# Robot Vision: Multi-view Stereo

# Prof. Friedrich Fraundorfer

SS 2022

# Outline

- Multi-view stereo principle
- Feature extraction networks
- Cost volume generation
- Cost volume regularization
- Depth inference
- Data sets
- Results

# Multi-View Stereo

- Input: set of images + camera poses (from SFM)
- Output: 3D model (as dense point cloud)



🔿 Meshlab_64bit v1.3.3 - [Project_1]	٥	×
🧔 File Edit Filters Render View Windows Tools Help		. 8 ×
□ 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2		Q





#### Plane-sweep multi-view stereo

- Classical plane sweeping stereo [8]
- Sweep family of planes at different depths with respect to reference camera
- With CNNs: Warp deep features instead of raw pixel values



$$H_i(d) = K_i \cdot R_i \cdot \left( I - \frac{(t_{ref} - t_i) \cdot n_{ref}^T}{d} \right) \cdot R_{ref}^T \cdot K_{ref}^T$$

- Advantages:
  - fast
  - usually works better in terms of completeness
  - can work on non-lambertian surfaces
- Disadvantages:
  - often huge (GPU) memory requirements
  - needs large amount of data to train on
  - might fail in a completely new environment

# **Deep Learning for MVS: Features**

- Hand-crafted Features:
  - Designed by human experts to extract a given set of chosen characteristics
  - Trade-off between accuracy and computational efficiency
  - e.g.: Census
- Learned Features:
  - Extracted via Convolutional Neural Network (CNN)
  - Learned from data

# **Deep Learning for MVS: Regularization**

- Needed to filter incorrect correspondences (e.g. from occlusions, noise)
- Traditional Regularization:
  - Find local correspondences
  - Apply regularization methods
    - Semi global matching
    - Belief propagation
    - Graph cut
    - Smoothness priors
  - Apply filters
- Learned Regularization:
  - Network learns to regularize raw feature output
  - Often 3D convolutions

- Crop/scale to fit network requirements
  - Due to convolutions, width/height usually need to be a multiple of 2<sup>n</sup> (e.g. 32 or 64)
  - Adjust camera parameters accordingly!
- Images usually need to be stacked in network -> need same sizes!
- Augment data for training: Change brightness, contrast, etc



# **Deep Feature Extraction**

- Acquired from RGB image via CNN
- Encode image information in a way that it can be compared to other images
- Can have many layers
  - Usually a combination of 2D convolutions, Normalization and ReLU
- Original neighboring information can be encoded to smaller resolution
  - Save memory for next step



# **Deep Feature Extraction: 2D Convolution**

Example: Kernel=3x3, Stride = 1



# Deep Feature Extraction: 2D Convolution

Example: Kernel=3x3, Stride = 3



# **Deep Feature Extraction: 2D Convolution**

Input and output channels can be arbitrary (modelled through more kernel weights)





# Ex. Deep Feature Extraction: Simple Feature Net





# Ex. Deep Feature Extraction: Unet





#### **Cost Volume**

- Aggregate N feature volumes to one cost volume C via homograp warping (plane sweep)
- Variance cost metric using the average feature volume:

$$C = \frac{\sum_{i=1}^{N} (F_i - \bar{F}_i)^2}{N}$$

Each point in the cost volume can be seen as a similarity measure

$$\overrightarrow{\mathbf{F}} \rightarrow \overrightarrow{\mathbf{C}}$$



# **Cost Volume Regularization**

- Raw cost volume
  - could be noise-contaminated
  - has no smoothness constraint
- Use CNNs to regularize the obtained cost volume variance
- Usually 3D convolutions



# Cost Volume Regularization

- Last 3D convolution layer maps output to single channel
- Search for lowest cost / highest probability



- Classification:
  - Predicts label
  - Discrete output: Class with highest probability
  - Can be filtered through probability threshold
  - Example: Class 4 has highest prob -> Result: 4
- Regression:
  - Predicts quantity
  - Continuous output
  - Can be filtered through entropy threshold
  - Example: 0.1\*1 + 0.1\*2 + 0.1\*3 + 0.5\*4 + 0.2\*5 = 3.6

