impa Instituto de Matemática Pura e Aplicada

Visgraf Vision and Graphics Laboratory

Paper Presentation — RomniStereo

360 e-2-e: Analysis and Synthesis of Omnidirectional Video

Rafael Romeiro
IMPA

November 11, 2025

Paper

RomniStereo: Recurrent Omnidirectional Stereo Matching

Hualie Jiang, Rui Xu, Minglang Tan and Wenjie Jiang
Insta360 Research, Shenzhen, China

IEEE Robotics and Automation Letters, 2024

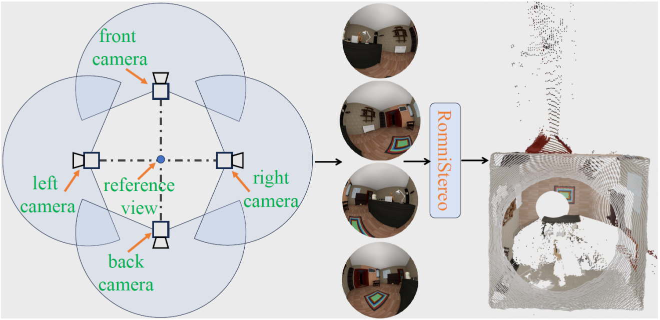
Problem Statement

Omnidirectional Stereo Matching

- Depth sensing

Rig of 4 fisheye cameras

- Wide baseline
- FoV $> 180^\circ$



Previous Work

SweepNet: Wide-baseline Omnidirectional Depth Estimation
Changhee Won, Jongbin Ryu and Jongwoo Lim
Hanyang University, Seoul, Korea
International Conference on Robotics and Automation (ICRA), 2019

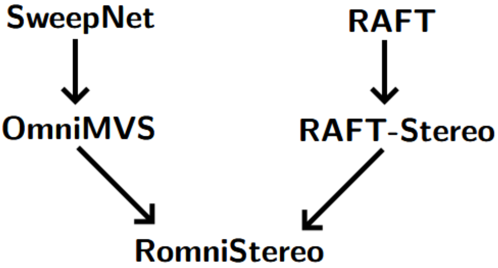
OmniMVS: End-to-End Learning for Omnidirectional Stereo Matching
Changhee Won, Jongbin Ryu and Jongwoo Lim
Hanyang University, Seoul, Korea
IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2020

Previous Work

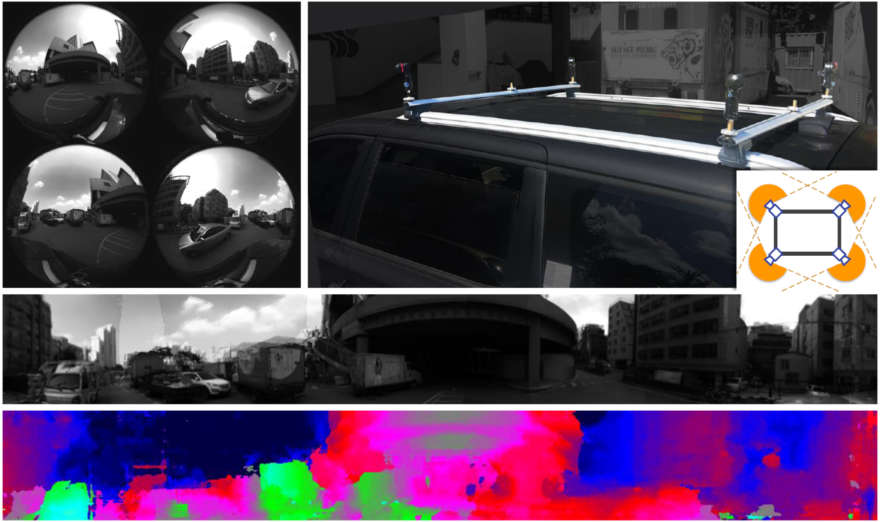
RAFT: Recurrent All-Pairs Field Transforms for Optical Flow
Zachary Teed and Jia Deng
Princeton University, New Jersey, United States
European Computer Vision Association (ECVA), 2020

RAFT-Stereo: Multilevel Recurrent Field Transforms for Stereo Matching
Lahav Lipson, Zachary Teed and Jia Deng
Princeton University, New Jersey, United States
International Conference on 3D Vision (3DV), 2021

