Geometry in the Deep Learning Era

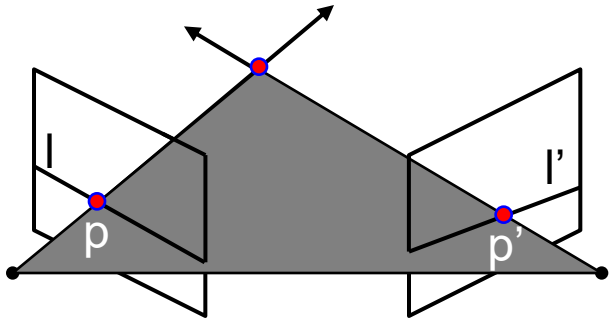
Agenda

- Quiz 2 recap
- Multi-view geometry recap
- “Classical” Stereo
- Dust3r
- VGGT
- DepthAnything V3

Quiz 2

Fundamental matrix

Let *p* be a point in left image, *p'* in right image



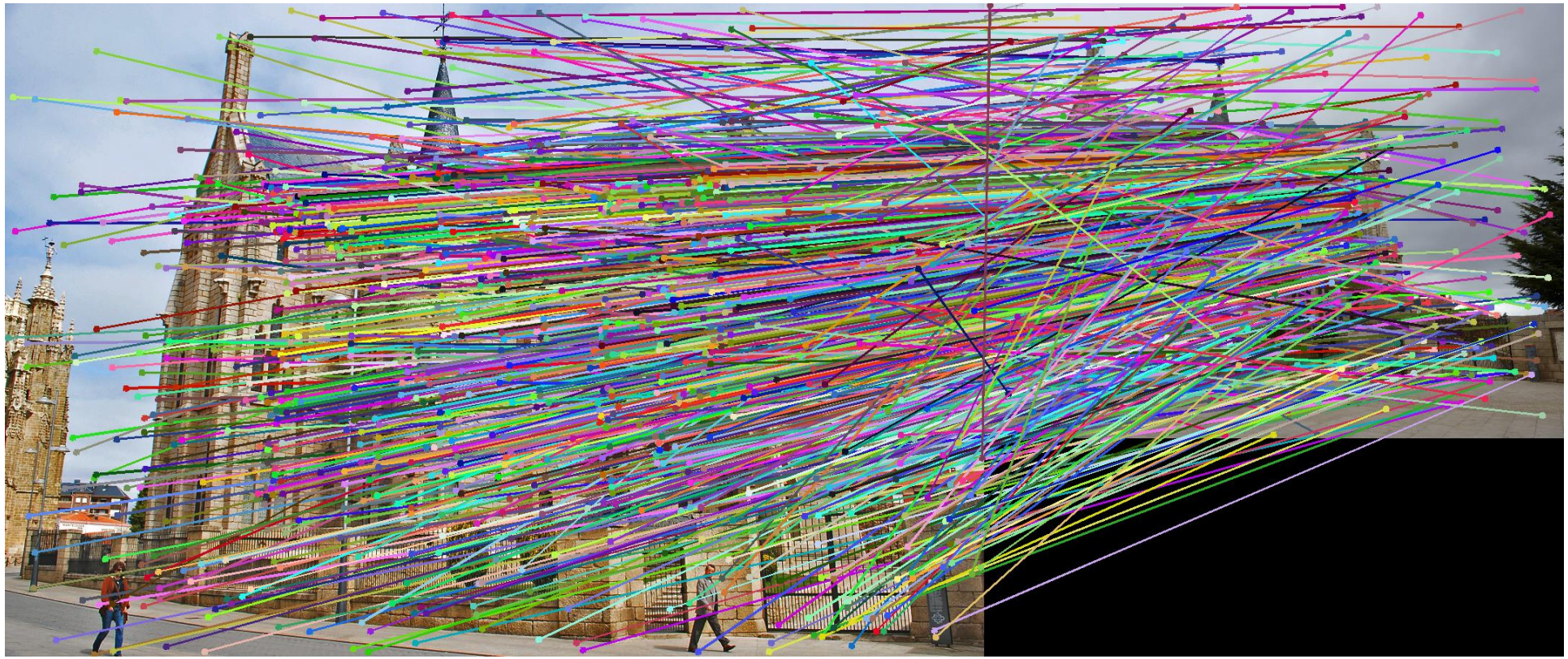
Epipolar relation

- *p* maps to epipolar line *l'*
- *p'* maps to epipolar line *l*

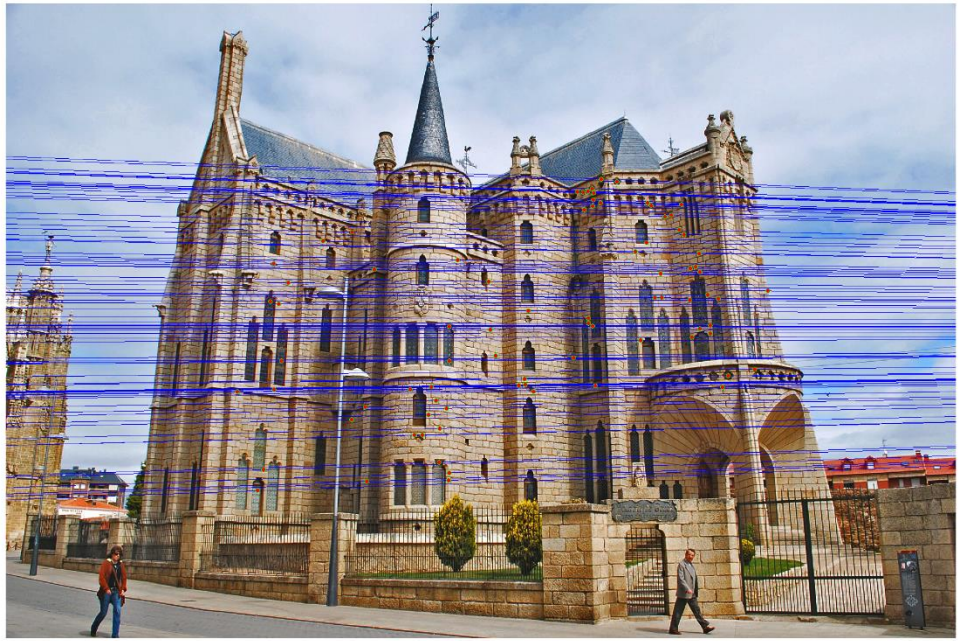
Epipolar mapping described by a 3x3 matrix *F*

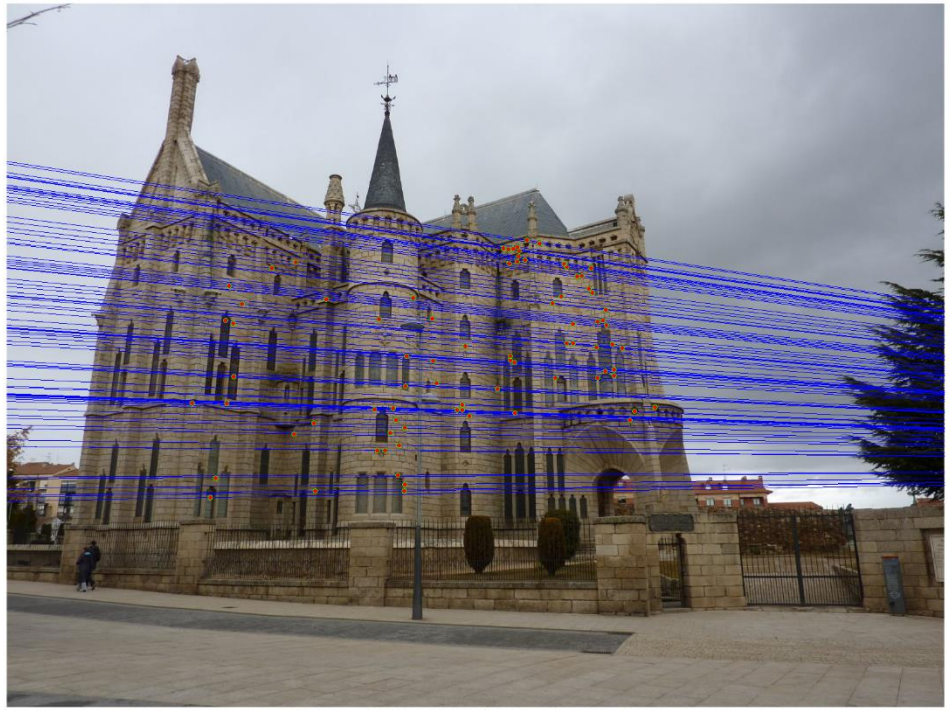
$$p'^T F p = 0$$

How to test for outliers?

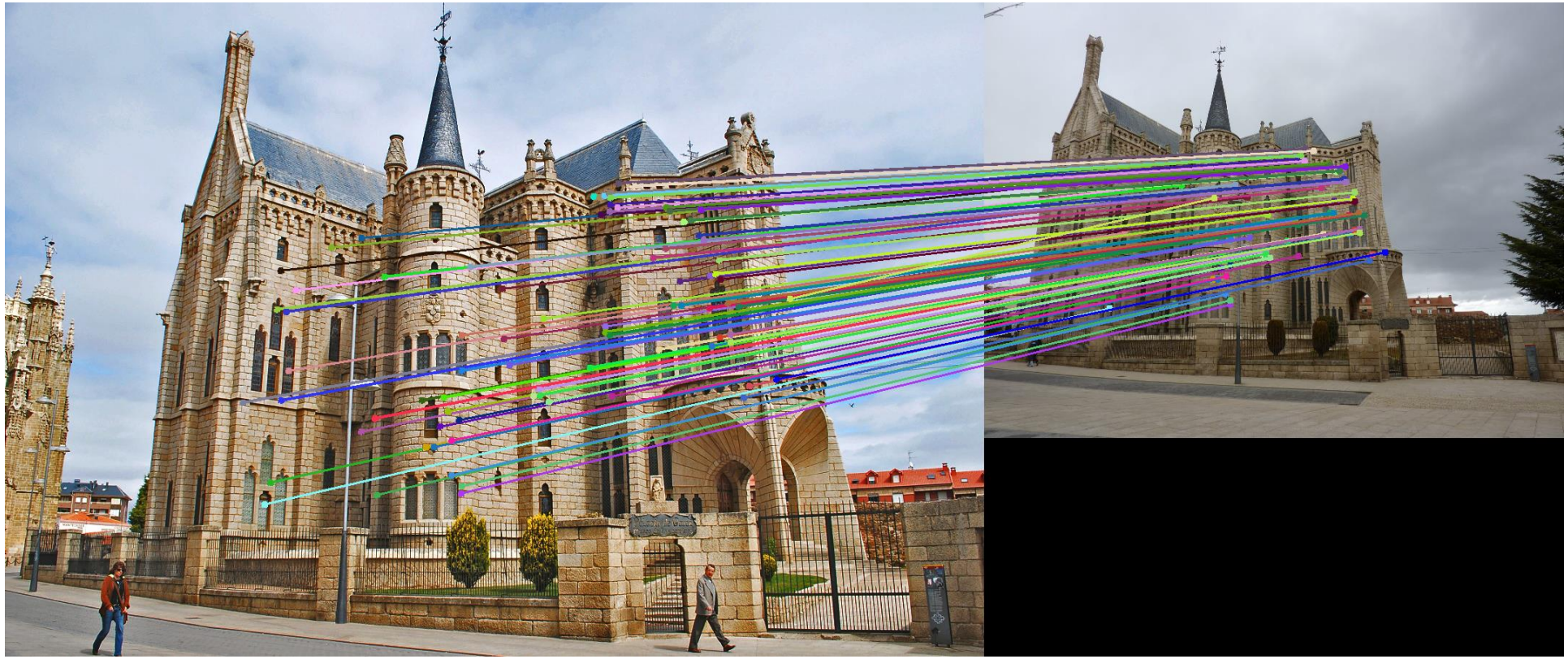


Epipolar lines





Keep only the matches at are “inliers” with respect to the “best” fundamental matrix

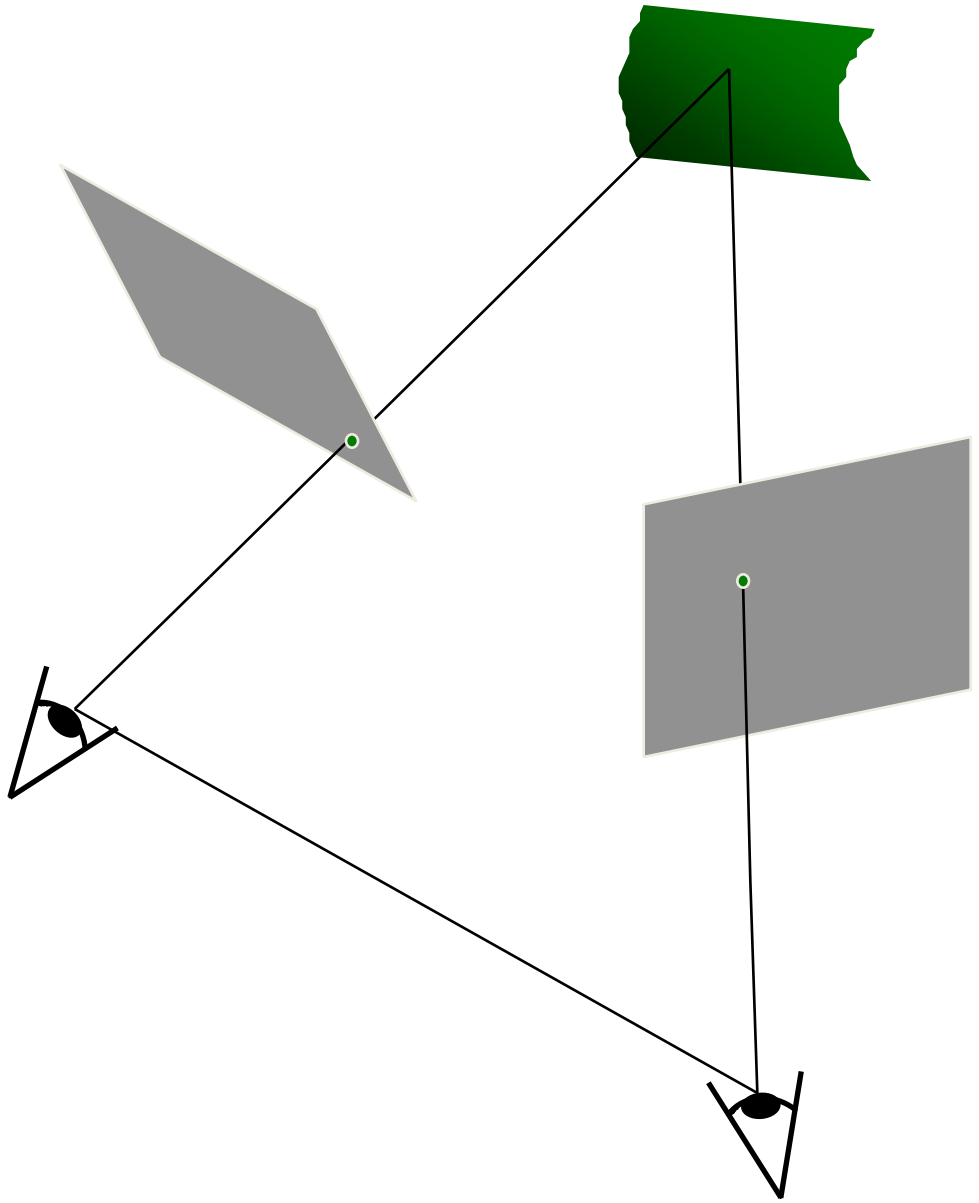


“Classical” 3D from two images

- Depth Estimation from Stereo Matching
- Keypoint matching and structure-from-motion gave us *sparse* matches.
- Stereo / Multi-view Stereo gives us *dense* correspondences and depths.

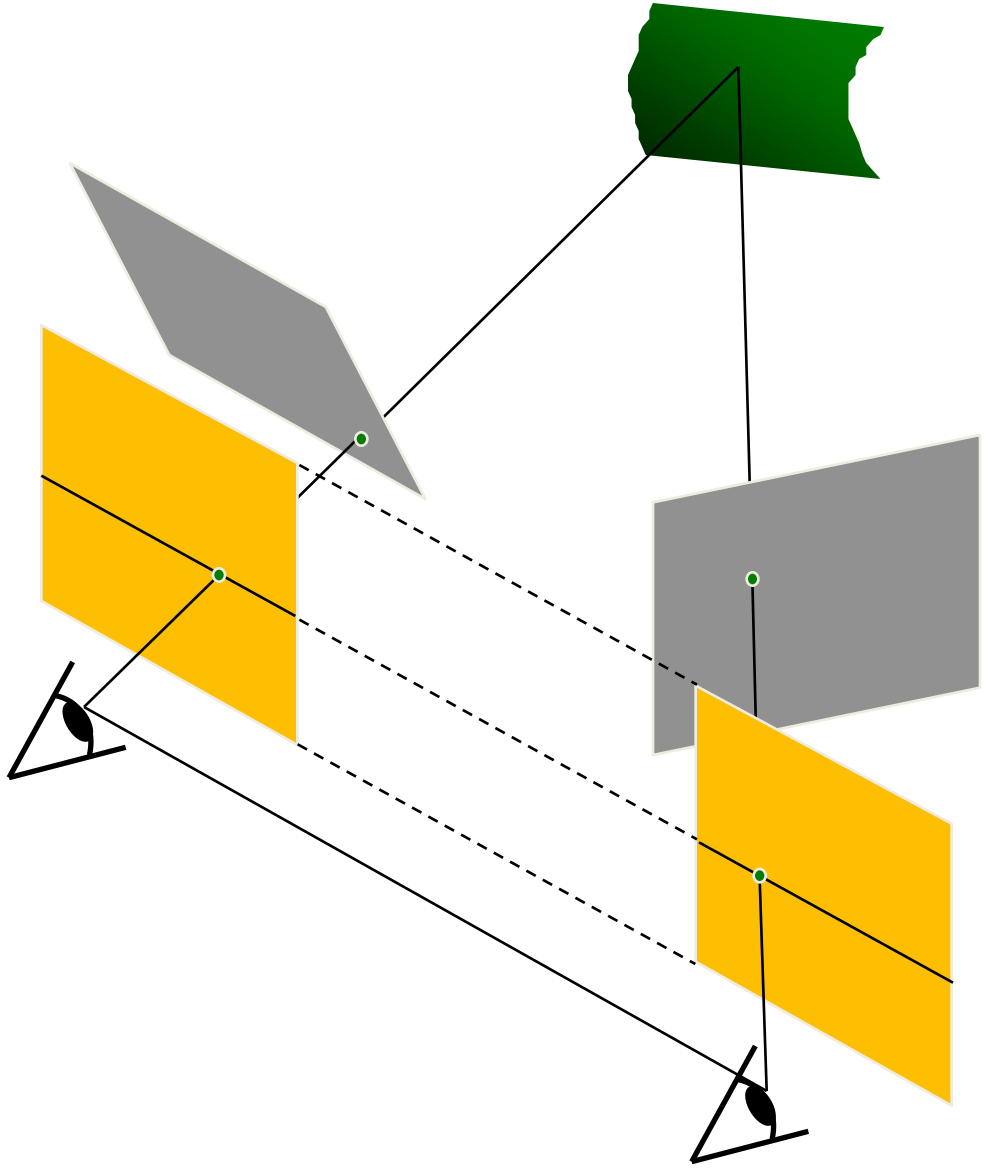
Stereo Matching

Stereo image rectification



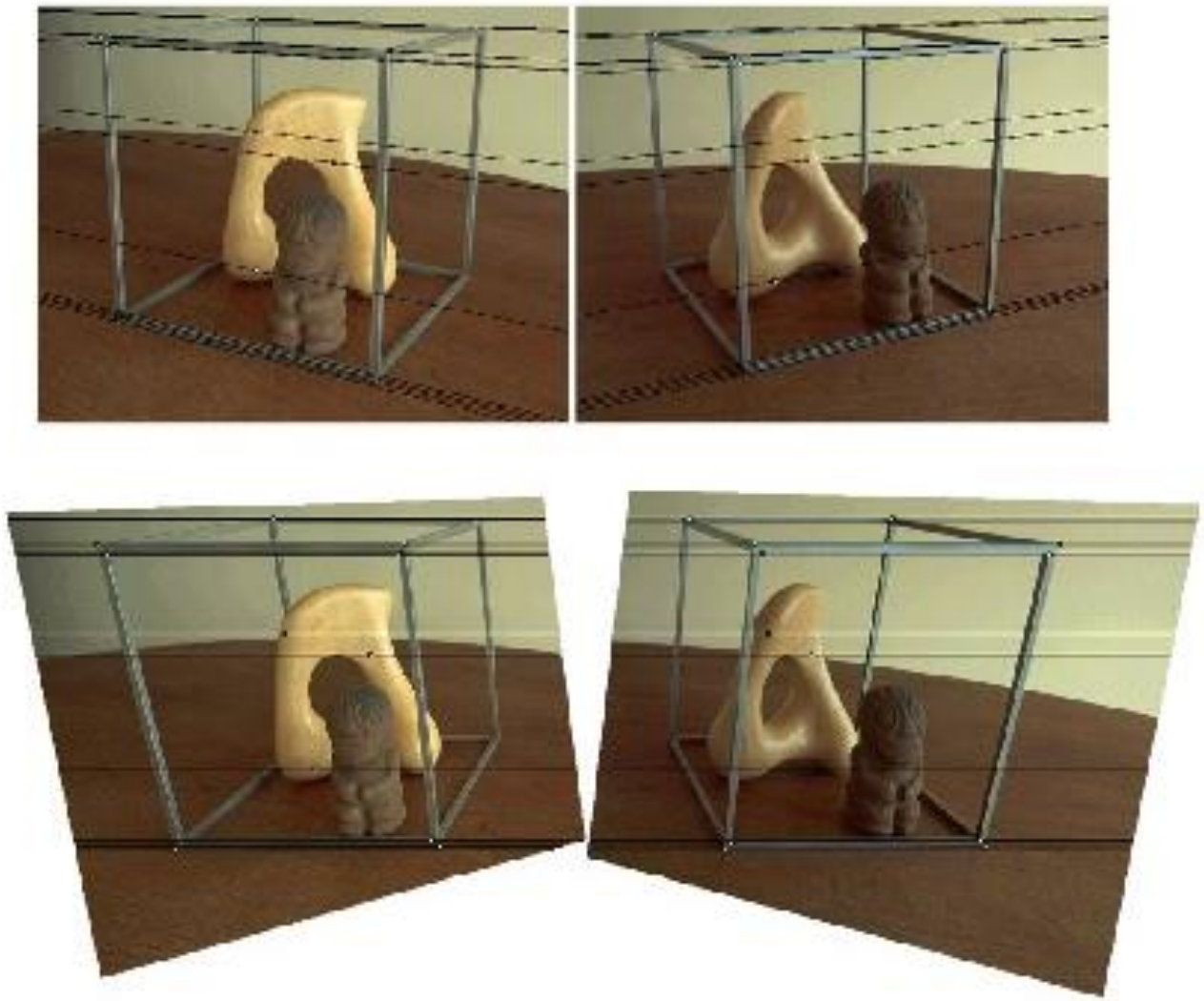
Stereo image rectification

- Reproject image planes onto a common plane parallel to the line between camera centers
- Pixel motion is horizontal after this transformation
- Two homographies (3x3 transform), one for each input image reprojection



➢ C. Loop and Z. Zhang. Computing Rectifying Homographies for Stereo Vision. IEEE Conf. Computer Vision and Pattern Recognition, 1999.

Rectification example

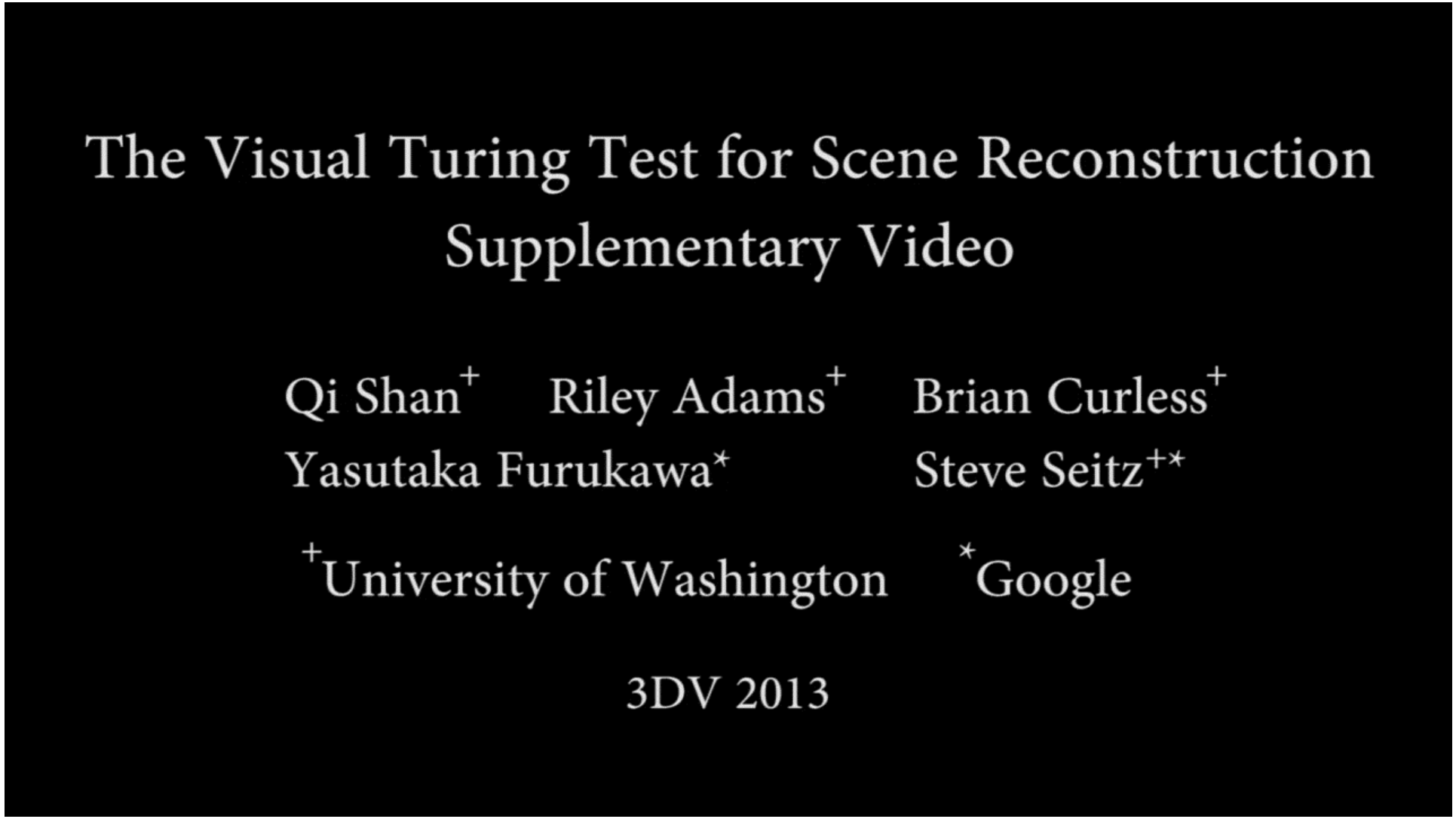


The correspondence problem

- Epipolar geometry constrains our search, but we still have a difficult correspondence problem.

