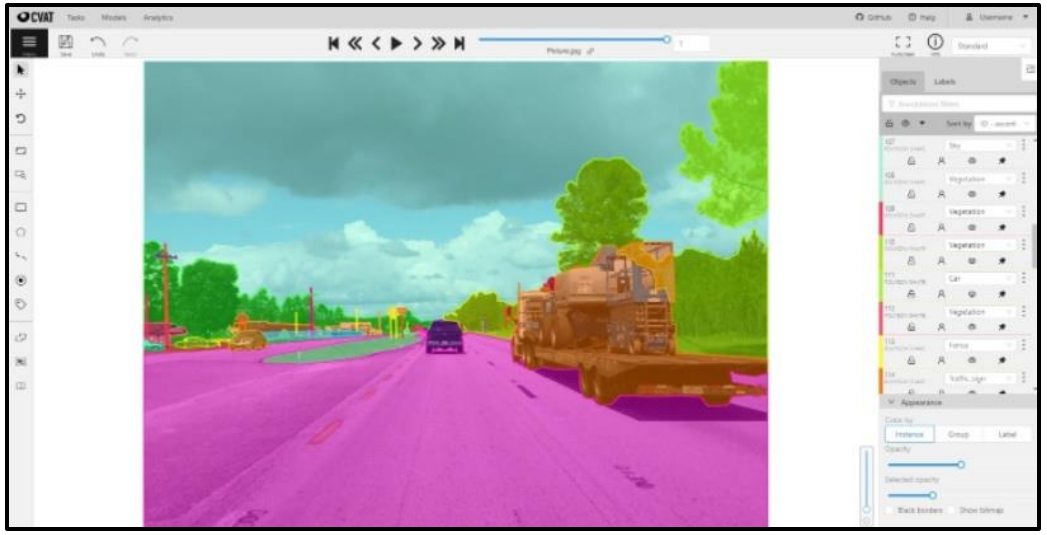


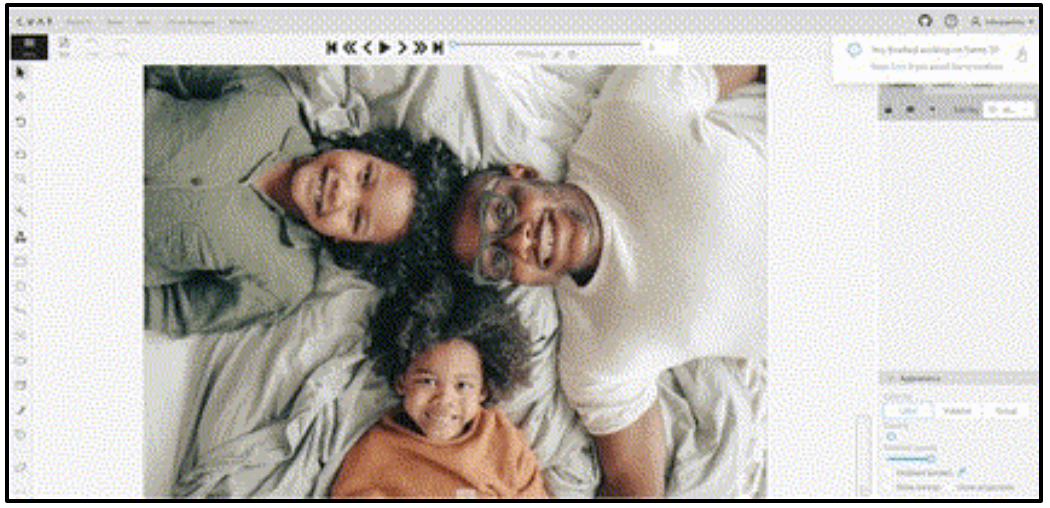
CLIP [1] leverages contextual information about the specified task, making it effective as a model-assisted annotation tool for labeling specific object classes or shapes.

[1] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... Sutskever, I. (2021). Learning Transferable Visual Models From Natural Language Supervision. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2103.00020

CVAT (Computer Vision Annotation Tool)

CVAT is an open-source, web-based model-assisted image and video annotation tool for labelling data for tasks of supervised machine learning: **object detection**, **image classification**, and **image segmentation**. It offers four basic types of annotation: boxes, polygons, polylines, and points.





Key features:

- Semi-automatic annotation
- Interpolation of shapes between keyframes
- Dashboard with a list of annotation projects and tasks
- CVAT supports PyTorch models through frameworks like Detectron2 and MMDetection for automatic annotation, allowing users to load and use models trained in PyTorch for annotation tasks.
- Exports the annotations in various formats (COCO, VOC, PASCAL, YOLO, KITTI, MOT)

https://github.com/cvat-ai/cvat
https://www.cvat.ai/post/an-introduction-to-automated-data-annotation-with-cvat-ai-cli
https://www.cvat.ai/

CVAT - Example

In this example, CVAT's SAM2 and YOLOv7 models are used in parallel to analyze the game screenshot. YOLOv7 is applied for object detection, specifically identifying a person, while SAM2 generates segmentation masks for various objects within the image.

Class: Person

Model: YOLOv7, SAM2

Game: Elden Ring

Roboflow/Autodistill

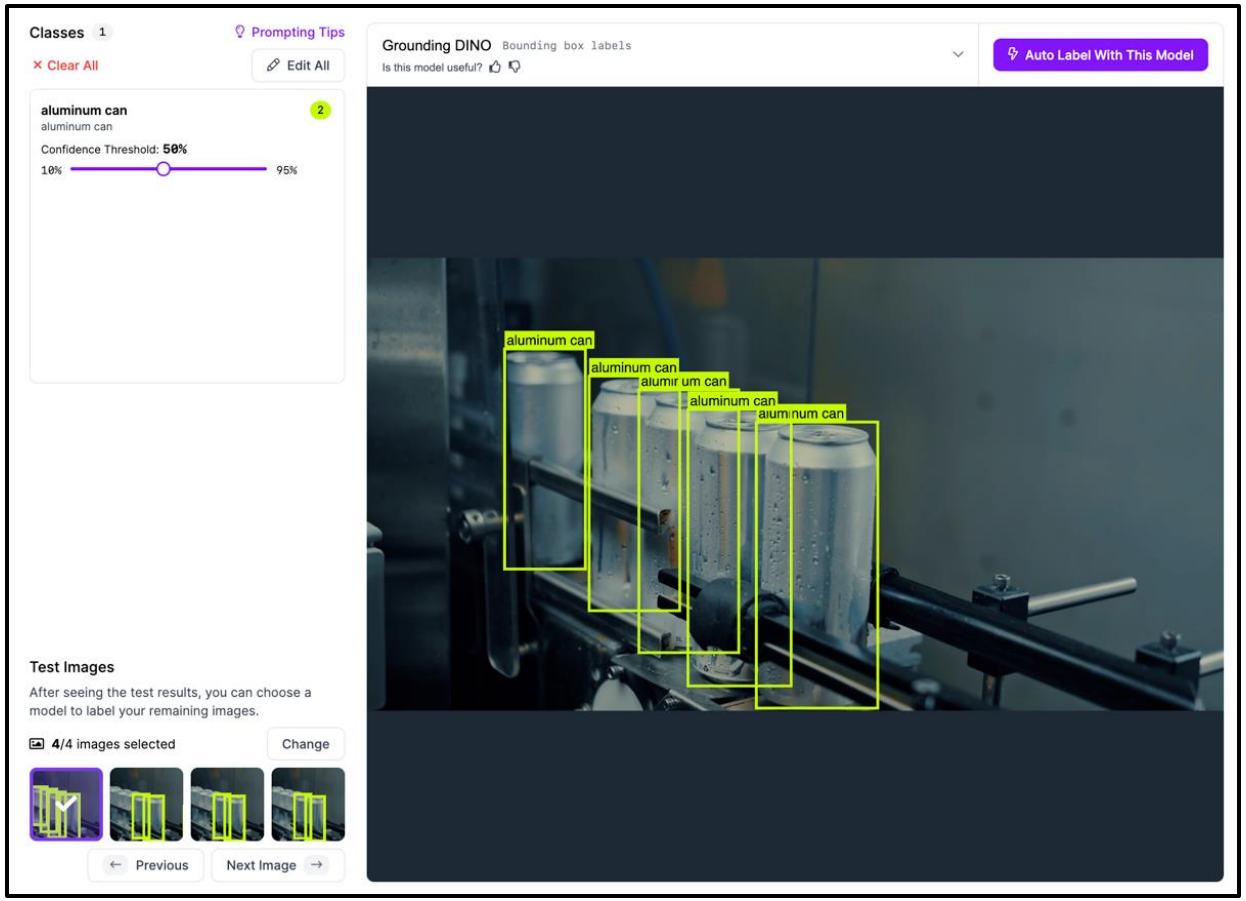
Model-Assisted annotation using foundation models:

- **Grounding DINO [2]** for **Object Detection**
- **Grounded SAM [3]** for **Segmentation**
- **Any other model** trained in Roboflow

Best suited **for labeling common objects**, such as vehicles, but may not perform well with specific or niche variants of objects.

An **ontology** (a structured representation of concepts and relationships) is used to map various prompts to object classes. The model takes a prompt and produces the labeling output with the specified class label.

https://docs.roboflow.com/annotate/automated-annotation-with-autodistill

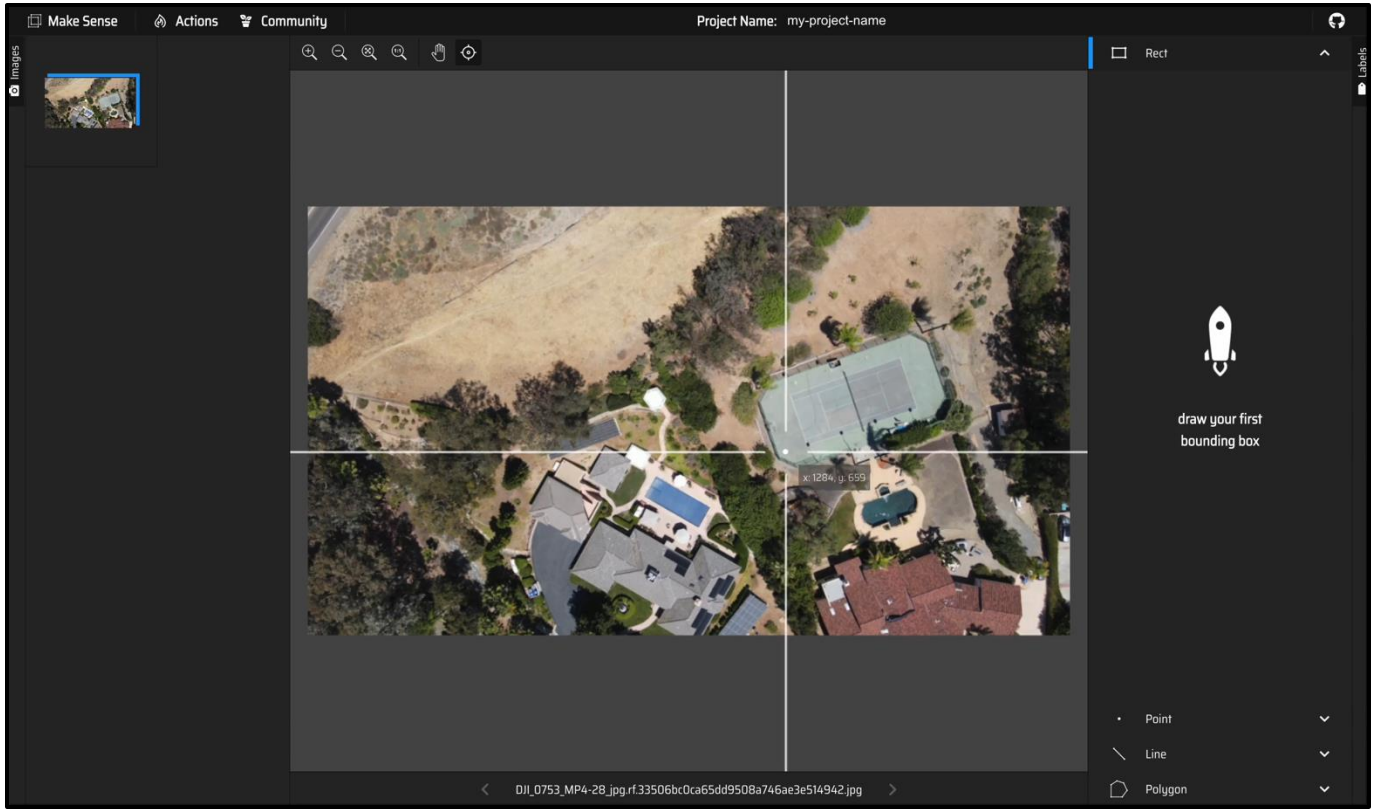


[2] Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., ... Others. (2023). Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv Preprint arXiv:2303. 05499.
[3] Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., ... Zhang, L. (2024). Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2401.14159

Make Sense

Make Sense is a free, open-source computer vision annotation tool.

- Licensed under a GPLv3 license.
- Does not require sing up or installation.
- Runs entirely in the browser, data never leaves
- Supports many formats: YOLO, VOC XML, VGG JSON etc.
- Annotations using boxes, lines, points and polygons.
- Integration with Roboflow as labelling assistant (private or public models)

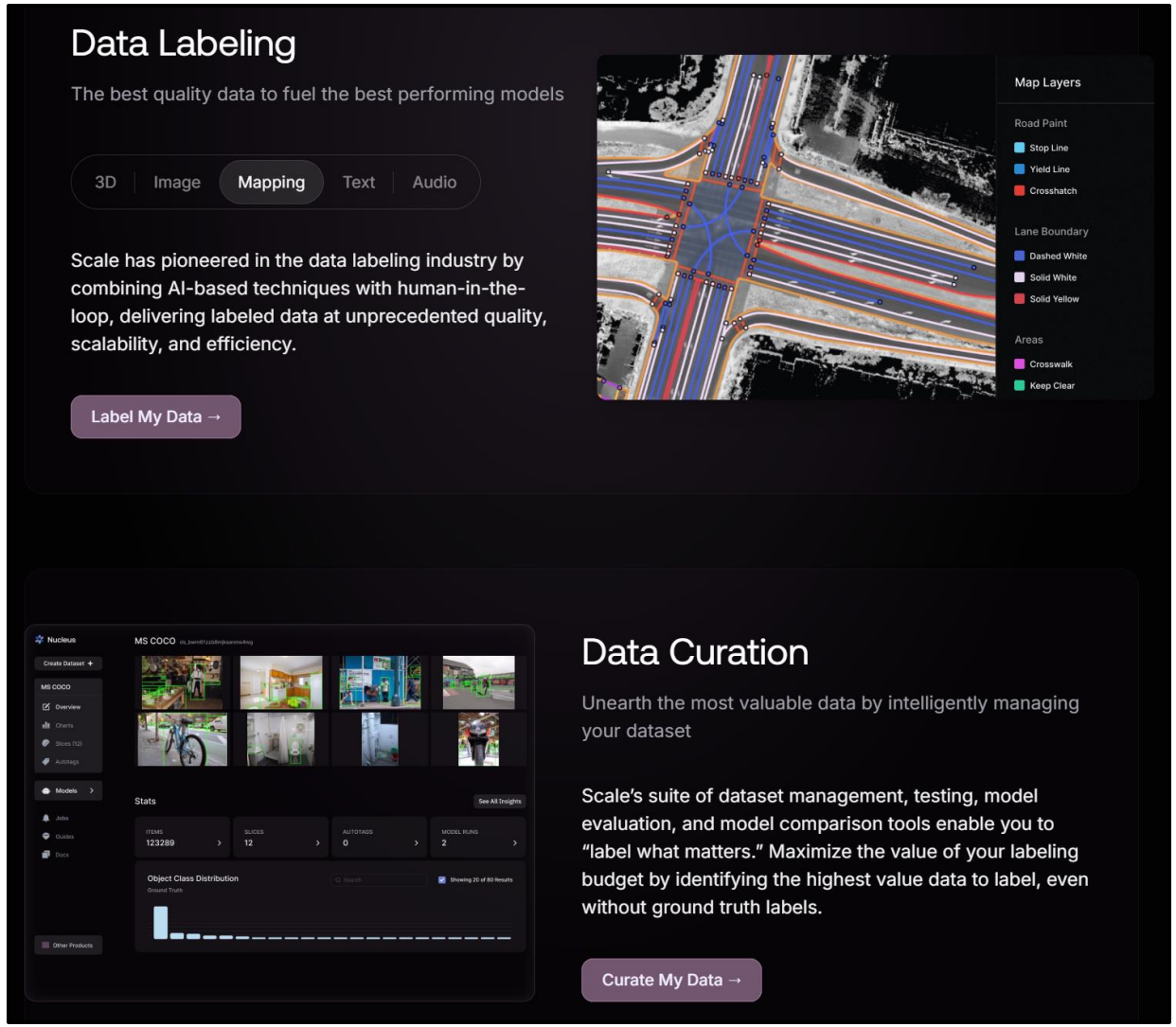


https://www.makesense.ai/ (Piotr Skalski)
https://github.com/SkalskiP/make-sense

Scale AI

Scale AI is a paid, closed-source enterprise annotation tool.