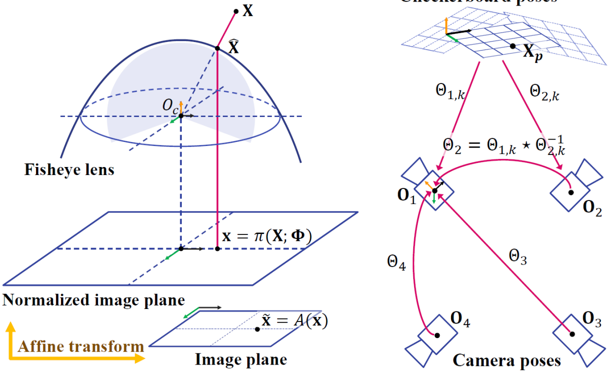
Previous Work



SweepNet



SweepNet - Camera Calibration



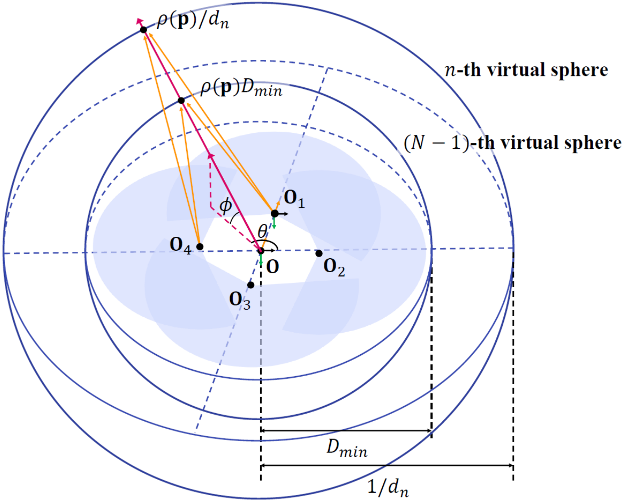
$$\min_{\substack{\mathbf{\Phi}_i, A_i \\ \Theta_i, \Theta_k}} \sum_{(i,k)} \sum_{p} \left\| \tilde{\mathbf{x}}_{i,p} - A_i\left(\Pi\left(M(\Theta_i \star \Theta_k)\begin{bmatrix}\mathbf{X}_p \\ 1\end{bmatrix}; \mathbf{\Phi}_i\right)\right)\right\|^2$$

SweepNet - Spherical Sweeping

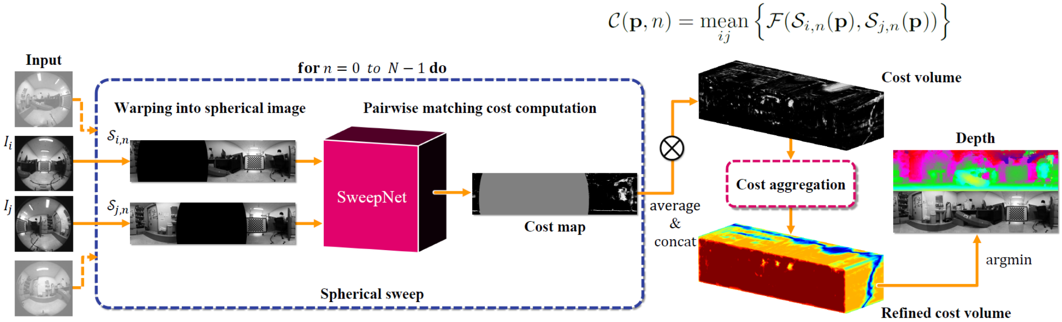
$$\mathbf{p} = (\theta, \phi)$$

$$\rho(\mathbf{p}) = (\cos(\phi)\cos(\theta),\ \sin(\phi),\ \cos(\phi)\sin(\theta))^\top$$

$$\mathcal{S}_{i,n}(\mathbf{p}) = I_i\left(\Pi_i\left(M(\Theta_i^*)\begin{bmatrix}\rho(\mathbf{p})/d_n \\ 1\end{bmatrix}\right)\right)$$