Fundamental Matrix + Sparse correspondence

Fundamental Matrix + Dense correspondence



SIFT + Fundamental Matrix + RANSAC

Despite their scale invariance and robustness to appearance changes, SIFT features are *local* and do not contain any global information about the image or about the location of other features in the image. Thus feature matching based on SIFT features is still prone to errors. However, since we assume that we are dealing with rigid scenes, there are strong geometric constraints on the locations of the matching features and these constraints can be used to clean up the matches. In particular, when a rigid scene is imaged by two pinhole cameras, there exists a 3×3 matrix F, the *Fundamental matrix*, such that corresponding points x_{ij} and x_{ik} (represented in homogeneous coordinates) in two images j and k satisfy[10]:

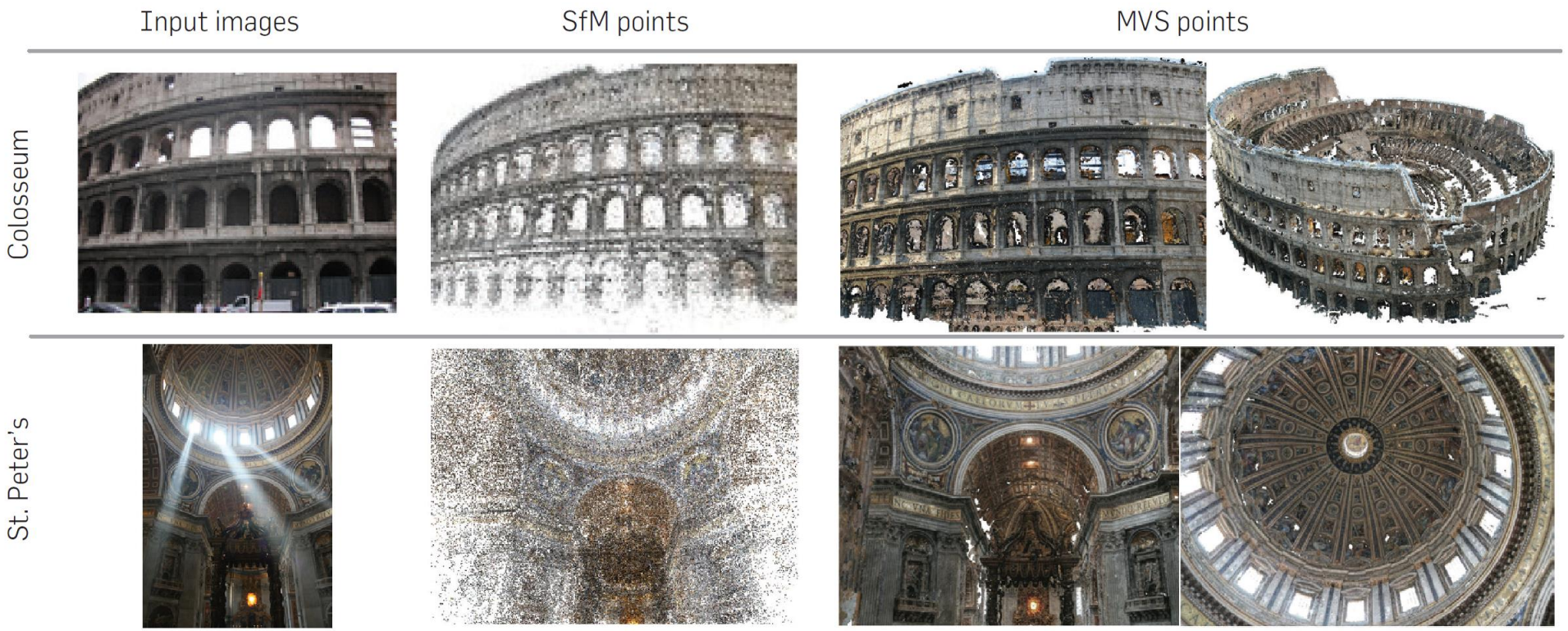
$$x_{ij}^{\top} F x_{ij} = 0. \quad (3)$$

A common way to impose this constraint is to use a greedy randomized algorithm to generate suitably chosen random estimates of F and choose the one that has the largest support among the matches, i.e., the one for which the most matches satisfy (3). This algorithm is called Random Sample Consensus (RANSAC)[6] and is used in many computer vision problems.

Building Rome in a Day

By Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, Richard Szeliski
Communications of the ACM, Vol. 54 No. 10, Pages 105-112

Sparse to Dense Correspodence

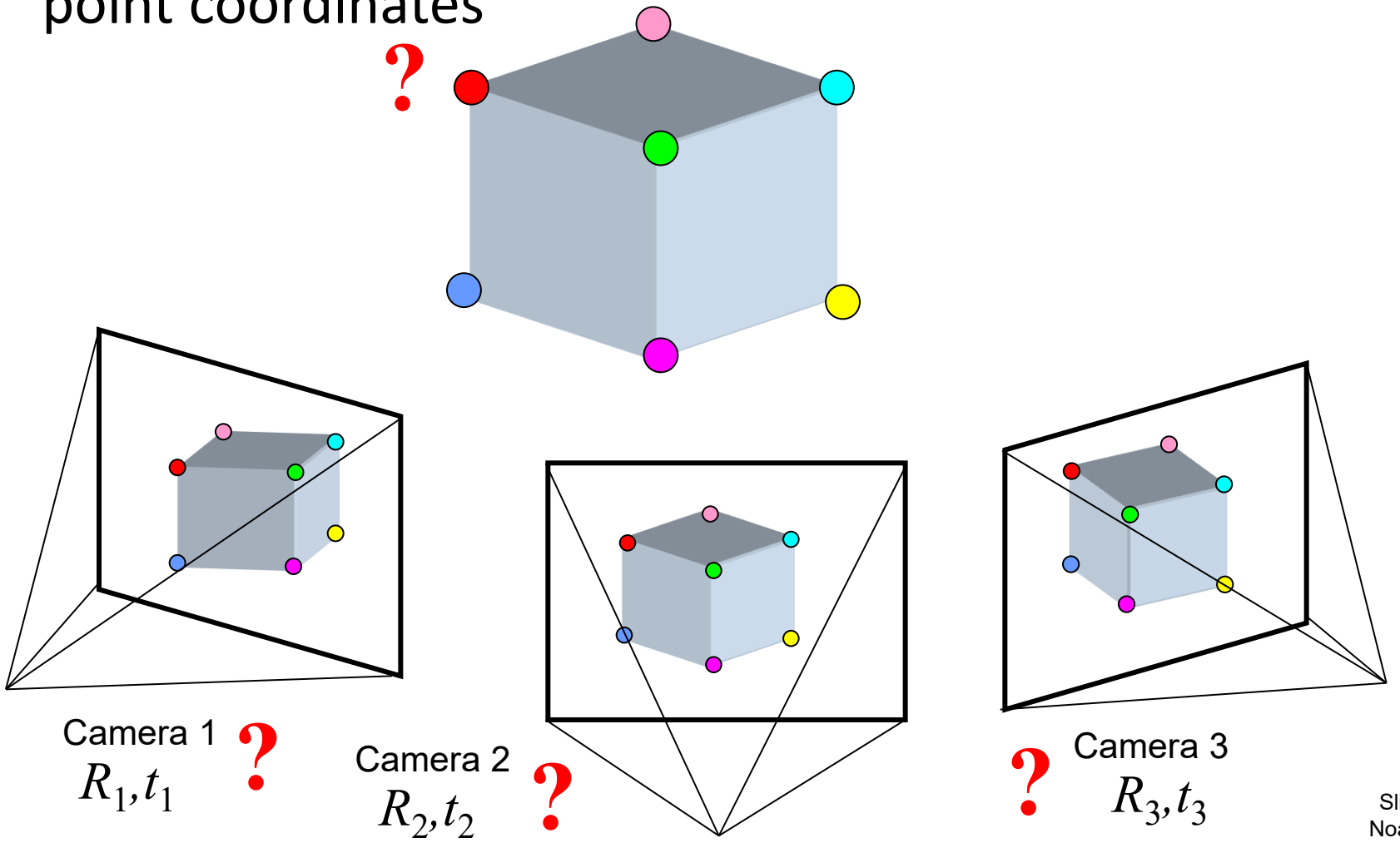


Building Rome in a Day

By Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, Richard Szeliski
Communications of the ACM, Vol. 54 No. 10, Pages 105-112

Structure from motion (or SLAM)

- Given a set of corresponding points in two or more images, compute the camera parameters and the 3D point coordinates

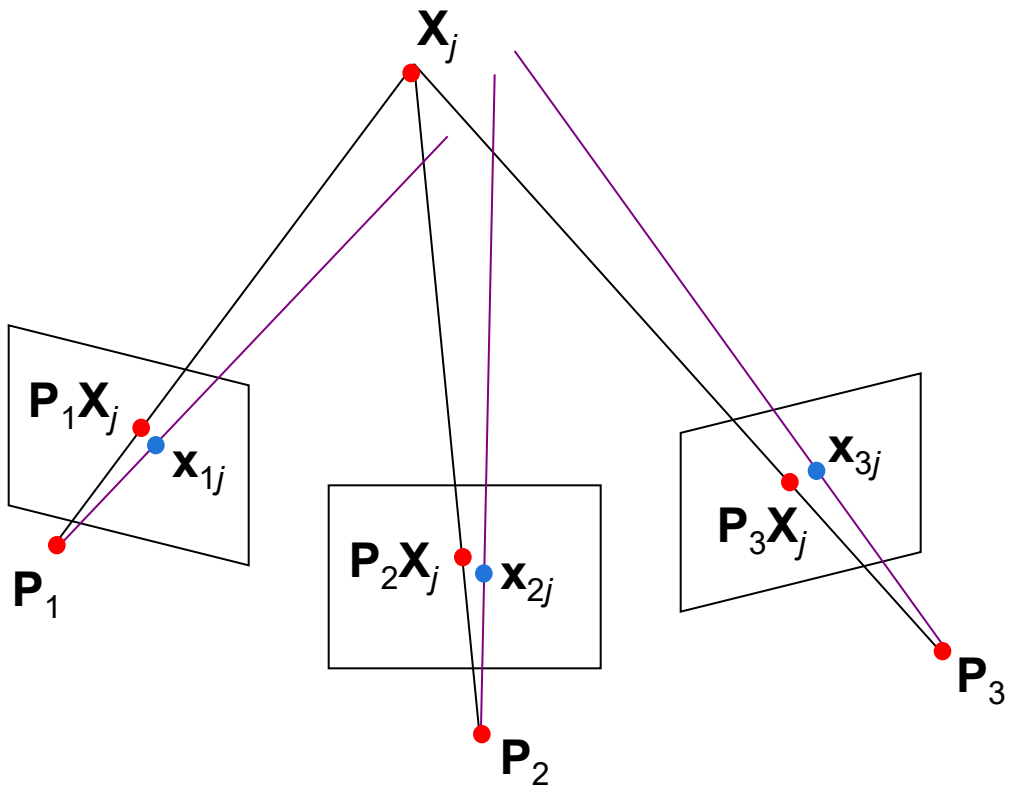


Slide credit:
Noah Snavely

Bundle adjustment – the core optimization problem inside classical Structure-from-Motion

- Non-linear method for refining structure and motion
- Minimizing reprojection error

$$E(\mathbf{P},\mathbf{X}) = \sum_{i=1}^{m}\sum_{j=1}^{n} D\left(\mathbf{x}_{ij}, \mathbf{P}_i\mathbf{X}_j\right)^2$$



How do we get dense stereo correspondences?

Correspondence problem

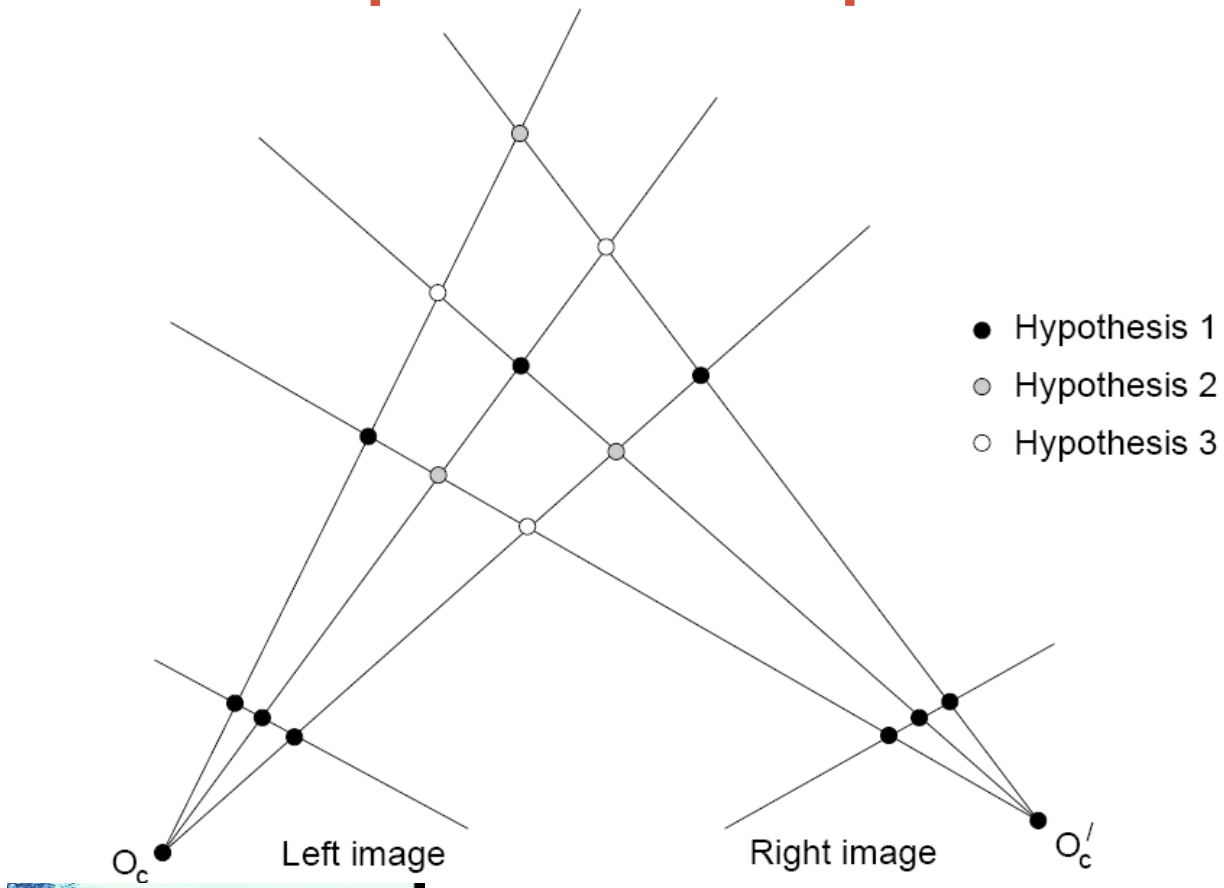


Figure from Gee & Cipolla 1999

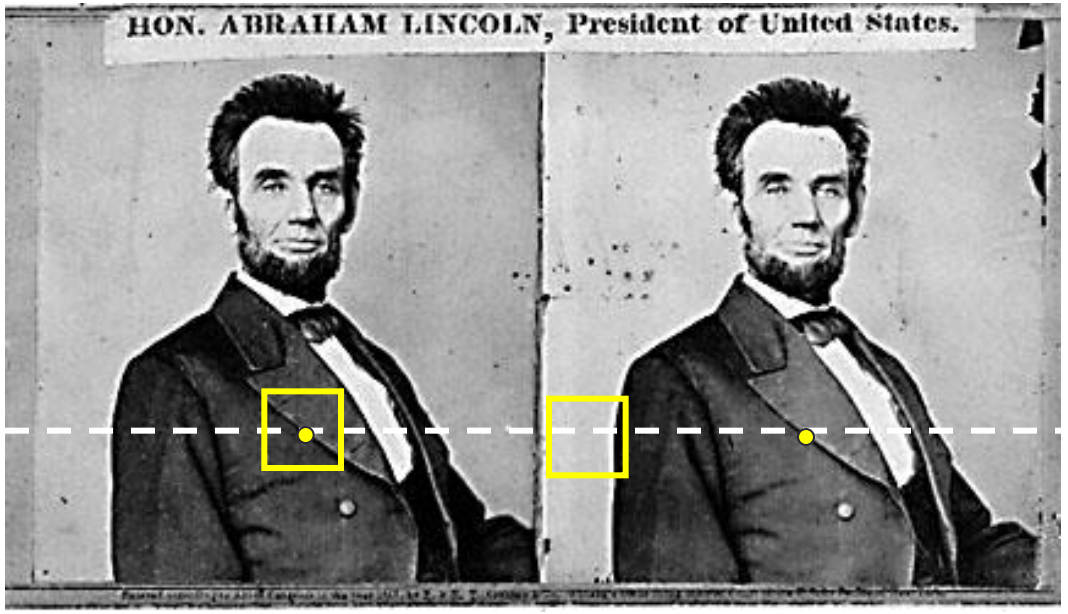
Multiple match hypotheses satisfy epipolar constraint, but which is correct?

Correspondence problem

- Beyond the hard constraint of epipolar geometry, there are “soft” constraints to help identify corresponding points
  - Similarity
  - Uniqueness
  - Ordering
  - Disparity gradient

- To find matches in the image pair, we will assume
  - Most scene points visible from both views
  - Image regions for the matches are similar in appearance

Dense correspondence search



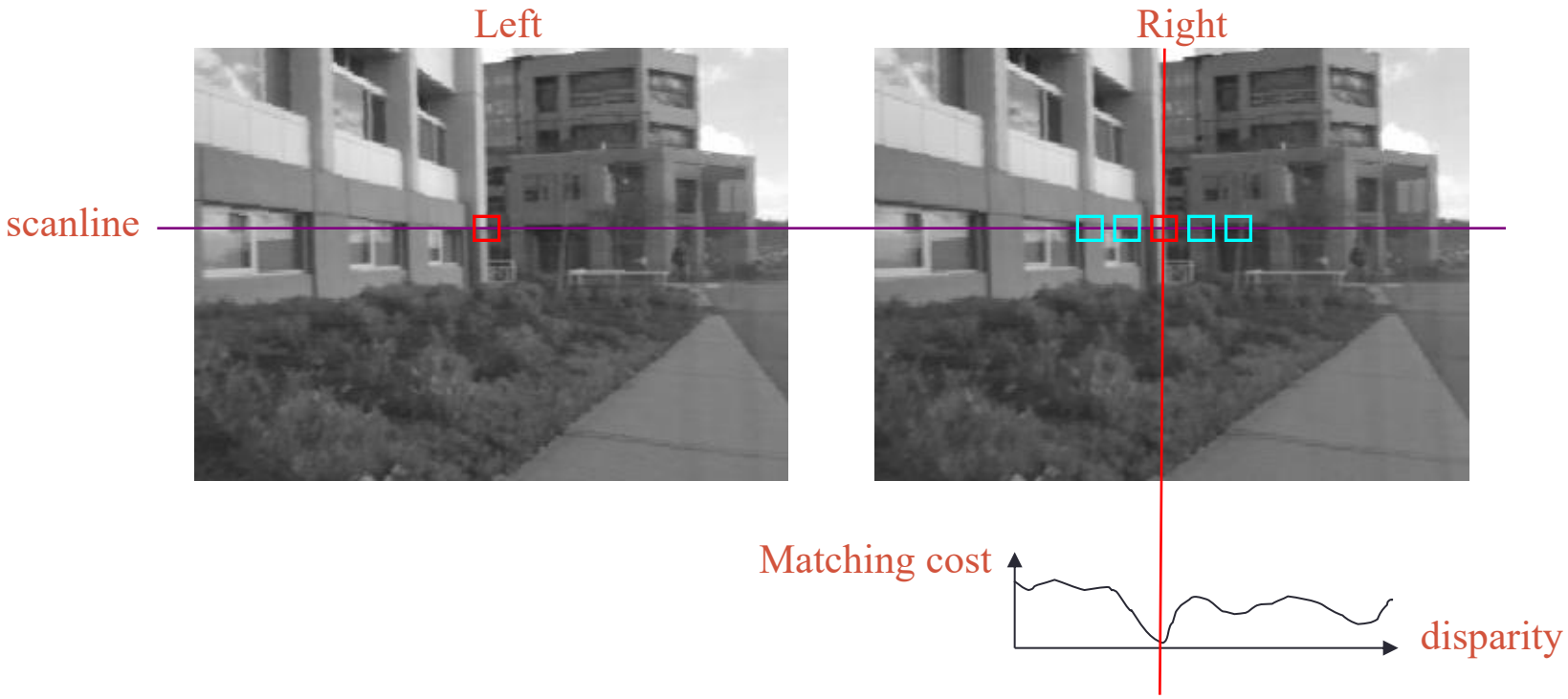
For each epipolar line

For each pixel / window in the left image

- compare with every pixel / window on same epipolar line in right image
- pick position with minimum match cost (e.g., SSD, normalized correlation)

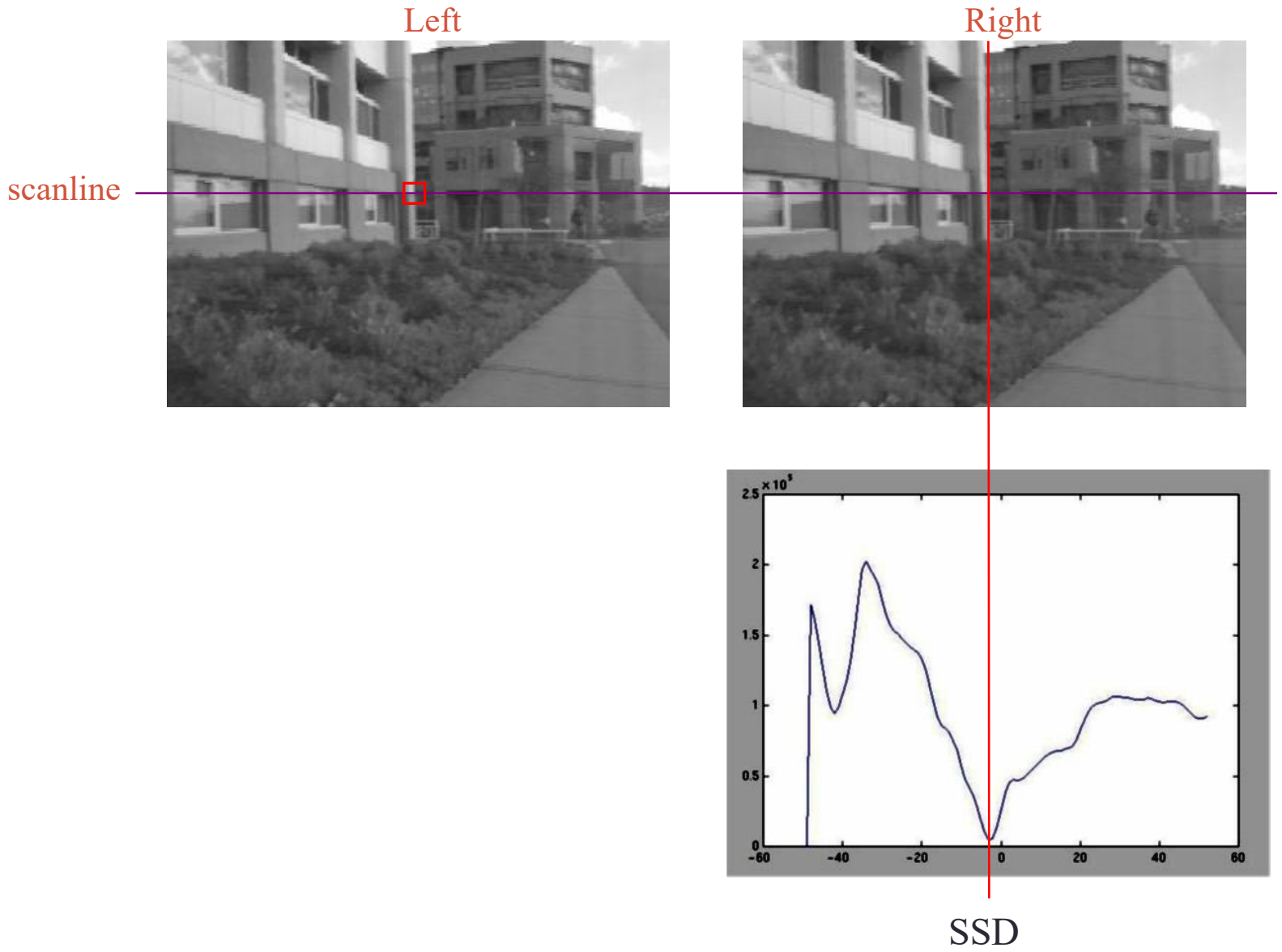
Adapted from Li Zhang

Correspondence search with similarity constraint

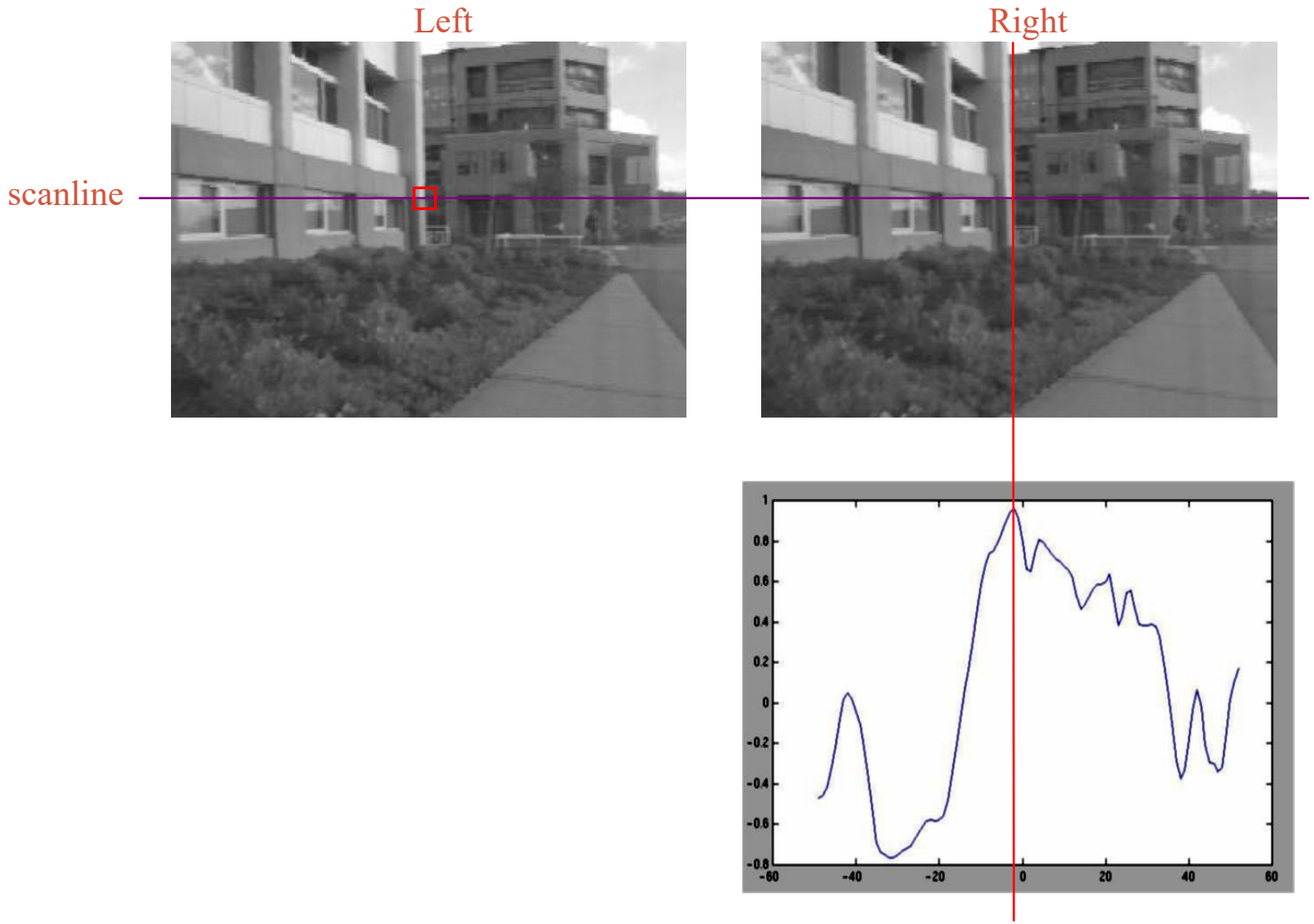


- Slide a window along the right scanline and compare contents of that window with the reference window in the left image
- Matching cost: SSD or normalized correlation