Networ Outpu	rk It V	Class /ectoi
0.1		1
0.1		2
0.1		3
0.5		4
0.2		5

# Training loss: Classification

Multi-class classification problem with cross entropy loss:

$$loss = \sum_{\mathbf{p}} \left( \sum_{i=1}^{D} -\mathbf{P}(i, \mathbf{p}) \cdot \log \mathbf{Q}(i, \mathbf{p}) \right)$$

where:

 $\begin{array}{ll} \mathbf{p} &= \mathrm{spatial\ image\ coordinate} \\ D &= \mathrm{maximum\ depth\ value} \\ \mathbf{P}(i,\mathbf{p}) &= \mathrm{voxel\ in\ the\ probability\ volume\ } \mathbf{P} \\ \mathbf{Q}(i,\mathbf{p}) &= \mathrm{ground\ truth\ voxel} \end{array}$ 

# Training loss: Regression

Regress depth outputs using the soft argmin [7] operation and I1 loss:

soft argmin := 
$$\sum_{d=1}^{D_{max}} d \times \sigma(-c_d)$$

where:

 $D_{max} =$ maximum depth value  $c_d =$ predicted cost  $\sigma(\cdot) =$ softmax operation

$$loss = \frac{1}{N} \sum_{n=1}^{N} ||d_{n,gt} - d_{n,pred}||_{1}$$

# Post-Processing and Filtering

- Geometric verification
  - Project each pixel into different view and back
  - Check if reprojected image lies within some threshold
- Photometric verification
  - Measures matching quality for each pixel
  - Directly implemented in network: probability, standard deviation or entropy



#### Datasets

- Quality of dataset very important for training
- Benchmarks for evaluation

Examples: DTU, Tanks and Temples, ETH3D, Blended MVS

# **DTU** dataset

#### http://roboimagedata.compute.dtu.dk

- Recorded using industrial robot arm with a structured light scanner
- Indoor, small scale, different light settings, 49 or 64 images per scene
- Ground-truth available as point clouds
- "Ground-truth" depth maps available from MVSNet
  - Screened Poisson surface reconstruction: point cloud -> mesh
  - Render mesh to each viewpoint
  - Not perfect: holes and wrong labelling in depth maps
    - Attention-Aware MVS [6]: improve ground-truth depth maps

# DTU dataset



# Tanks and Temples dataset

#### https://www.tanksandtemples.org/

- Ground-truth point cloud captured with industrial laser scanner
- Outdoor and indoor environments
- high-res video available for each scene

# Tanks and Temples dataset











# ETH3D dataset

- https://www.eth3d.net/datasets
- Ground-truth point cloud from laser scan
- 13 training and 12 test scenes in high resolution
- 5 training and 5 test videos in low resolution
- Challenging
  - large image size
  - large viewpoint change
  - small amount of images
- Deep learning methods not (yet) competitive

# ETH3D dataset









# **Evaluation**

- Overall Score: mean of accuracy and completeness (DTU)
  - Measures the mean distance to the groundtruth point cloud
  - Lower is better
- F-Score: harmonic mean of precision and recall (TaT, ETH3D)
  - Measured at a certain distance threshold d
  - If either  $P(d) \rightarrow 0$  or  $R(d) \rightarrow 0$ , then  $F(d) \rightarrow 0$
  - Better summary measure than the arithmetic mean

$$F(d)=rac{2P(d)R(d)}{P(d)+R(d)}$$

### Examples

- MVSNet (ECCV 2018): CostRegNet after volume variance calculation
- R-MVSNet (CVPR 2019): regularizes 2D costmaps along depth direction via GRU to save memory
- MVSCRF (ICCV 2019): CRF after cost volume regularization
- CasMVSNet (CVPR 2020): Multiscale feature extraction, refine depth values in every step
- Cost Volume Pyramid (CVPR 2020)
- Attention-Aware MVS (CVPR 2020)