SweepNet - Flowchart



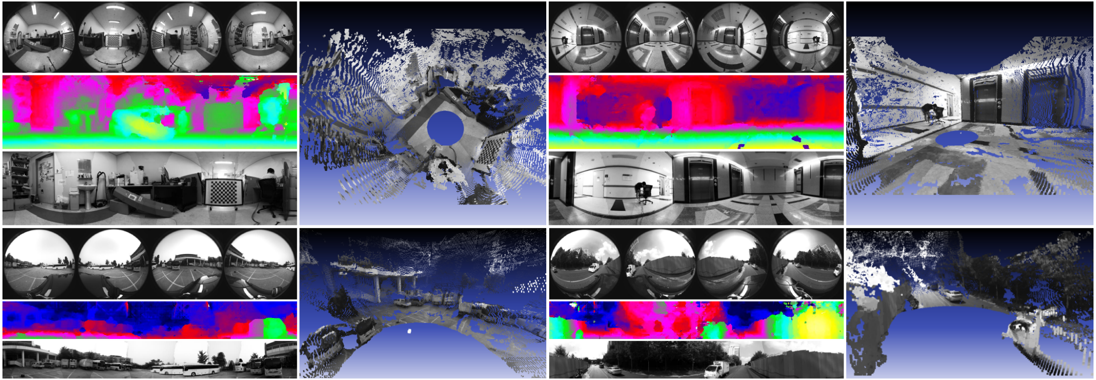
SweepNet - Architecture

The architecture of the proposed network is detailed in Table I. As shown in Fig. 4, the input of the network is a pair of gray scale spherical images acquired from (1). To ensure that the horizontal ends are connected, we add the circular column padding to the input spherical images. The conv1∼18 layers are Siamese residual blocks [31] for learning the unary feature extraction. We reduce the size of the input image in half for the larger receptive field, which helps the network learns from global context. The output feature maps are concatenated, and then the features are upsampled using transposed convolution. Finally, the network outputs the $W \times H$ cost map which ranges from 0 to 1, through fully connected layers and a sigmoid layer.

Layer	Property	Output Dim.
input	add circular column padding	$(W + 4) \times H$
conv1	5×5, 32, s 2, p_W 0, p_H 2	
conv2	3×3, 32, s 1, p 1	
conv3	3×3, 32, s 1, p 1, add conv1	$\frac{1}{2}W \times \frac{1}{2}H \times 32$
conv4-17	repeat conv2-3	
conv18	3×3, 32, s 1, p 1	
concat		$\frac{1}{2}W \times \frac{1}{2}H \times 64$
conv19	3×3, 128, s 1, p 1	$\frac{1}{2}W \times \frac{1}{2}H \times 128$
deconv1	3×3, 128, s 2, p 1	$W \times H \times 128$
conv20	3×3, 128, s 1, p 1	$W \times H \times 128$
fc1-4	1×1, 256	$W \times H \times 256$
fc5	1×1, 1, no ReLu	$W \times H$
sigmoid		$W \times H$

TABLE I: SweepNet has 20 convolutional layers and a transposed convolutional layer followed by 5 fully connected layers. Each properties (s, p) means (stride, padding) in the convolutional block.

SweepNet - Results

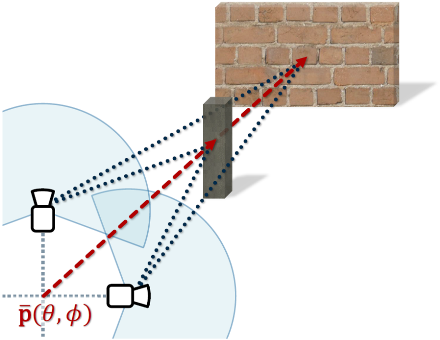


SweepNet - Core Idea and Limitations

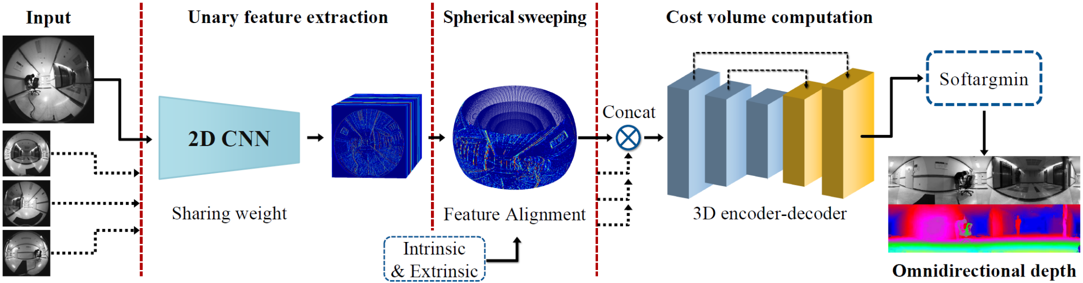
- Introduced spherical sweeping
- At each depth hypothesis, pairs of spherical images are compared using a 2D CNN that scores the matching cost
- Network ignores spatial smoothness of neighboring pixels' depth
- No learning-based depth regression