Correspondence search with similarity constraint

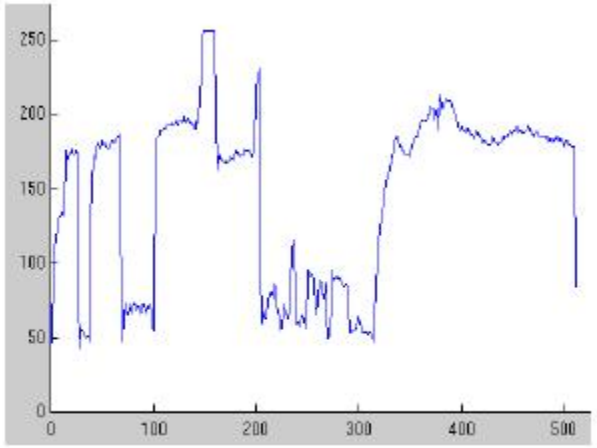


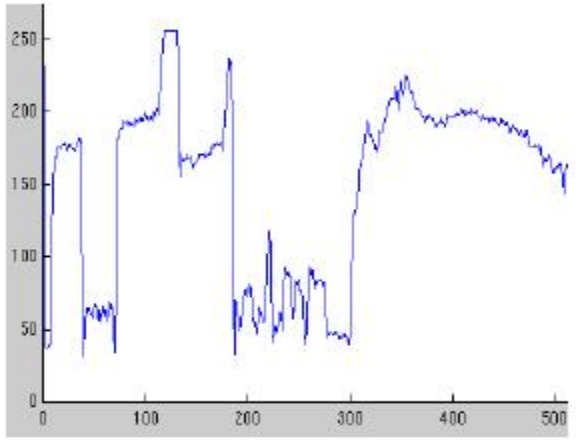
Correspondence search with similarity constraint



Correspondence problem

Intensity profiles

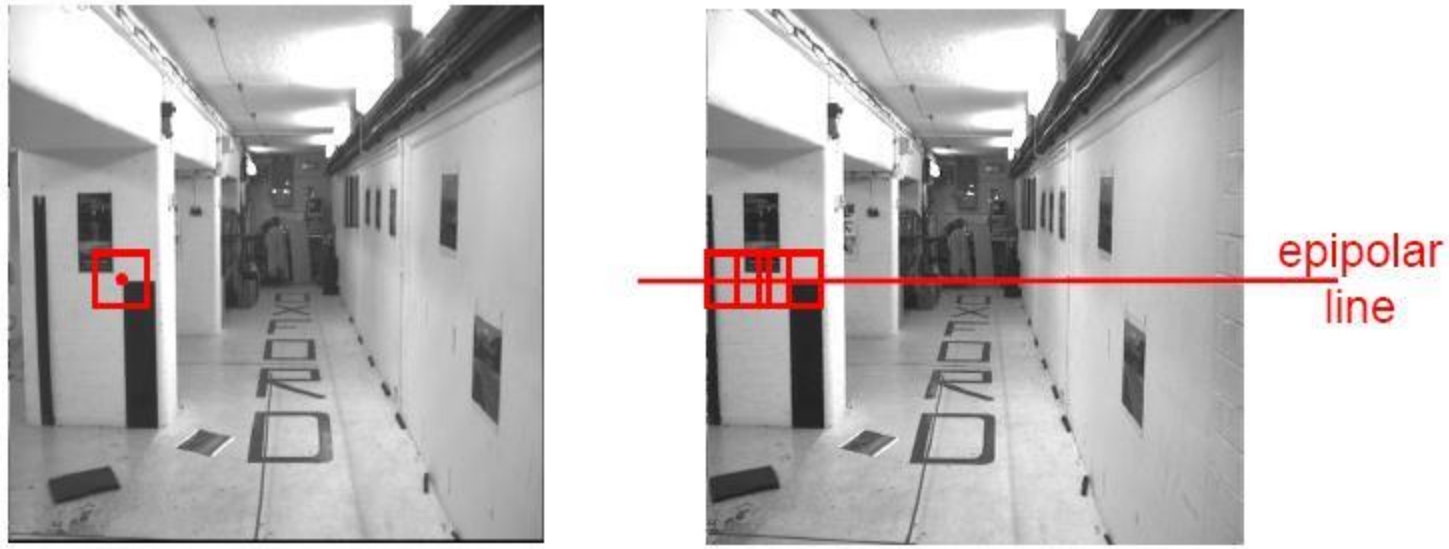




- Clear correspondence between intensities, but also noise and ambiguity

Source: Andrew Zisserman

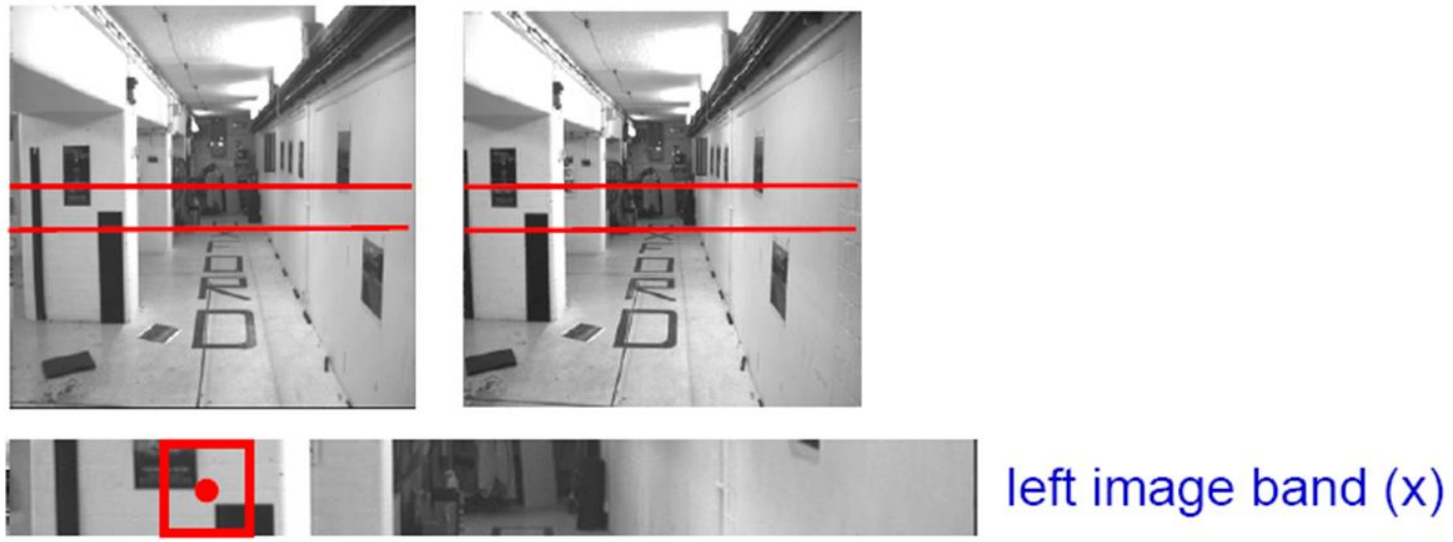
Correspondence problem



Neighborhoods of corresponding points are similar in intensity patterns.

Source: Andrew Zisserman

Correlation-based window matching

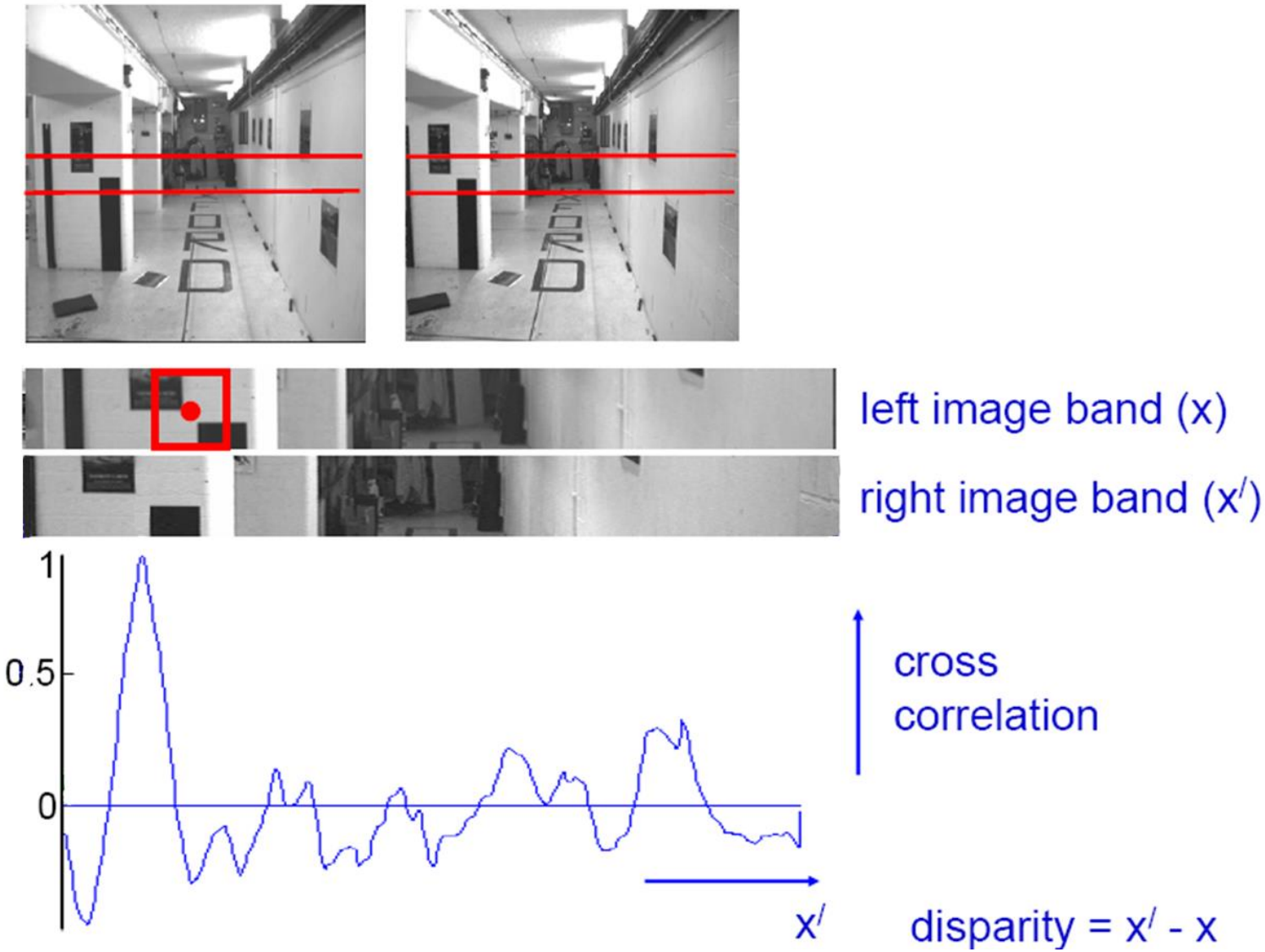


Correlation-based window matching

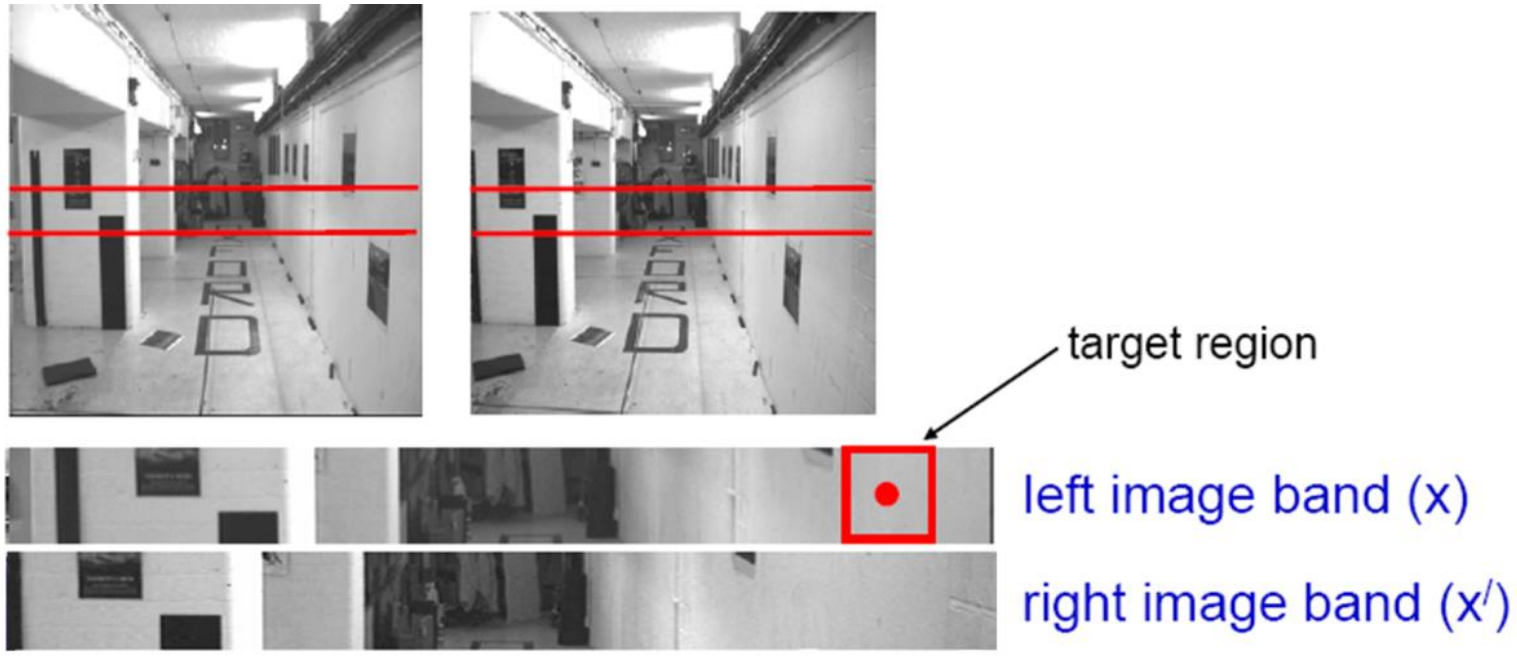
left image band (x)

right image band (x')

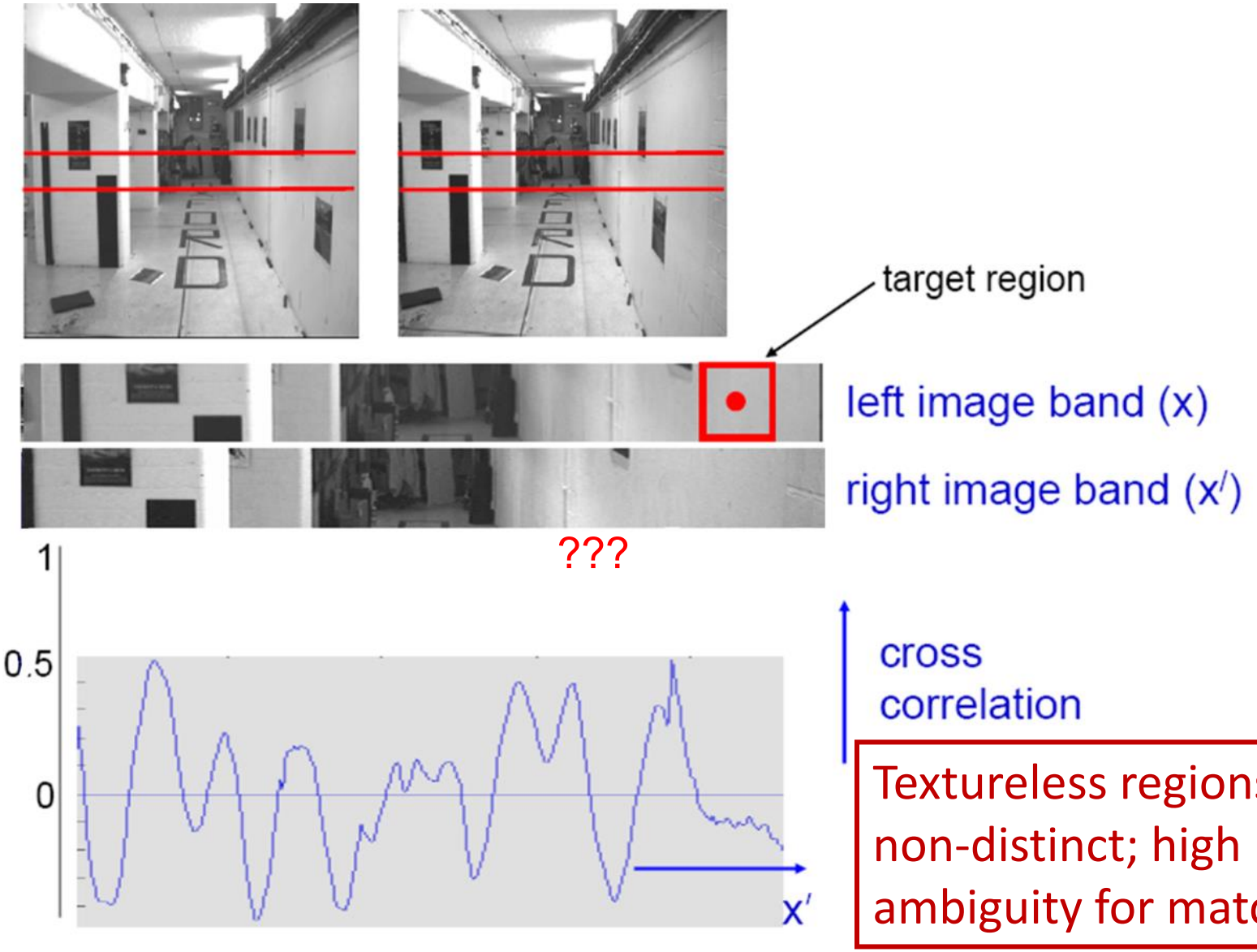
Correlation-based window matching



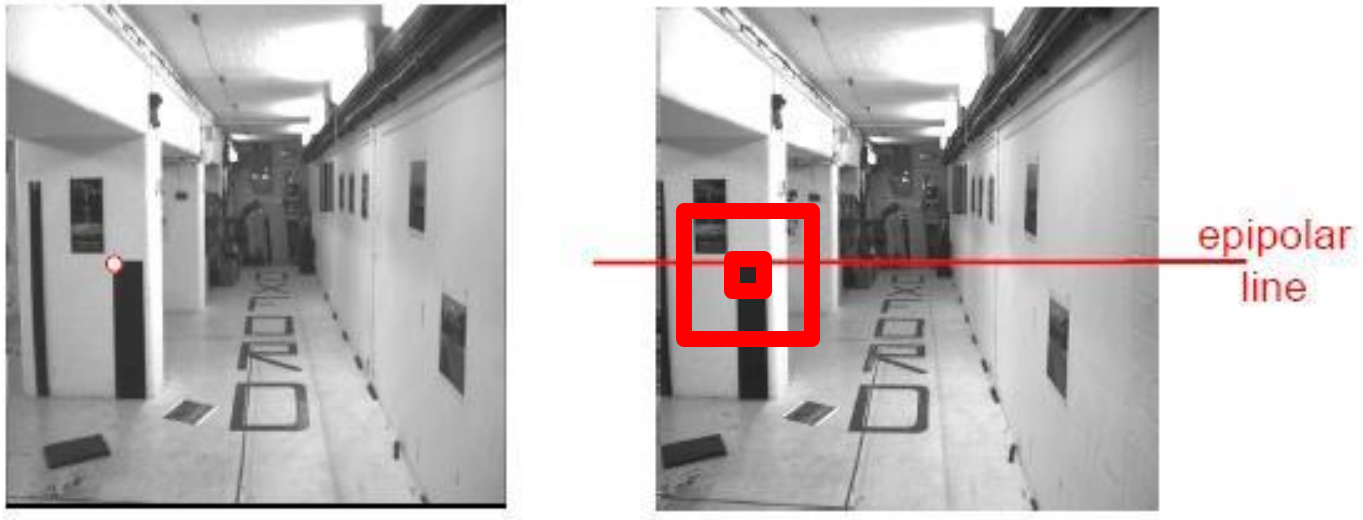
Correlation-based window matching



Correlation-based window matching



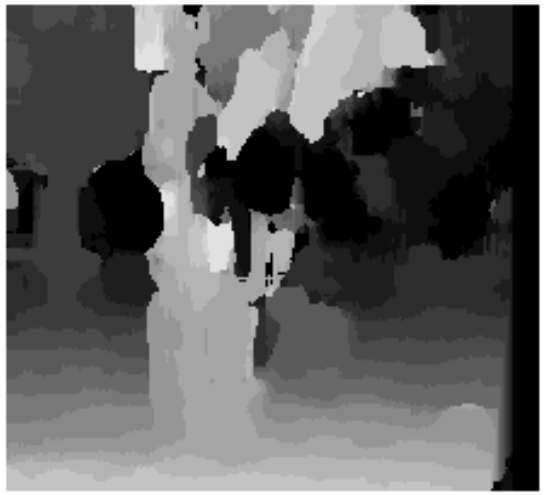
Effect of window size



Source: Andrew Zisserman

Effect of window size

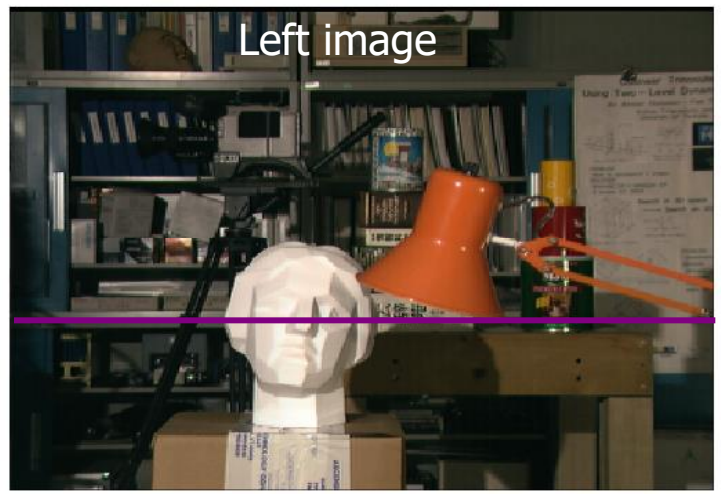
W = 3

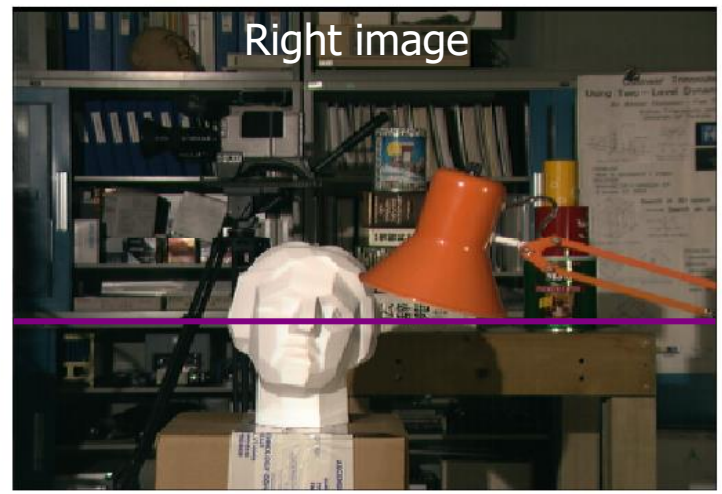


W = 20

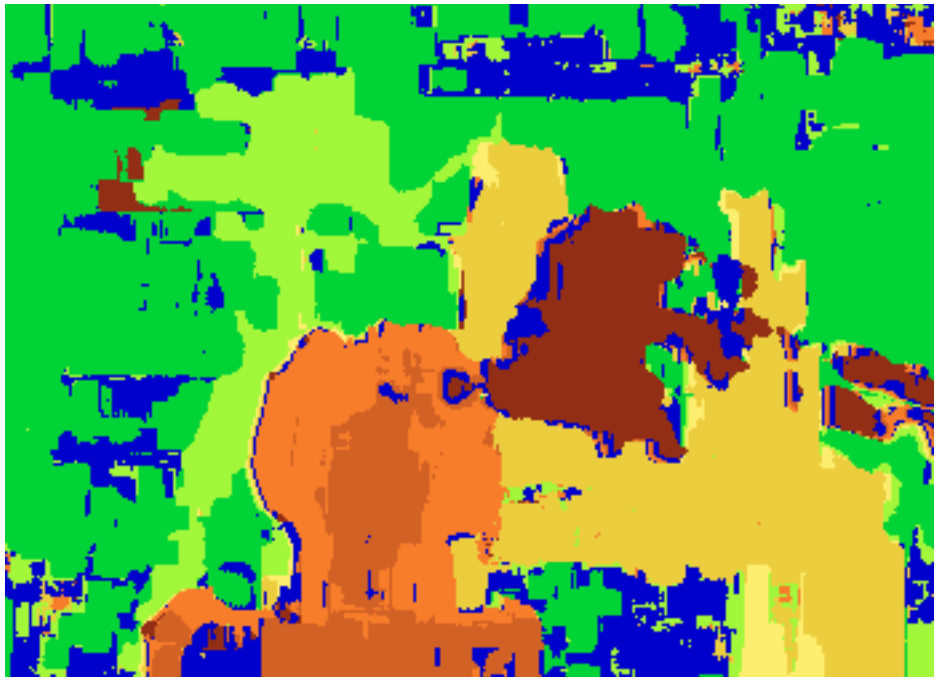
Want window large enough to have sufficient intensity variation, yet small enough to contain only pixels with about the same disparity.