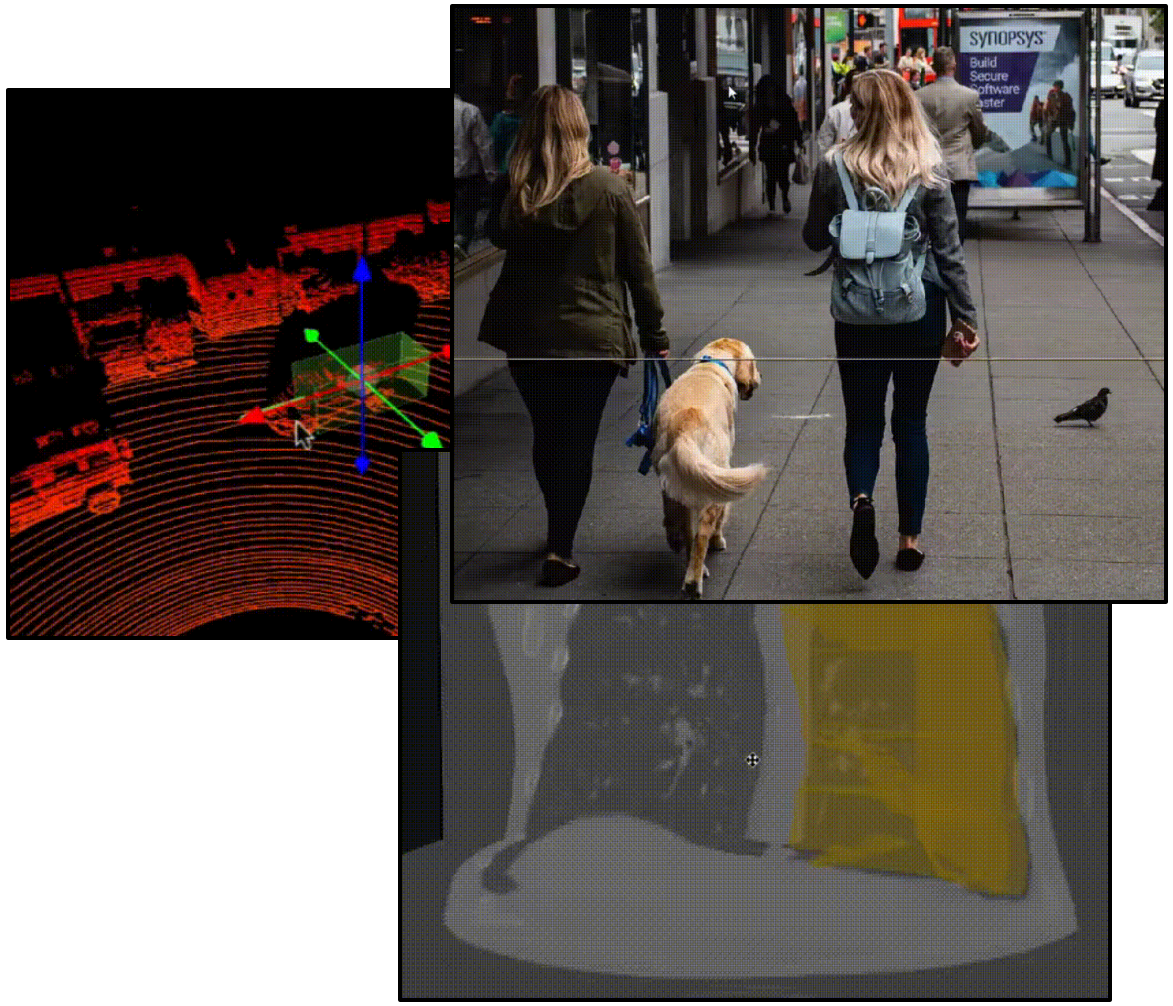
- Suite for annotating data across domains of text, video, mapping, 3D and even audio.
- Comes with a built-in automatic evaluation tool that checks data annotation quality.
- Suggests areas in which you can improve your dataset.
- Supports many formats.
- Offers annotation teams and many auto-annotation tools as well.



https://scale.com/

Automatic Annotation – Other Tools

- Supervisely:
  - Can be used for LiDAR 3D sensor fusion data and DICOM volumes
  - Offers a community edition that is free for open-source projects by individuals
  - It provides a comprehensive manual annotation suite
- V7 (V7 Darwin):
  - primarily designed for larger teams and companies requiring annotation services
  - Offers a community license, though it is more limited compared to Supervisely, Roboflow, or CVAT
  - Supports multi-channel fluorescence and time-lapse imagery, as well as cell imaging
  - Provides voxel annotation tools for 3D and multi-planar views of medical images
- Datasaur:
  - For text datasets
  - Supports sentiment analysis, text classification, named entity recognition, and other related tasks



Deep Network Development

Lecture 11.

Deep Learning Tools For Computer Vision part 1

Budapest, 02nd May 2025

1 Annotation Tools

2 Depth Estimation

3 Optical Flow

Depth Estimation

Depth estimation (in a deep learning context) is the process of predicting the distance of objects from the camera using a neural network trained to infer depth information from visual input, such as RGB images or sequences.

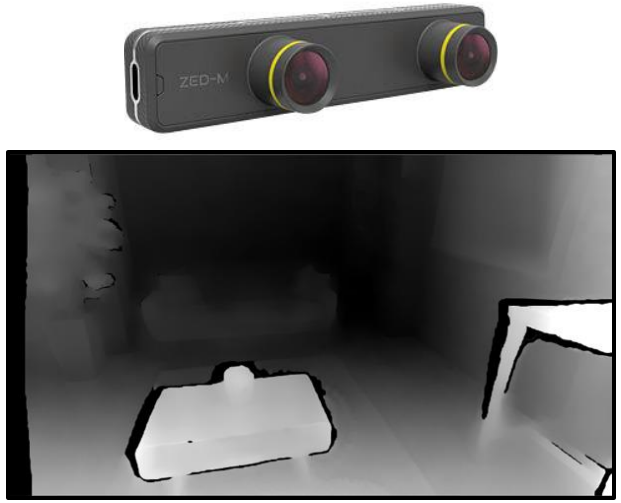
- **Monocular** Depth Estimation:
  - Estimating depth from a single image
  - Utilizes spatial cues (monoscopic depth cues) and scene understanding from a single viewpoint
  - Also referred to as Single-image depth estimation or SIDE
- **Stereo** Depth Estimation:
  - Uses two images from slightly different viewpoints (2 camera setup)
  - Estimates depth by calculating the disparity between matching points in each image (stereopsis)
  - Usually done by epipolar geometry and block matching
  - Disparity can also be learned by vision models
- Depth (Structure) **from Motion**:
  - Multi-view scenario
  - Estimates depth by matching keypoints across multiple images and triangulating their positions in 3D space
  - Sensitive from motion blur

Monoscopic Depth Cues	Examples	Appear Nearer	Appear Farther
Size of objects	Tree	Larger	Smaller
Texture	Grass patch	High quality texture	Low quality, blurry
Linear Perspective	Curb line	-	Converge to horizon

Why Depth Estimation Matters

3D point sensors like LIDAR, ToF and depth cameras are often expensive, bulky, and power-hungry, making them impractical for many consumer devices.

In contrast, monocular or stereo depth estimation can be achieved with a single or pair of RGB cameras, which are affordable, compact, and already widely integrated into these devices, enabling depth perception without additional hardware costs.



Stereo Camera

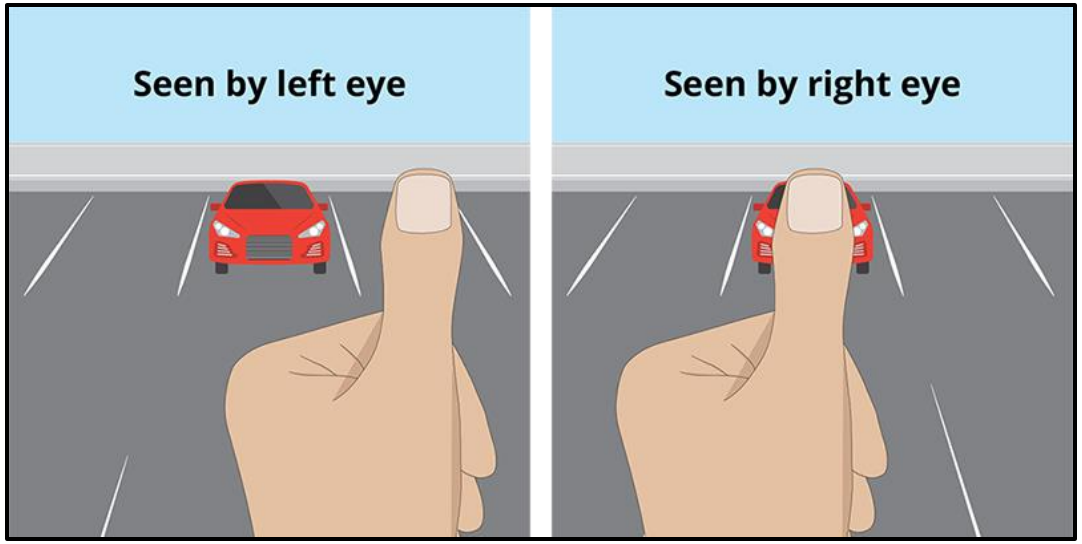
Structured Light Camera

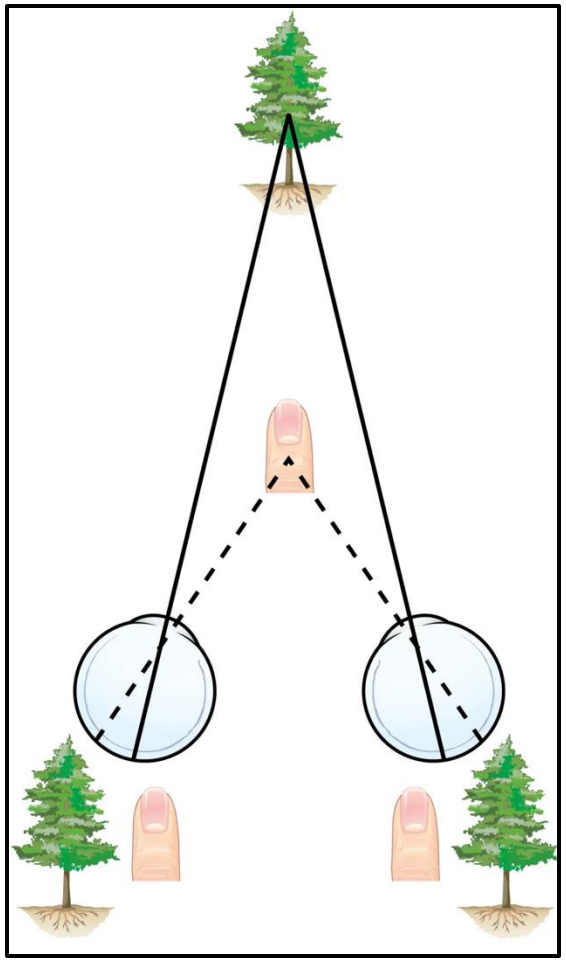
LIDAR

Stereo Vision

The human eye uses stereopsis, which is a crucial aspect of depth perception, relying on binocular disparity to create a 3D view of the world. It involves the brain fusing slightly different images from each eye, allowing us to judge distances and navigate our environment effectively.

Retinal disparity refers to the slight differences in the position of an object's image on the left and right retinas due to the horizontal separation of the eyes. Objects at different distances from the observer will have different amounts of retinal disparity, with closer objects having larger disparities than distant objects

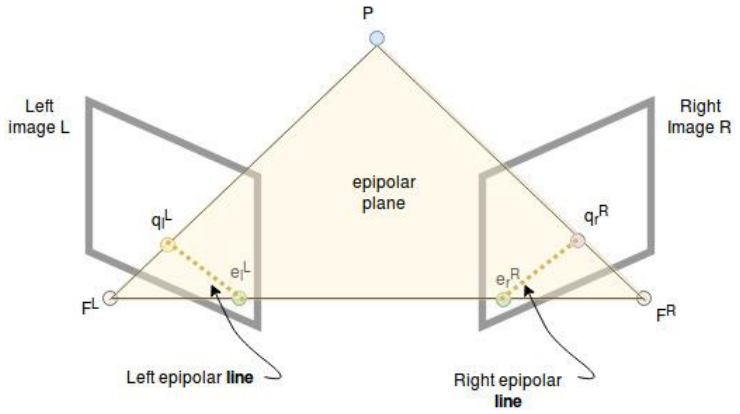


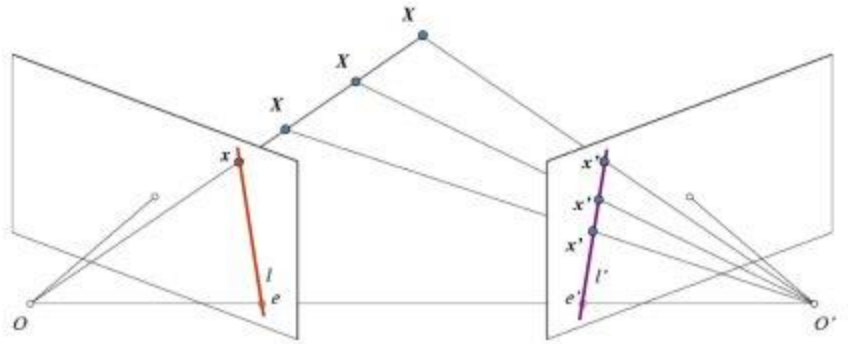


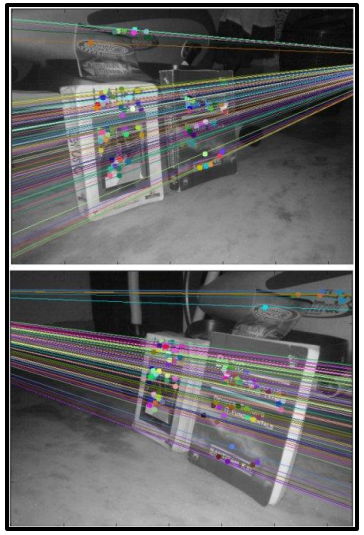
Stereo Depth Estimation

Stereo depth estimation is a technique that determines the depth of objects in a scene by analyzing the disparity between two images captured from slightly different viewpoints.

The earliest algorithms for stereo depth estimation, developed in the 1990s, achieved impressive results by leveraging geometric constraints to mathematically replicate the concept of stereopsis, enabling real-time processing.







The core approach to depth estimation with stereo cameras relies on triangulation and stereo matching. Accurate results depend on proper calibration and rectification, which constrain the problem to a 2D plane called the epipolar plane, reducing the search to a line along the epipolar line.

Monocular Depth Estimation

Depth computation from a single image is inherently ambiguous, as there are multiple ways to project the same 3D scene onto the 2D plane of an image (taking a photo of an object). To address this ambiguity, various cues such as object size, occlusion, and perspective must be considered.

Before the advent of deep learning and monocular depth estimation, special sensors or aligned stereo sensors were used to calculate depth by measuring discrepancies between stereo pairs. These solutions were mostly limited to lab conditions.

The demo uses **DPT** (Dense-Prediction Transformer) showcasing its versatility in estimating depth on any 2D image that represents a projection of a 3D scene.

Why Monocular Depth Estimation?