OmniMVS - The end-to-end approach

- Integrates spherical sweeping into an end-to-end deep stereo pipeline
- Instead of predicting costs per sweep independently, constructs a full spherical cost volume and processes it with a 3D CNN exploiting spatial and depth consistency
- Learns to regress continuous depth rather than picking the minimum cost



OmniMVS - Flowchart



OmniMVS - Architecture

The architecture of the proposed network is detailed in Table 1. The input of the network is a set of grayscale fisheye images. We use the residual blocks [9] for the unary feature extraction, and the dilated convolution for the larger receptive field. The output feature map size is half ($r = 2$) of the input image. Each feature map is aligned by the spherical sweeping (Sec. 3.2), and transferred to the spherical feature by a 3×3 convolution. The spherical feature maps are concatenated and fused into the 4D initial cost volume by a $3 \times 3 \times 3$ convolution. We then use the 3D encoder-decoder architecture [14] to refine and regularize the cost volume using the global context information.

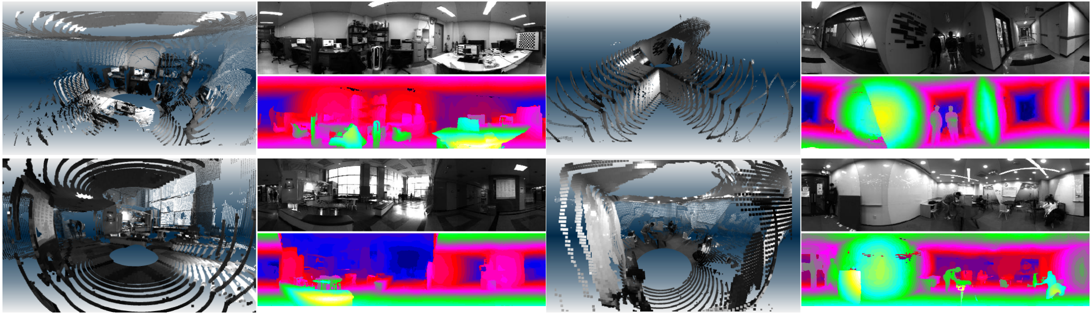
Finally, the inverse depth index $\hat{n}$ can be computed by the softargmin [14] as

$$\hat{n}(\theta, \phi) = \sum_{n=0}^{N-1} n \times \frac{e^{-\mathcal{C}(\phi,\theta,n)}}{\sum_{\nu} e^{-\mathcal{C}(\phi,\theta,\nu)}}$$

where $\mathcal{C}$ is the $(H \times W \times N)$ regularized cost volume.

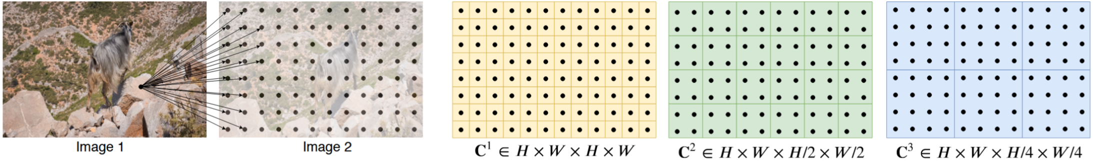
	Name	Layer Property	Output (H, W, N, C)
Unary feature extraction	Input		$H_I \times W_I$
	conv1	$5 \times 5, 32$	$\frac{1}{2}H_I \times \frac{1}{2}W_I \times 32$
	conv2	$3 \times 3, 32$	
	conv3	$3 \times 3, 32$, add conv1	
	conv4-11	repeat conv2-3	
	conv12-17	repeat conv2-3 with dilate = 2, 3, 4	
Spherical sweeping	warp		$H \times W \times \frac{1}{2}N \times 32$
	transference	$3 \times 3 \times 1, 32$	$\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times 32$
	concat(4)*		$\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times 128$
	fusion	$3 \times 3 \times 3, 64$	$\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times 64$
Cost volume computation	3Dconv1-3	$3 \times 3 \times 3, 64$	$\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times 64$
	3Dconv4-6	from 1, $3 \times 3 \times 3, 128$	$\frac{1}{4} \times \frac{1}{4} \times \frac{1}{4} \times 128$
	3Dconv7-9	from 4, $3 \times 3 \times 3, 128$	$\frac{1}{8} \times \frac{1}{8} \times \frac{1}{8} \times 128$
	3Dconv10-12	from 7, $3 \times 3 \times 3, 128$	$\frac{1}{16} \times \frac{1}{16} \times \frac{1}{16} \times 128$
	3Dconv13-15	from 10, $3 \times 3 \times 3, 256$	$\frac{1}{32} \times \frac{1}{32} \times \frac{1}{32} \times 256$
	3Ddeconv1	$3 \times 3 \times 3, 128$, add 3Dconv12	$\frac{1}{16} \times \frac{1}{16} \times \frac{1}{16} \times 128$
	3Ddeconv2	$3 \times 3 \times 3, 128$, add 3Dconv9	$\frac{1}{8} \times \frac{1}{8} \times \frac{1}{8} \times 128$
	3Ddeconv3	$3 \times 3 \times 3, 128$, add 3Dconv6	$\frac{1}{4} \times \frac{1}{4} \times \frac{1}{4} \times 128$
	3Ddeconv4	$3 \times 3 \times 3, 64$, add 3Dconv3	$\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times 64$
	3Ddeconv5	$3 \times 3 \times 3, 1$	$H \times W \times N$
	softargmin		$H \times W$