Figures from Li Zhang


Left image


Right image

Results with window search



Window-based matching
(best window size)

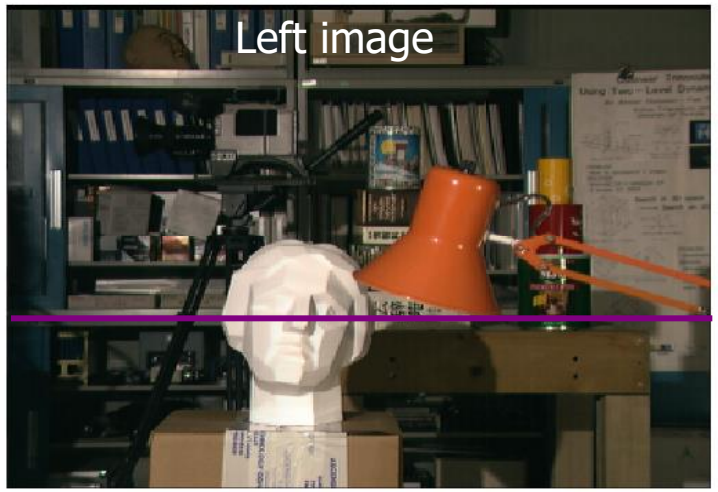
Ground truth

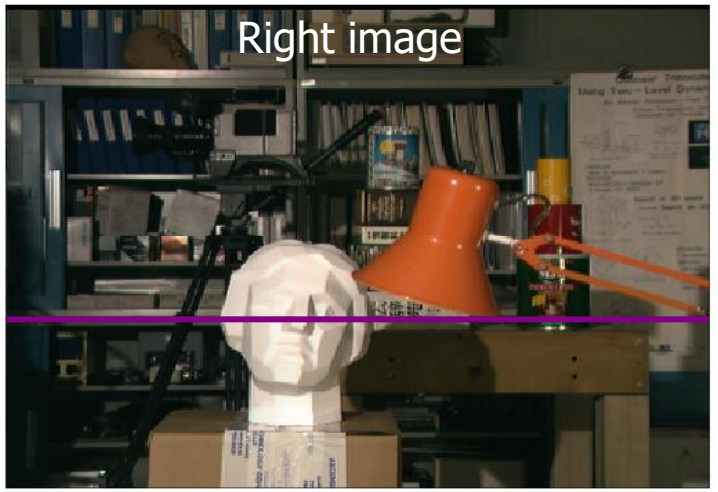
Better solutions

- Beyond individual correspondences to estimate disparities:
- Optimize correspondence assignments jointly
  - Scanline at a time (e.g. dynamic programming)
  - Full 2D grid (e.g. graph cuts)
  - Approximate 2D solution (e.g. semi-global matching)

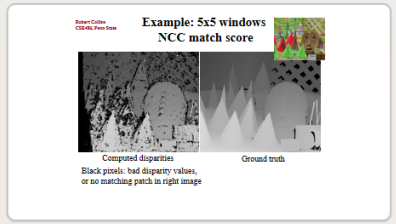
Scanline stereo

- Try to coherently match pixels on the entire scanline
- Different scanlines are still optimized independently

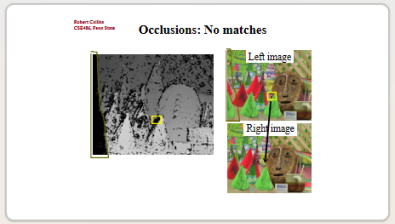




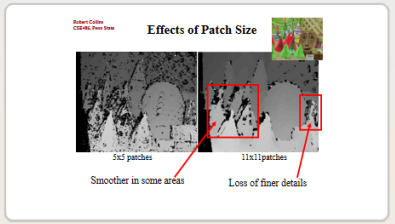
Skipping over details of Dynamic Program Scanline Stereo



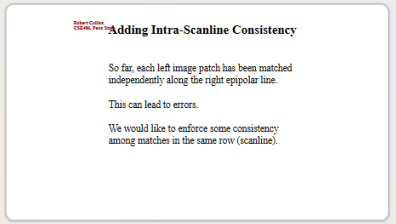
66



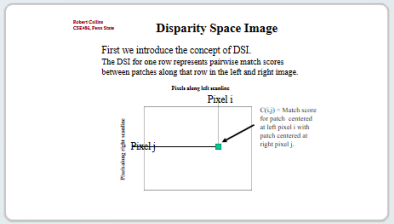
67



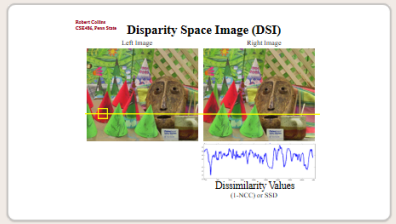
68



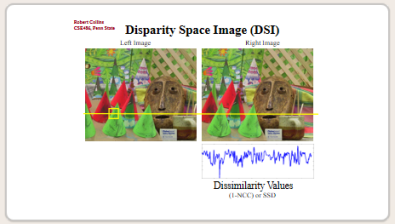
69



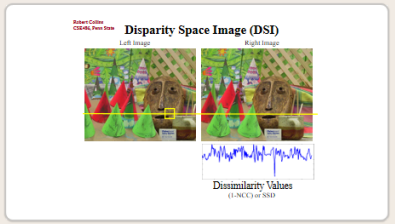
70



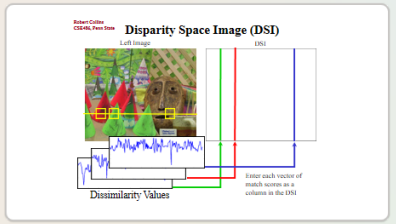
71



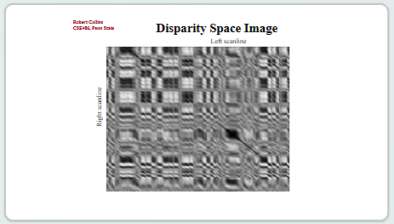
72



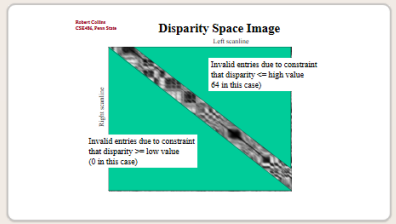
73



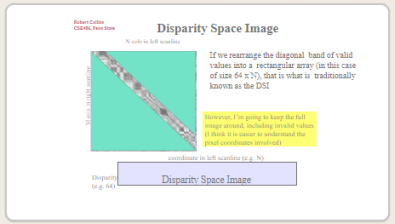
74



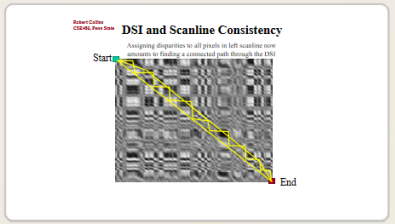
75



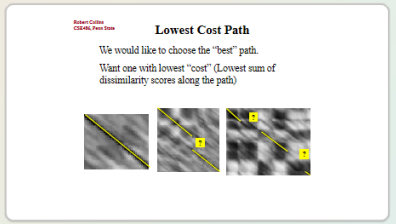
76



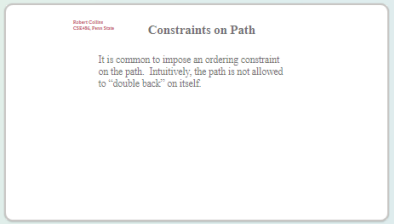
77



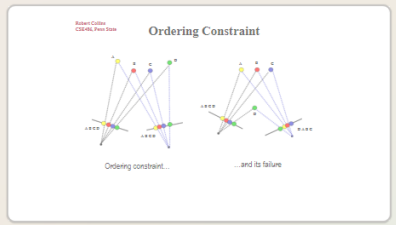
78

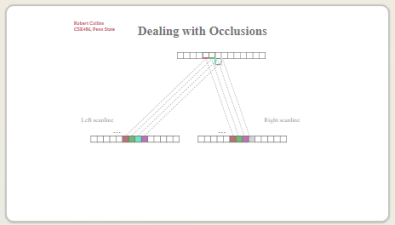


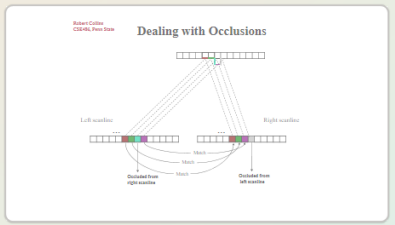
79

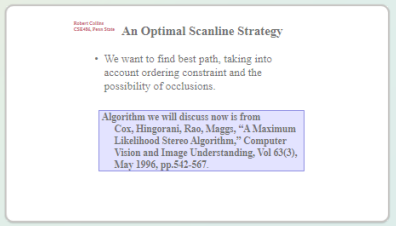


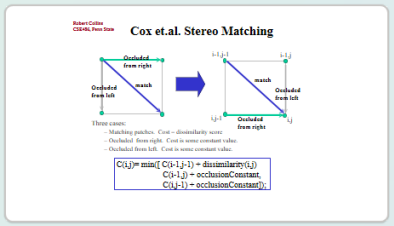
80









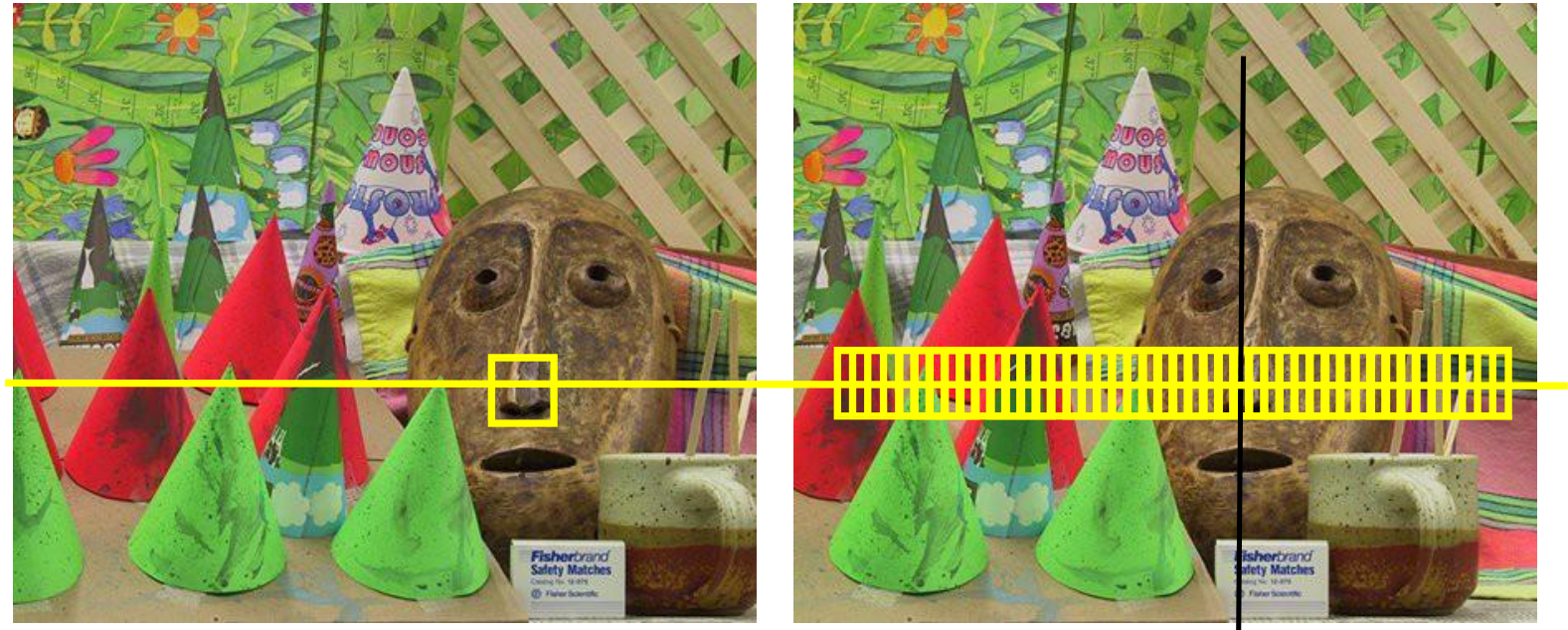


Robert Collins
CSE486, Penn State

Matching using Epipolar Lines

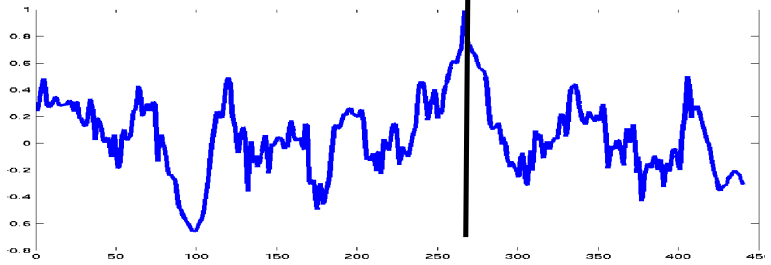
Left Image

Right Image



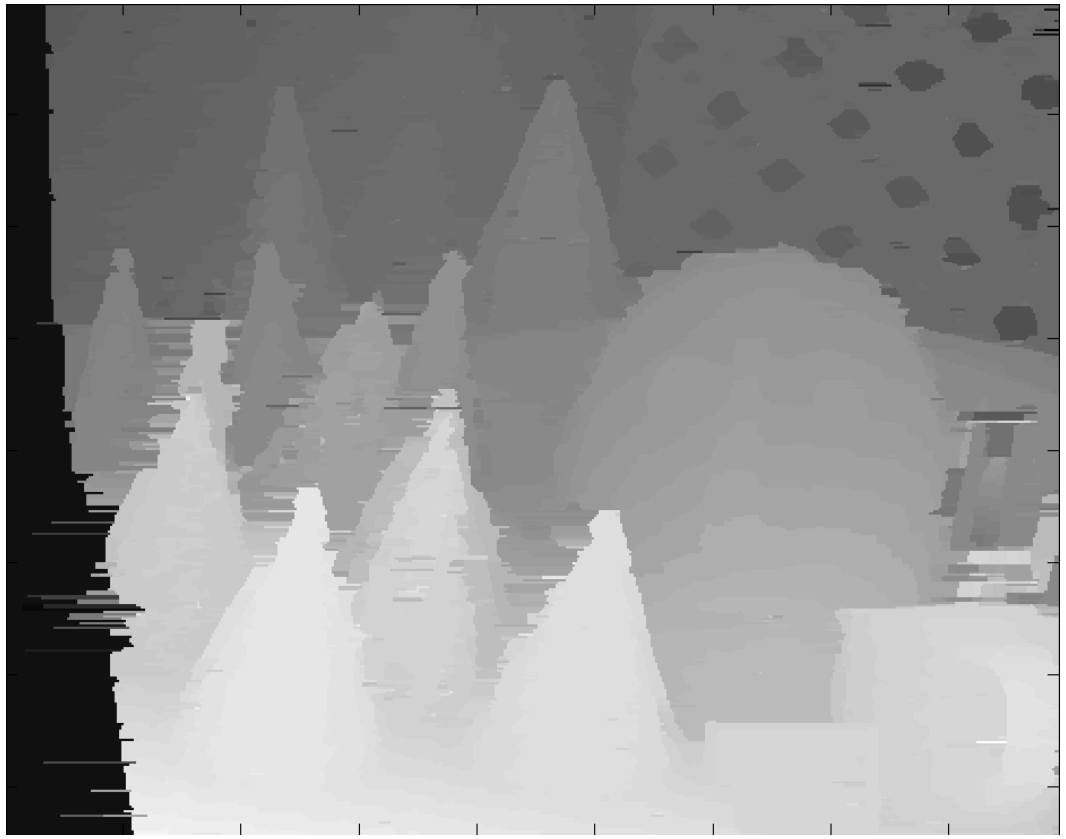
For a patch in left image

Compare with patches along same row in right image



Match Score Values

Example

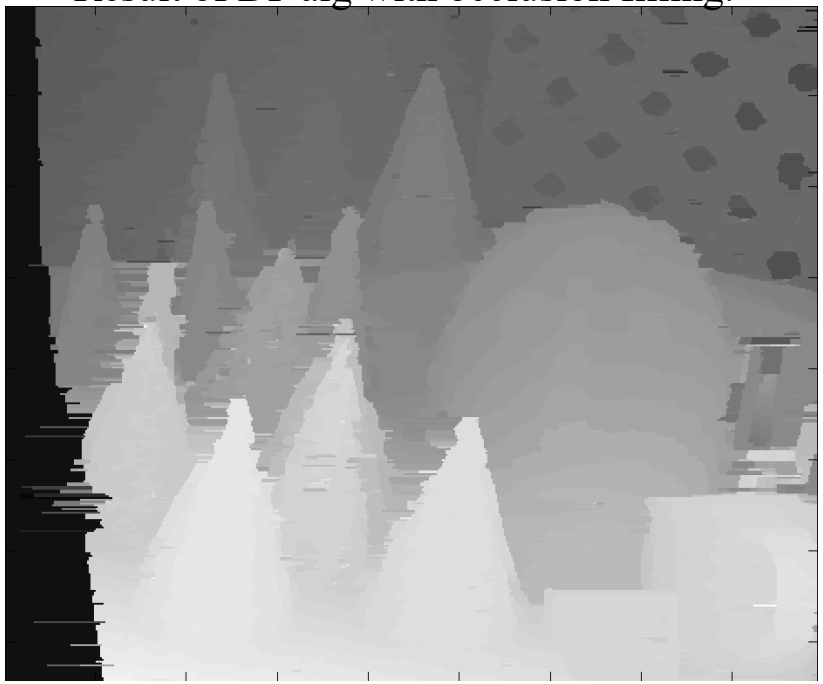


Result of DP alg with occlusion filling.

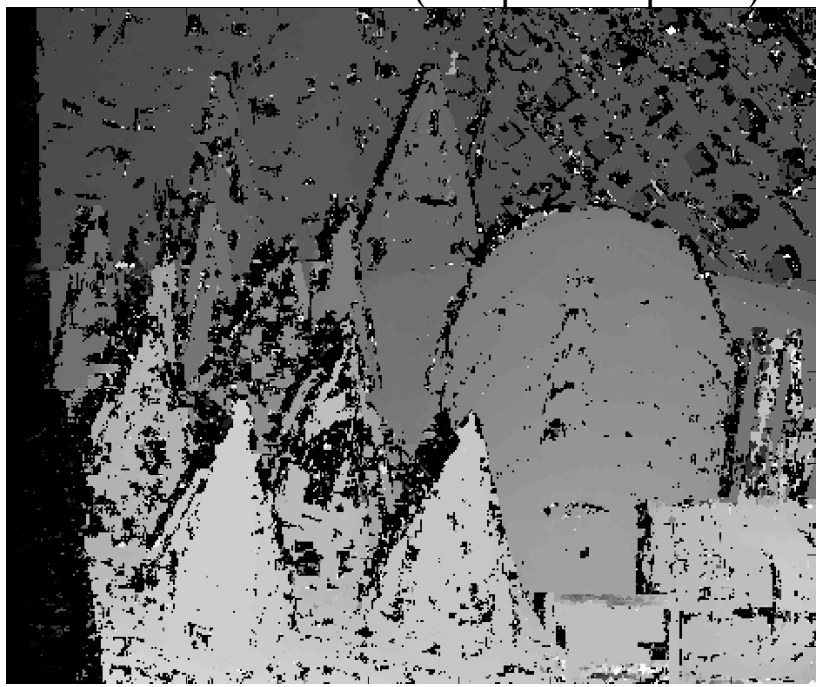
Robert Collins
CSE486, Penn State

Example

Result of DP alg with occlusion filling.



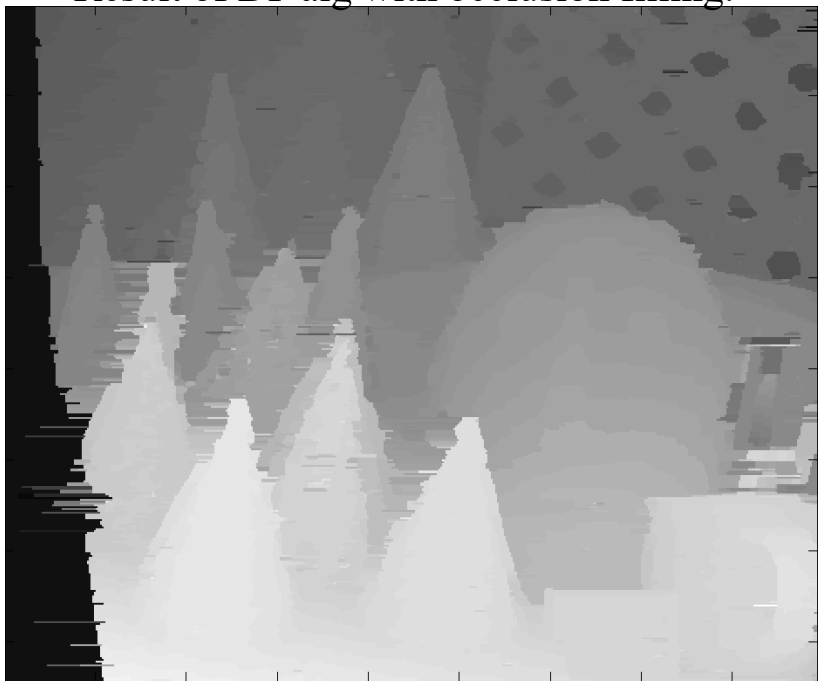
Result without DP (independent pixels)



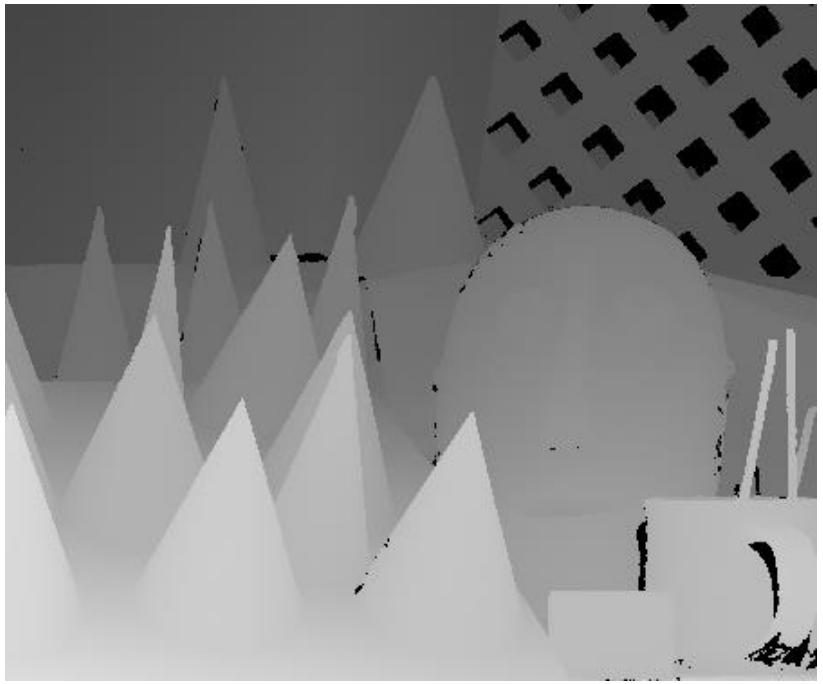
Robert Collins
CSE486, Penn State

Example

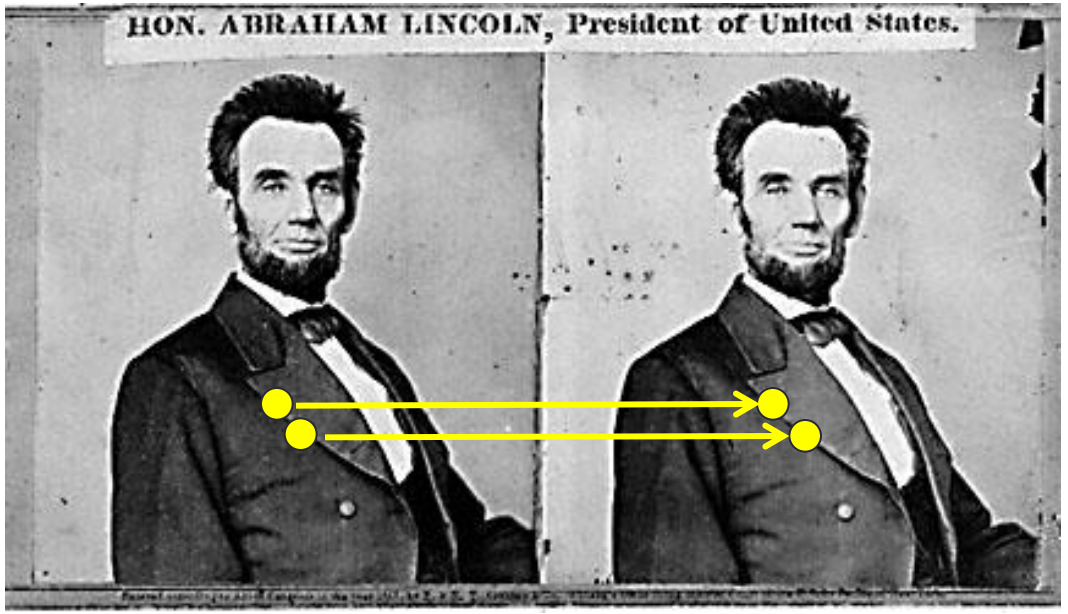
Result of DP alg with occlusion filling.



Ground truth

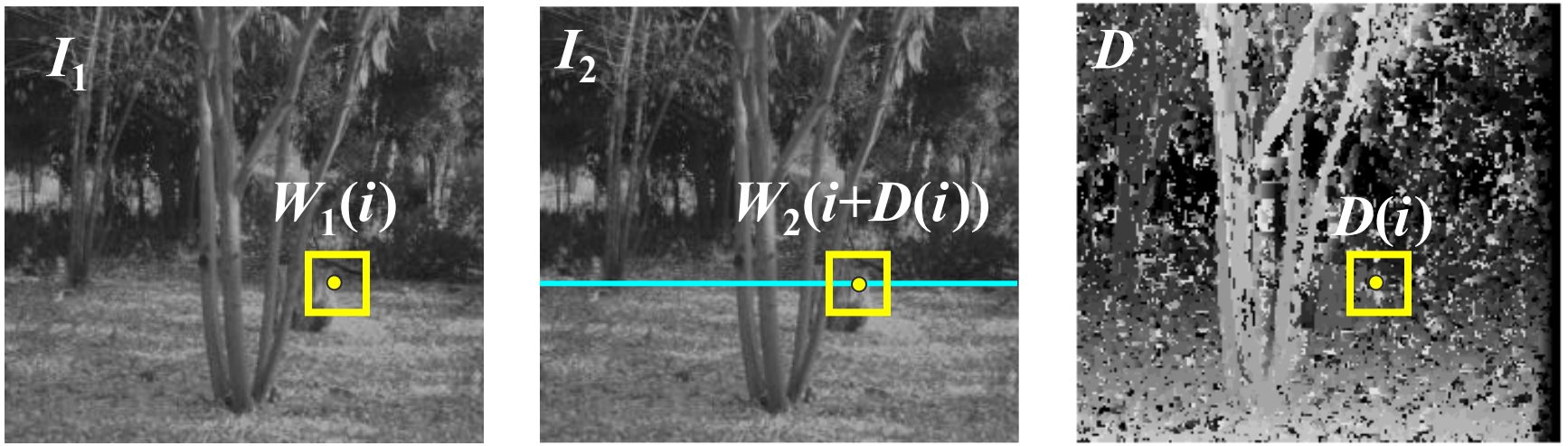


Stereo with 2D smoothness constraint



- What defines a good stereo correspondence?
  1. Match quality
     - Want each pixel to find a good match in the other image
  2. Smoothness
     - If two pixels are adjacent, they should (usually) move about the same amount

Optimizing for match quality *and* smoothness (in any direction)



$$E = \alpha E_{\text{data}}(I_1, I_2, D) + \beta E_{\text{smooth}}(D)$$

$$E_{\text{data}} = \sum_i \left(W_1(i) - W_2(i + D(i))\right)^2$$

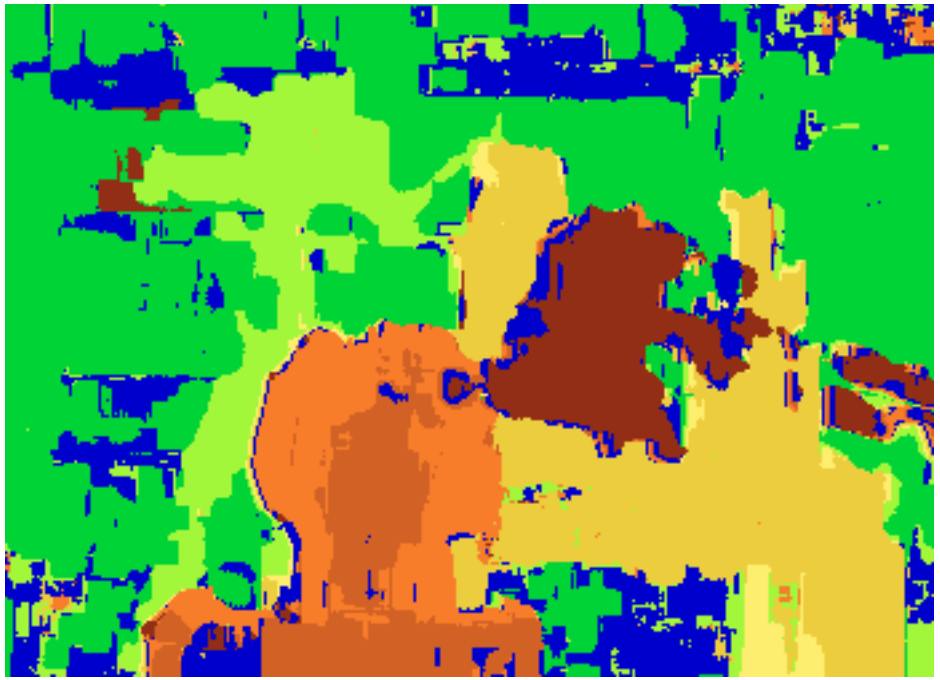
$$E_{\text{smooth}} = \sum_{\text{neighbors } i,j} \rho\left(D(i) - D(j)\right)$$

- Energy functions of this form can be minimized using *graph cuts*

Y. Boykov, O. Veksler, and R. Zabih, Fast Approximate Energy Minimization via Graph Cuts, PAMI 2001

Source: Steve Seitz

Results with window search



Window-based matching
(best window size)

Ground truth

Better results…

Graph cut method

Ground truth

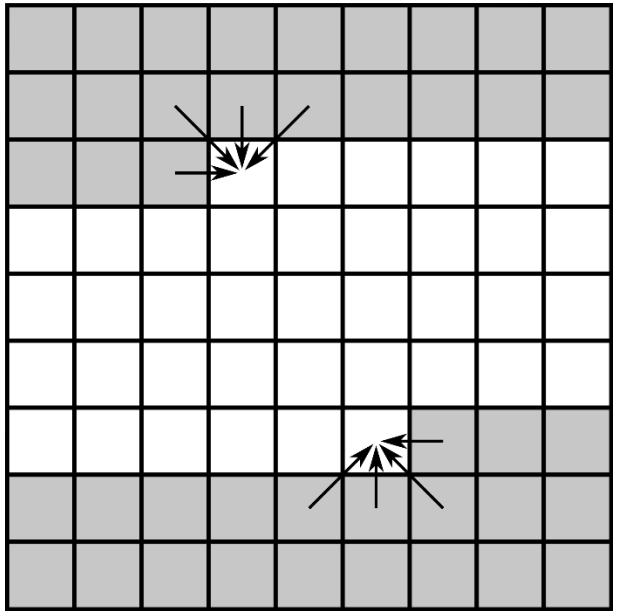
Boykov et al., Fast Approximate Energy Minimization via Graph Cuts, International Conference on Computer Vision, September 1999.