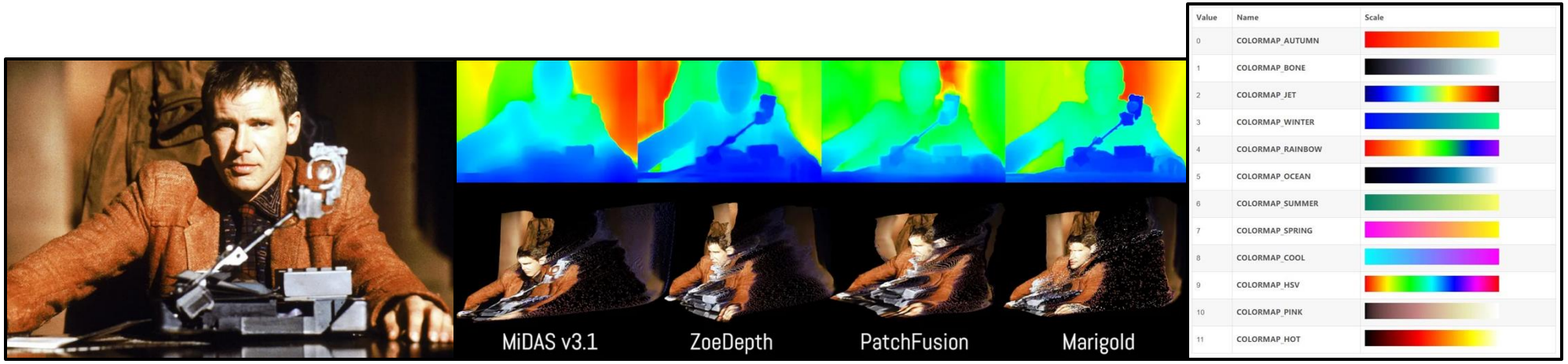
A single-camera setup is significantly cheaper than using LiDAR or stereo cameras (camera parameters are not required)

Monocular depth estimation can work with any standard camera in various settings, including mobile devices, drones, and surveillance

Monocular models typically allow for faster inference speeds compared to multi-camera setups

Monocular depth estimation models trained with diverse datasets tend to generalize well across different environments and lighting conditions

Vision Transformers have revolutionized monocular depth estimation with models such as **MiDAS v3.1**, **ZoeDepth**, **PatchFusion**, and **Marigold**.

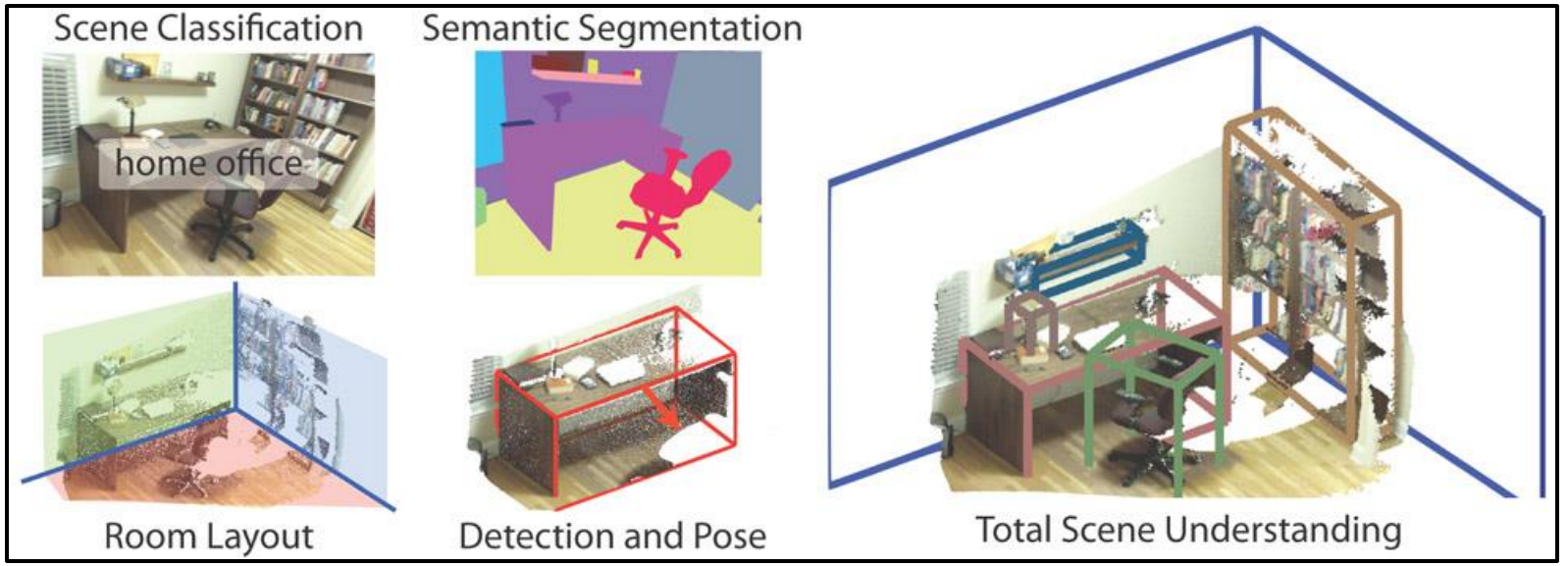


Benchmark Datasets

SUN RGB-D [4]: An RGB-D Scene Understanding Benchmark Suite. A dataset captured by four different sensors containing 10,000 RGB-D images. The whole dataset is densely annotated and includes 146,617 2D polygons and 58,657 3D bounding boxes with accurate object orientations, as well as a 3D room layout and category for scenes.

Contains images from the NYU depth v2, Berkeley B3DO, and SUN3D datasets.

One of the most popular datasets for depth estimation and segmentation tasks.



[4] Song, S., Lichtenberg, S. P., & Xiao, J. (2015). SUN RGB-D: A RGB-D scene understanding benchmark suite. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 567–576. doi:10.1109/CVPR.2015.7298655

Benchmark Datasets

KITTI Vision Benchmark Suite: The **KITTI Vision Benchmark Suite** is a comprehensive real-world dataset for benchmarking computer vision tasks, including stereo depth estimation, optical flow, visual odometry, and 3D object detection.

Collected by driving around the mid-size city of Karlsruhe, rural areas, and highways, the dataset captures scenes with up to 15 cars and 30 pedestrians per image.

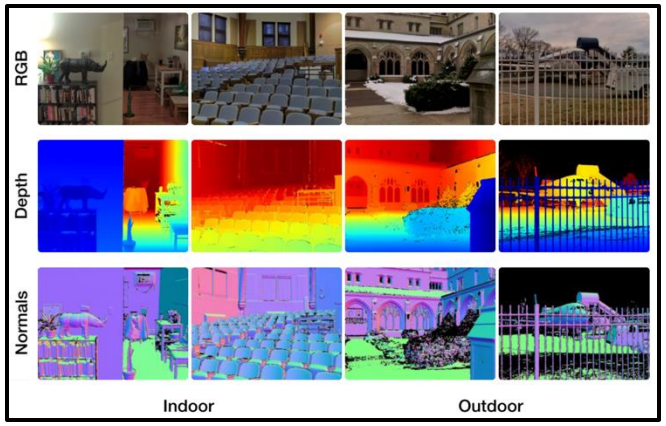
High-precision ground truth data is generated using a Velodyne laser scanner and a GPS localization system.

Benchmark Datasets

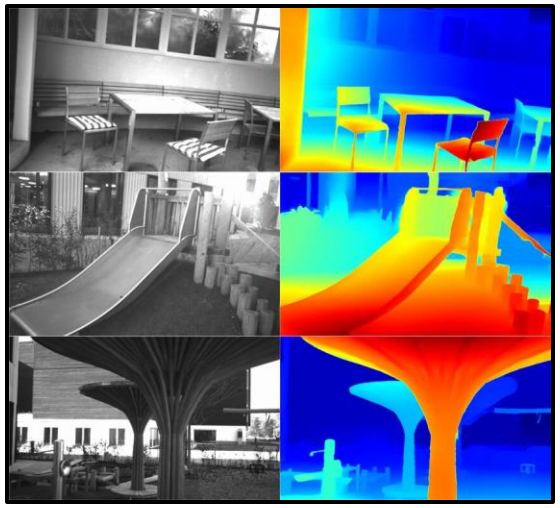
Many Others:

- Sintel (Animation)
- DDAD (Dense Depth for Autonomous Driving)
- ETH3D (Multi-view stereo benchmark)
- DIODE (Dense Indoor and Outdoor Depth)

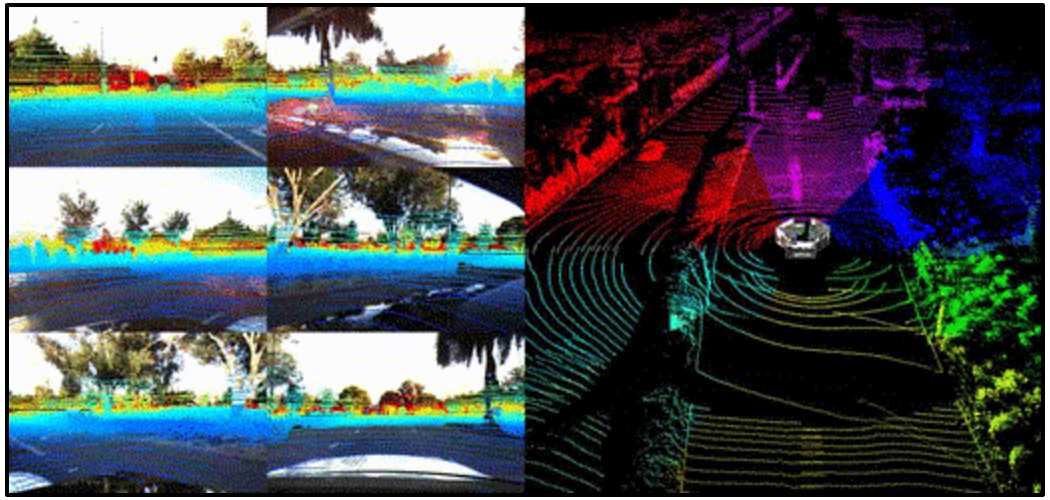
Sintel



DIODE



ETH3D

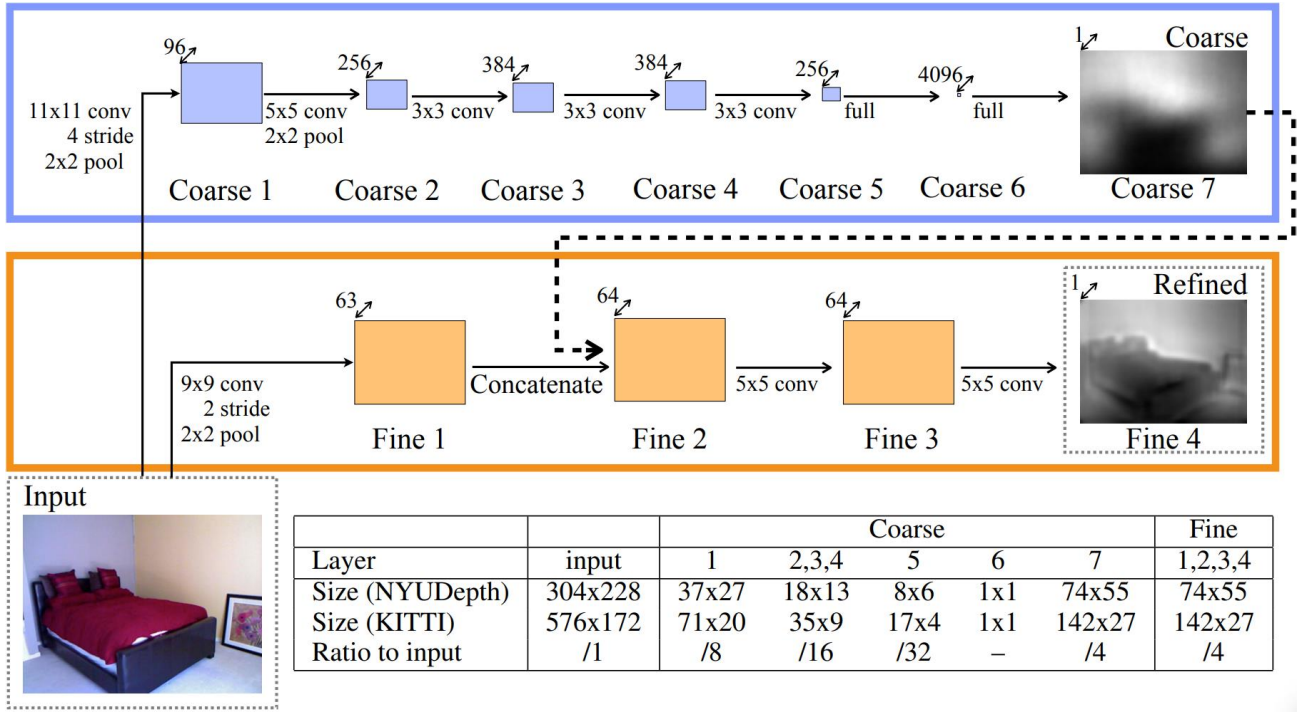


DDAD

Depth Estimation – Naïve Approach (2014)

The original paper that started the hype behind deep learning based single-image depth estimation was *"Depth Map Prediction from a Single Image using a Multi-Scale Deep Network"* [5] introducing CNNs to estimate depth from RGB images without the need for multi-camera setup.

- A two-scale CNN architecture captures both **global** (coarse network) and **local** (fine network) **depth cues**
- The **end-to-end** approach relies solely on a CNN backbone
- The lower and middle layers use max-pooling to **aggregate information** from various parts of the image, reducing it to a smaller spatial dimension
- The final output, at a **quarter of the input resolution ***, is reshaped from the last fully connected layer



		Coarse					Fine
Layer	input	1	2,3,4	5	6	7	1,2,3,4
Size (NYUDepth)	304x228	37x27	18x13	8x6	1x1	74x55	74x55
Size (KITTI)	576x172	71x20	35x9	17x4	1x1	142x27	142x27
Ratio to input	/1	/8	/16	/32	–	/4	/4

[5] Eigen, D., Puhrsch, C., & Fergus, R. (2014). Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/1406.2283

Naïve Approach - Loss

Due to the issues in different input and output sizes a pixel-wise loss function cannot be used. Traditional loss functions, like Mean Squared Error (MSE), calculate the error by comparing each pixel's predicted depth with its true depth value.

This means they heavily penalize any differences between the predicted depth and the exact ground truth depth but because there's only one viewpoint, **the model doesn't have enough information to determine the exact distances** (absolute depth) of objects.

Mean Error Squared

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$$

Where Y_i is the target absolute depth value for a pixel, and $\hat{Y}_i$ is the predicted depth. The Mean Squared Error (MSE) is calculated for each pixel.

Naïve Approach - Loss

For example, if the true depth of object A is 5 meters and object B is 10 meters, the model might predict them as 2.5 meters and 5 meters, respectively. This prediction is **correct in a relative sense** (object A is still closer than object B) but is **off in an absolute sense** (distances are half of the actual values).

The scale-invariant loss focuses on the **relative differences** between predicted and true depth values across the image, reducing sensitivity to overall scale mismatches. It compares depth values in a way that penalizes inconsistencies in relative depth relationships rather than absolute depth values.

$$D(y, y^*) = \frac{1}{n^2} \sum_{i,j} \left((\log y_i - \log y_j) - (\log y_i^* - \log y_j^*) \right)^2$$

Where the terms $\log y_i$ and $\log y_i^*$ represent the true and predicted depth values for pixel i.

The pairwise differences capture **relative depth relationships** by comparing how the depth of one pixel relates to another.

The sum is taken over all pixel pairs, and the result is averaged (scales poorly with the size of the image).

Naïve Approach – Examples

[5] Eigen, D., Puhrsch, C., & Fergus, R. (2014). Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/1406.2283

Depth Estimation – Unsupervised Approach (2016-17)

The paper *"Unsupervised Monocular Depth Estimation with Left-Right Consistency"* [6] proposes an unsupervised depth estimation approach using stereo image pairs.