OmniMVS - Results

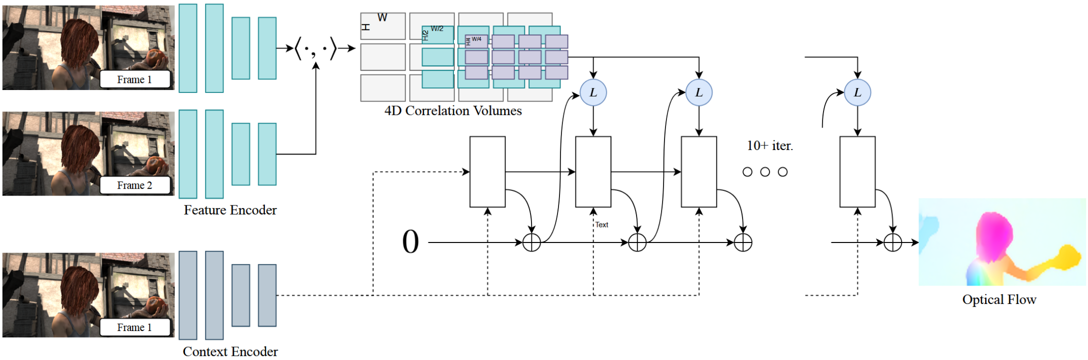


RAFT

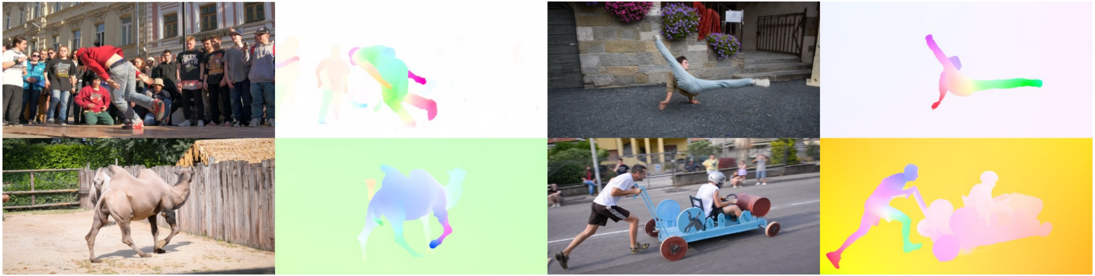
- Estimate 2D motion vectors between two consecutive images (optical flow)
- Compute correlations between all pixel pairs (dense 4D cost volume)