Semi-global matching

$$E(D) = \sum_{\mathbf{p}} (C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1]$$

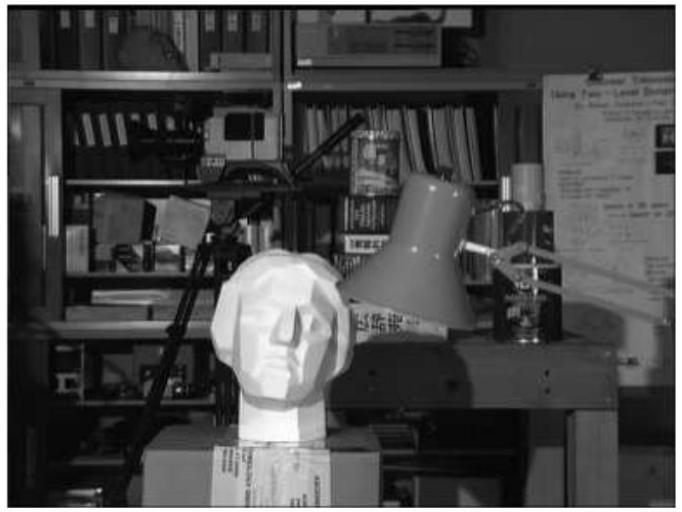
$$+ \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1])$$

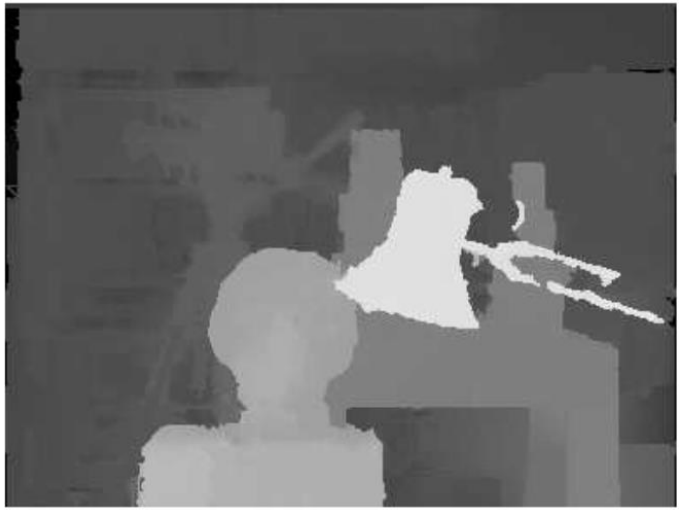
- Approximate the full smoothness optimization by considering 8 or 16 directions in two or three passes.
- Optimization looks like scanline, dynamic programming stereo, but with a 2d notion of smoothness

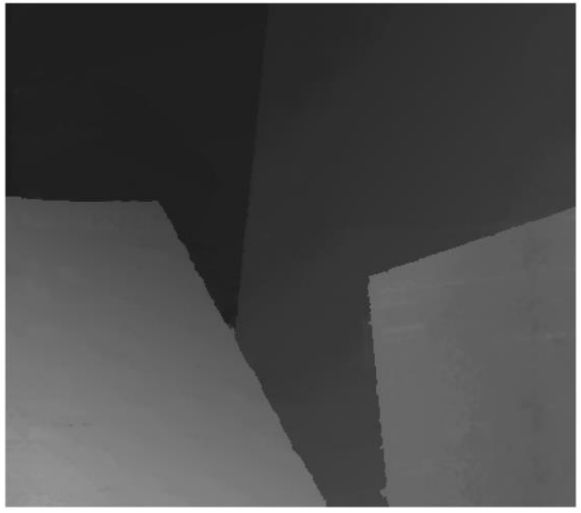


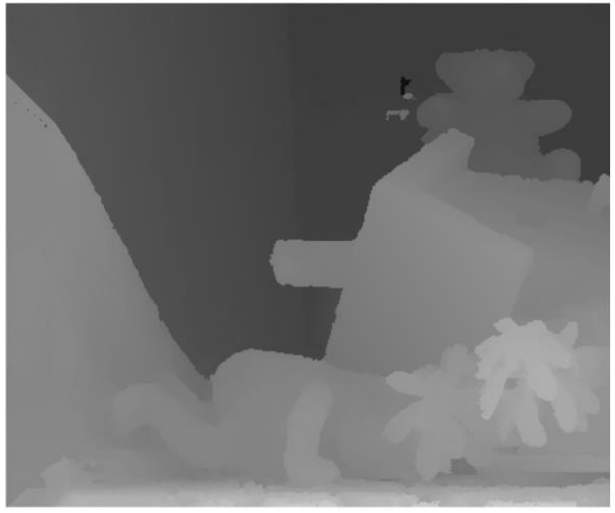
Stereo Processing by Semi-Global Matching and Mutual Information. Hirschmuller, PAMI 2007. **3500+ citations**

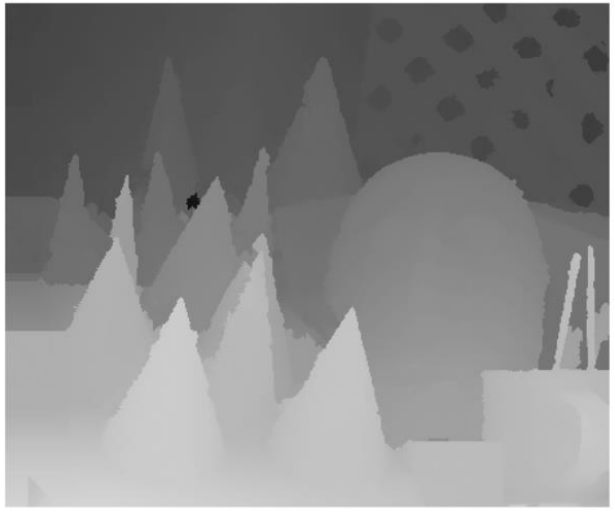
Semi-global matching











https://vision.middlebury.edu/stereo/eval3/

Stereo | Evaluation • Datasets • Code • Submit

Middlebury Stereo Evaluation - Version 3

Mouseover the table cells to see the produced disparity map. Clicking a cell will blink the ground truth for comparison. To change the table type, click the links below. For more information, please see the description of new features.

Submit and evaluate your own results.

Set: test dense | test sparse | training dense | training sparse
Metric: bad 0.5 | bad 1.0 | bad 2.0 | bad 4.0 | avgerr | rms | A50 | A90 | A95 | A99 | time | time/MP | time/GD
Mask: nonocc | all

plot selected | show invalid | Reset sort | Reference list

bad 2.0 (%)

Date	Name	Res	Weight Avg	Austr MP: 5.6 nd: 290 im0 im1 GT nonocc	AustrP MP: 5.6 nd: 290 im0 im1 GT nonocc	Bicyc2 MP: 5.6 nd: 250 im0 im1 GT nonocc	Class MP: 5.7 nd: 610 im0 im1 GT nonocc	ClassE MP: 5.7 nd: 610 im0 im1 GT nonocc	Compu MP: 1.5 nd: 256 im0 im1 GT nonocc	Crusa MP: 5.5 nd: 800 im0 im1 GT nonocc	CrusaP MP: 5.5 nd: 800 im0 im1 GT nonocc	Djemb MP: 5.7 nd: 320 im0 im1 GT nonocc	DjembL MP: 5.7 nd: 320 im0 im1 GT nonocc	Hoops MP: 5.7 nd: 410 im0 im1 GT nonocc	Livgrm MP: 5.9 nd: 320 im0 im1 GT nonocc	Nkuba MP: 5.5 nd: 570 im0 im1 GT nonocc	Plants MP: 5.6 nd: 320 im0 im1 GT nonocc	Stairs MP: 5.2 nd: 450 im0 im1 GT nonocc
08/07/24	AIO-Stereo	F	**2.36** 1	2.38 3	**1.71** 1	3.22 26	**0.85** 1	5.83 23	1.24 6	1.42 4	1.32 5	1.03 8	4.49 10	4.81 5	2.43 2	**3.61** 1	2.12 2	3.63 7
08/01/24	RSM	F	2.40 2	2.66 6	1.88 4	3.18 24	0.91 2	5.80 22	1.34 10	1.35 2	1.16 2	0.93 4	3.35 4	**3.96** 1	2.88 3	4.38 3	**2.01** 1	4.15 9
11/13/23	Selective-IGEV	F	2.51 3	2.54 5	1.86 3	2.51 14	1.12 6	7.22 35	1.23 5	1.36 3	1.17 3	1.16 12	4.48 9	4.83 6	2.99 4	3.79 2	2.26 5	4.72 13
08/12/24	PointerNet	F	2.69 4	2.67 7	1.84 2	3.21 25	1.51 10	7.52 40	1.29 7	1.54 6	1.17 3	1.09 10	3.59 5	**3.96** 1	3.10 5	5.60 5	2.29 6	4.27 10
11/10/22	DLNR	F	3.20 5	2.91 8	2.37 9	2.18 10	1.67 13	3.21 4	1.37 11	1.66 7	1.66 9	1.11 11	6.25 22	7.07 16	3.45 7	8.90 21	4.43 25	2.91 3
06/14/24	IGEV++	F	3.23 6	3.24 11	2.46 10	4.12 41	1.15 7	6.71 32	1.38 12	1.53 5	1.52 6	1.02 7	4.57 11	4.68 4	5.41 27	7.68 15	2.22 3	4.68 12
06/27/24	CAS++	F	3.33 7	4.27 37	3.72 46	3.17 23	2.17 24	**2.44** 1	1.33 9	2.24 17	2.01 14	1.47 24	4.04 8	8.15 19	4.97 23	5.80 7	3.73 22	3.04 4
02/21/24	ClearDepth	F	3.48 8	4.14 34	3.16 30	2.81 17	1.95 16	4.55 12	2.36 26	1.73 10	1.70 12	1.25 19	5.46 16	11.2 48	3.12 6	7.30 13	3.70 21	3.45 6
06/06/24	MoCha-V2	F	3.51 9	2.52 4	1.95 5	2.25 11	1.47 9	4.61 13	0.98 3	7.35 69	8.07 79	**0.66** 1	2.95 2	4.18 3	4.46 16	5.70 6	2.54 8	2.70 2
06/05/24	MGS-Stereo	F	3.57 10	3.62 17	2.93 19	3.43 29	2.66 35	6.24 29	2.54 29	2.04 14	2.15 18	1.23 18	5.81 19	8.40 22	3.56 8	6.48 10	3.18 13	4.81 14
10/28/23	SAMTormer	F	3.63 11	3.84 28	2.95 21	1.85 7	0.98 3	3.31 5	2.02 22	1.71 8	1.67 10	0.82 2	5.50 17	10.4 38	4.57 18	11.7 38	3.92 23	3.41 5
06/13/22	EAI-Stereo	F	3.68 12	4.02 30	3.32 33	2.48 13	1.42 8	4.19 10	2.37 27	2.18 16	2.01 14	1.16 12	10.2 42	8.84 25	4.00 11	7.15 12	3.14 12	6.44 25
11/10/21	CREStereo	F	3.71 13	4.73 44	3.94 49	5.07 60	1.96 17	3.02 3	1.42 13	2.28 18	2.05 17	1.51 26	6.86 23	6.35 9	4.25 14	6.01 9	4.60 28	5.49 20
09/08/24	RSD	F	3.73 14	**2.13** 1	1.98 6	1.71 2	2.03 20	2.63 2	0.87 2	8.66 77	9.69 96	0.96 6	**2.54** 1	6.82 13	**2.34** 1	7.76 16	2.23 4	**2.57** 1
02/28/24	AKD_Stereo	F	3.87 15	4.21 36	3.53 41	3.91 34	1.08 4	7.63 42	4.75 53	1.72 9	1.60 7	1.18 15	5.26 15	9.62 29	3.66 9	7.63 14	3.23 15	5.37 19
03/04/24	ET_Stereo	F	4.00 16	4.38 38	3.33 34	2.85 18	1.53 12	7.84 45	2.61 30	1.91 12	1.82 13	1.05 9	5.08 14	8.72 24	7.52 42	8.81 20	3.09 11	4.82 15
10/09/23	EGLCR-Stereo	F	4.03 17	4.69 41	2.46 10	3.70 31	2.99 47	10.7 71	2.48 28	1.95 13	1.63 8	0.94 5	5.76 18	8.17 20	3.84 10	10.3 30	2.99 10	4.87 16
03/04/24	AEACV	F	4.15 18	5.53 62	2.98 23	2.54 15	3.23 51	3.42 7	1.57 14	2.85 22	2.99 27	1.22 16	4.63 12	5.96 8	4.36 15	12.9 51	5.41 46	4.08 8
10/30/23	LoS	F	4.20 19	5.85 68	4.92 95	4.64 52	2.77 40	3.92 8	1.32 8	2.36 19	2.17 19	1.81 32	8.18 30	6.58 10	4.55 17	8.57 18	4.57 26	5.06 18
01/31/24	HART	F	4.24 20	3.13 9	2.24 8	4.16 43	1.10 5	4.01 9	2.03 23	1.86 11	1.68 11	0.85 3	9.83 37	11.0 44	8.71 51	9.65 25	3.26 16	6.96 30
09/26/24	GCAP_Stereo	F	4.31 21	5.32 54	3.40 37	2.38 12	2.16 23	11.2 78	4.44 49	2.13 15	2.04 16	1.32 21	7.16 24	8.97 26	5.03 25	8.38 17	3.22 14	6.08 24

Medium | Popout

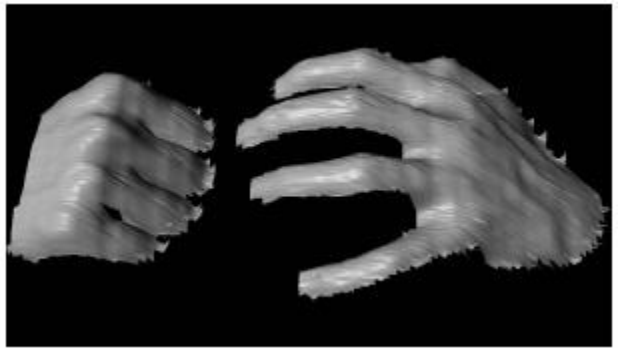
Right View Newkuba

Stereo Depth Estimation Challenges

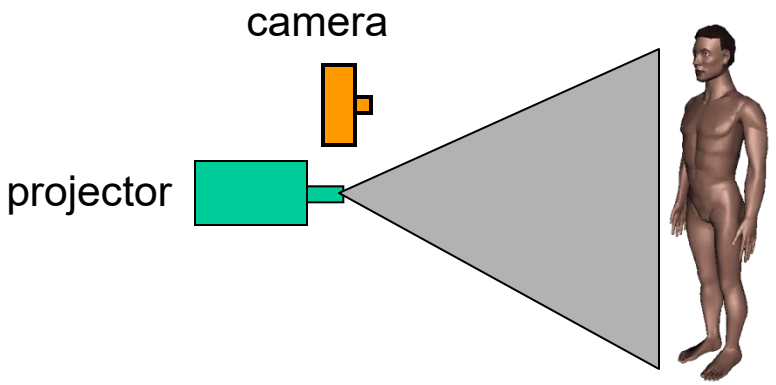
- Low-contrast ; textureless image regions
- Occlusions
- Violations of brightness constancy (e.g., specular reflections)
- Really large baselines (foreshortening and appearance change)
- Camera calibration errors

Active stereo with structured light





- Project “structured” light patterns onto the object
  - Simplifies the correspondence problem
  - Allows us to use only one camera

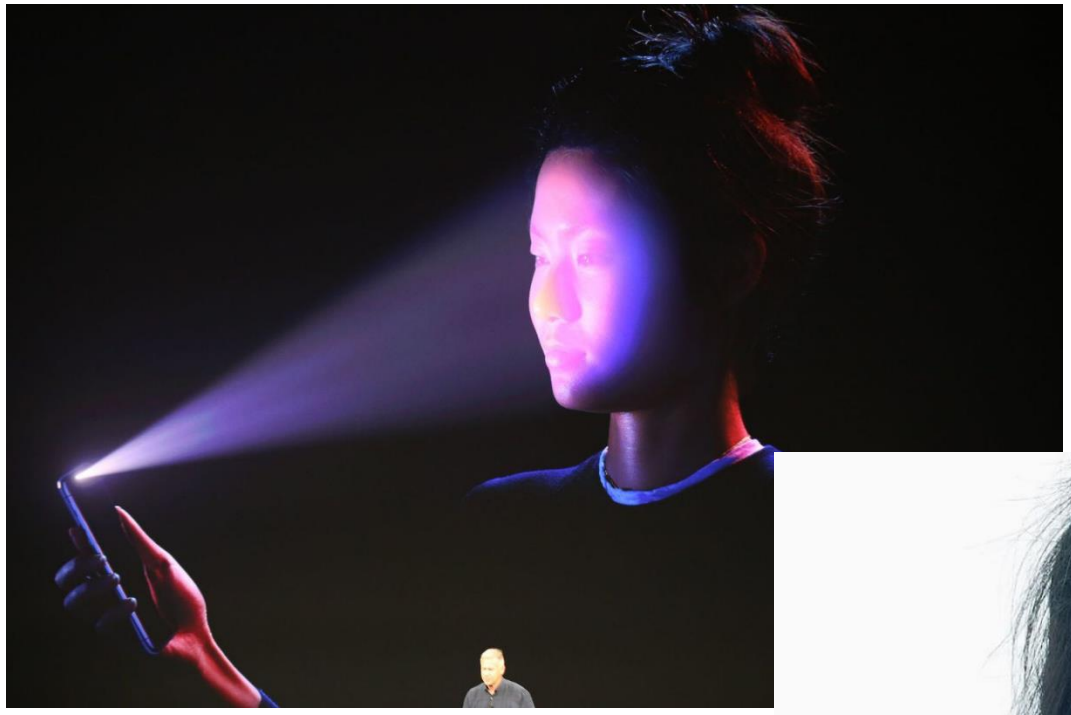


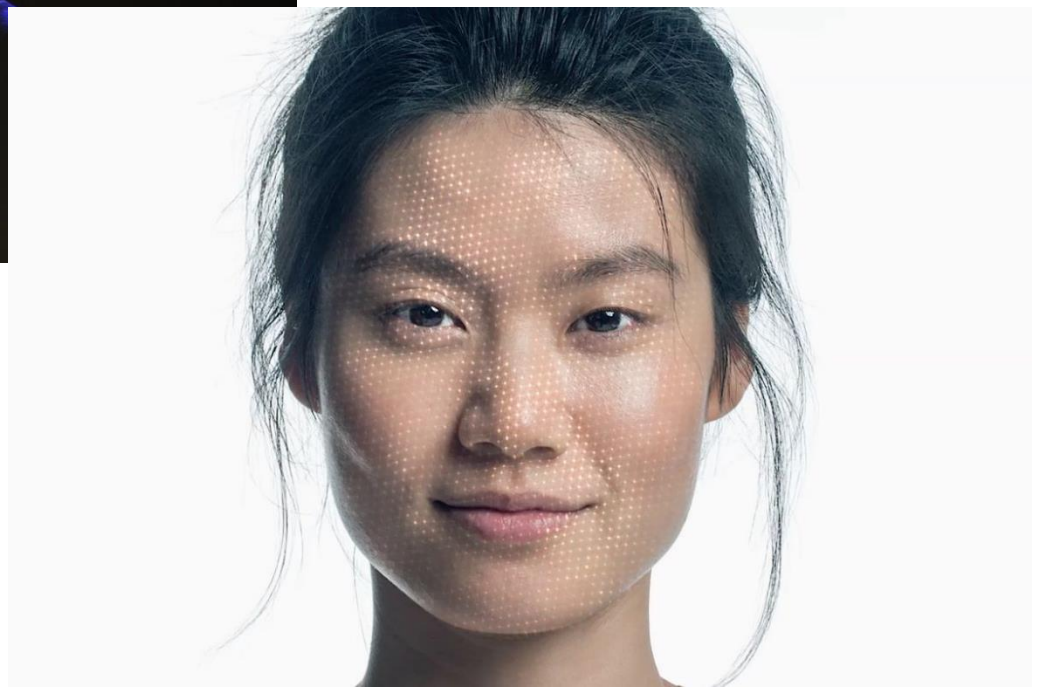
L. Zhang, B. Curless, and S. M. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. *3DPVT* 2002

Kinect: Structured infrared light

http://bbzippo.wordpress.com/2010/11/28/kinect-in-infrared/

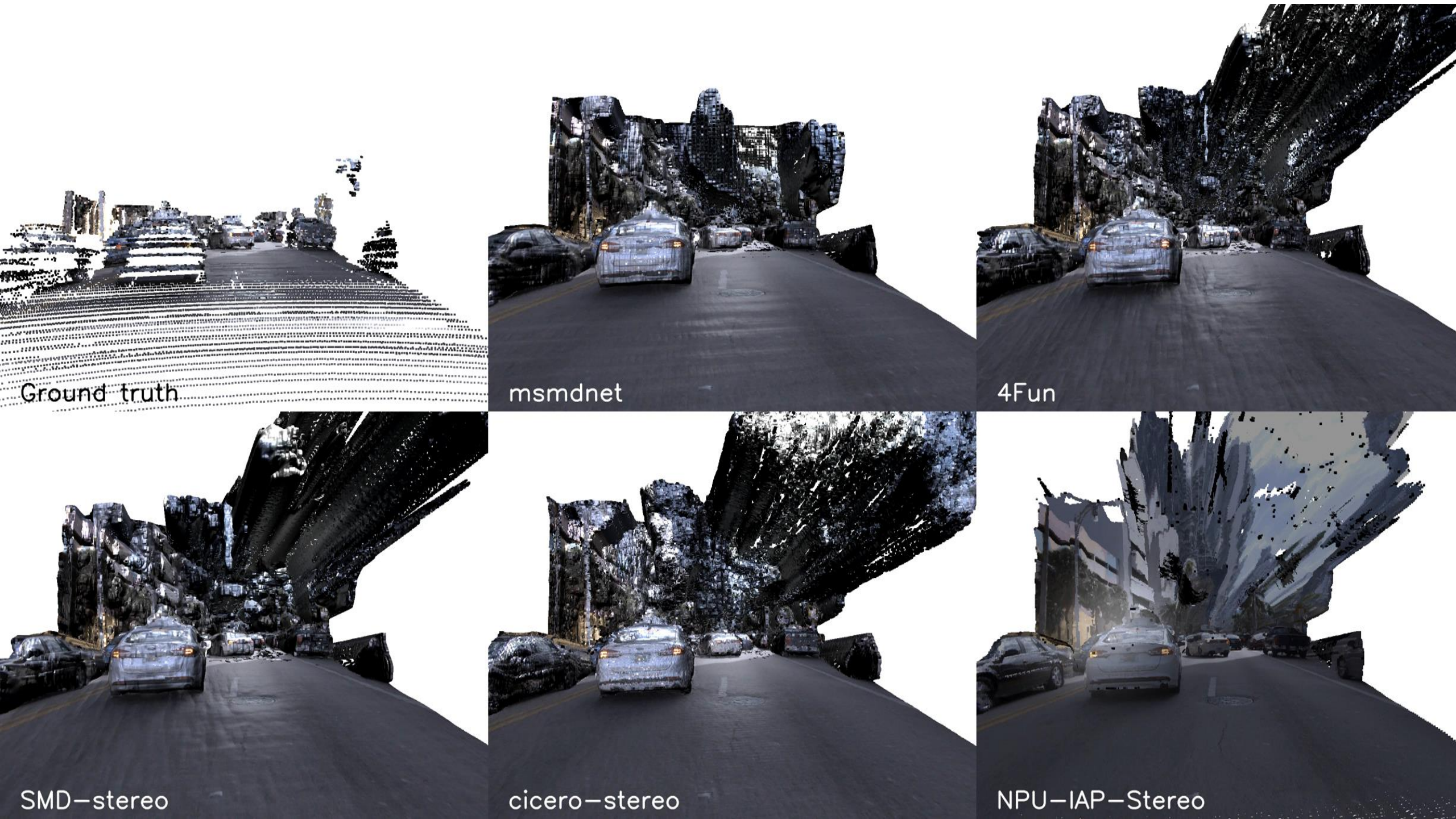
iPhone X





iPhone 12 switched to lidar
(time of flight)

Self-driving efforts use both lidar and stereo


Ground truth
msmdnet
4Fun
SMD−stereo
cicero−stereo
NPU−IAP−Stereo

Can we train a deep network to go straight from images to 3d in one forward pass?