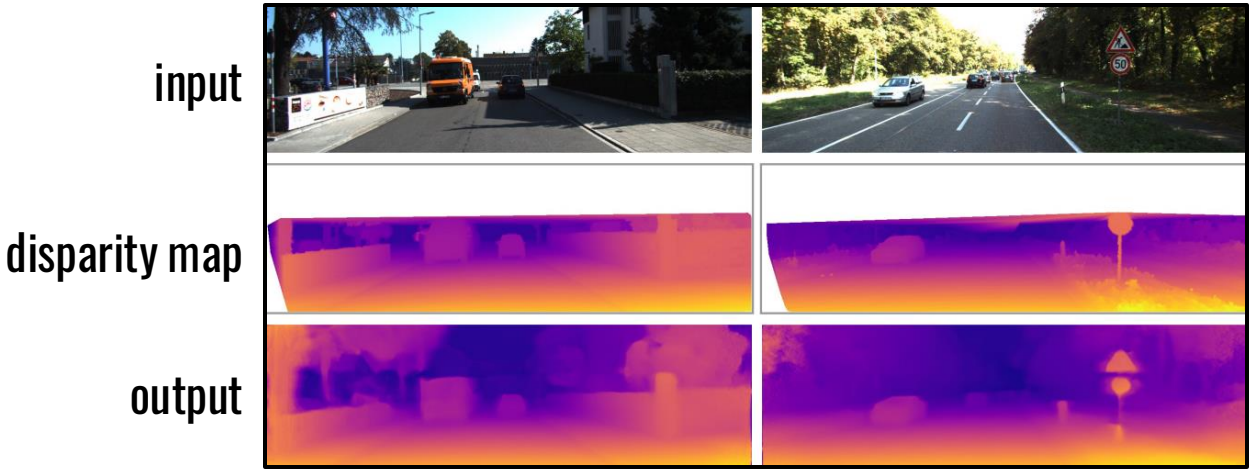
During training, the model predicts depth from a single image by leveraging **left-right consistency**. Although depth is predicted for each image individually, a consistency constraint ensures that the depth map for the left image aligns with the right image's view and vice versa, enforcing coherence without ground truth depth maps.

At test time, the model requires only one image to produce robust, generalized depth estimates.

No ground truth depth data are needed.

"Encoder"						
layer	k	s	chns	in	out	input
conv1	7	2	3/32	1	2	left
conv1b	7	1	32/32	2	2	conv1
conv2	5	2	32/64	2	4	conv1b
conv2b	5	1	64/64	4	4	conv2
conv3	3	2	64/128	4	8	conv2b
conv3b	3	1	128/128	8	8	conv3
conv4	3	2	128/256	8	16	conv3b
conv4b	3	1	256/256	16	16	conv4
conv5	3	2	256/512	16	32	conv4b
conv5b	3	1	512/512	32	32	conv5
conv6	3	2	512/512	32	64	conv5b
conv6b	3	1	512/512	64	64	conv6
conv7	3	2	512/512	64	128	conv6b
conv7b	3	1	512/512	128	128	conv7

"Decoder"						
layer	k	s	chns	in	out	input
upconv7	3	2	512/512	128	64	conv7b
iconv7	3	1	1024/512	64	64	upconv7+conv6b
upconv6	3	2	512/512	64	32	iconv7
iconv6	3	1	1024/512	32	32	upconv6+conv5b
upconv5	3	2	512/256	32	16	iconv6
iconv5	3	1	512/256	16	16	upconv5+conv4b
upconv4	3	2	256/128	16	8	iconv5
iconv4	3	1	128/128	8	8	upconv4+conv3b
disp4	3	1	128/2	8	8	iconv4
upconv3	3	2	128/64	8	4	iconv4
iconv3	3	1	130/64	4	4	upconv3+conv2b+disp4*
disp3	3	1	64/2	4	4	iconv3
upconv2	3	2	64/32	4	2	iconv3
iconv2	3	1	66/32	2	2	upconv2+conv1b+disp3*
disp2	3	1	32/2	2	2	iconv2
upconv1	3	2	32/16	2	1	iconv2
iconv1	3	1	18/16	1	1	upconv1+disp2*
disp1	3	1	16/2	1	1	iconv1



[6] Godard, C., Aodha, O. M., & Brostow, G. J. (2017). Unsupervised Monocular Depth Estimation with Left-Right Consistency. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/1609.03677

Dense Prediction Transformer (DPT) [7]

The Dense Prediction Transformer (DPT) was introduced in Vision Transformers for Dense Prediction (2021) from Intel Labs research. Estimatates relative depth.

DPT has **an encoder-decoder structure** including transformer layers, while the overall architecture is **hybrid with convolutional layers.**

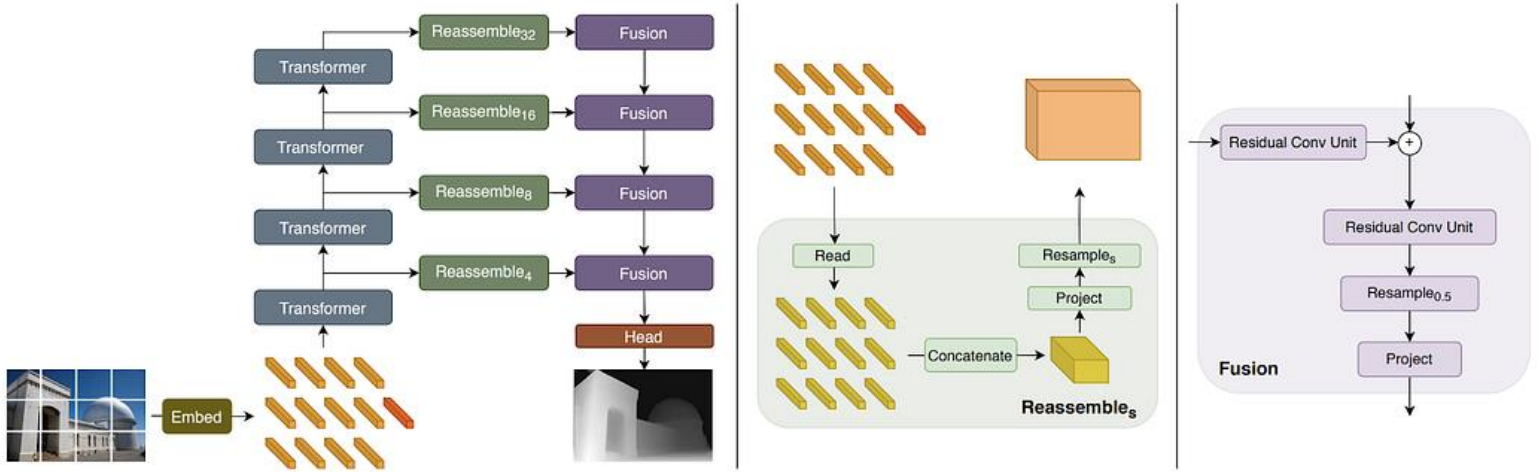
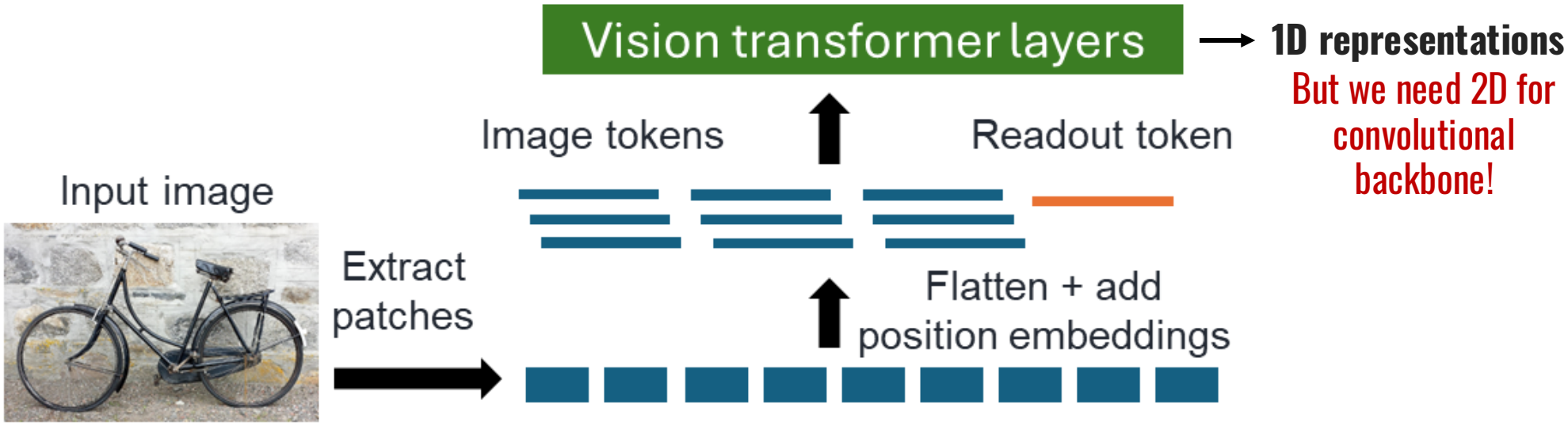


Figure 1. *Left*: Architecture overview. The input image is transformed into tokens (orange) either by extracting non-overlapping patches followed by a linear projection of their flattened representation (DPT-Base and DPT-Large) or by applying a ResNet-50 feature extractor (DPT-Hybrid). The image embedding is augmented with a positional embedding and a patch-independent readout token (red) is added. The tokens are passed through multiple transformer stages. We reassemble tokens from different stages into an image-like representation at multiple resolutions (green). Fusion modules (purple) progressively fuse and upsample the representations to generate a fine-grained prediction. *Center*: Overview of the Reassemble$_s$ operation. Tokens are assembled into feature maps with $\frac{1}{s}$ the spatial resolution of the input image. *Right*: Fusion blocks combine features using residual convolutional units [23] and upsample the feature maps.

[7] Ranftl, R., Bochkovskiy, A., & Koltun, V. (2021). Vision Transformers for Dense Prediction. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2103.13413

Dense Prediction Transformer (DPT) [7]

1. Encoder is the ViT architecture from previous lecture. **CLS** token changed to **redout** token.



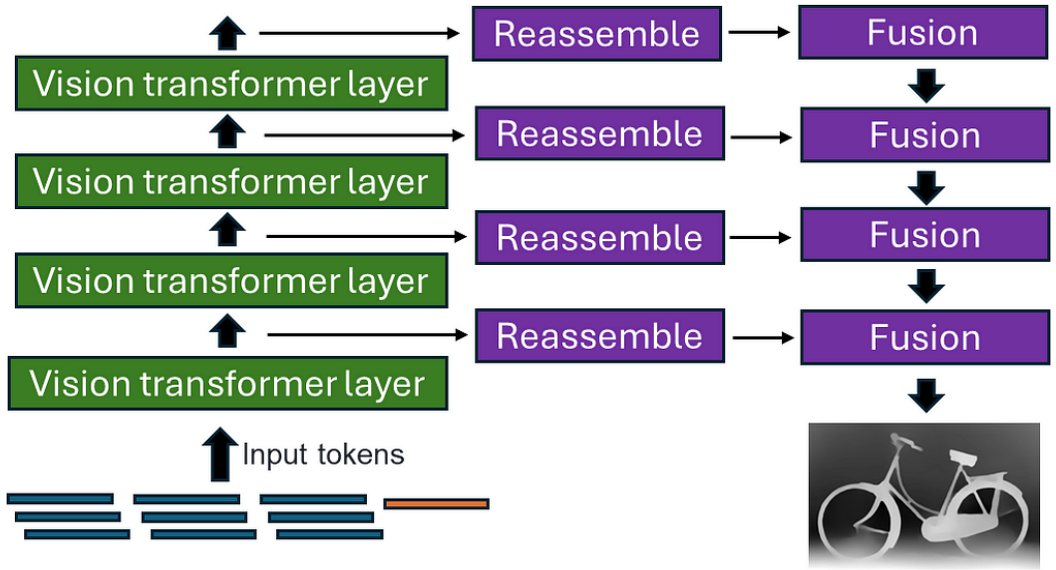
[7] Ranftl, R., Bochkovskiy, A., & Koltun, V. (2021). Vision Transformers for Dense Prediction. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2103.13413

Dense Prediction Transformer (DPT) [7]

2. To pass linear representations into the decoder, image-like representations are recovered using the reassemble operation.
   - During reassemble, image tokens are read, concatenated following the positions of the initial image patch, and then resampled.
   - Given D as the feature dimension of each of N image tokens, patch size p^2, then the spatial concatenation resulted in the representation shape with height and width as:

$$ND = \left(\frac{H}{p}\right)\left(\frac{W}{p}\right)D$$

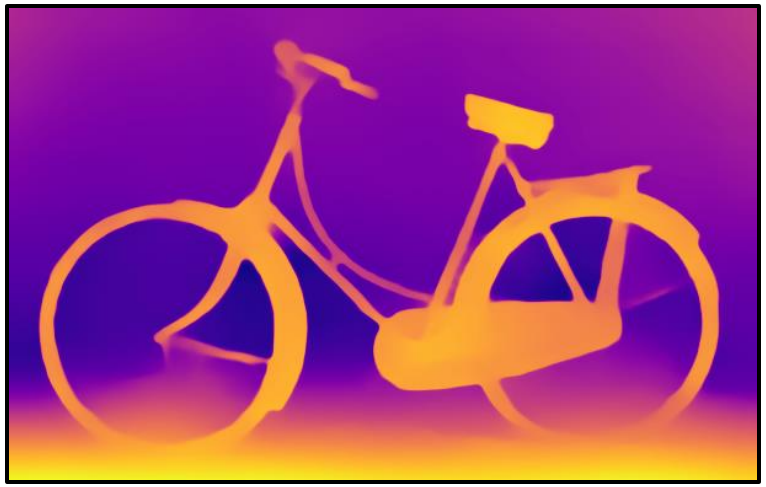
   - The decoder uses RefineNet-based [5] feature fusion from reassembled image-like representations sampled at four different stages from four different resolutions



[5] Lin, G., Milan, A., Shen, C., & Reid, I. (2016). RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/1611.06612

Dense Prediction Transformer (DPT) [7]

DPT - Hybrid



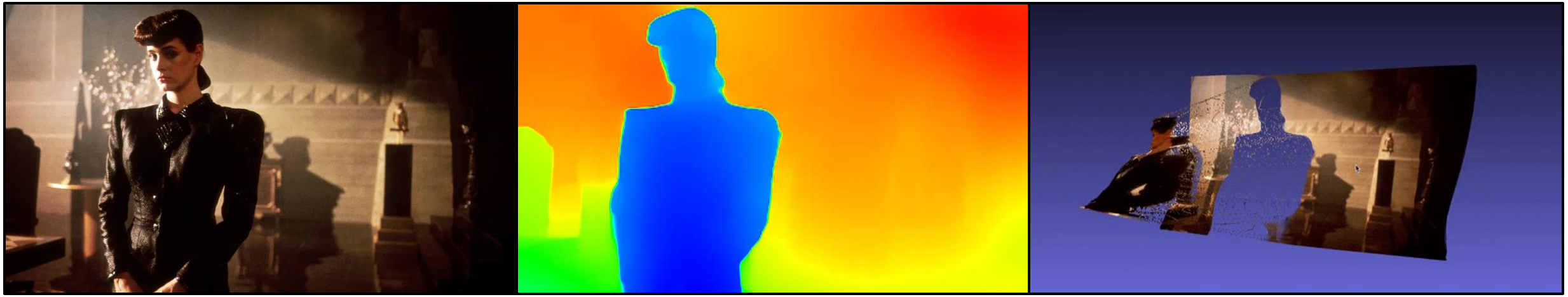
DPT - Large

Monocular Depth Estimation – MiDAS (2019-22) [8]

MiDAS was another pioneering model that sparked interest in monocular depth estimation using CNNs. Although it is no longer considered state-of-the-art, newer models continue to build upon its backbone.

MiDAS has an important characteristic: the inferred depth is relative (relative depth estimation, **RDE**). In other words, it excels at estimating the relative position of pixels (e.g., one pixel being behind another), but it isn't designed to provide consistent, absolute depth measurements.

MiDAS v3.1



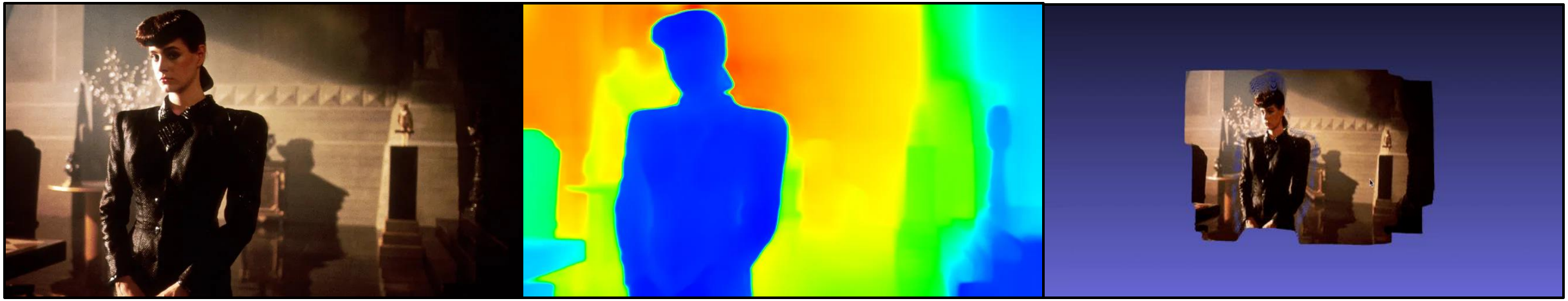
Original image | Depth map | 3D point cloud using pixel extrusion

[8] Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., & Koltun, V. (2020). Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/1907.01341

MDE – ZoeDepth (2023) [9]

ZoeDepth was built on top of **MiDAS** as a clear improvement, designed to make inferences in metric units (metric depth estimation, **MDE**). The advancements over **MiDAS** become evident when projecting onto a 3D space and observing from novel viewpoints.

