Adaptive Patch Deformation for Textureless-Resilient Multi-View Stereo

Yuesong Wang, Zhaojie Zeng, Tao Guan, et al.

JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

Advantage: Global-context awareness

Disadvantage: High memory consumption

Learning-based methods

Advantage: Low memory consumption

Disadvantage: Limited feature receptive field

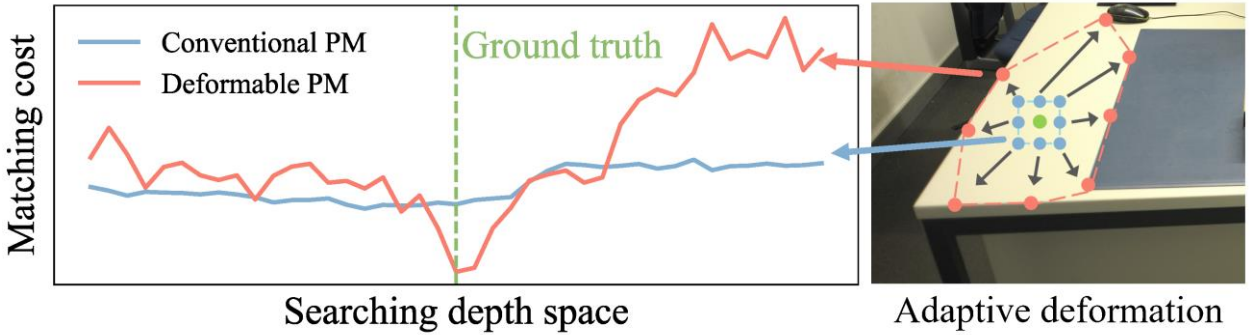
Traditional PatchMatch-based methods

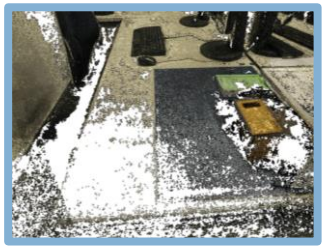
?: How to combine the advantages and avoid the disadvantages

Learning-based: Spirit of deformable convolution

Traditional : PatchMatch-based Multi-view stereo

Our adaptive patch deformation

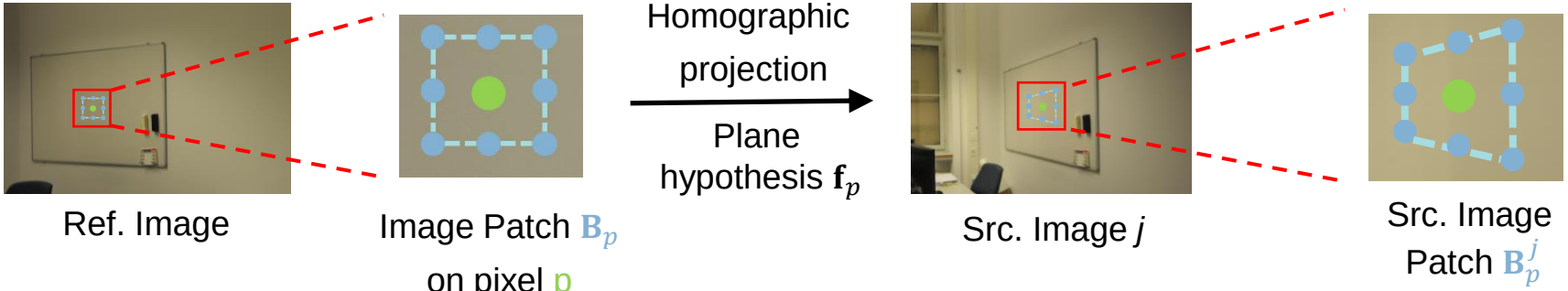




Result of conventional PatchMatch

Result of Deformable PatchMatch

Brief review of the PatchMatching:



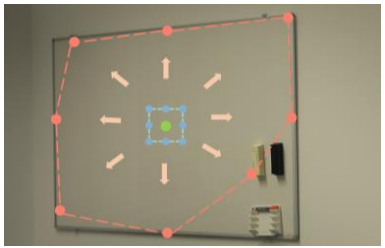
Matching cost between ref. and src. on pixel p given a hypothesis $\mathbf{f}_p$:

$$m_j(\mathbf{p}, \mathbf{f}_p, \mathbf{B}_p)$$

$$= 1 - \frac{cov\left(\mathbf{B}_p, \mathbf{B}_p^j\right)}{\sqrt{cov(\mathbf{B}_p, \mathbf{B}_p)cov\left(\mathbf{B}_p^j, \mathbf{B}_p^j\right)}}$$

?: How to eliminate matching ambiguity when dealing with textureless regions?