DUSt3R: Geometric 3D Vision Made Easy

Shuzhe Wang*, Vincent Leroy†, Yohann Cabon†, Boris Chidlovskii† and Jerome Revaud†
*Aalto University
shuzhe.wang@aalto.fi
†Naver Labs Europe
firstname.lastname@naverlabs.com

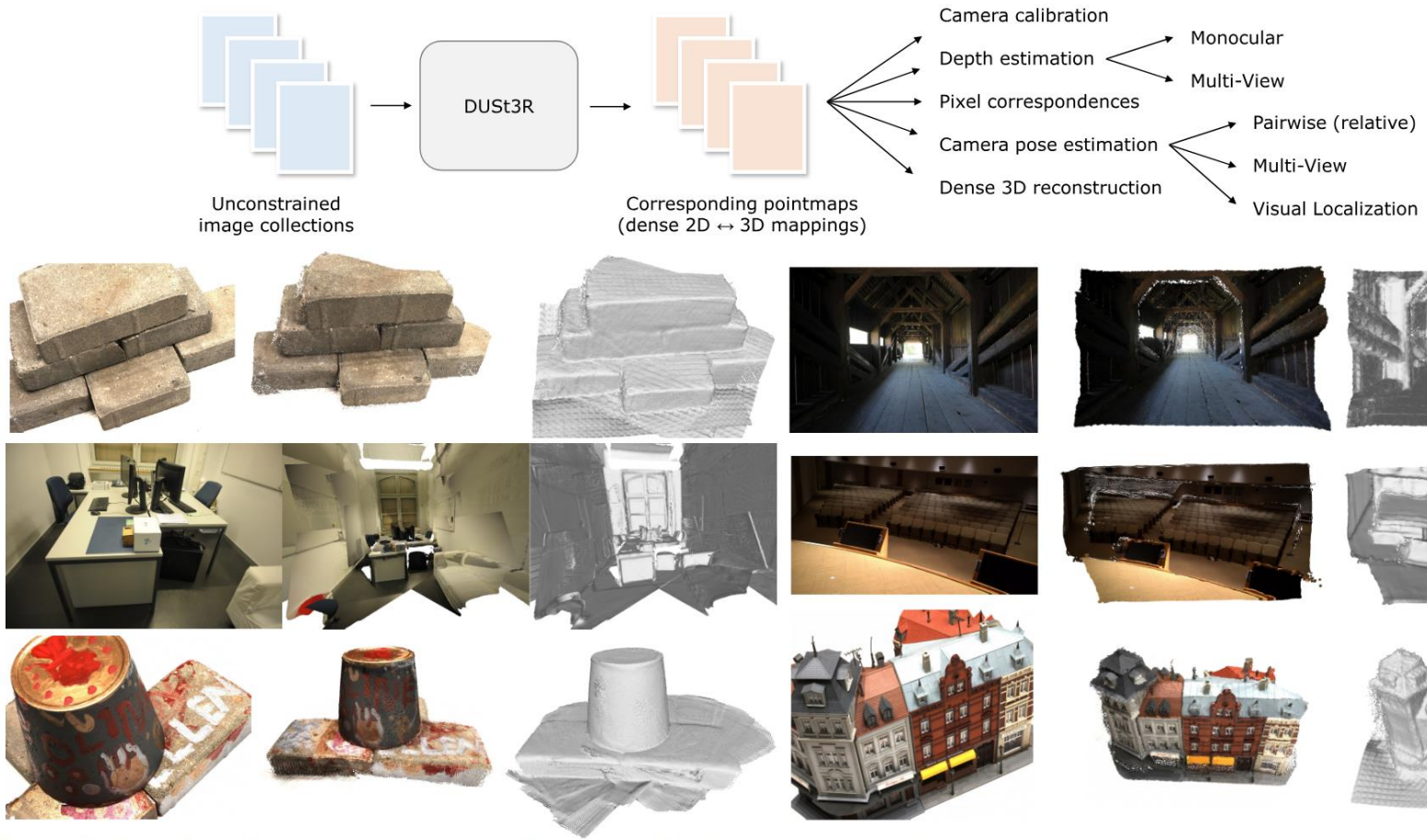


Figure 1. **Overview**: Given an unconstrained image collection, *i.e.* a set of photographs with unknown camera poses and intrinsics, our proposed method **DUSt3R** outputs a set of corresponding *pointmaps*, from which we can straightforwardly recover a variety of geometric quantities normally difficult to estimate all at once, such as the camera parameters, pixel correspondences, depthmaps, and fully-consistent 3D reconstruction. Note that DUSt3R also works for a single input image (*e.g.* achieving in this case monocular reconstruction). We also show **qualitative examples** on the DTU, Tanks and Temples and ETH-3D datasets [1, 51, 108] obtained **without** known camera parameters.

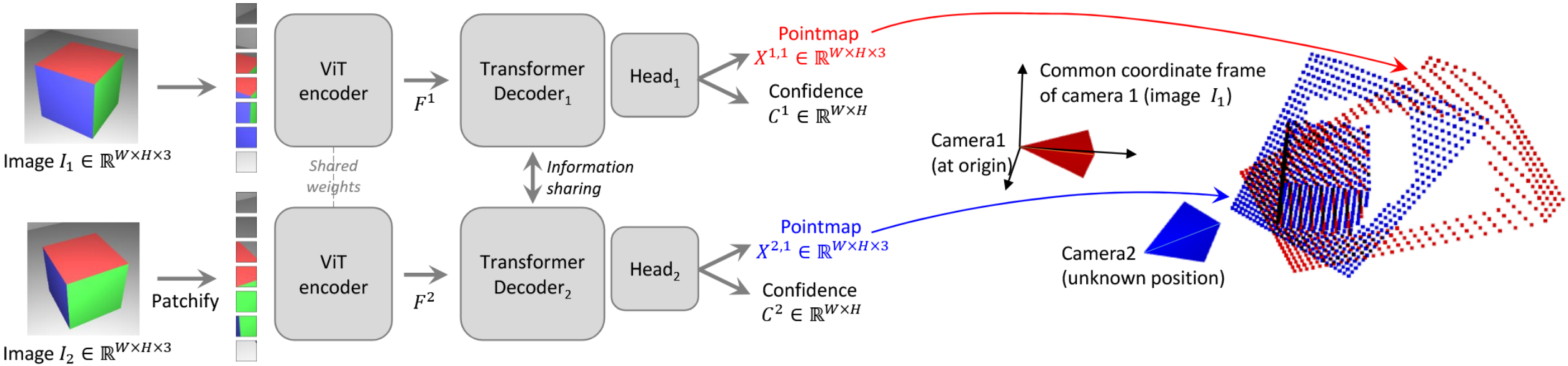
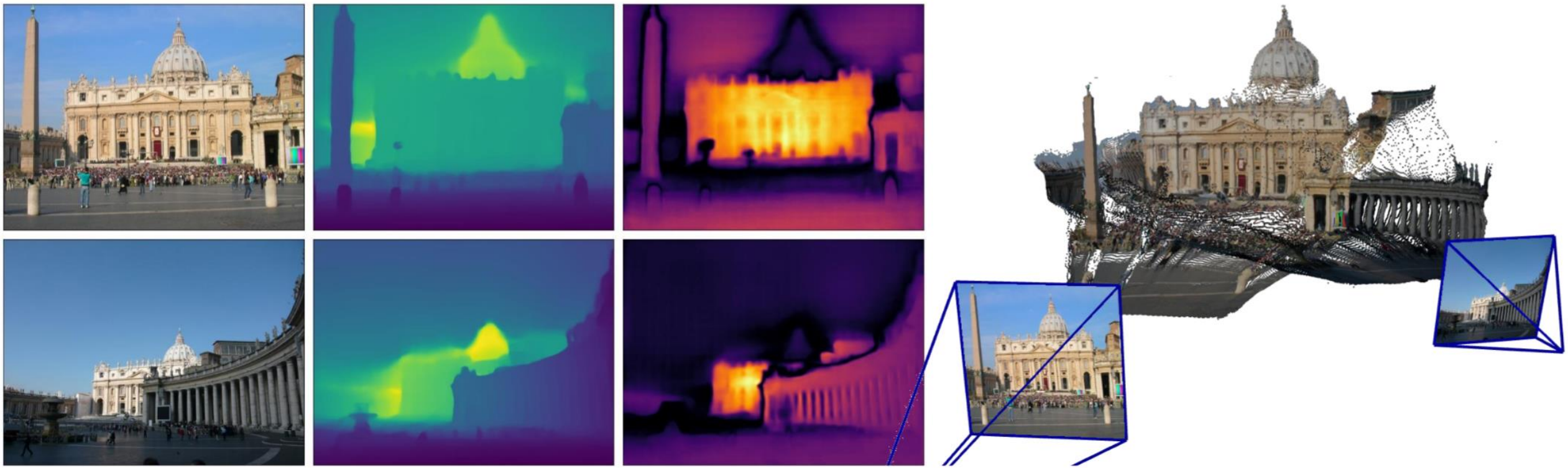
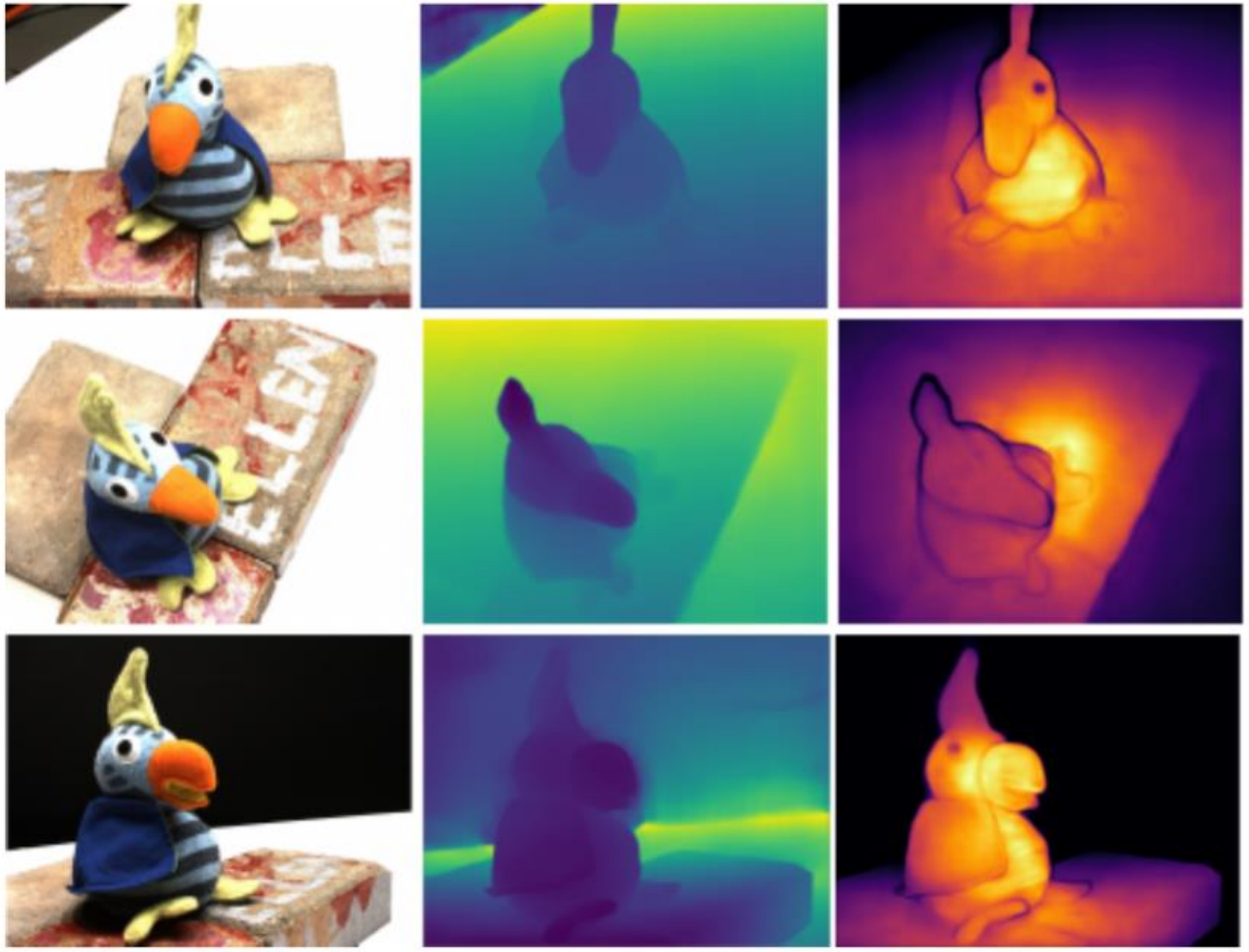


Figure 2. **Architecture of the network $\mathcal{F}$.** Two views of a scene (I^1, I^2) are first encoded in a Siamese manner with a shared ViT encoder. The resulting token representations F^1 and F^2 are then passed to two transformer decoders that constantly exchange information via cross-attention. Finally, two regression heads output the two corresponding pointmaps and associated confidence maps. Importantly, the two pointmaps are expressed in the same coordinate frame of the first image I^1. The network $\mathcal{F}$ is trained using a simple regression loss (Eq. (4))





More than 2 images – not a single forward pass

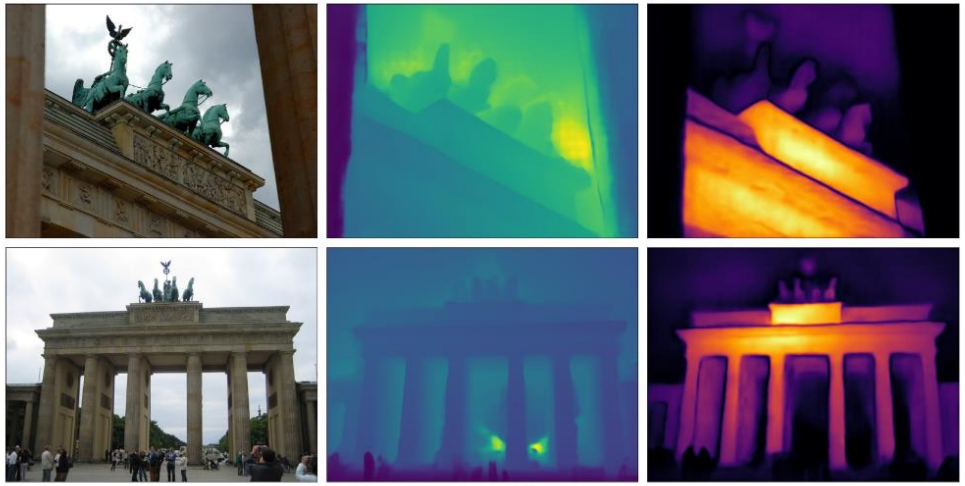
Methods	Train	Outdoor				Indoor					
		DDAD[41]		KITTI [35]		BONN [80]		NYUD-v2 [115]		TUM [119]	
		Rel↓	$\delta_{1.25}$ ↑	Rel↓	$\delta_{1.25}$ ↑	Rel↓	$\delta_{1.25}$ ↑	Rel↓	$\delta_{1.25}$ ↑	Rel ↓	$\delta_{1.25}$ ↑
DPT-BEiT[91]	D	10.70	**84.63**	9.45	89.27	-	-	**5.40**	**96.54**	**10.45**	**89.68**
NeWCRFs[174]	D	**9.59**	82.92	**5.43**	**91.54**	-	-	6.22	95.58	14.63	82.95
Monodepth2 [37]	SS	23.91	75.22	11.42	86.90	56.49	35.18	16.19	74.50	31.20	47.42
SC-SfM-Learners [6]	SS	16.92	77.28	11.83	86.61	21.11	71.40	13.79	79.57	22.29	64.30
SC-DepthV3 [121]	SS	**14.20**	**81.27**	11.79	86.39	**12.58**	**88.92**	**12.34**	**84.80**	**16.28**	**79.67**
MonoViT[182]	SS	-	-	**09.92**	**90.01**	-	-	-	-	-	
RobustMIX [92]	T	-	-	18.25	76.95	-	-	11.77	90.45	15.65	**86.59**
SlowTv [117]	T	**12.63**	79.34	(6.84)	(56.17)	-	-	11.59	87.23	15.02	80.86
DUSt3R 224-NoCroCo	T	19.63	70.03	20.10	71.21	14.44	86.00	14.51	81.06	22.14	66.26
DUSt3R 224	T	16.32	77.58	16.97	77.89	11.05	89.95	10.28	88.92	17.61	75.44
DUSt3R 512	T	13.88	81.17	**10.74**	**86.60**	**8.08**	**93.56**	**6.50**	94.09	**14.17**	79.89

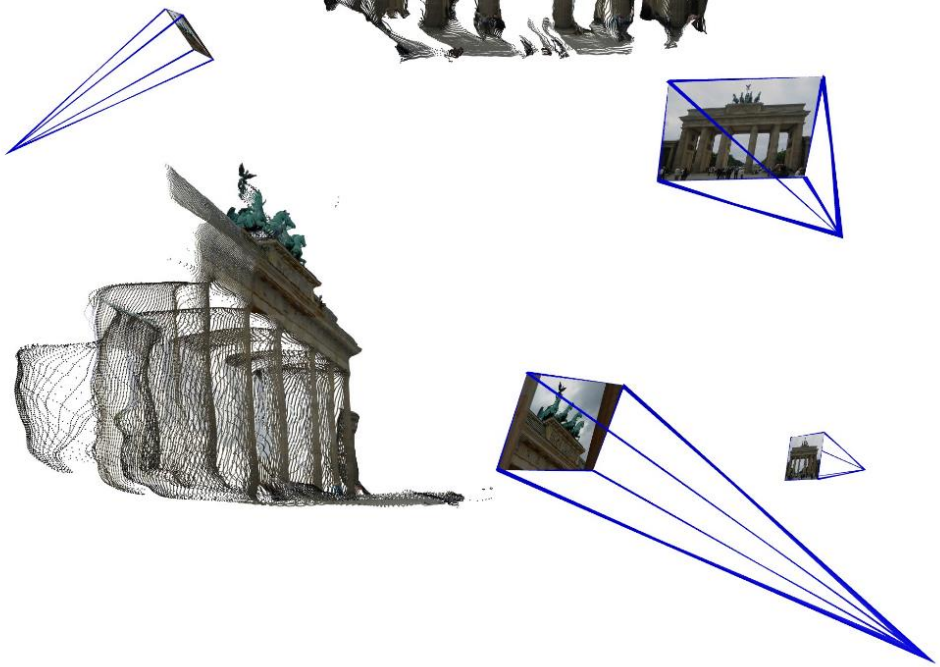
Methods	Co3Dv2 [94]			RealEstate10K
	RRA@15	RTA@15	mAA(30)	mAA(30)
RelPose [177]	57.1	-	-	-
Colmap+SPSG [26, 100]	36.1	27.3	25.3	45.2
PixSfM [59]	33.7	32.9	30.1	49.4
PosReg [140]	53.2	49.1	45.0	-
PoseDiffusion [140]	80.5	79.8	66.5	48.0
DUSt3R 512 (w/ PnP)	94.3	**88.4**	**77.2**	61.2
DUSt3R 512 (w/ GA)	**96.2**	86.8	76.7	**67.7**

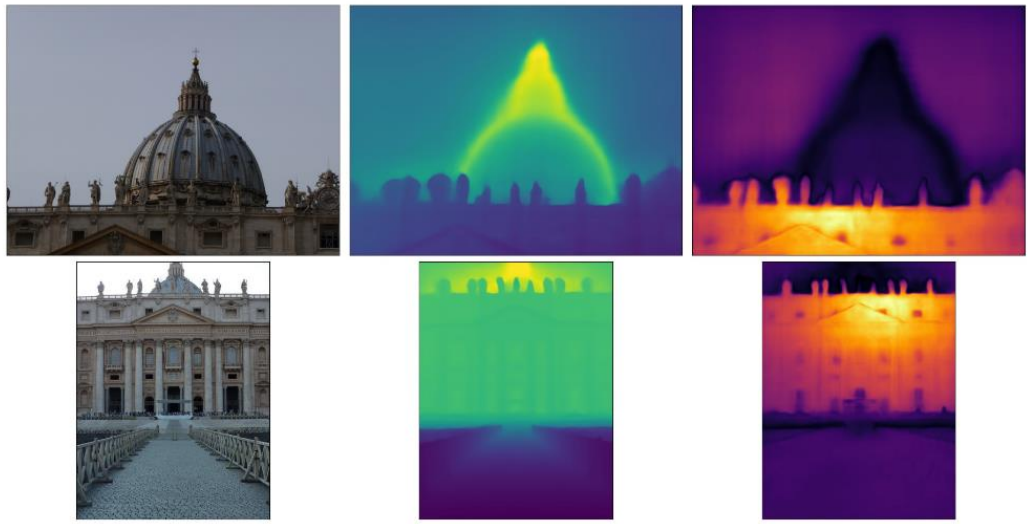
Table 2. **Left**: Monocular depth estimation on multiple benchmarks. D-Supervised, SS-Self-supervised, T-transfer (zero-shot). (Parentheses) refers to training on the same set. **Right:** Multi-view pose regression on the CO3Dv2 [94] and RealEst10K [186] with 10 random frames.

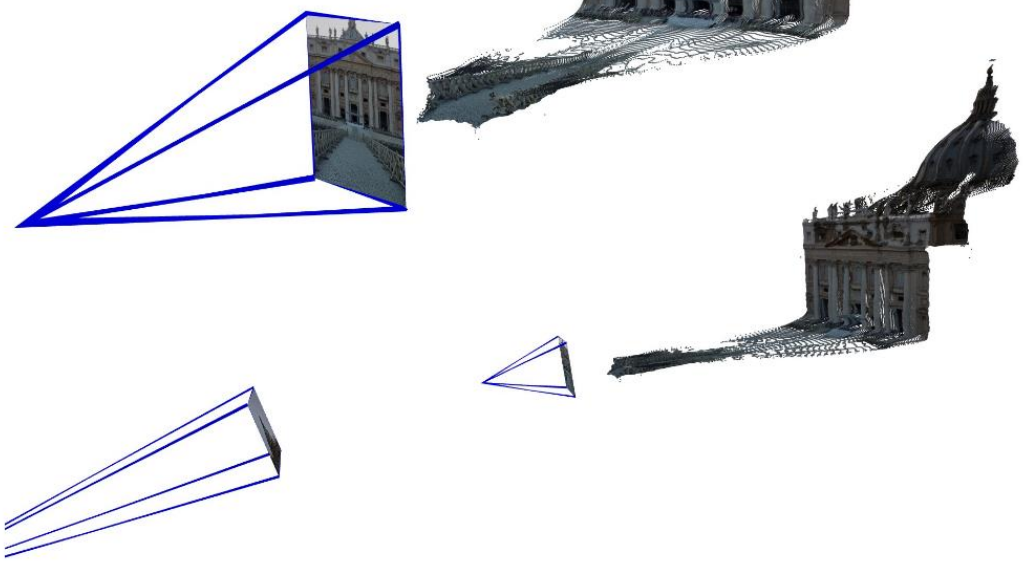
	Methods	GT Pose	GT Range	GT Intrinsics	Align	KITTI rel ↓	KITTI τ ↑	ScanNet rel ↓	ScanNet τ ↑	ETH3D rel ↓	ETH3D τ ↑	DTU rel ↓	DTU τ ↑	T&T rel↓	T&T τ ↑	Average rel↓	Average τ ↑	Average time (s)↓
(a)	COLMAP [106, 107]	✓	×	✓	×	**12.0**	**58.2**	**14.6**	**34.2**	**16.4**	**55.1**	**0.7**	**96.5**	**2.7**	95.0	**9.3**	**67.8**	≈ 3 min
	COLMAP Dense [106, 107]	✓	×	✓	×	26.9	52.7	38.0	22.5	89.8	23.2	20.8	69.3	25.7	76.4	40.2	48.8	≈ 3 min
(b)	MVSNet [161]	✓	✓	✓	×	22.7	36.1	24.6	20.4	35.4	31.4	(1.8)	(86.0)	8.3	73.0	18.6	49.4	0.07
	MVSNet Inv. Depth [161]	✓	✓	✓	×	18.6	30.7	22.7	20.9	21.6	35.6	(1.8)	(86.7)	6.5	74.6	14.2	49.7	0.32
	Vis-MVSSNet [176]	✓	✓	✓	×	**9.5**	**55.4**	8.9	33.5	**10.8**	**43.3**	**(1.8)**	**(87.4)**	**4.1**	**87.2**	**7.0**	**61.4**	0.70
	MVS2D ScanNet [160]	✓	✓	✓	×	21.2	8.7	(27.2)	(5.3)	27.4	4.8	17.2	9.8	29.2	4.4	24.4	6.6	**0.04**
	MVS2D DTU [160]	✓	✓	✓	×	226.6	0.7	32.3	11.1	99.0	11.6	(3.6)	(64.2)	25.8	28.0	77.5	23.1	0.05
(c)	DeMon [136]	✓	×	✓	×	16.7	13.4	75.0	0.0	19.0	16.2	23.7	11.5	17.6	18.3	30.4	11.9	0.08
	DeepV2D KITTI [131]	✓	×	✓	×	(20.4)	(16.3)	25.8	8.1	30.1	9.4	24.6	8.2	38.5	9.6	27.9	10.3	1.43
	DeepV2D ScanNet [131]	✓	×	✓	×	61.9	5.2	(3.8)	(60.2)	18.7	28.7	9.2	27.4	33.5	38.0	25.4	31.9	2.15
	MVSNet [161]	✓	×	✓	×	14.0	35.8	1568.0	5.7	507.7	8.3	(4429.1)	(0.1)	118.2	50.7	1327.4	20.1	0.15
	MVSNet Inv. Depth [161]	✓	×	✓	×	29.6	8.1	65.2	28.5	60.3	5.8	(28.7)	(48.9)	51.4	14.6	47.0	21.2	0.28
	Vis-MVSNet [176]	✓	×	✓	×	10.3	**54.4**	84.9	15.6	51.5	17.4	(374.2)	(1.7)	21.1	65.6	108.4	31.0	0.82
	MVS2D ScanNet [160]	✓	×	✓	×	73.4	0.0	(4.5)	(54.1)	30.7	14.4	5.0	57.9	56.4	11.1	34.0	27.5	**0.05**
	MVS2D DTU [160]	✓	×	✓	×	93.3	0.0	51.5	1.6	78.0	0.0	(1.6)	(92.3)	87.5	0.0	62.4	18.8	0.06
	Robust MVD Baseline [110]	✓	×	✓	×	**7.1**	41.9	**7.4**	**38.4**	**9.0**	**42.6**	**2.7**	**82.0**	**5.0**	**75.1**	**6.3**	**56.0**	0.06
(d)	DeMoN [136]	×	×	✓	$\|\|t\|\|$	15.5	15.2	12.0	21.0	17.4	15.4	21.8	16.6	13.0	23.2	16.0	18.3	0.08
	DeepV2D KITTI [131]	×	×	✓	med	(3.1)	(74.9)	23.7	11.1	27.1	10.1	24.8	8.1	34.1	9.1	22.6	22.7	2.07
	DeepV2D ScanNet [131]	×	×	✓	med	10.0	36.2	**(4.4)**	(54.8)	11.8	29.3	7.7	33.0	8.9	46.4	8.6	39.9	3.57
	DUSt3R 224-NoCroCo	×	×	×	med	15.14	21.16	7.54	40.00	9.51	40.07	3.56	62.83	11.12	37.90	9.37	40.39	**0.05**
	DUSt3R 224	×	×	×	med	15.39	26.69	(5.86)	(50.84)	4.71	61.74	**2.76**	**77.32**	5.54	56.38	6.85	54.59	**0.05**
	DUSt3R 512	×	×	×	med	**9.11**	**39.49**	(4.93)	**(60.20)**	**2.91**	**76.91**	3.52	69.33	**3.17**	**76.68**	**4.73**	**64.52**	0.13

Table 3. **Multi-view depth evaluation** with different settings: a) Classical approaches; b) with poses and depth range, without alignment; c) absolute scale evaluation with poses, without depth range and alignment; d) without poses and depth range, but with alignment. (Parentheses) denote training on data from the same domain. The best results for each setting are in **bold**.









VGGT: Visual Geometry Grounded Transformer

- Are we approaching a 3D foundation model?

Jianyuan Wang, Minghao Chen, Nikita Karaev,

Andrea Vedaldi, Christian Rupprecht, David Novotny

🏛️ VGGT: Visual Geometry Grounded Transformer

🌟 GitHub Repository | 🚀 Project Page

Upload a video or a set of images to create a 3D reconstruction of a scene or object. VGGT takes these images and generates all key 3D attributes, including extrinsic and intrinsic camera parameters, point maps, depth maps, and 3D point tracks.

Getting Started:

1. **Upload Your Data:** Use the "Upload Video" or "Upload Images" buttons on the left to provide your input. Videos will be automatically split into individual frames (one frame per second).
2. **Preview:** Your uploaded images will appear in the gallery on the left.
3. **Reconstruct:** Click the "Reconstruct" button to start the 3D reconstruction process.
4. **Visualize:** The 3D reconstruction will appear in the viewer on the right. You can rotate, pan, and zoom to explore the model, and download the GLB file. Note the visualization of 3D points may be slow for a large number of input images.
5. **Adjust Visualization (Optional):** After reconstruction, you can fine-tune the visualization using the options below (**click to expand**):

Please note: Our model itself usually only needs less than 1 second to reconstruct a scene. However, visualizing 3D points may take tens of seconds due to third-party rendering, which are independent of VGGT's processing time. Please be patient or, for faster visualization, use a local machine to run our demo from our GitHub repository.