ZoeDepth



Original image | Depth map | 3D point cloud using pixel extrusion

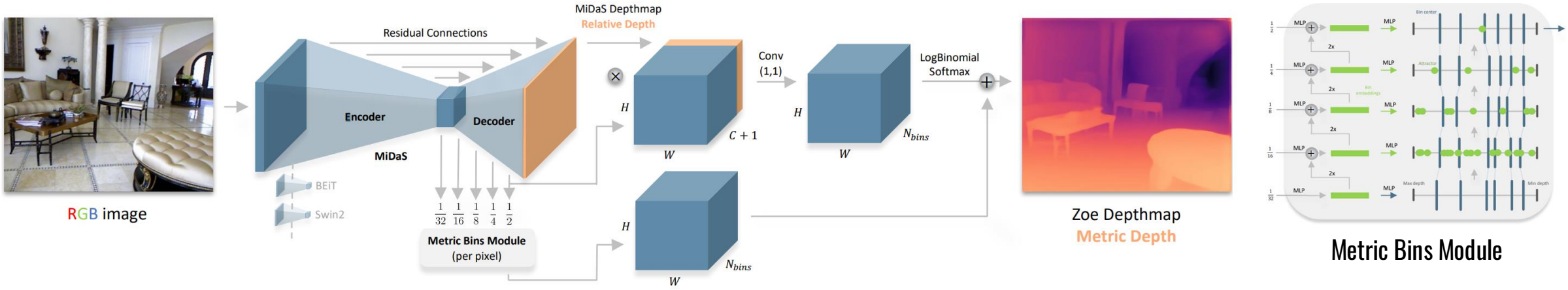
[9] Bhat, S. F., Birkl, R., Wofk, D., Wonka, P., & Müller, M. (2023). ZoeDepth: Zero-shot Transfer by Combining Relative and Metric Depth. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2302.12288

MDE – ZoeDepth (2023) [9]

ZoeDepth was built on top of **MiDAS** as a clear improvement, designed to make inferences in metric units. The advancements over **MiDAS** become evident when projecting onto a 3D space and observing from novel viewpoints.

ZoeDepth attempts to **combine the two tasks of relative depth estimation and metric depth estimation**.

Transition from CNNs to Vision Transformer encoders, such as BEiT and Swin2.



Metric Bins Module

[9] Bhat, S. F., Birkl, R., Wofk, D., Wonka, P., & Müller, M. (2023). ZoeDepth: Zero-shot Transfer by Combining Relative and Metric Depth. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2302.12288

MDE – PatchFusion (2023-24) [10]

PatchFusion is built on top of ZoeDepth and its core principle of the algorithm is that instead of making a single estimation of an entire image, we iteratively create multiple estimations of different regions—patches—of the entire image.

These patches of estimated depth are “stitched” together.

[10] Ke, B., Obukhov, A., Huang, S., Metzger, N., Daudt, R. C., & Schindler, K. (2024). Repurposing Diffusion-Based Image Generators for Monocular Depth Estimation. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2312.02145

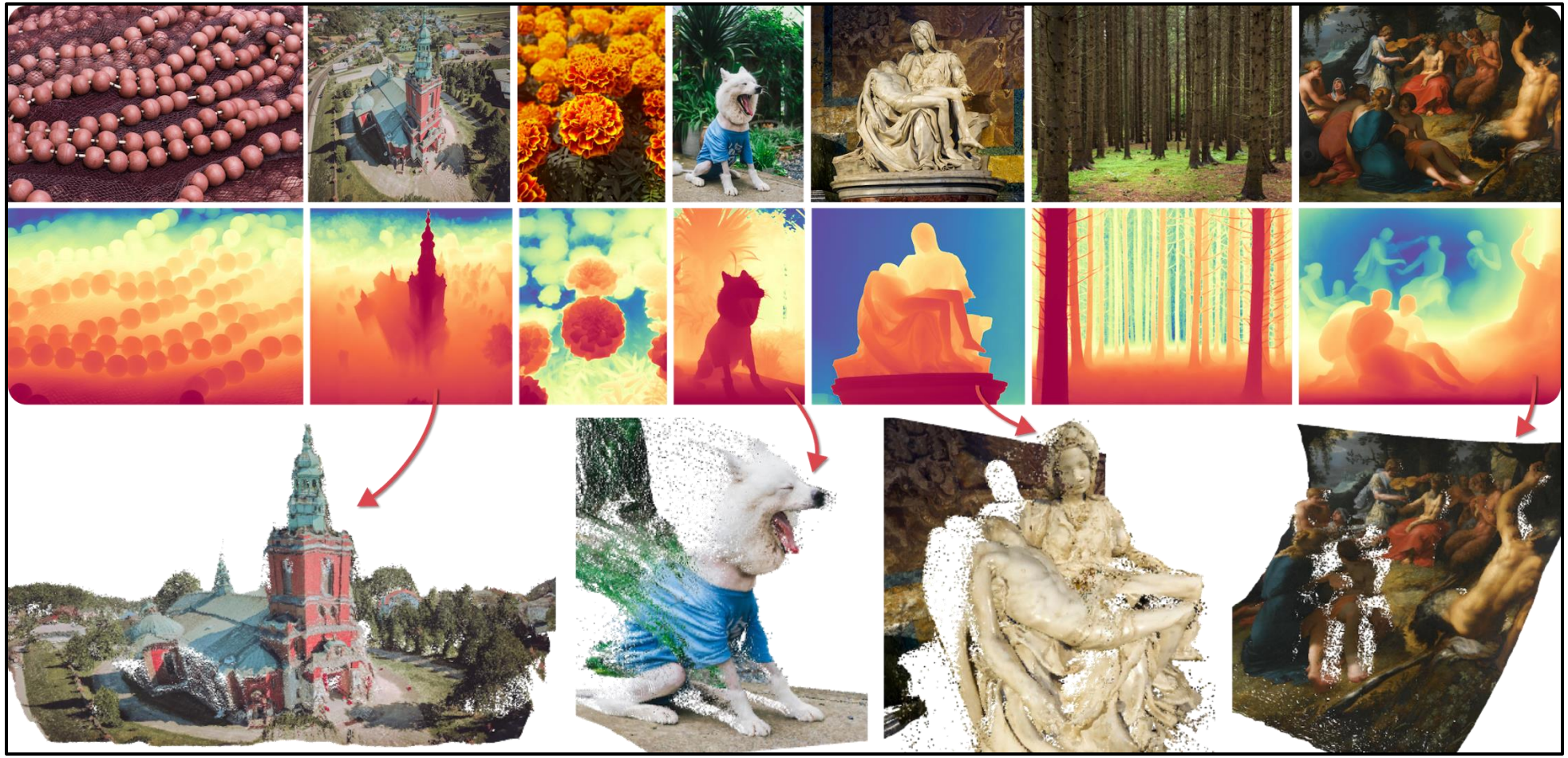
MDE – Marigold (2023-24) [11]

Marigold leverages the extensive visual knowledge embedded in modern generative models. Built on **Stable Diffusion** and fine-tuned with synthetic data, it achieves state-of-the-art monocular depth estimation with impressive zero-shot transfer capabilities on unseen data

- **Best Paper Award** - CVPR 2024
- **SOTA**

[11] Ke, B., Obukhov, A., Huang, S., Metzger, N., Daudt, R. C., & Schindler, K. (2024). Repurposing Diffusion-Based Image Generators for Monocular Depth Estimation. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2312.02145

MDE – Marigold (2023-24) [11]

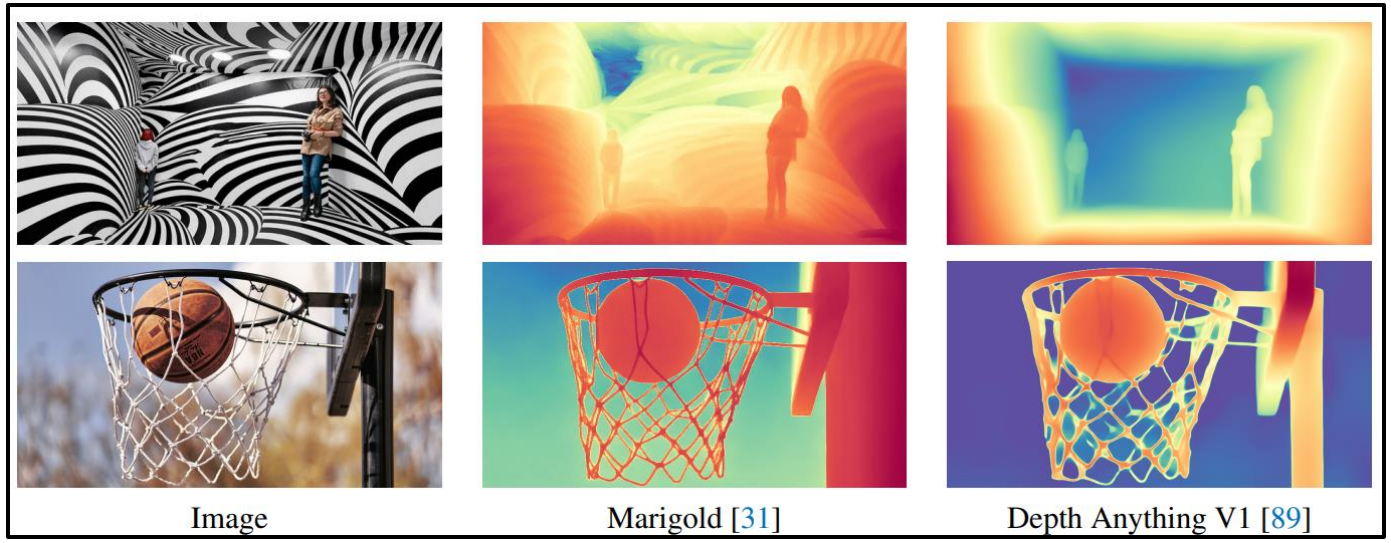


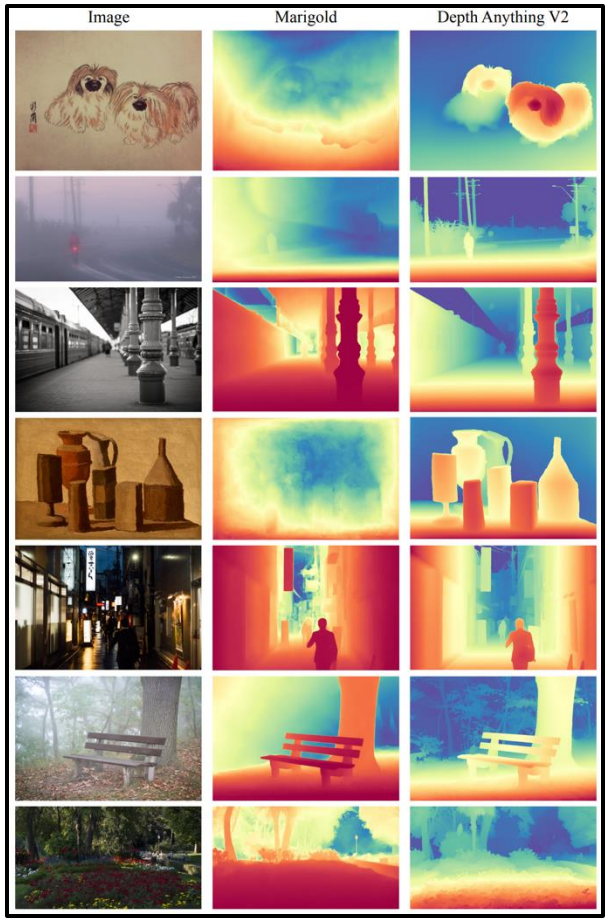
[11] Ke, B., Obukhov, A., Huang, S., Metzger, N., Daudt, R. C., & Schindler, K. (2024). Repurposing Diffusion-Based Image Generators for Monocular Depth Estimation. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2312.02145

MDE – Depth Anything v1-2 (2024-25) [12]

A discriminative model utilizing a **Dense Prediction Transformer (DPT)** architecture with a DINOv2 backbone. It focuses on direct depth prediction from image features. The authors trained a reliable teacher model based on DINOv2-G purely on high-quality synthetic images, produced precise pseudo depths on real images and finally trained a student model on the pseudo-labeled real images.

- **CVPR 2024**
- **SOTA**





https://huggingface.co/spaces/depth-anything/Depth-Anything-V2

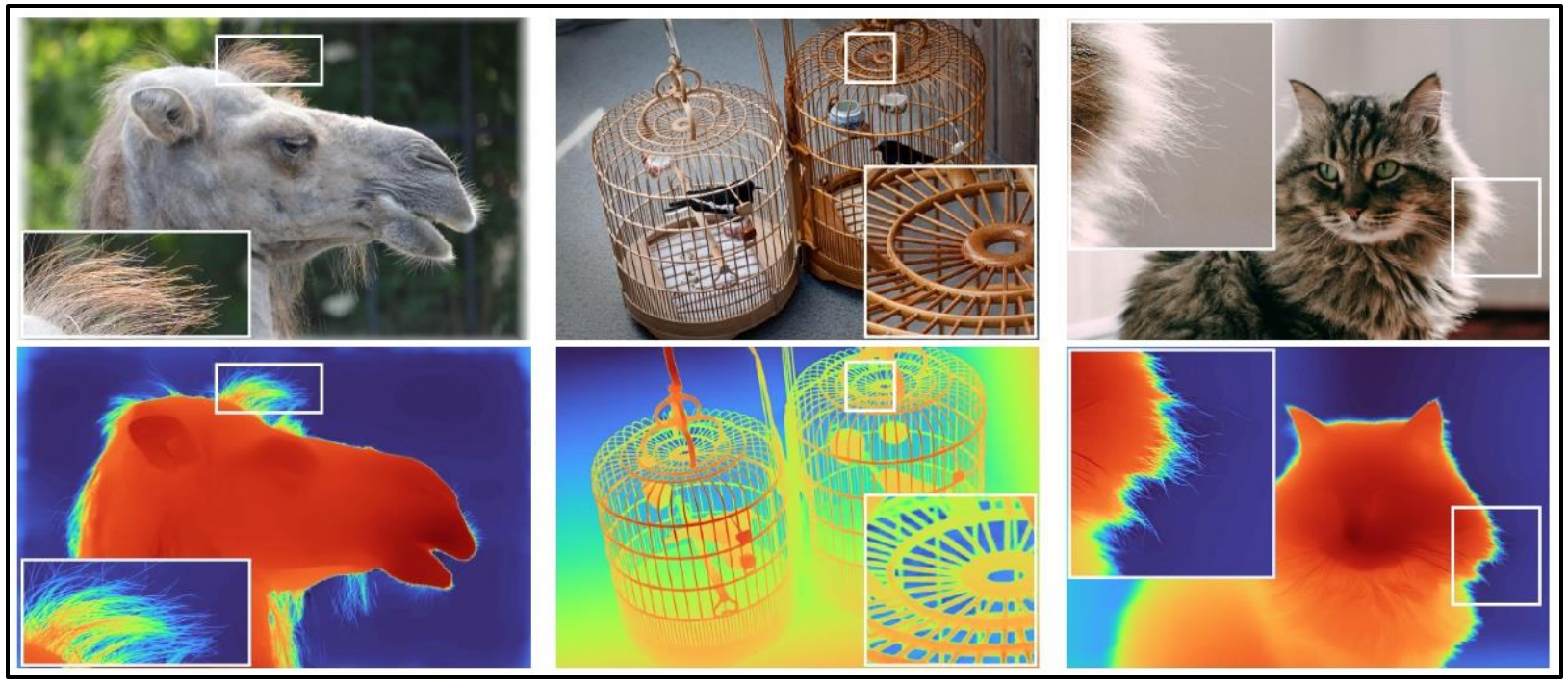
[12] Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., & Zhao, H. (2024). Depth Anything V2. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2406.09414

MDE – Depth Pro (2025) [13]