RAFT - Flowchart



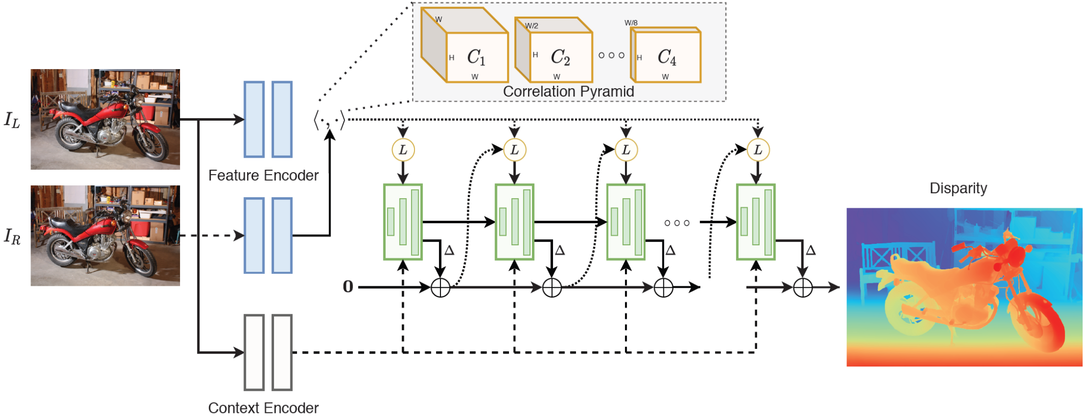
RAFT - Results



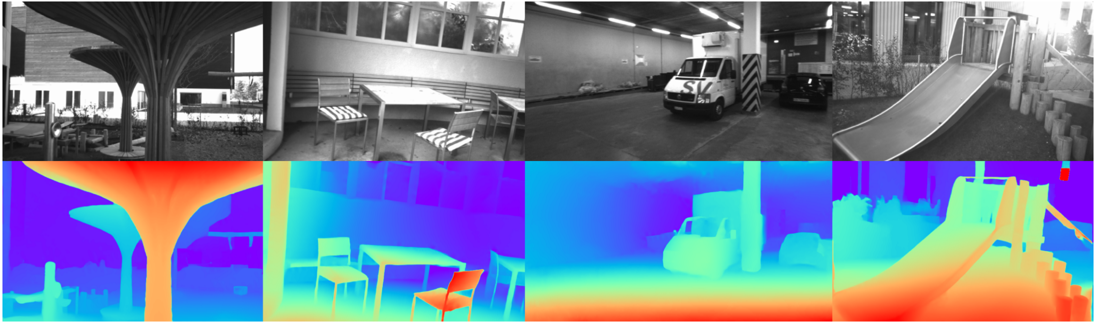
RAFT-Stereo

- RAFT assumes **temporal** consistency; stereo has **geometric** consistency
- Adapt the recurrent update framework from 2D optical flow to 1D stereo disparity
- Redesig the correlation computation and update rules to exploit the epipolar constraint
- Introduces a multi-level recurrent framework: estimates disparity at coarse resolution, refines iteratively at higher resolutions

RAFT-Stereo - Flowchart



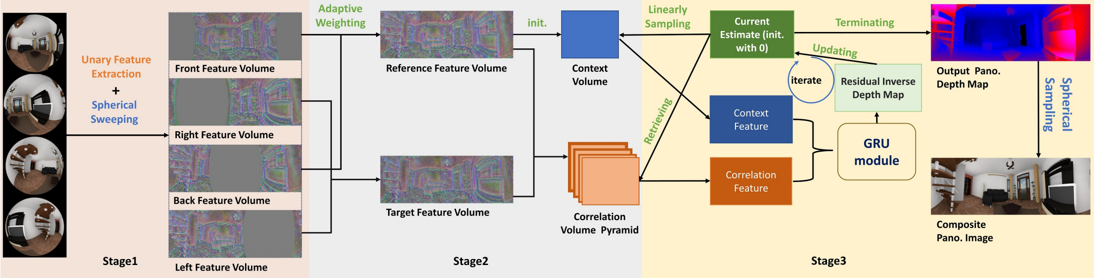
RAFT-Stereo - Results



RomniStereo

- Proposes an opposite adaptive weighting scheme that transforms the output of spherical sweeping into the inputs required by the recurrent update network
- Bridge between OmniMVS and RAFT-Stereo
- Avoid heavy 3D encoder-decoder cost-volume regularisation