Upload Video

Drop Video Here
- or -
Click to Upload

Upload Images

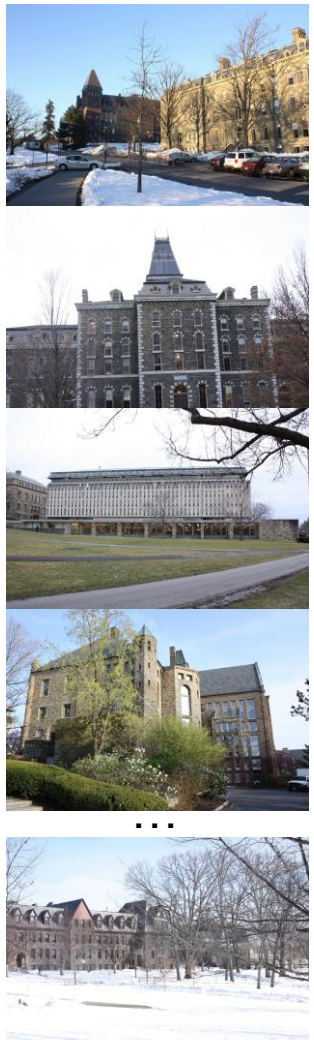
Drop File Here
- or -
Click to Upload

3D Reconstruction (Point Cloud and Camera Poses)

Please upload a video or images, then click Reconstruct.

3D Model

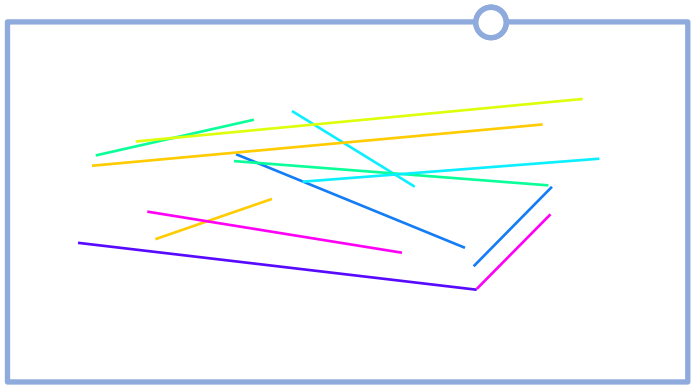
Reconstruction: Core of 3D



Video Source: COLMAP, Schönberger et al. 2016

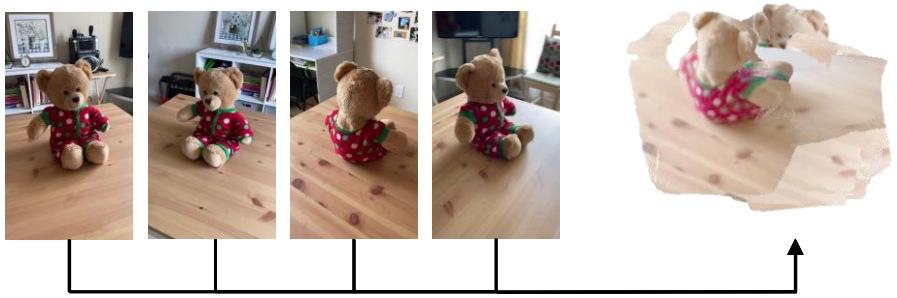
Previous Frameworks

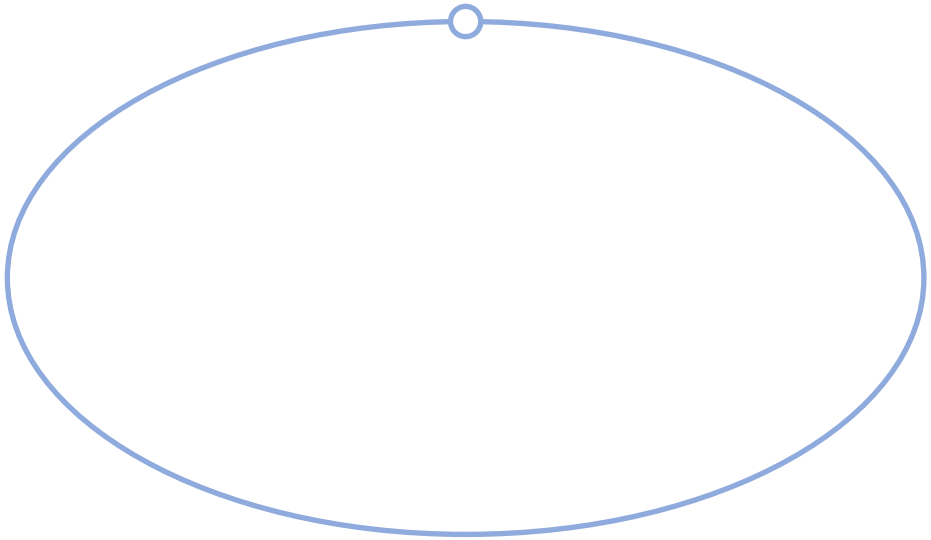
COLMAP



Bundle Adjustment

DUSt3R





Global Alignment

Optimization:

Bundle Adjustment

Global Alignment

Optimization: Bottleneck for 3D in the Deep Era

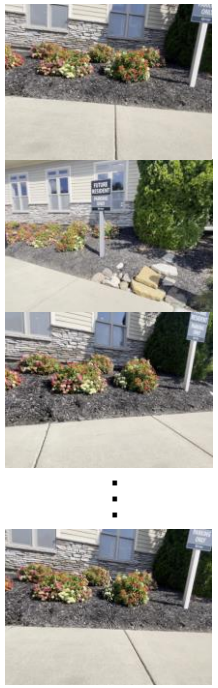
- Time-consuming
- Poor Compatibility with Deep Learning
  - Not inherently "plug-and-play"
  - Often non-differentiable
- Complexity
  - Scary for non-experts

Bundle Adjustment

Global Alignment

Let’s Reconstruct in One Go!

Images

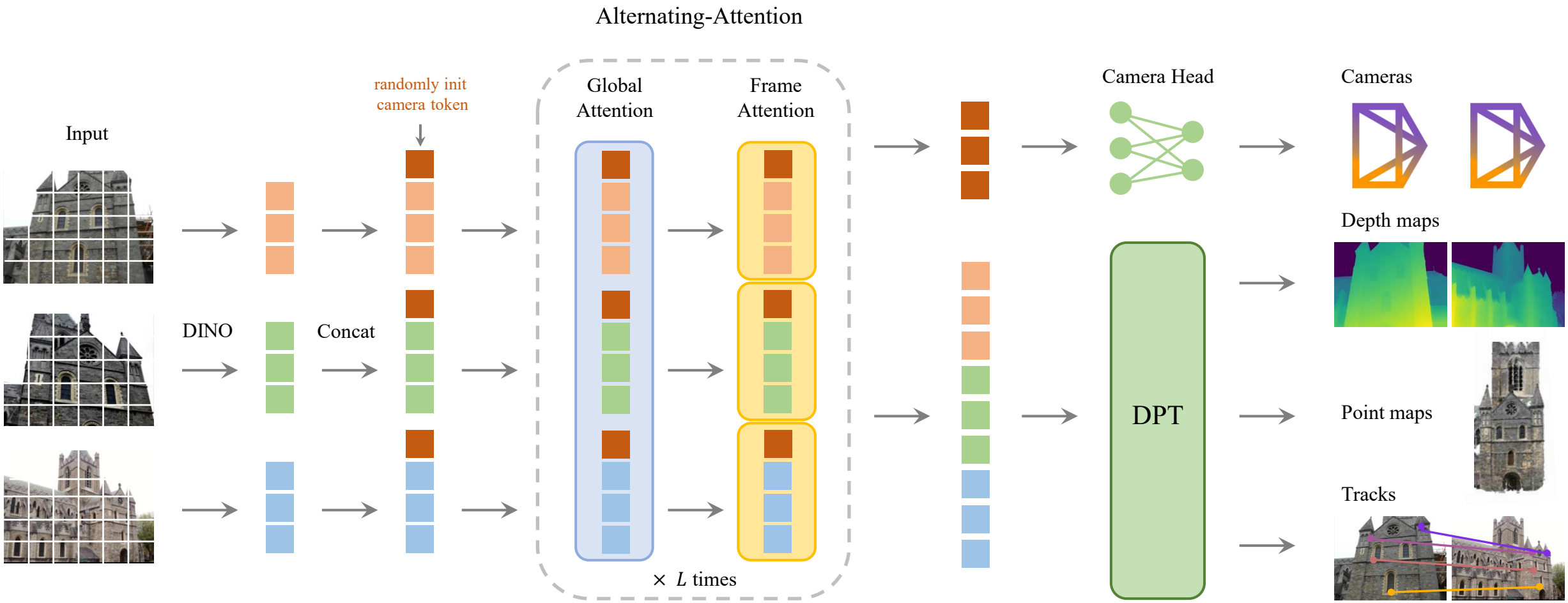


Neural Network

Reconstruction

Cameras, Depths, Points, and Correspondences

VGG Transformer



What is DPT?

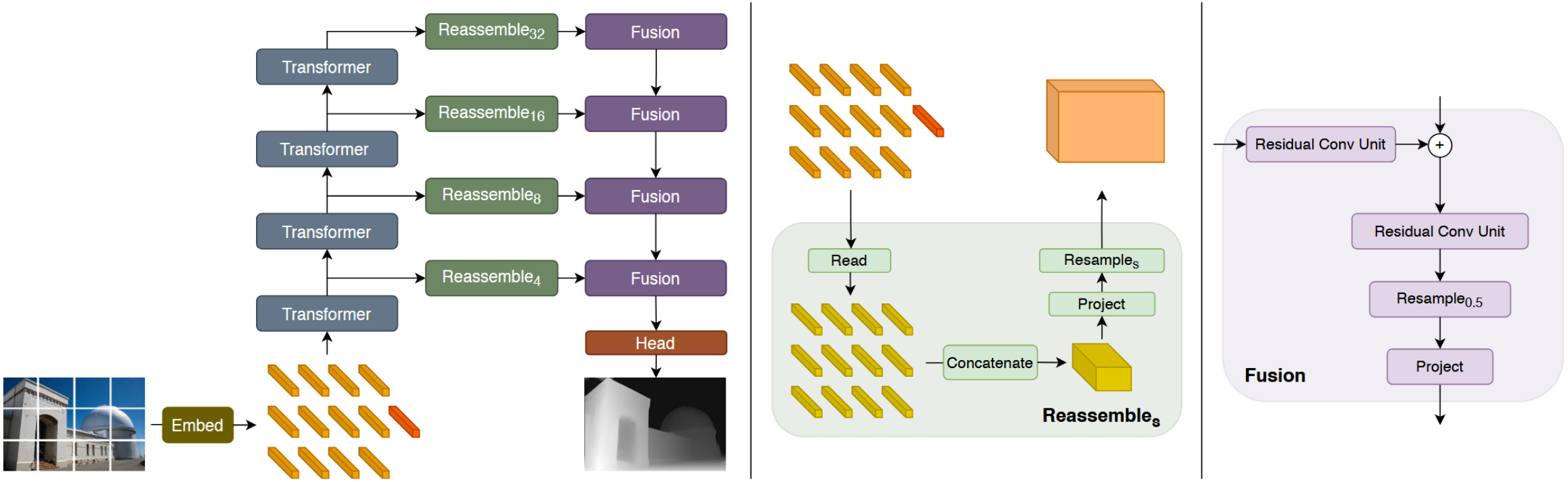


Figure 1. *Left*: Architecture overview. The input image is transformed into tokens (orange) either by extracting non-overlapping patches followed by a linear projection of their flattened representation (DPT-Base and DPT-Large) or by applying a ResNet-50 feature extractor (DPT-Hybrid). The image embedding 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.

What is DPT?

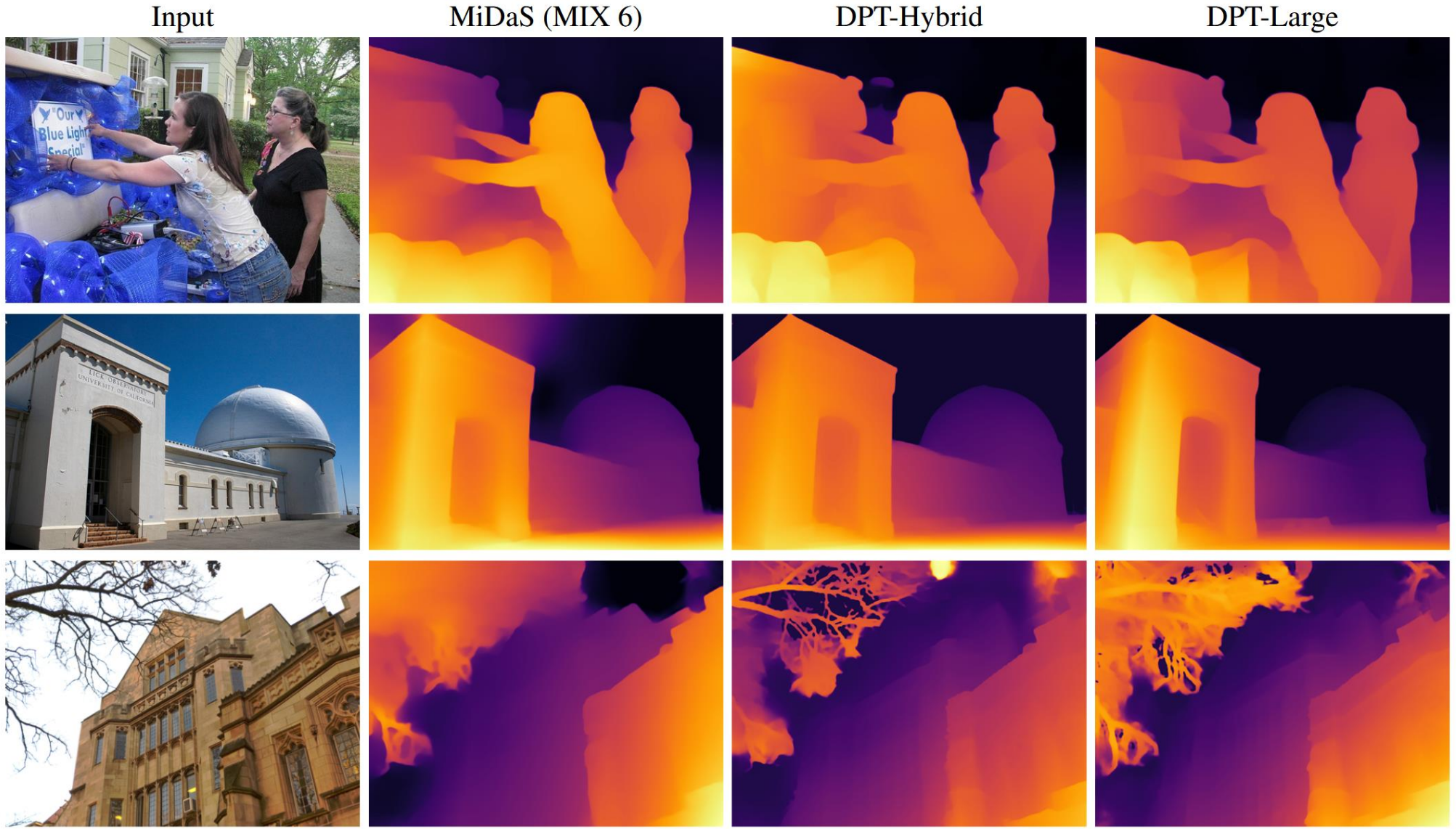
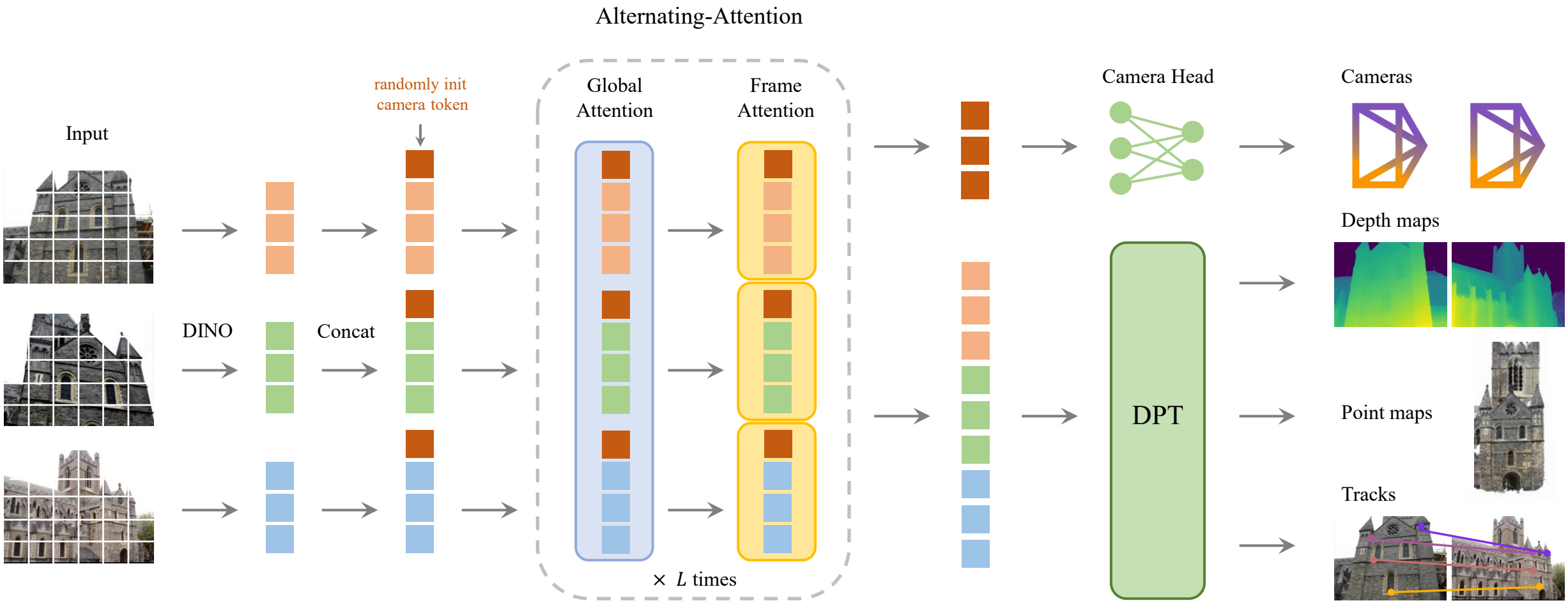


Figure 2. Sample results for monocular depth estimation. Compared to the fully-convolutional network used by MiDaS, DPT shows better global coherence (e.g., sky, second row) and finer-grained details (e.g., tree branches, last row).

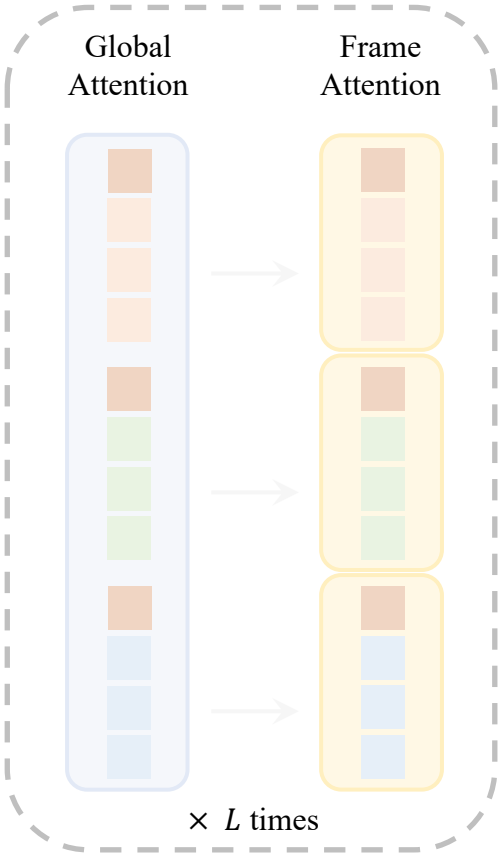
VGG Transformer



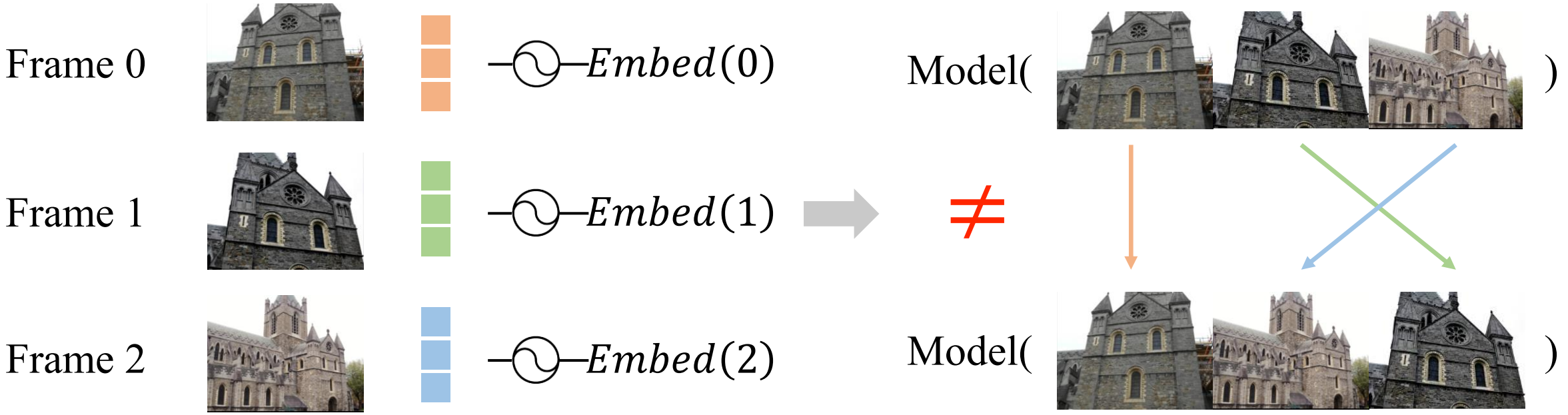
Why Alternating-Attention?

- Global Attention
  - Ensures scene-level coherence

- Frame-wise Attention
  - Eliminates **frame index embedding**
    - For permutation equivariance
    - For flexible input length



Why Alternating-Attention?



Not permutation equivariant

Why Alternating-Attention?

Frame 0  $Embed(0)$

Frame 1  $Embed(1)$

Frame 2  $Embed(2)$

⋮

Frame 842 But model never sees $Embed(842)$ during training

Why Alternating-Attention?

Frame 0

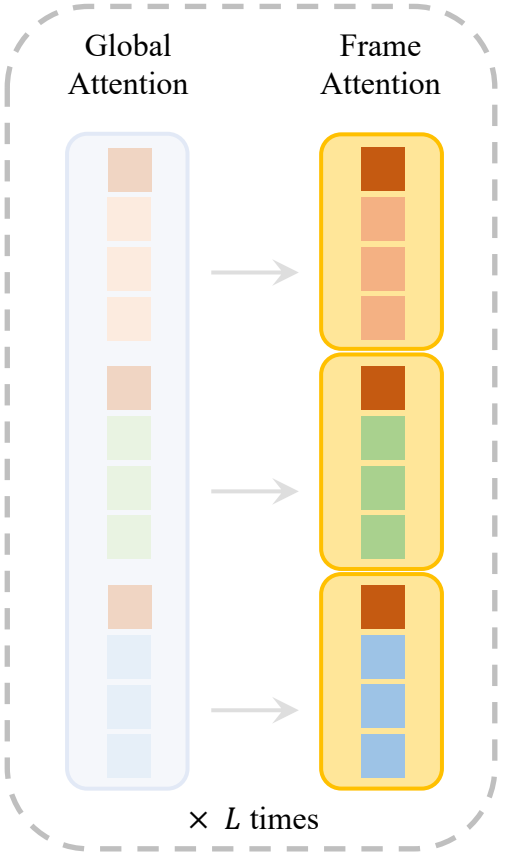
$Embed(0)$

Frame 1

$Embed(1)$

Frame 2

$Embed(2)$



Replaces frame index embedding by Frame-wise Attention

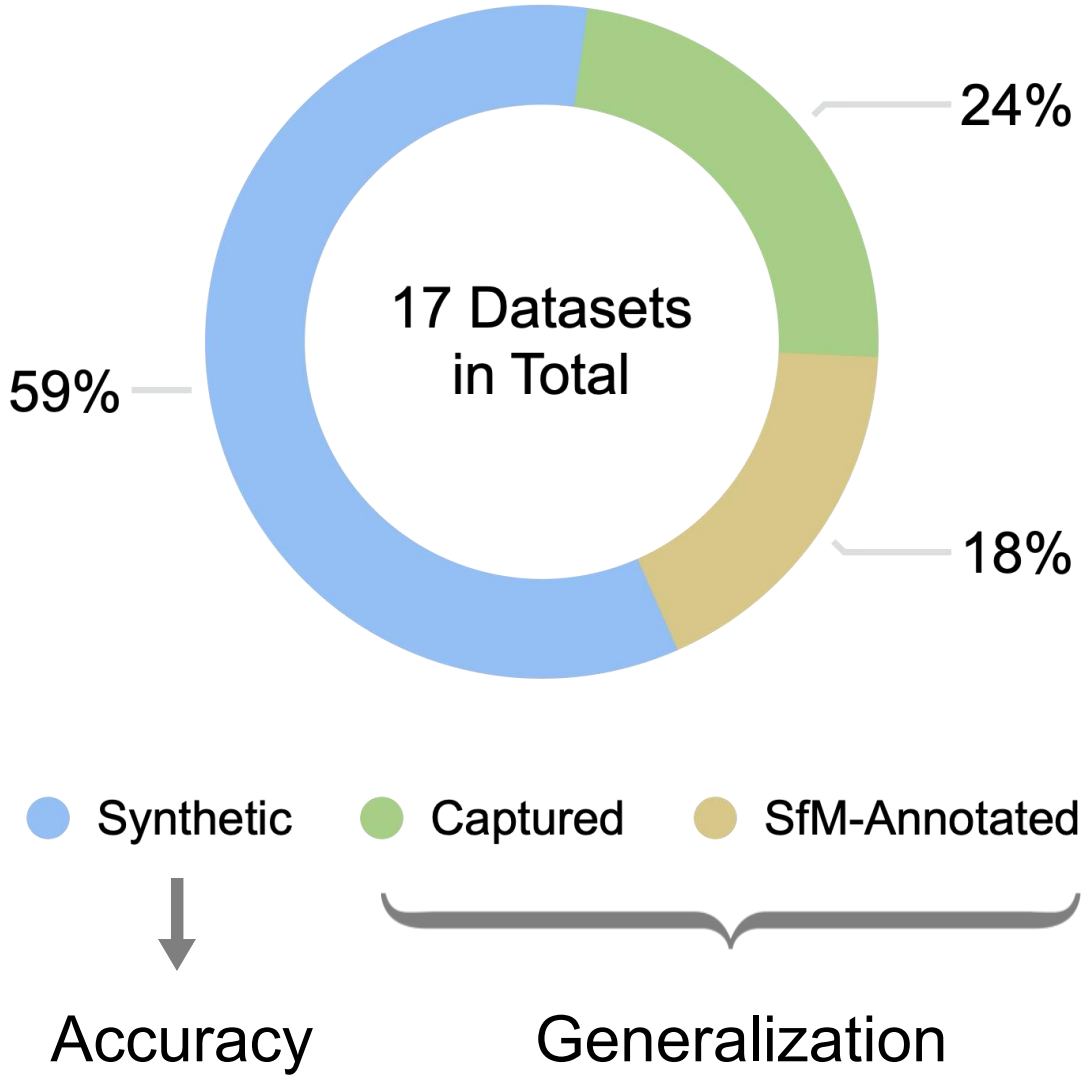
Training and Data

💡 Training:

2 to 24 frames

🔍 Inference:

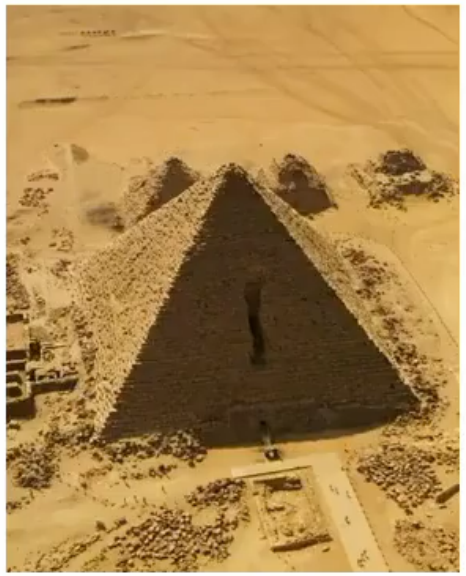
1 to 300+ frames



Results

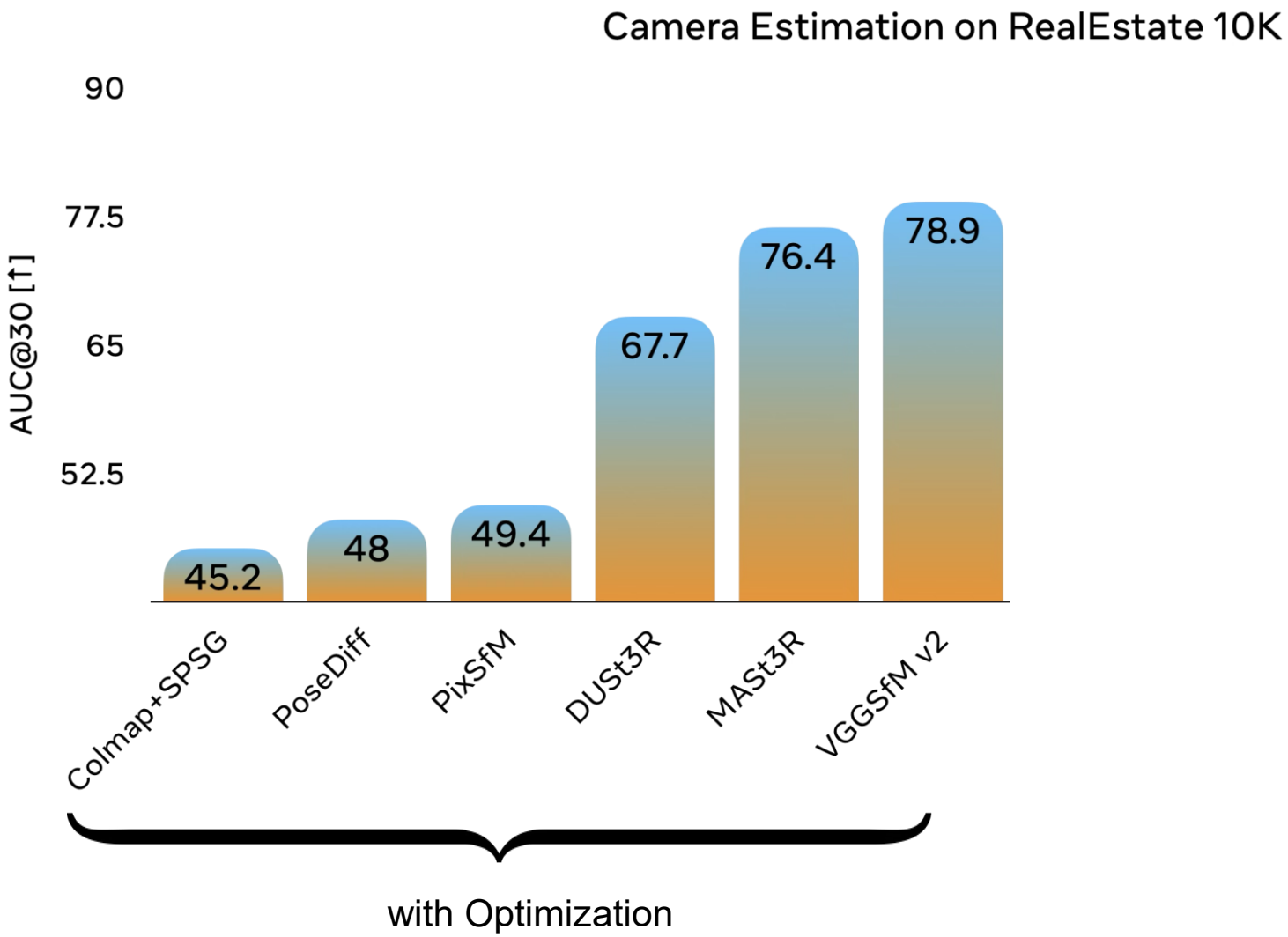
Qualitative

32 Views

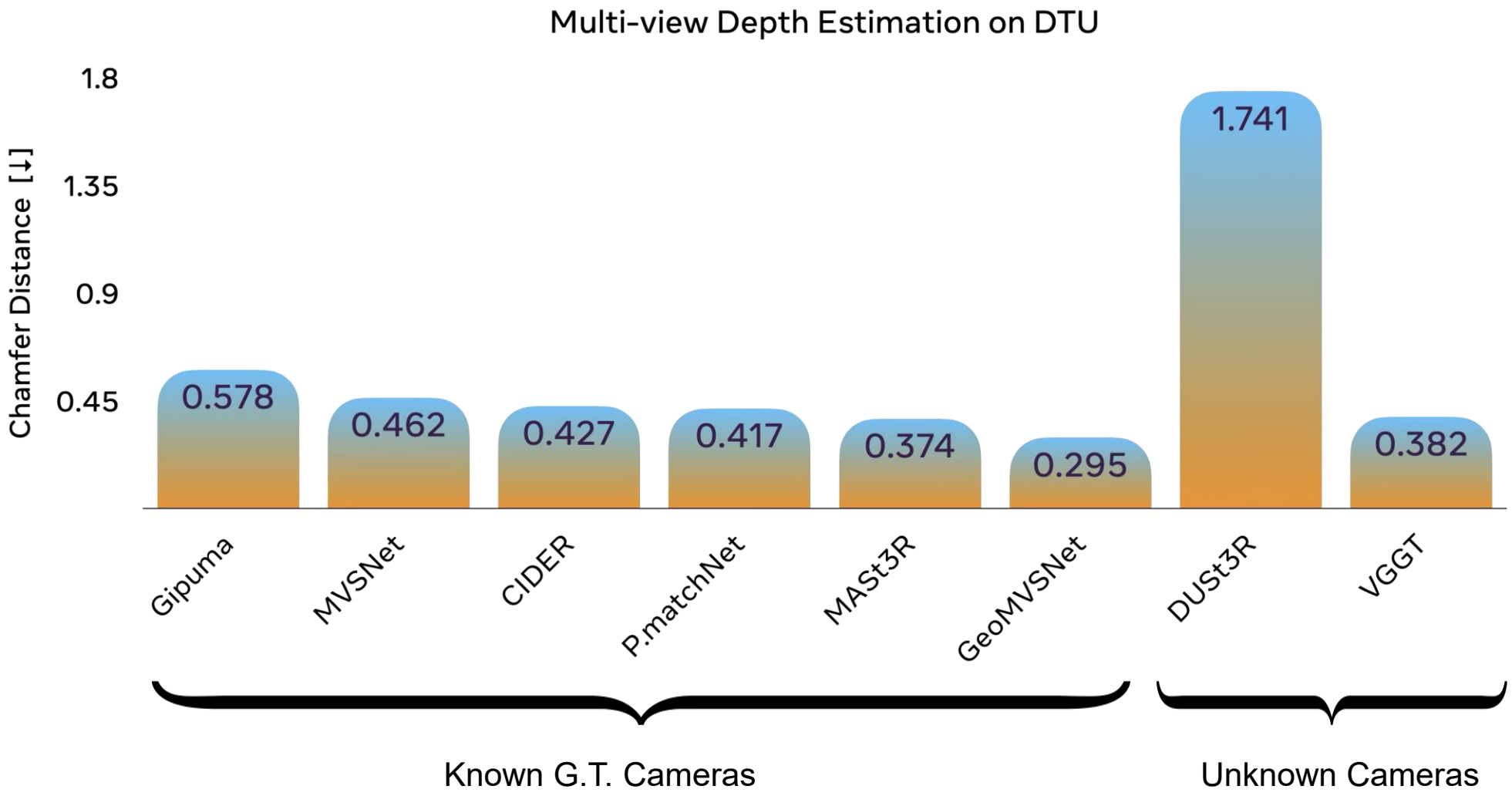


VGGT Is Accurate

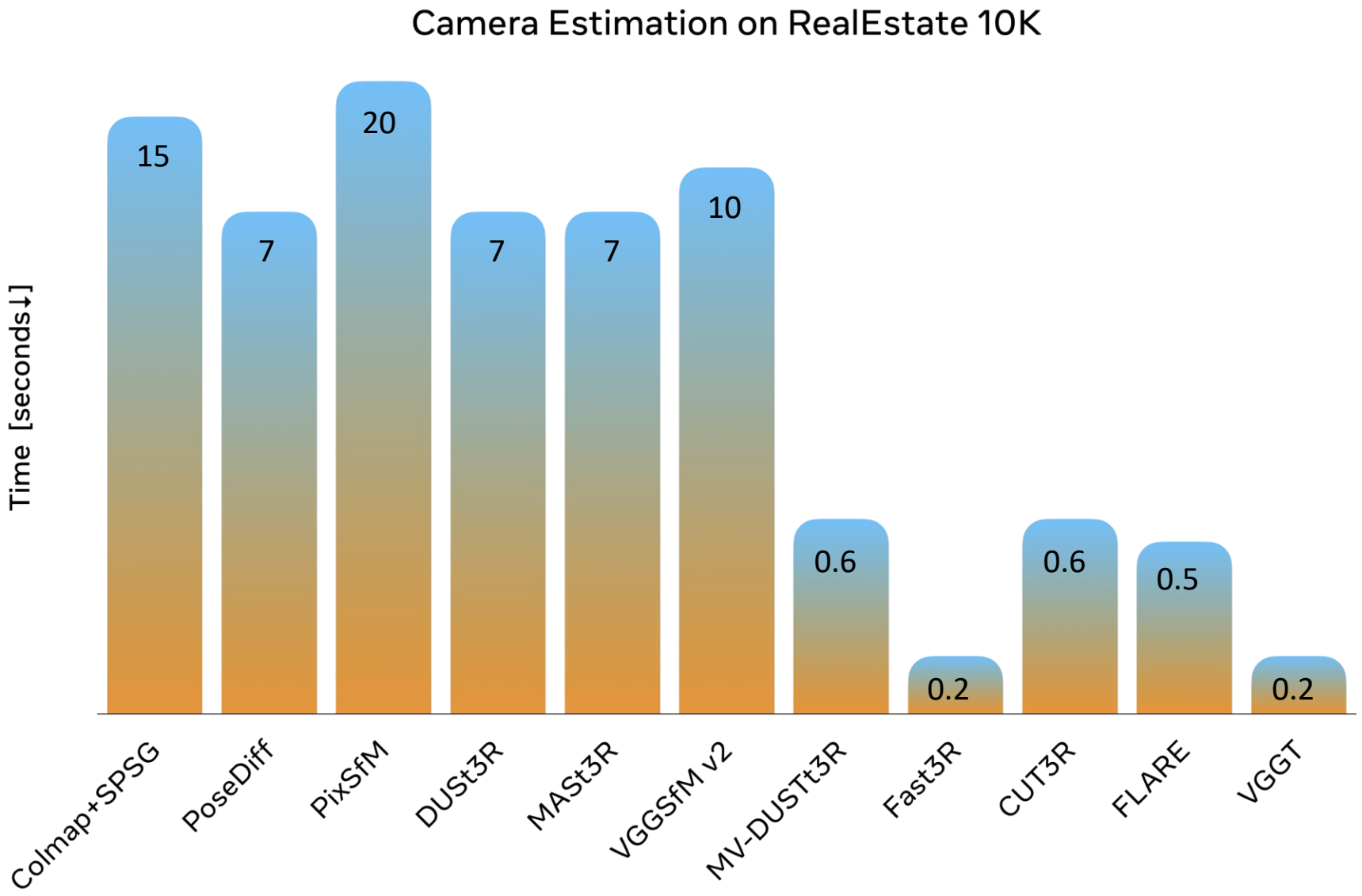
VGGT Is Accurate



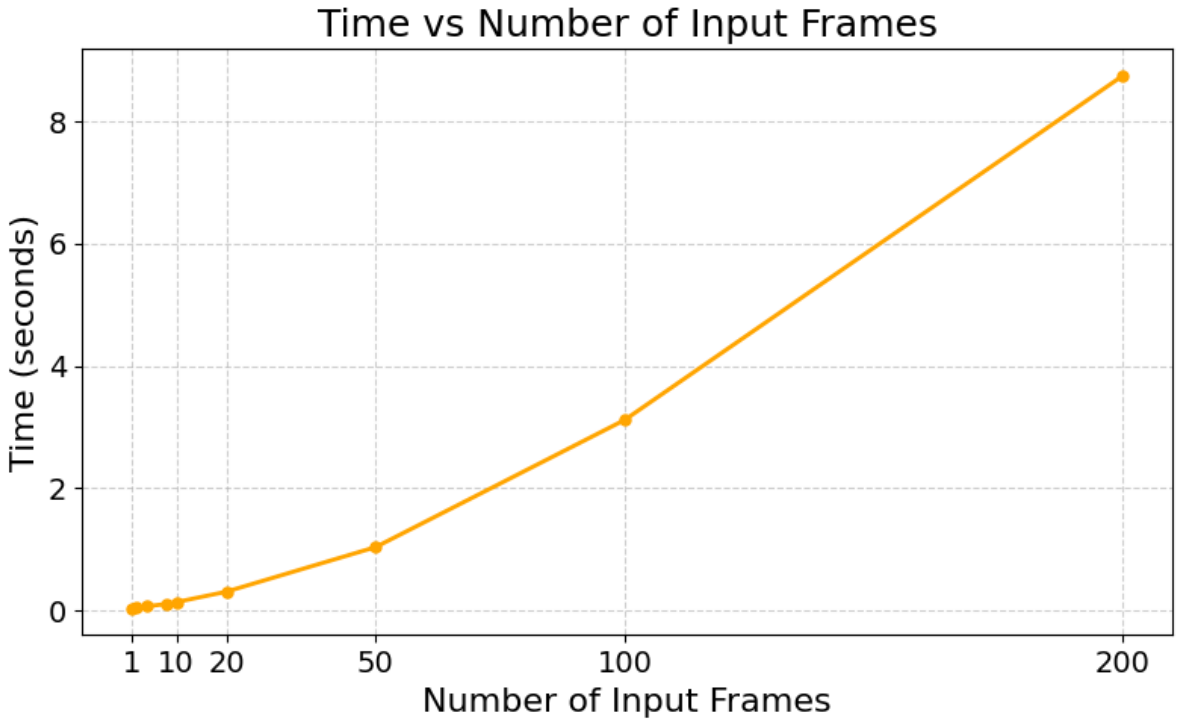
VGGT Is Accurate

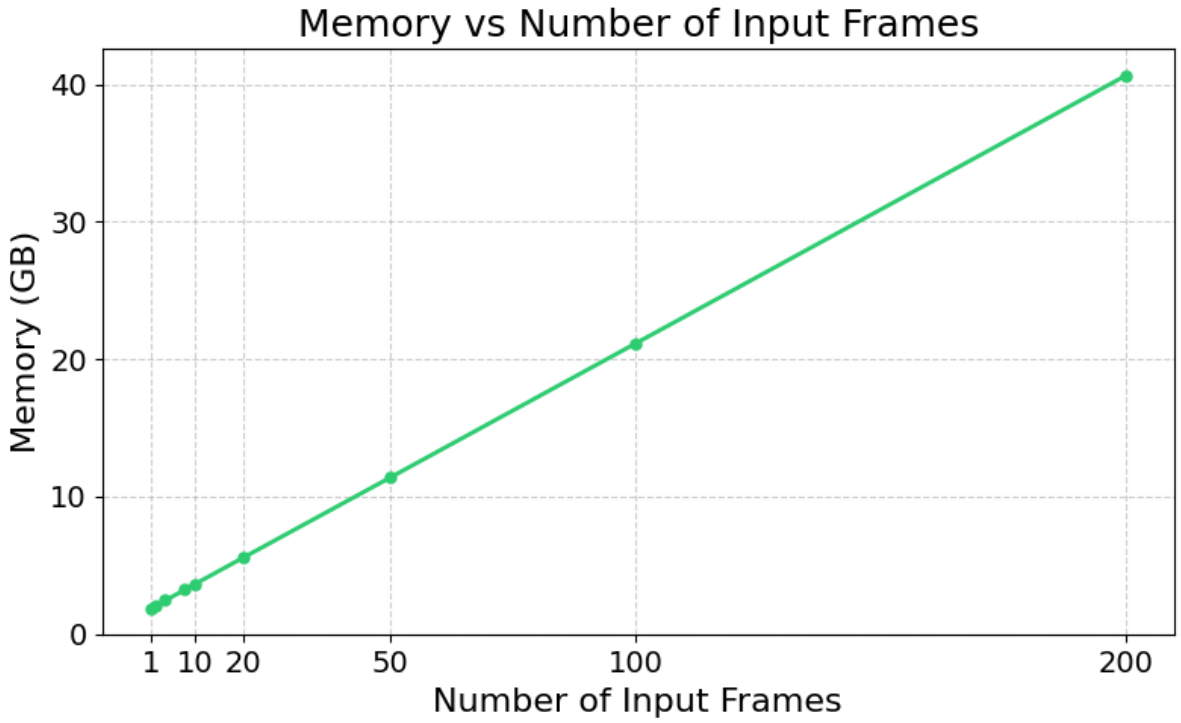


VGGT Is Fast


Camera Estimation on RealEstate 10K
Time [seconds↓]
15
7
20
7
7
10
0.6
0.2
0.6
0.5
0.2
Colmap+SPSG
PoseDiff
PixSfM
DUSt3R
MASt3R
VGGSfM v2
MV-DUSTt3R
Fast3R
CUT3R
FLARE
VGGT

Runtime and Memory

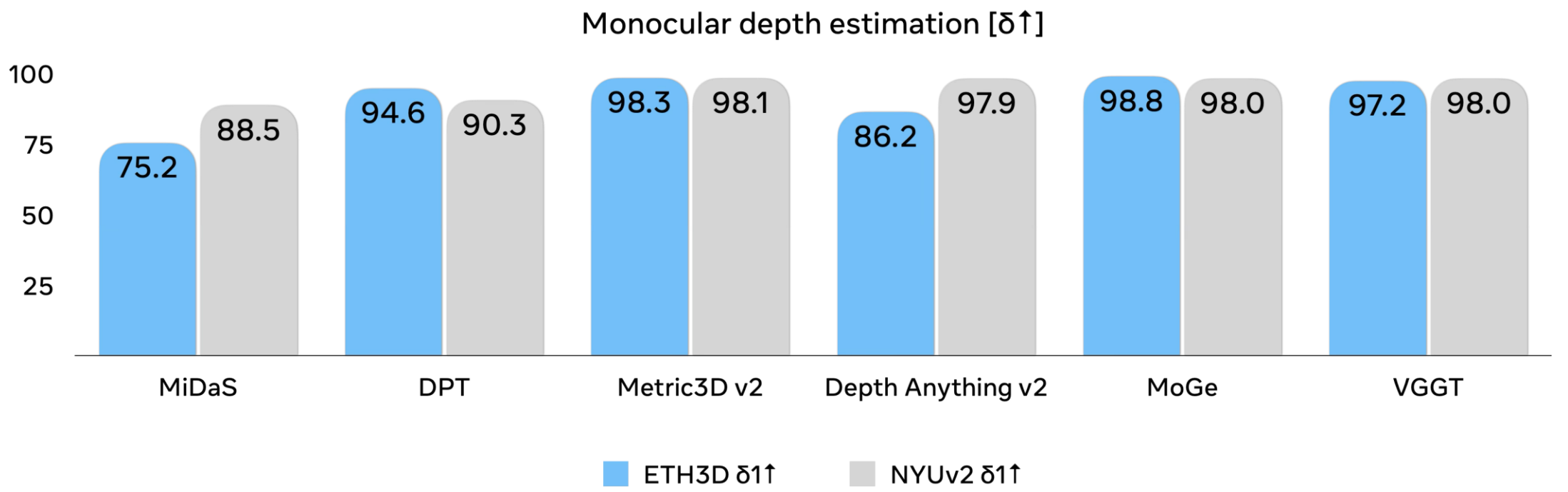




- Memory usage scales roughly linearly with input frames
- The time usage is around $O(N^{1.5})$

Zero-Shot and Finetuning

Zero-shot Monocular Depth Estimation



As good as SoTA experts – but VGGT was never trained for monocular

Zero-shot Monocular Depth Estimation

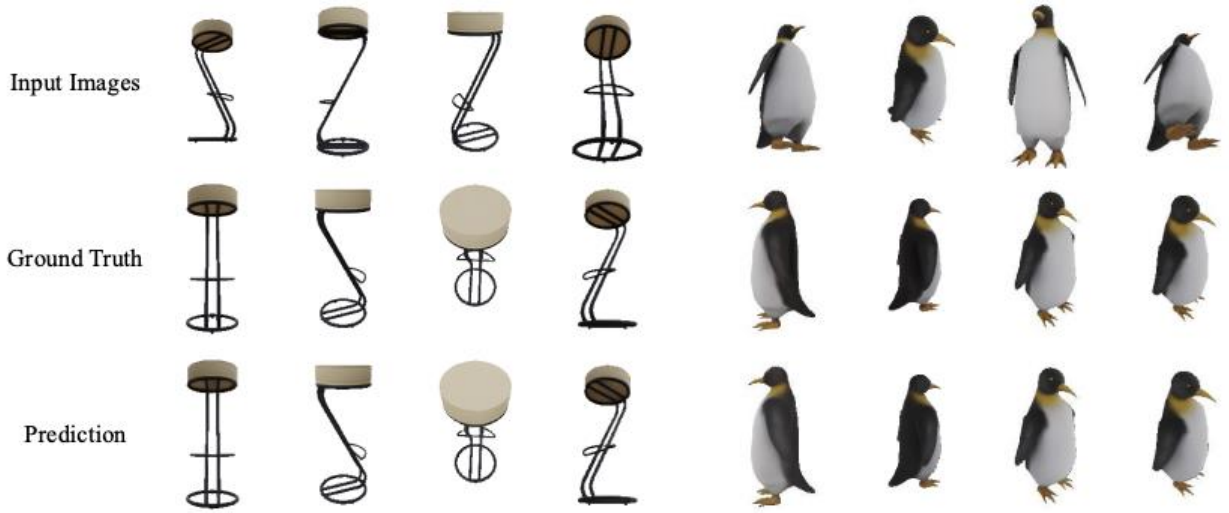
Single View

VGGT Helps Downstream Tasks

Dynamic Point Tracking

VGGT’s feature backbone
boosts CoTracker to SoTA

Novel View Synthesis



VGGT enables NVS without camera inputs
retaining comparable quality

VGGT Is General, Seamless and Practical

General	Seamless	Practical
• Diverse images • Single to hundreds of views	• Just a neural network • Standard components	• Fast and accurate • Addresses all core 3D tasks

Let’s Reconstruct in One Go!

You no longer have to be “Zisserman” for 3D Reconstruction 😉

Depth Anything 3: Recovering the Visual Space from Any Views

Haotong Lin*, **Sili Chen***, **Jun Hao Liew***, **Donny Y. Chen***, **Zhenyu Li**, **Guang Shi**, **Jiashi Feng**, **Bingyi Kang***,†

ByteDance Seed

†Project Lead, *Equal Contribution

Abstract

We present Depth Anything 3 (DA3), a model that predicts spatially consistent geometry from an arbitrary number of visual inputs, with or without known camera poses. In pursuit of minimal modeling, DA3 yields two key insights: a single plain transformer (*e.g.*, vanilla DINO encoder) is sufficient as a backbone without architectural specialization, and a singular depth-ray prediction target obviates the need for complex multi-task learning. Through our teacher-student training paradigm, the model achieves a level of detail and generalization on par with Depth Anything 2 (*DA2*). We establish a new visual geometry benchmark covering camera pose estimation, any-view geometry and visual rendering. On this benchmark, DA3 sets a new state-of-the-art across all tasks, surpassing prior SOTA *VGGT* by an average of 35.7% in camera pose accuracy and 23.6% in geometric accuracy. Moreover, it outperforms *DA2* in monocular depth estimation. All models are trained exclusively on public academic datasets.

Correspondence: Bingyi Kang

Project Page: depth-anything-3.github.io

Table 2 Comparisons with SOTA methods on pose accuracy. We report both Auc3 ↑ and Auc30 ↑ metrics. The top-3 results are highlighted as first, second, and third.

Methods	Params	HiRoom		ETH3D		DTU		7Scenes		ScanNet++	
		Auc3	Auc30	Auc3	Auc30	Auc3	Auc30	Auc3	Auc30	Auc3	Auc30
DUSt3R	0.57B	17.6	54.3	4.30	27.3	4.00	74.3	6.90	61.6	8.10	33.9
Fast3R	0.65B	25.9	77.0	8.10	44.4	9.50	79.1	19.0	78.6	17.9	72.5
MapAnything	0.56B	17.9	82.8	19.2	77.4	6.50	72.7	12.6	79.7	20.2	84.1
Pi3	0.96B	67.0	94.8	35.2	87.3	62.5	94.9	25.5	86.3	50.7	92.1
VGGT	1.19B	49.1	88.0	26.3	80.8	79.2	99.8	23.9	85.0	62.6	95.1
DA3-Giant	1.10B	80.3	95.9	48.4	91.2	94.1	99.4	28.5	86.8	85.0	98.1
DA3-Large	0.36B	58.7	94.2	32.2	86.9	70.2	96.7	29.2	86.6	60.2	94.7
DA3-Base	0.11B	19.0	83.2	15.1	74.6	60.1	95.9	20.1	82.9	25.1	83.4
DA3-Small	0.03B	9.49	75.2	8.59	62.1	30.6	91.2	14.0	78.7	10.9	71.9

Table 3 Comparisons with SOTA methods on reconstruction accuracy. For all datasets except DTU, we report the F-Score (**F1** ↑). For DTU, we report the chamfer distance (**CD** ↓, unit: mm). w/o p. and w/ p. denote without pose and with pose, indicating whether ground-truth camera poses are provided for reconstruction. The top-3 results are highlighted as first , second , and third .

Methods	Params	HiRoom		ETH3D		DTU		7Scenes		ScanNet++	
		w/o p.	w/ p.	w/o p.	w/ p.	w/o p.	w/ p.	w/o p.	w/ p.	w/o p.	w/ p.
DUSt3R	0.57B	30.1	39.5	19.7	18.8	7.60	7.97	26.6	39.8	18.9	27.3
Fast3R	0.65B	40.7	48.2	38.5	50.3	6.88	8.20	41.0	49.8	37.1	53.7
MapAnything	0.56B	32.4	69.2	54.8	71.9	7.91	3.97	44.8	55.2	39.4	71.3
Pi3	0.96B	75.8	85.0	72.7	80.6	3.28	1.72	44.2	57.5	63.1	73.3
VGGT	1.19B	56.7	70.2	57.2	66.7	2.05	1.44	47.9	51.4	66.4	70.7
DA3-Giant	1.10B	85.1	95.6	79.0	87.1	1.85	1.85	53.5	56.5	77.0	79.3
DA3-Large	0.36B	69.5	87.1	65.8	75.2	2.08	1.23	56.3	49.2	67.9	75.7
DA3-Base	0.11B	25.9	71.4	49.5	66.7	2.87	2.36	49.9	50.6	47.2	67.8
DA3-Small	0.03B	18.3	52.2	41.6	63.4	5.83	2.49	41.0	46.8	32.3	53.8


ADB SAFEGATE
BERGLING ENGEL
BAKER HOUSE GALLERY
UNTERGESCHOSS
ZUSER
ENGEL
Any number of images