Depth Pro (Apple) foundation model for zero-shot metric monocular depth estimation. Synthesizes high-resolution depth maps with unparalleled sharpness and high-frequency details. The predictions are metric, with absolute scale, without relying on the availability of metadata such as camera intrinsics. Model is fast, producing a **2.25-megapixel depth map in 0.3 seconds on a standard GPU.**

Contains a training protocol that combines real and synthetic datasets to achieve high metric accuracy alongside fine boundary tracing

- **CVPR 2025**



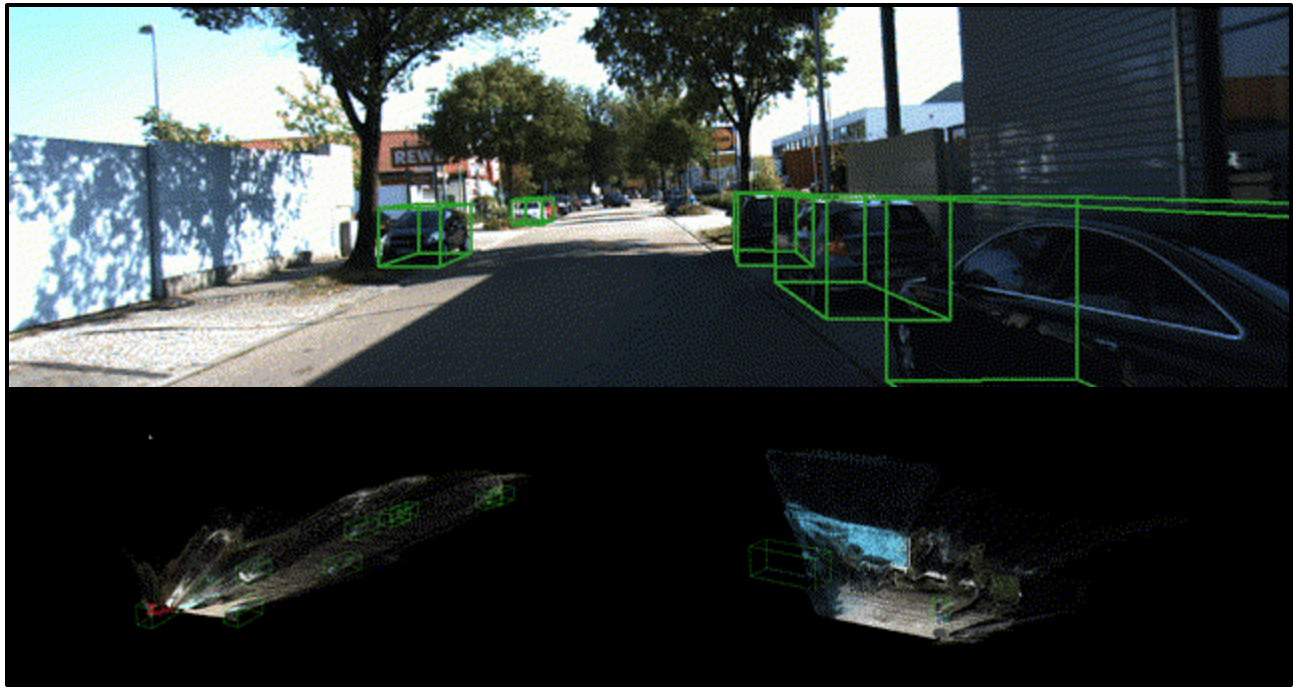
https://github.com/apple/ml-depth-pro

[13] Bochkovskii, A., Delaunoy, A., Germain, H., Santos, M., Zhou, Y., Richter, S. R., & Koltun, V. (2025). Depth Pro: Sharp Monocular Metric Depth in Less Than a Second. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2410.02073

Depth From Motion - Monocular Images (2023)

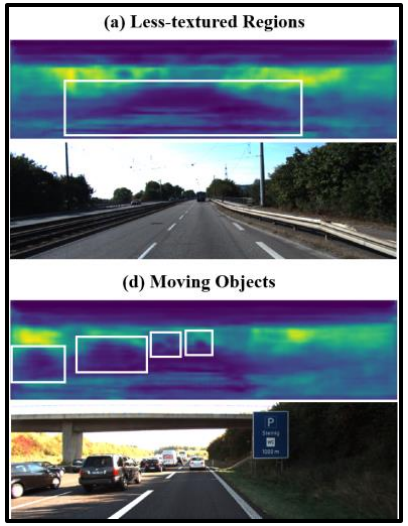
The paper *"Monocular 3D Object Detection with Depth from Motion"* [14] presents a technique for estimating depth by leveraging motion in a sequence of monocular images (i.e., images from a single moving camera).

The model estimates depth by analyzing the apparent motion of objects between consecutive frames. When the camera moves, objects closer to the camera appear to move faster across the image than those farther away, providing cues about their relative depth.



Uses CNN backbone

https://github.com/Tai-Wang/Depth-from-Motion



[14] Wang, T., Pang, J., & Lin, D. (2023). Monocular 3D Object Detection with Depth from Motion. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2207.12988

Deep Network Development

Lecture 11.

Deep Learning Tools For Computer Vision part 1

Budapest, 02nd May 2025

1 Annotation Tools

2 Depth Estimation

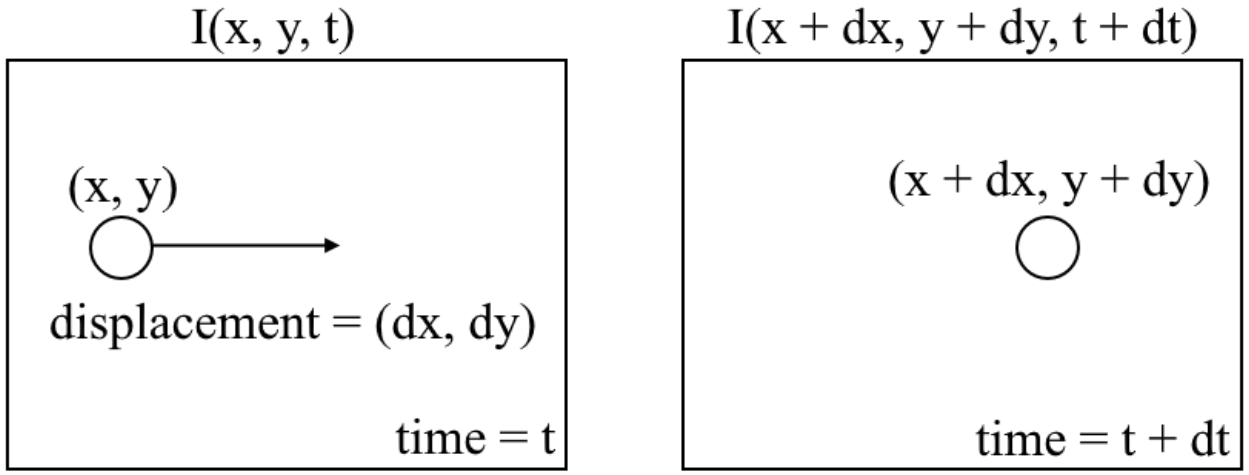
3 Optical Flow

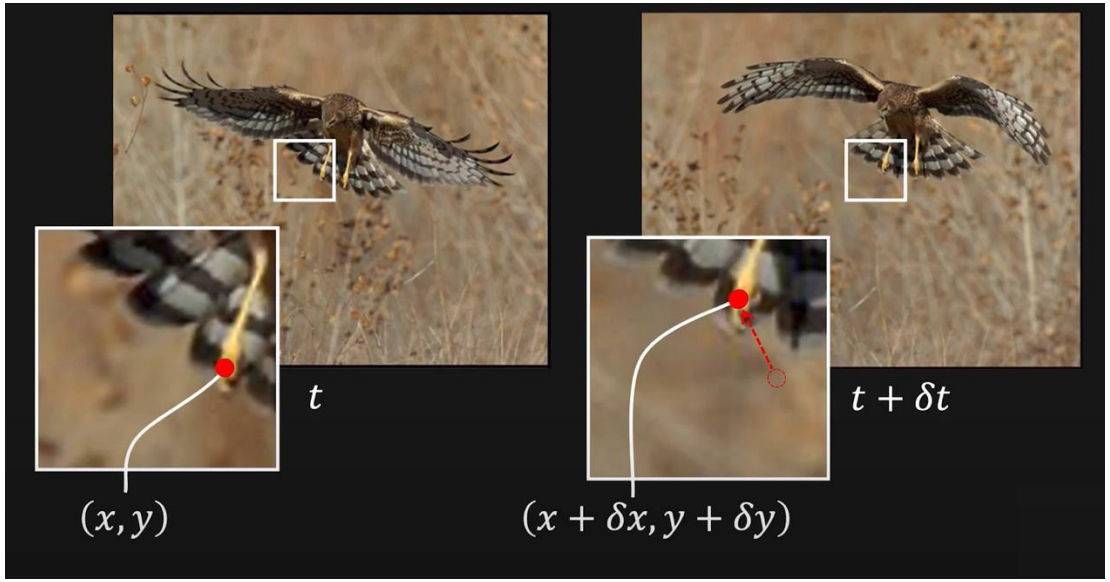
Optical Flow

Optical flow is the motion of objects between consecutive frames of sequence, caused by the relative movement between the object and camera.
Basically, it's about understanding how things are moving in an image.

Optical Flow

The problem of optical flow may be expressed as:





Where between consecutive frames, we can express the image intensity (I) as a function of space (x, y) and time (t). In other words, if we take the first image $I(x, y, t)$ and move its pixels by (dx, dy) over time t, we obtain the new image $I(x + dx, y + dy, t + dt)$.

Optical Flow

Optical flow is the apparent motion of brightness patterns in the image. Generally, optical flow corresponds to the motion field, but not always. For example, the motion field and optical flow of a rotating barber's pole are different:

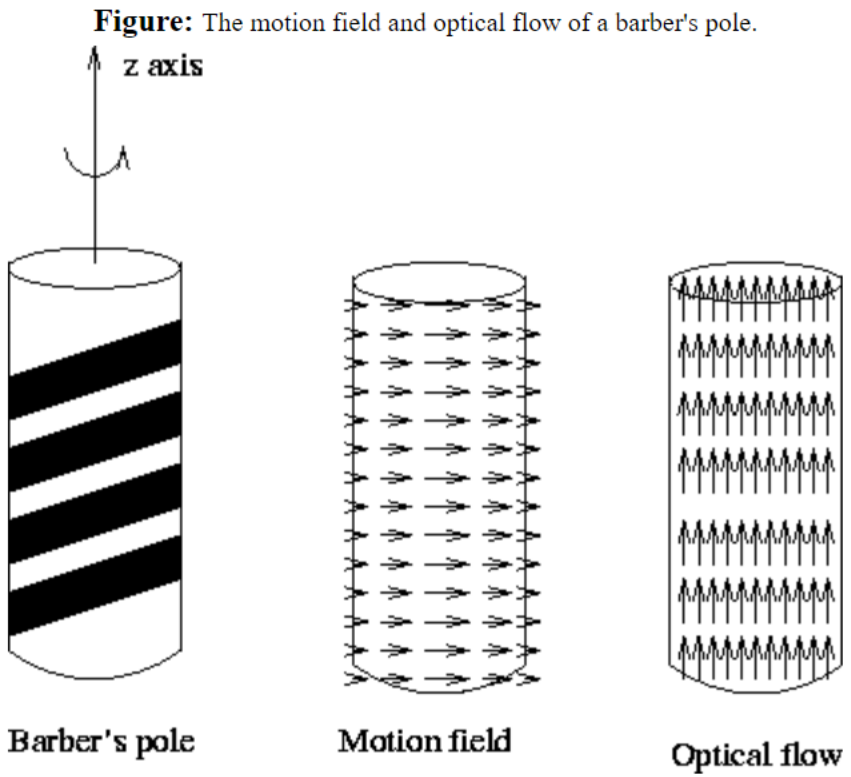
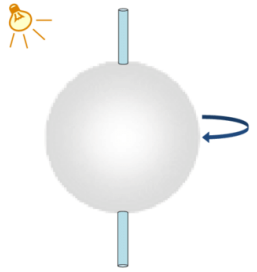


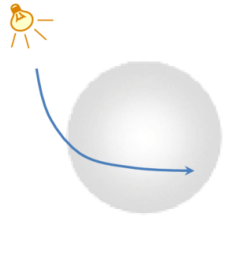
Figure: The motion field and optical flow of a barber's pole.

Optical flow is the **motion of each brightness pattern**. Conversely, the motion field represents the **motion of each pixel in the frame**.

In general, such cases are unusual, so assuming optical flow corresponds to the motion field is not unusual in literature.



A smooth sphere is rotating under constant illumination. Thus, the optical flow field is zero, but the motion field is not (no visible brightness change)



A fixed sphere is illuminated by a moving source and the shading of the image changes. Thus, the motion field is zero, but the optical flow field is not.

Optical Flow

Optical flow constraint equations describe the relationship between the motion of objects in a scene and the observed changes in image intensity over time. The most fundamental of these is the **Optical Flow Constraint Equation** (OFCE).

1. Brightness Constancy Assumption:

   This assumption states that the brightness (intensity) of a pixel remains constant over time as it moves across the image.

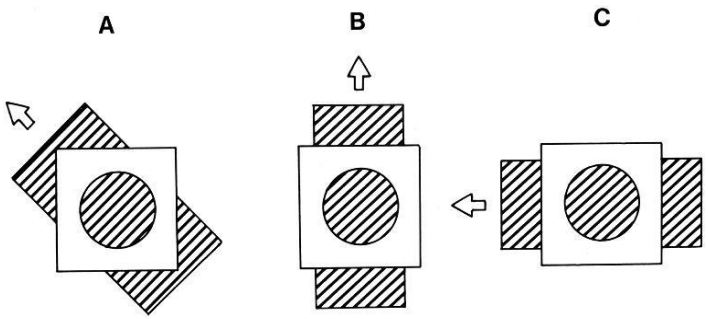
$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t)$$

2. Optical Flow Constraint Equation:

   By taking the first-order **Taylor expansion** of the brightness constancy assumption and assuming small movements, we get:

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{dy}v + \frac{\partial I}{\partial t} = 0$$

   It provides **one constraint per pixel but has two unknowns** (u and v, horizontal and vertical motion), leading to what is known as the **aperture problem**. (one eq., two unknowns -> infinite solutions)



3. Displacement Size:

   Displacement $(\partial x, \partial y)$ and time step ∂t are small.

Optical Flow – Lucas-Kanade Solution

A classic algorithm for optical flow estimation that mitigates the aperture problem by analyzing a local neighborhood of pixels to infer motion. It is based on the **Optical Flow Constraint Equation** (OFCE)

1. Instead of estimating motion at each pixel independently, the Lucas-Kanade method considers a small neighborhood (e.g., a 5x5 or 7x7 window) around each pixel. By assuming that all pixels within this window have the same motion vector (same u and v), the method gets multiple equations from this local patch, helping to constrain the problem.

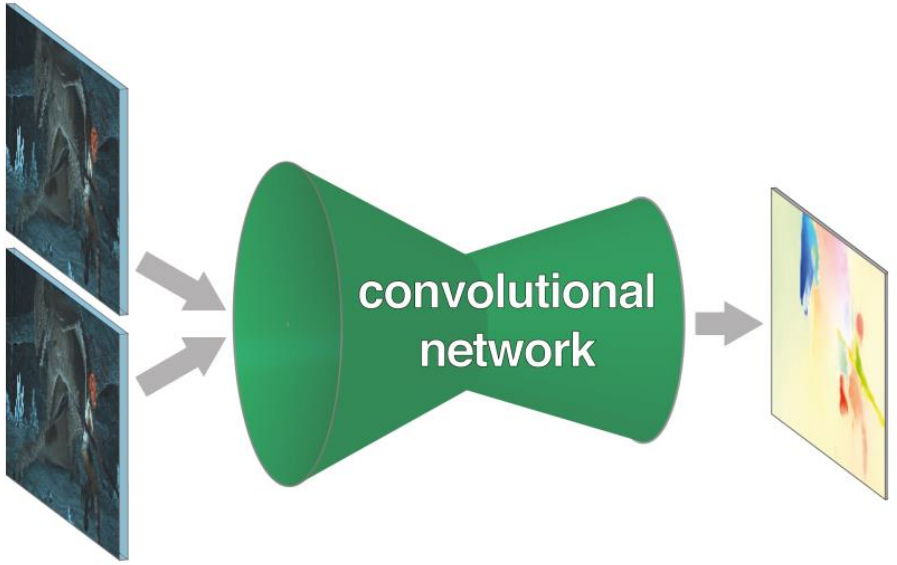
2. With multiple pixels in the neighborhood, the method builds a system of linear equations: $Av = b$, where A is a matrix of spatial gradients of all pixels in neighborhood, v is the optical flow vector we want to solve for and b contains the temporal gradients of each pixel.