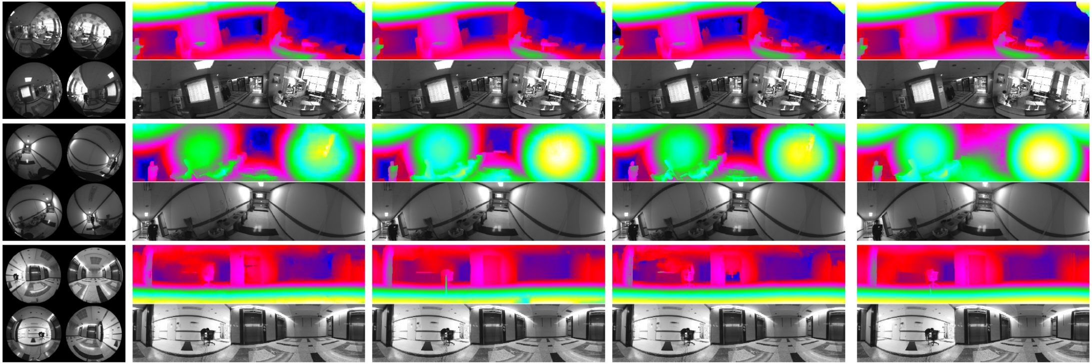
RomniStereo - Flowchart



RomniStereo - Opposite adaptive weighting

- Multi-Layered Perceptron (MLP) with sigmoid activation
- Grid Embedding: Concatenate the the normalized spherical grid coordinates $G_i(\theta, \phi, n)$ to the feature volumes input to the MLP.
  That is, input to MLP $= [S_a, S_b, G]$ at each location.
- The weighting allows the network to learn which camera view offers better feature information at each spherical cell (depending on occlusion, coverage, view-angle, etc.)
- Smooth blending avoids harsh seams or visible switching artifacts

RomniStereo - Qualitative Comparison



Input | OmniMVS$^{+}_{4}$-ft | RomniStereo$_4$-ft | OmniMVS$^{+}_{32}$-ft | RomniStereo$_{64}$-ft

RomniStereo - Quantitative Comparison

Dataset	OmniThings					OmniHouse					Run Time
Metric	>1	>3	>5	MAE	RMS	>1	>3	>5	MAE	RMS	(s)
Non-learning based method											
Sphere-Stereo [23]	80.01	56.67	44.06	9.14	14.06	65.84	27.29	12.84	2.82	4.60	0.21
Trained on OmniThings only											
OmniMVS$^{+}_{4}$ [12]	46.01	21.00	13.59	2.97	6.48	37.77	13.80	7.43	1.88	3.93	0.11
RomniStereo$_{4}$	35.61	17.05	11.46	2.52	6.13	21.82	9.24	5.67	1.33	2.96	0.09
OmniMVS$^{+}_{8}$ [12]	32.26	13.36	8.67	2.05	5.21	29.52	10.34	5.96	1.62	3.53	0.19
RomniStereo$_{8}$	28.67	12.90	8.64	1.99	5.31	20.02	8.00	4.70	1.17	2.66	0.10
OmniMVS [11]	47.72	15.12	8.91	2.40	5.27	30.53	10.29	6.27	1.72	4.05	0.82
S-OmniMVS [13]	28.03	10.40	6.33	1.48	**3.68**	18.86	8.05	4.90	1.06	2.41	-
OmniMVS$^{+}_{32}$-IS [12]	24.11	9.38	5.84	1.45	4.14	23.91	8.97	5.63	1.41	3.33	0.72
OmniMVS$^{+}_{32}$ [12]	20.70	8.18	5.49	1.37	4.11	19.89	5.89	3.99	1.30	2.64	0.82
RomniStereo$_{32}$	20.42	8.49	5.81	1.39	4.22	12.13	4.73	3.02	0.80	1.85	0.21
RomniStereo$_{64}$	**17.77**	**7.52**	**5.00**	**1.22**	3.90	**10.52**	**4.05**	**2.69**	**0.74**	**1.73**	0.44
Finetuned on OmniHouse and Sunny											
OmniMVS$^{+}_{4}$-ft [12]	53.99	35.38	27.57	5.68	9.98	15.40	5.00	2.85	0.86	1.98	0.11
RomniStereo$_{4}$-ft	50.01	33.22	26.30	5.38	9.59	11.45	4.52	2.89	0.77	1.92	0.09
RomniStereo$_{8}$-ft	44.50	28.61	22.05	4.43	8.46	8.66	3.36	2.14	0.59	1.56	0.10
OmniMVS-ft [11]	50.28	22.78	15.60	3.52	7.44	21.09	4.63	2.58	1.04	1.97	0.82
S-OmniMVS-ft [13]	-	-	-	-	-	6.99	**1.79**	**0.97**	**0.42**	**1.06**	-
OmniMVS$^{+}_{32}$-ft [12]	44.79	27.17	20.41	4.23	8.42	9.70	3.51	2.13	0.64	1.69	0.82
RomniStereo$_{32}$-ft	34.32	19.76	14.22	2.81	6.47	6.02	2.49	1.73	0.49	1.31	0.21
RomniStereo$_{64}$-ft	**29.84**	**16.21**	**11.28**	**2.26**	**5.60**	**5.28**	2.22	1.51	**0.42**	1.14	0.44