	Method	Acc.	Comp.	Overall
Geometric	Furu [6]	0.613	0.941	0.777
	Tola [27]	0.342	1.190	0.766
	Camp [2]	0.835	0.554	0.695
	Gipuma [7]	0.283	0.873	0.578
	COLMAP [25, 26]	0.400	0.664	0.532
Learning	MVSNet [32]	0.396	0.527	0.462
	R-MVSNet [33]	0.383	0.452	0.417
	SurfaceNet [14]	0.450	1.040	0.745
	MVSCRF [29]	0.371	0.426	0.398
	Point-MVSNet [4]	0.342	0.411	0.376
	CasMVSNet [9]	0.346	0.351	<u>0.348</u>
	CVP-MVSNet [31]	<u>0.296</u>	0.406	0.351
	AttMVS [20]	0.383	0.329	0.356
	Fast-MVSNet [35]	0.336	0.403	0.370
	Ours	0.354	0.393	0.373
	Ours(HR)	0.346	<u>0.345</u>	0.346

Groundtruth

Ours



Method	Mean	Family	Francis	Horse	Lighthouse	M60	Panther	Playground	Train
COLMAP [25, 26]	42.41	50.41	22.25	25.63	56.43	44.83	46.97	48.53	42.04
MVSNet [32]	43.48	55.99	28.55	25.07	50.79	53.96	50.86	47.90	34.69
R-MVSNet [33]	48.40	69.96	46.65	32.59	42.95	51.88	48.80	52.00	42.38
Point-MVSNet [4]	48.27	61.79	41.15	34.20	50.79	51.97	50.85	52.38	43.06
AttMVS [20]	60.05	73.90	62.58	44.08	64.88	56.08	59.39	63.42	56.06
CasMVSNet [9]	56.42	76.36	58.45	46.20	55.53	56.11	54.02	58.17	46.56
CVP-MVSNet [31]	54.03	76.50	47.74	36.34	55.12	57.28	54.28	57.43	47.54
MVSCRF [29]	45.73	59.83	30.60	29.93	51.15	50.61	51.45	52.60	39.68
Fast-MVSNet [35]	47.39	65.18	39.59	34.98	47.81	49.16	46.20	53.27	42.91
Ours	49.81	66.62	44.17	30.84	55.13	53.20	50.32	55.45	42.73

[1] [MVSNet] Yao, Y., Luo, Z., Li, S., Fang, T., & Quan, L. (2018). Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 767-783).

[2] [R-MVSNet] Yao, Y., Luo, Z., Li, S., Shen, T., Fang, T., & Quan, L. (2019). Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5525-5534).

[3] [MVSCRF] Xue, Y., Chen, J., Wan, W., Huang, Y., Yu, C., Li, T., & Bao, J. (2019). Mvscrf: Learning multi-view stereo with conditional random fields. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 4312-4321).

[4] [CasMVSNet] Gu, X., Fan, Z., Zhu, S., Dai, Z., Tan, F., & Tan, P. (2020). Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2495-2504).

[5] [Cost Volume Pyramid] Yang, J., Mao, W., Alvarez, J. M., & Liu, M. (2020). Cost Volume Pyramid Based Depth Inference for Multi-View Stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4877-4886).

[6] [Attention MVS] Luo, K., Guan, T., Ju, L., Wang, Y., Chen, Z., & Luo, Y. (2020). Attention-Aware Multi-View Stereo. In *Proceedings* of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 1590-1599).

#### References

[7] Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., & Bry, A. (2017). End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 66-75).

[8] Collins, R. T. (1996, June). A space-sweep approach to true multi-image matching. In *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 358-363).

[DTU] Aanæs, H., Jensen, R. R., Vogiatzis, G., Tola, E., & Dahl, A. B. (2016). Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, *120*(2), 153-168.

[Blended MVS] Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., ... & Quan, L. (2020). BlendedMVS: A Large-scale Dataset for Generalized Multi-view Stereo Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 1790-1799).

[ETH3D] Schöps, T., Schönberger, J. L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., & Geiger, A. (2018). Eth3d benchmark.

[Tanks and Temples] Knapitsch, A., Park, J., Zhou, Q. Y., & Koltun, V. (2017). Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, *36*(4), 1-13.