3. If there are enough pixels in the neighborhood (more equations than unknowns), **the system can be solved using least squares to minimize the error.**

Optical Flow – FlowNet (2015) [15]

In the seminal paper *"FlowNet: Learning Optical Flow with Convolutional Networks"* , **FlowNet** introduces a CNN-based architecture designed to learn optical flow from synthetic data.

FlowNet has two main variants: **FlowNetS** and **FlowNetC**, each with unique approaches to processing image pairs.



[15] Fischer, P., Dosovitskiy, A., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., ... Brox, T. (2015). FlowNet: Learning Optical Flow with Convolutional Networks. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/1504.06852

Benchmark Datasets

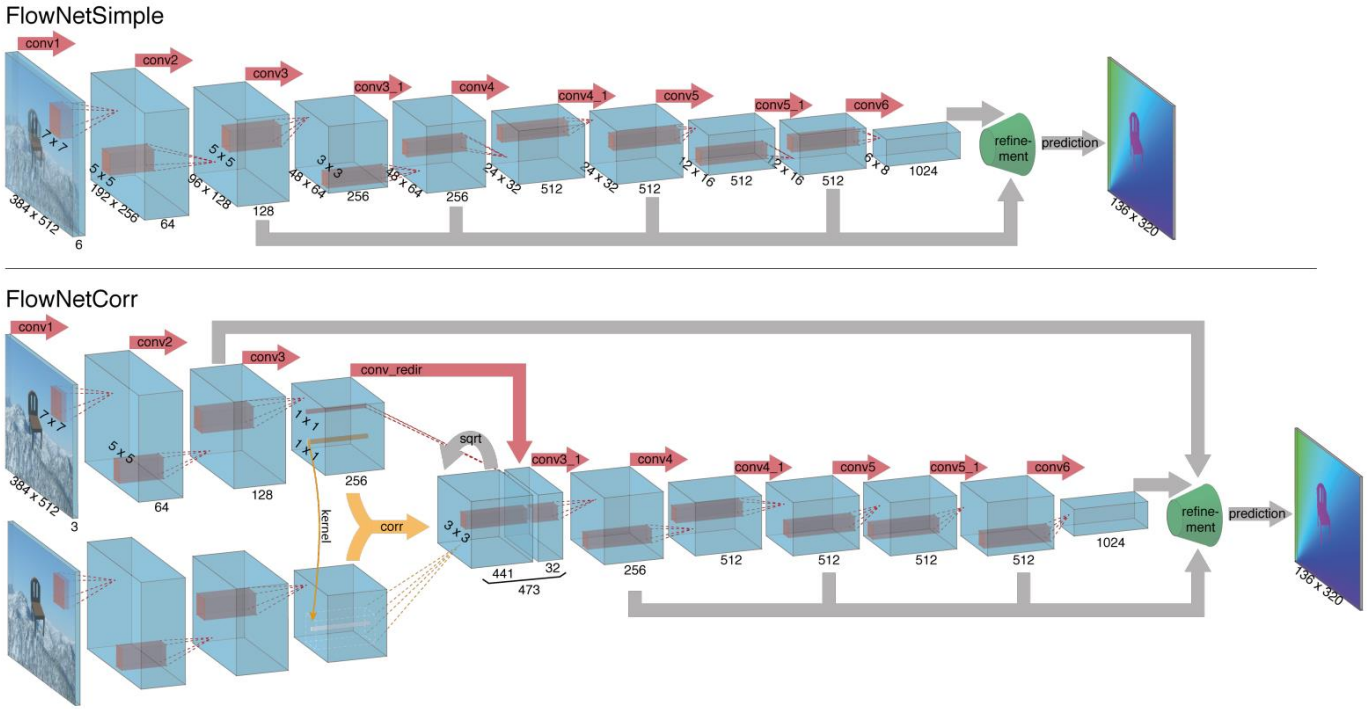
Sintel

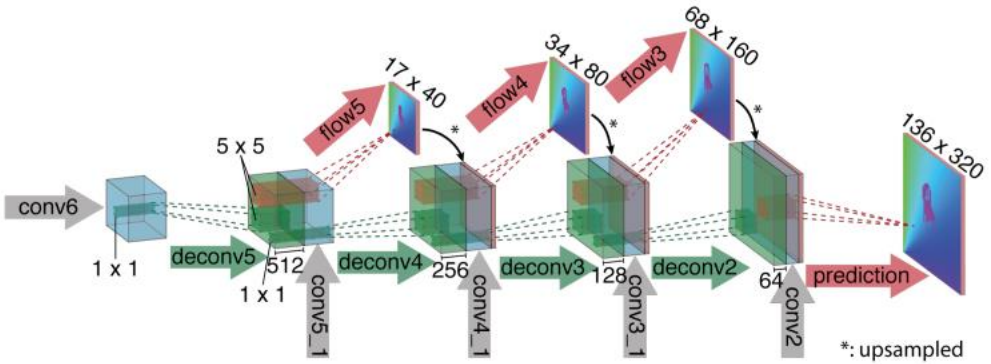
Kitti

Optical Flow – FlowNet (2015) [16]

For **FlowNetS**, The two input images (frames) are concatenated along the channel dimension, creating a 6-channel input. The concatenated image pair is passed through several convolutional layers to extract hierarchical features. The decoder upsamples the features back to the original image resolution. At each upsampling stage, the network produces intermediate flow estimates, which are refined as features are upsampled.

FlowNetC adds a correlation layer to compute explicit motion information between the two images

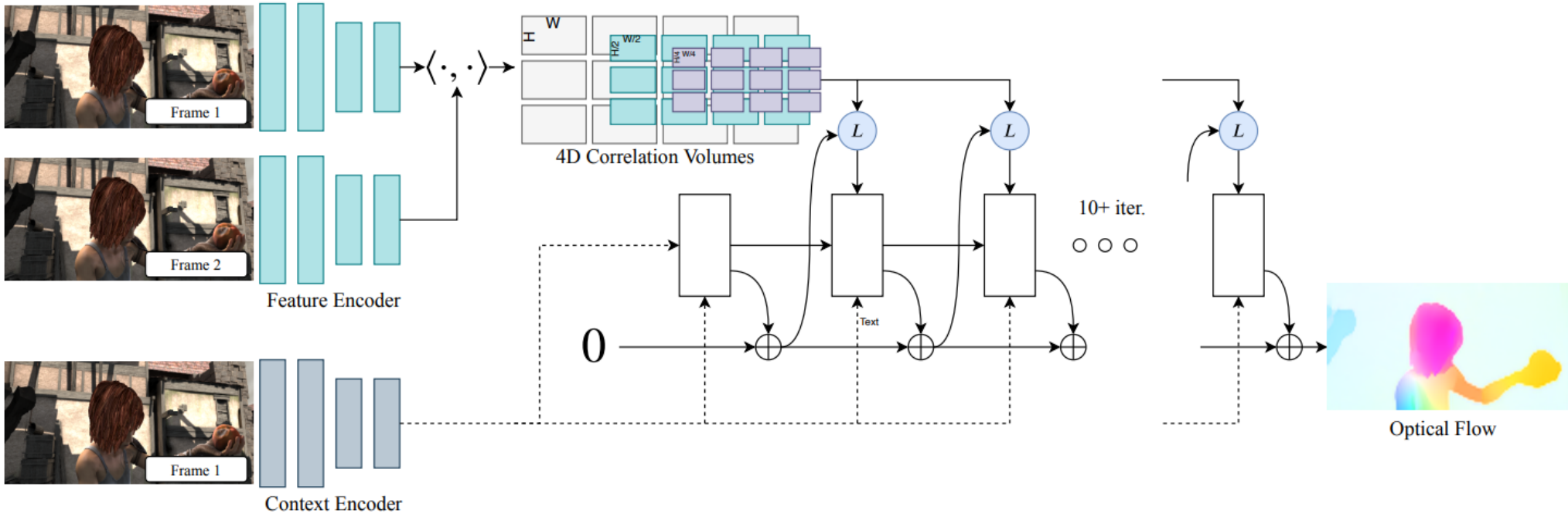




[16] Fischer, P., Dosovitskiy, A., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., ... Brox, T. (2015). FlowNet: Learning Optical Flow with Convolutional Networks. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/1504.06852

Optical Flow – RAFT (2020) [17]

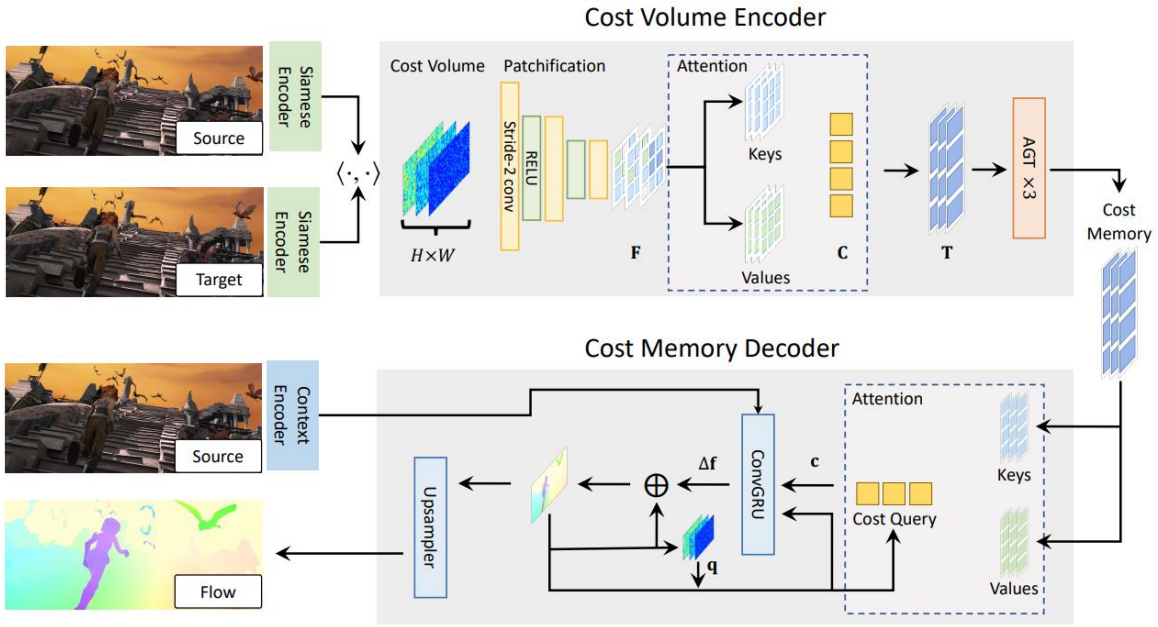
- **RAFT** combines CNN and RNN architectures and is still considered state-of-the-art.
- Like the previous architecture, it has two versions: **RAFT** and RAFT-S, with **RAFT-S** being a lightweight variant of the original RAFT model.
- **RAFT** consists of a feature encoder, a context encoder, a visual similarity module, and an iterative update module that utilizes RNNs.



[17] Teed, Z., & Deng, J. (2020). RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2003.12039

Optical Flow – FlowFormer (++) (2022-23) [18]

- **FlowFormer** introduces a transformer architecture into optical flow estimation and achieves state-of-the-art performance. The core component of **FlowFormer** is the transformer-based cost-volume encoder.
- Utilizes the success of **masked autoencoding (MAE) pretraining** in unleashing transformers' capacity of encoding visual representation
- **FlowFormer** tokenizes the 4D cost volume derived from image pairs and processes these tokens using alternate-group transformer layers to encode cost memory. A recurrent transformer decoder then refines the flow estimates using dynamic positional cost queries.



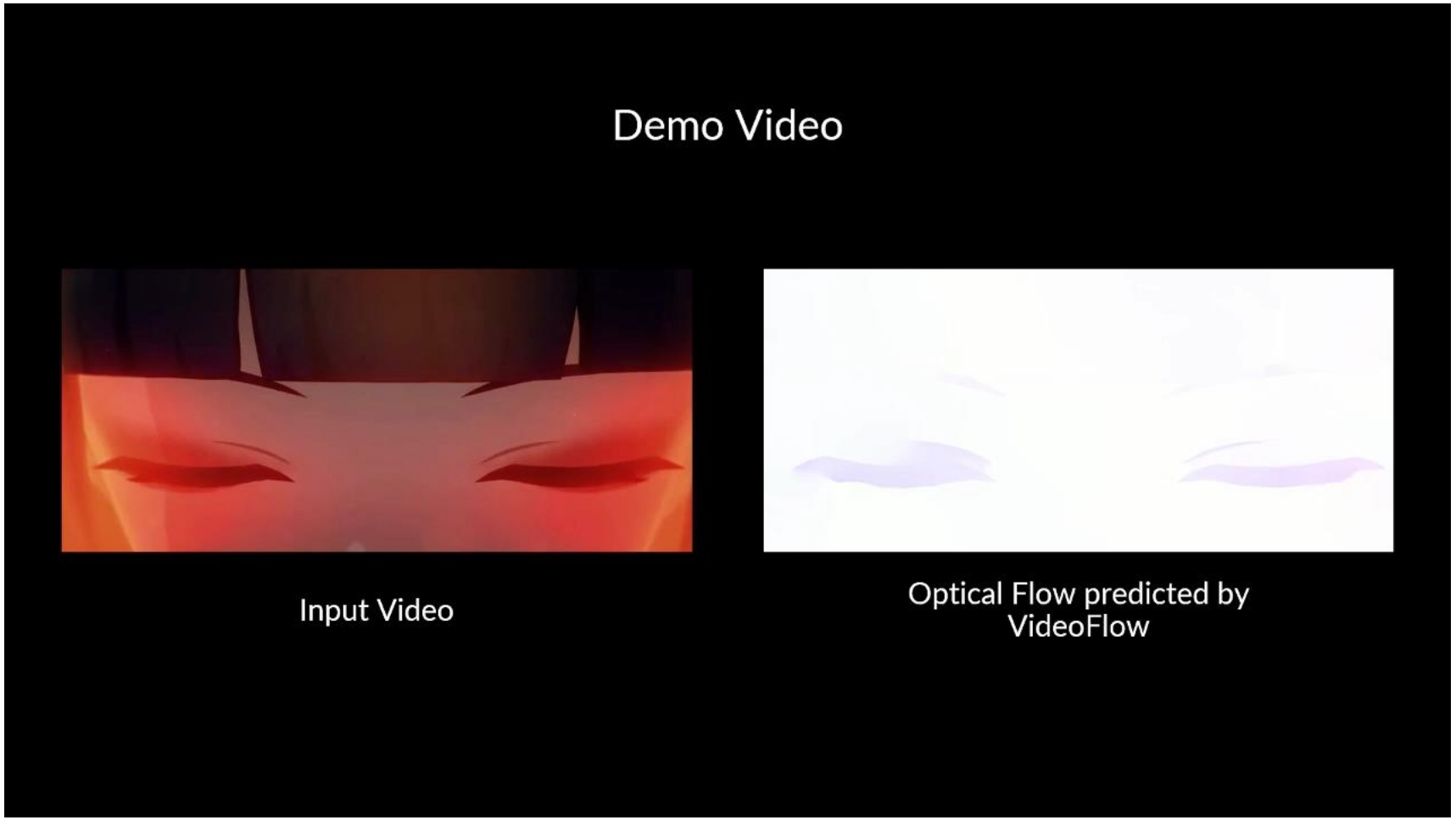
[18] Huang, Z., Shi, X., Zhang, C., Wang, Q., Cheung, K. C., Qin, H., ... Li, H. (2022). FlowFormer: A Transformer Architecture for Optical Flow. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2203.16194

Optical Flow – FlowFormer (++) (2022-23) [18]

(a) Input (b) FlowFormer (Ours)

[18] Huang, Z., Shi, X., Zhang, C., Wang, Q., Cheung, K. C., Qin, H., ... Li, H. (2022). FlowFormer: A Transformer Architecture for Optical Flow. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2203.16194

Optical Flow – VideoFlow (2023) [19]



[19] Shi, X., Huang, Z., Bian, W., Li, D., Zhang, M., Cheung, K. C., ... Li, H. (2023). VideoFlow: Exploiting Temporal Cues for Multi-frame Optical Flow Estimation. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2303.08340

Optical Flow – MambaFlow (2025) [20]

- **MambaFlow was designed to address the limitations of Transformer-based methods, particularly their high computational complexity**
- Utilizes the **Mamba** architecture, a state space model known for its linear computational complexity, **MambaFlow** achieves efficient and accurate motion estimation between consecutive video frames.
- Addresses occlusion challenges by effectively propagating flow information. Utilizes feature similarity to guide flow estimation in regions with missing data.