RomniStereo - Quantitative Comparison

Dataset	Sunny					Cloudy					Sunset				
Metric	>1	>3	>5	MAE	RMS	>1	>3	>5	MAE	RMS	>1	>3	>5	MAE	RMS
Non-learning based method															
Sphere-Stereo [23]	76.46	45.99	28.46	4.92	8.35	77.57	47.08	28.39	4.50	7.21	77.38	46.11	28.49	5.15	8.89
Trained on OmniThings only															
OmniMVS$^+_4$ [12]	26.18	7.06	4.37	1.24	3.06	28.50	6.62	3.93	1.23	2.92	25.29	6.92	4.18	1.22	3.06
RomniStereo$_4$	17.34	6.92	4.54	1.06	3.30	16.65	6.30	4.09	1.01	3.04	16.77	6.63	4.28	1.04	3.27
OmniMVS$^+_8$ [12]	18.49	6.13	3.93	1.10	3.07	18.85	5.89	3.72	1.08	2.94	17.99	6.08	3.85	1.09	3.02
RomniStereo$_8$	15.46	6.54	4.41	0.99	3.12	15.14	6.09	4.10	0.95	2.97	15.25	6.42	4.24	0.98	3.12
OmniMVS [11]	27.16	6.13	3.98	1.24	3.09	28.13	5.37	3.54	1.17	2.83	26.70	6.19	4.02	1.24	3.06
S-OmniMVS [13]	17.19	6.03	3.89	1.11	3.60	-	-	-	-	-	-	-	-	-	-
OmniMVS$^+_{32}$-IS [12]	17.46	5.73	3.60	0.99	2.76	17.67	5.84	3.82	1.04	3.00	17.28	5.63	3.42	0.98	2.71
OmniMVS$^+_{32}$ [12]	13.57	**4.81**	**3.10**	0.88	**2.56**	13.59	**4.81**	**3.15**	0.87	2.53	13.36	**4.71**	**2.93**	0.87	**2.50**
RomniStereo$_{32}$	12.28	5.59	3.79	0.80	2.68	11.86	5.08	3.44	0.75	2.50	12.30	5.45	3.48	0.78	2.67
RomniStereo$_{64}$	**11.25**	5.30	3.59	**0.75**	2.57	**10.97**	5.03	3.44	**0.73**	**2.47**	**10.94**	4.99	3.29	**0.72**	2.56
Finetuned on OmniHouse and Sunny															
OmniMVS$^+_4$-ft [12]	10.54	3.42	2.11	0.65	2.06	10.22	3.19	1.92	0.61	1.94	10.81	3.64	2.21	0.66	2.11
RomniStereo$_4$-ft	9.30	3.47	2.21	0.60	2.25	9.54	3.47	2.17	0.60	2.20	9.48	3.57	2.27	0.60	2.25
RomniStereo$_8$-ft	7.38	2.75	1.72	0.48	1.92	7.53	2.69	1.66	0.48	1.87	7.65	2.94	1.86	0.50	2.01
OmniMVS-ft [11]	13.93	2.87	1.71	0.79	2.12	12.20	2.48	1.46	0.72	1.85	14.14	2.88	1.71	0.79	2.04
S-OmniMVS-ft [13]	6.66	2.18	1.40	0.47	1.98	-	-	-	-	-	-	-	-	-	-
OmniMVS$^+_{32}$-ft [12]	7.48	3.57	2.42	0.57	2.42	7.29	3.38	2.30	0.54	2.31	7.82	3.60	2.42	0.58	2.36
RomniStereo$_{32}$-ft	5.19	1.98	1.23	0.36	1.55	5.63	2.03	1.29	0.39	1.72	5.53	2.13	1.34	0.37	1.61
RomniStereo$_{64}$-ft	**4.61**	**1.78**	**1.10**	**0.32**	**1.43**	**4.94**	**1.83**	**1.16**	**0.34**	**1.53**	**4.88**	**1.90**	**1.19**	**0.34**	**1.49**

Obrigado!

rafael.romeiro@impa.br