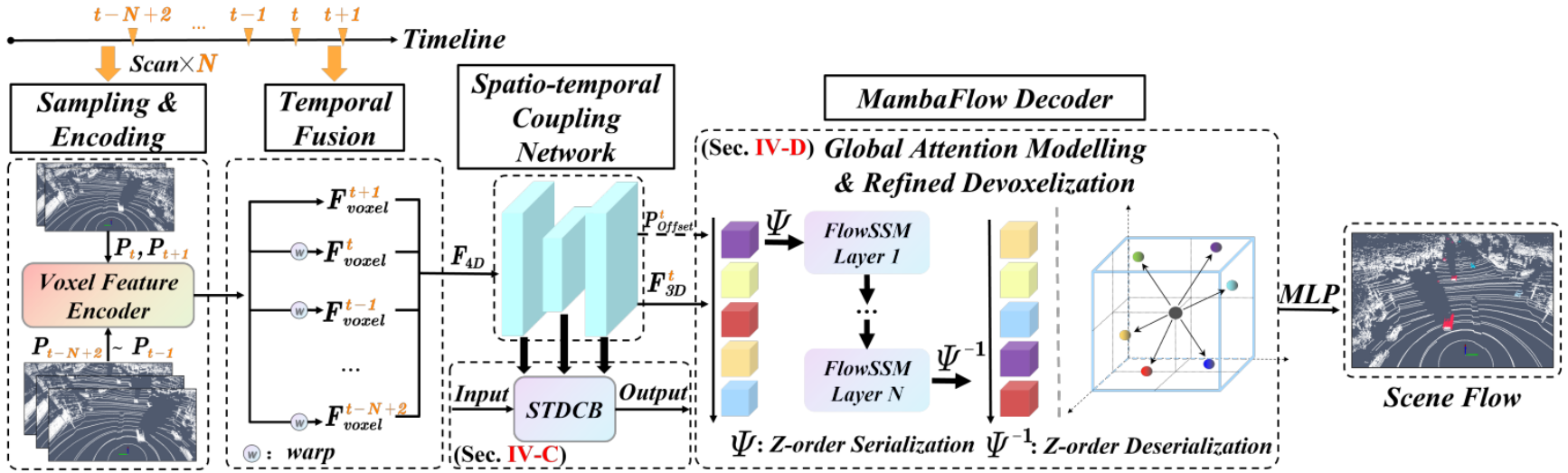
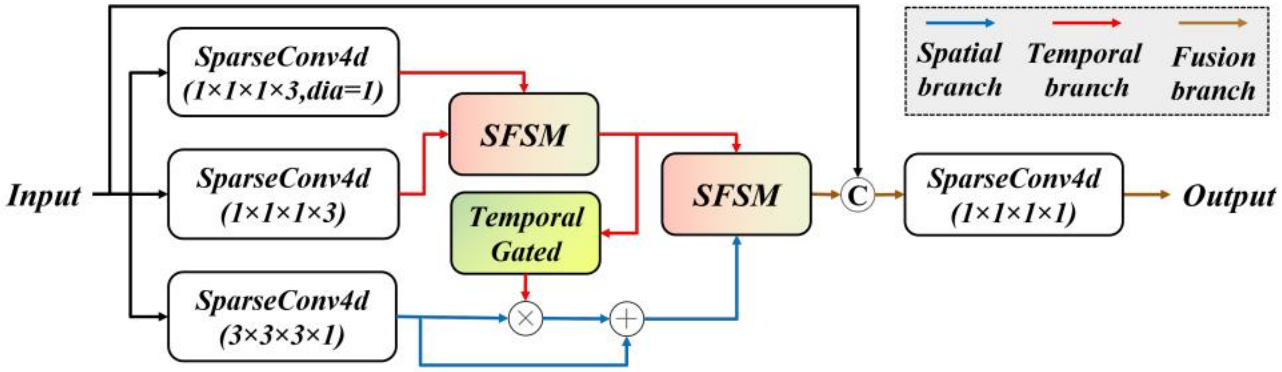


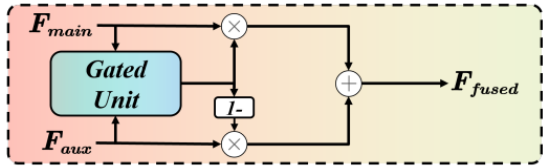
Fig. 2. **Overall architecture of MambaFlow.** The network first voxelizes and encodes five consecutive scans, forming 4D features by concatenating 3D voxel representations along the temporal dimension. These features are processed by our spatio-temporal coupling network for multi-scale feature learning. The decoder then learns voxel-to-point patterns through cascaded FlowSSM layers, enabling point-wise feature differentiation within the same voxel and generating the scene flow through an MLP layer.

[20] Luo, J., Cheng, J., Tang, X., Zhang, Q., Xue, B., & Fan, R. (2025). MambaFlow: A Novel and Flow-guided State Space Model for Scene Flow Estimation. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2502.16907

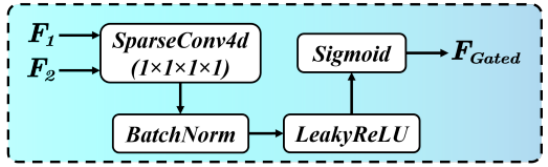
Optical Flow – MambaFlow (2025) [20]



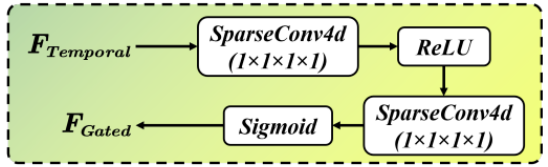
(b) Spatio-temporal Deep Coupling Block (Ours) (Sec. IV-B)



(c) Soft Feature Selection Mechanism (SFSM) (Sec. IV-B2)



(d) Gated Unit (Sec. IV-B2)



(e) Temporal Gated Block (Sec. IV-B3)

Fig. 3. **Architecture of the Spatio-temporal Deep Coupling Block.** (a) The baseline Spatio-temporal Decomposition Block [14] processes features through repeated convolutions at each stage. (b) Our proposed Spatio-temporal Deep Coupling Block achieves more efficient feature extraction by removing redundant convolutions modules and introducing a cross-timestep branch. The right panel (c)-(e) shows the detailed structures of Soft Feature Selection Mechanism and gating mechanisms used in Spatio-temporal Deep Coupling Block.

[20] Luo, J., Cheng, J., Tang, X., Zhang, Q., Xue, B., & Fan, R. (2025). MambaFlow: A Novel and Flow-guided State Space Model for Scene Flow Estimation. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2502.16907

Optical Flow – MambaFlow (2025) [20]

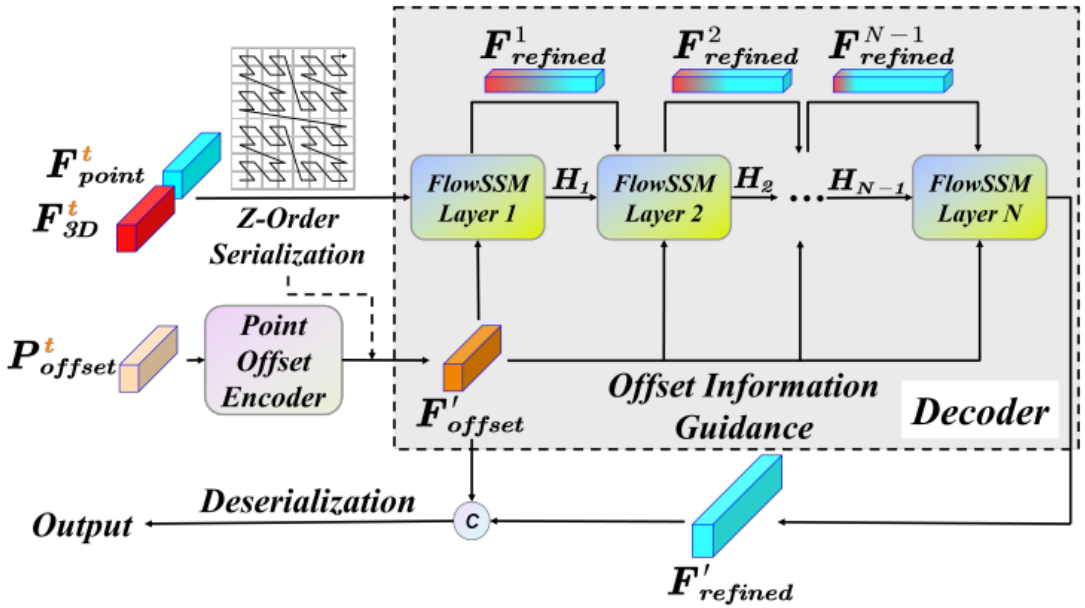


Fig. 4. **MambaFlow Decoder architecture.** Point-wise features and voxel features are first serialized through Z-order space-filling curves for spatial proximity preservation. The FlowSSM module consists of N cascaded FlowSSM layers, where point offset features guide the learning of voxel-to-point patterns in each layer for refined feature reconstruction. The final output is obtained through deserialization and feature fusion with point offset information.

[20] Luo, J., Cheng, J., Tang, X., Zhang, Q., Xue, B., & Fan, R. (2025). MambaFlow: A Novel and Flow-guided State Space Model for Scene Flow Estimation. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2502.16907

Optical Flow – MambaFlow (2025) [20]

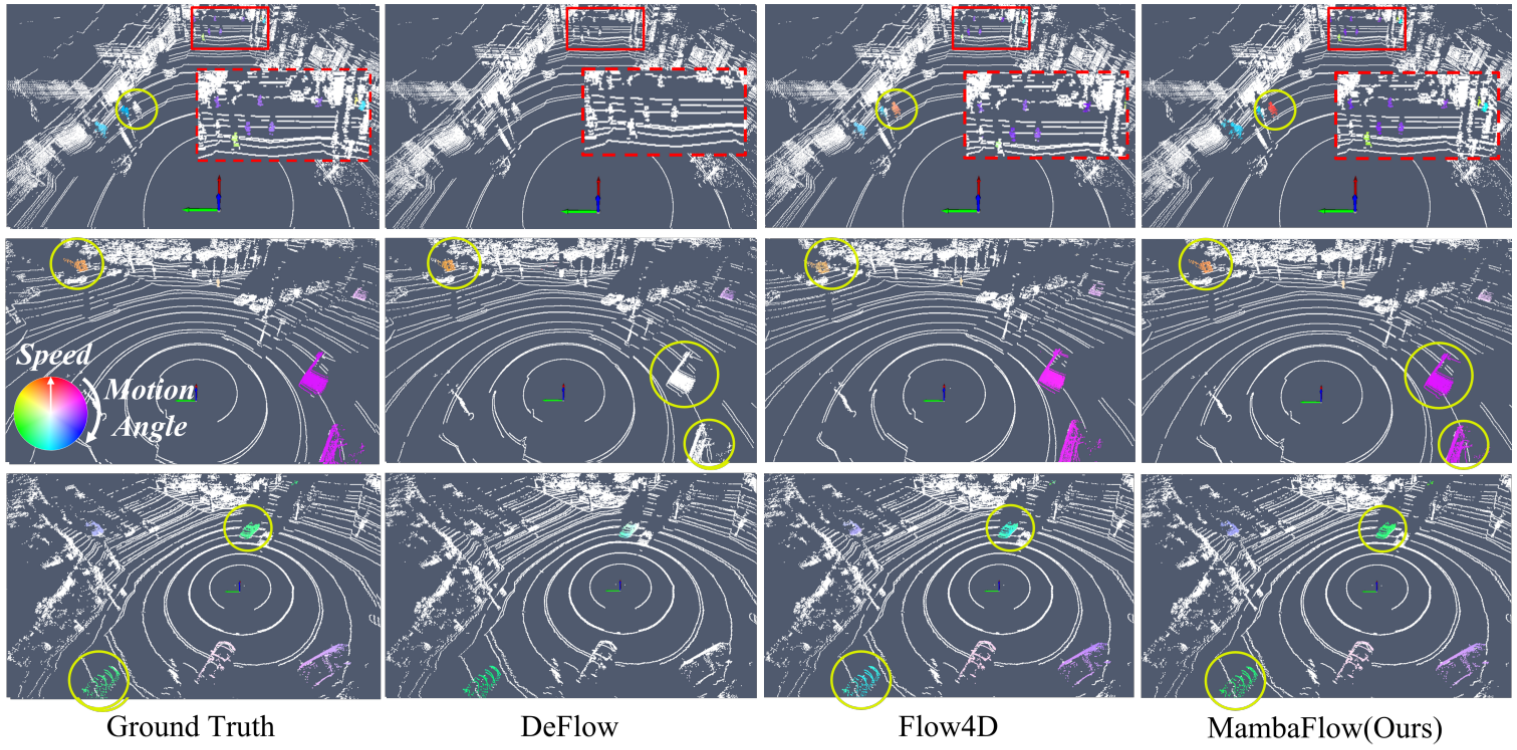


Fig. 5. **Qualitative results on the Argoverse 2 validation set.** From left to right: Ground Truth, DeFlow, Flow4D, and our proposed MambaFlow. The color legend indicates both speed (shown by color intensity) and motion angle (2D), aligned with the vehicle's forward direction. The highlighted regions (yellow circles) demonstrate our method's superior performance in capturing both static and dynamic object motions, especially for challenging cases with complex motion patterns.

[20] Luo, J., Cheng, J., Tang, X., Zhang, Q., Xue, B., & Fan, R. (2025). MambaFlow: A Novel and Flow-guided State Space Model for Scene Flow Estimation. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2502.16907

Summary

- Various **model-assisted annotation tools** can aid in the manual labeling of extensive datasets.
  - Tools like **CVAT** offer free services for individuals and open-source research, while automatic annotation tools can handle a wide range of data types, including images, **MRI**, and **DICOM** formats.

- In deep learning, **depth estimation** is the process of predicting the distance of objects from the camera using a neural network.
  - **Non-linear estimators** are important because they can replace expensive hardware with advanced software solutions that perform similarly.
  - **Stereo depth estimation** has long relied on epipolar geometry from stereo images, while **monocular** estimation offers a cheaper and more compact solution by leveraging the **power of CNNs and Transformers**.

- **Optical flow** refers to the motion of objects between consecutive frames in a sequence, resulting from relative movement between the object and the camera.
  - The **Lucas-Kanade** method provides a solution to the aperture problem (where there are more unknowns than constraints).
  - Neural networks, like **FlowNet**, **RAFT**, **FlowFormer** and **MambaFlow** can be used as approximators for flow vectors using **CNNs**.

Resources

Books:

- Courville, Goodfellow, Bengio: Deep Learning
  Freely available: https://www.deeplearningbook.org/
- Zhang, Aston and Lipton, Zachary C. and Li, Mu and Smola, Alexander J.: Dive into Deep Learning
  Freely available: https://d2l.ai/

Courses:

- Deep Learning specialization by Andrew NG
- https://www.coursera.org/specializations/deep-learning

Further Links + Resources

- The State of the Art of Depth Estimation from Single Images - https://medium.com/@patriciogv/the-state-of-the-art-of-depth-estimation-from-single-images-9e245d51a315
- How to Estimate Depth from a Single Image - https://towardsdatascience.com/how-to-estimate-depth-from-a-single-image-7f421d86b22d
- Monocular Depth Estimation using ZoeDepth : Our Experience - https://medium.com/@bhaskarbose1998/monocular-depth-estimation-using-zoedepth-our-experience-42fa5974cb59
- Epipolar Geometry - https://amroamroamro.github.io/mexopencv/opencv/epipolar_geometry_demo.html
- Optical Flow - https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT12/node4.html
- RAFT: Optical Flow estimation using Deep Learning - https://learnopencv.com/optical-flow-using-deep-learning-raft
- FlowFormer: A Transformer Architecture for Optical Flow - https://github.com/drinkingcoder/FlowFormer-Official
- Depth Pro: Sharp Monocular Metric Depth in Less Than a Second - https://github.com/apple/ml-depth-pro?tab=readme-ov-file
- Dept Anything V2 - https://github.com/DepthAnything/Depth-Anything-V2

That’s all for today!