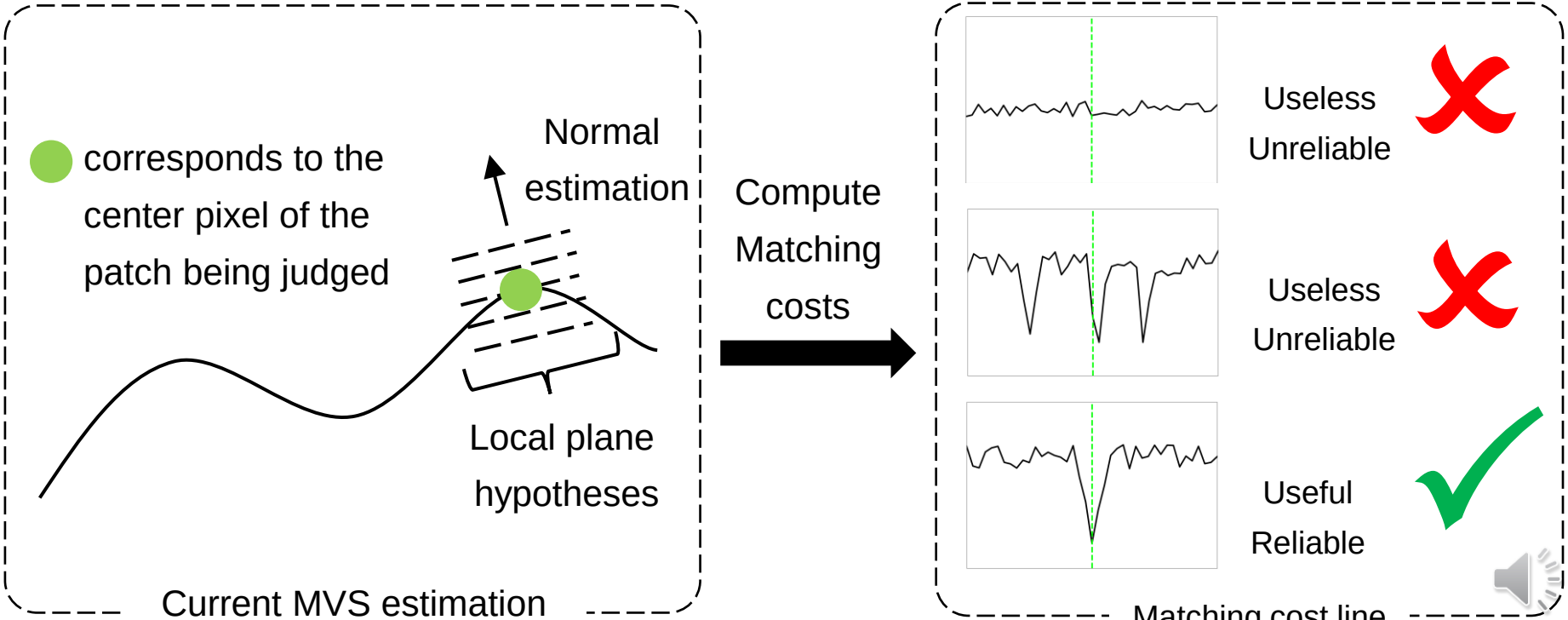
Adaptive patch deformation:
Making the patch contain enough useful feature information



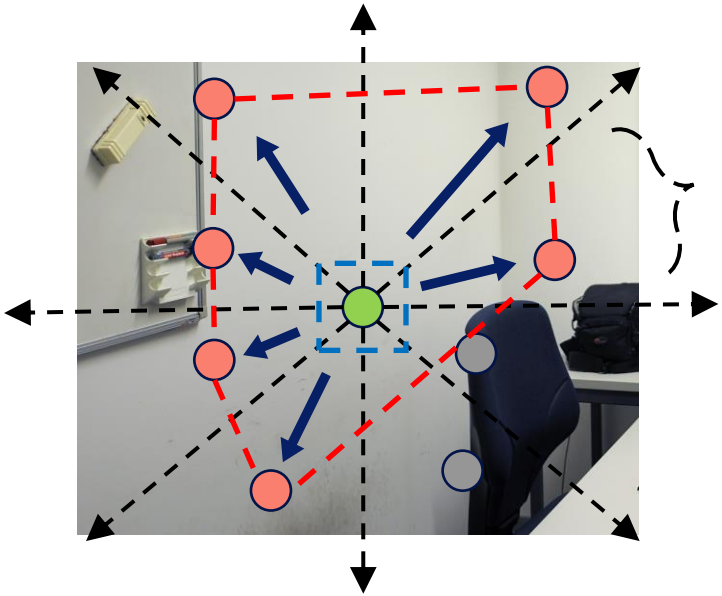
?:

- How to find the useful feature information?
- How to adaptively deform the patch to cover enough useful feature information?
- How to compute matching cost for the deformed patch?

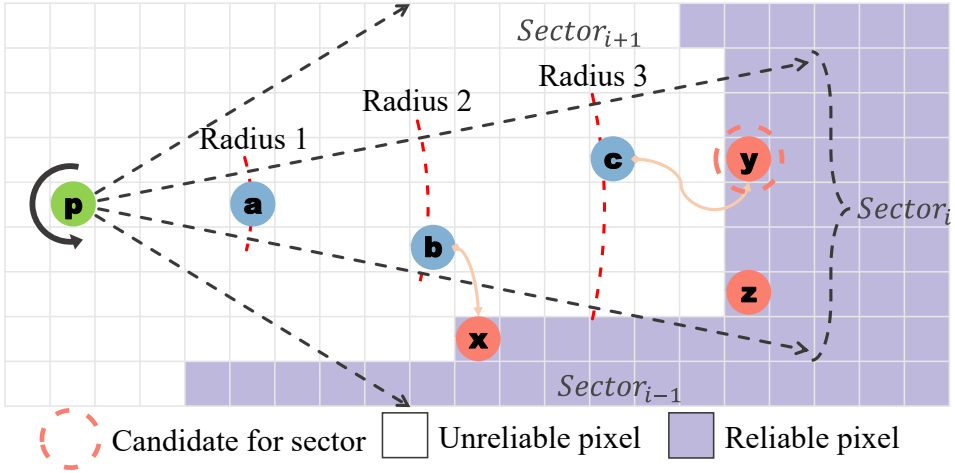
Finding useful feature information:



Adaptive deformation:

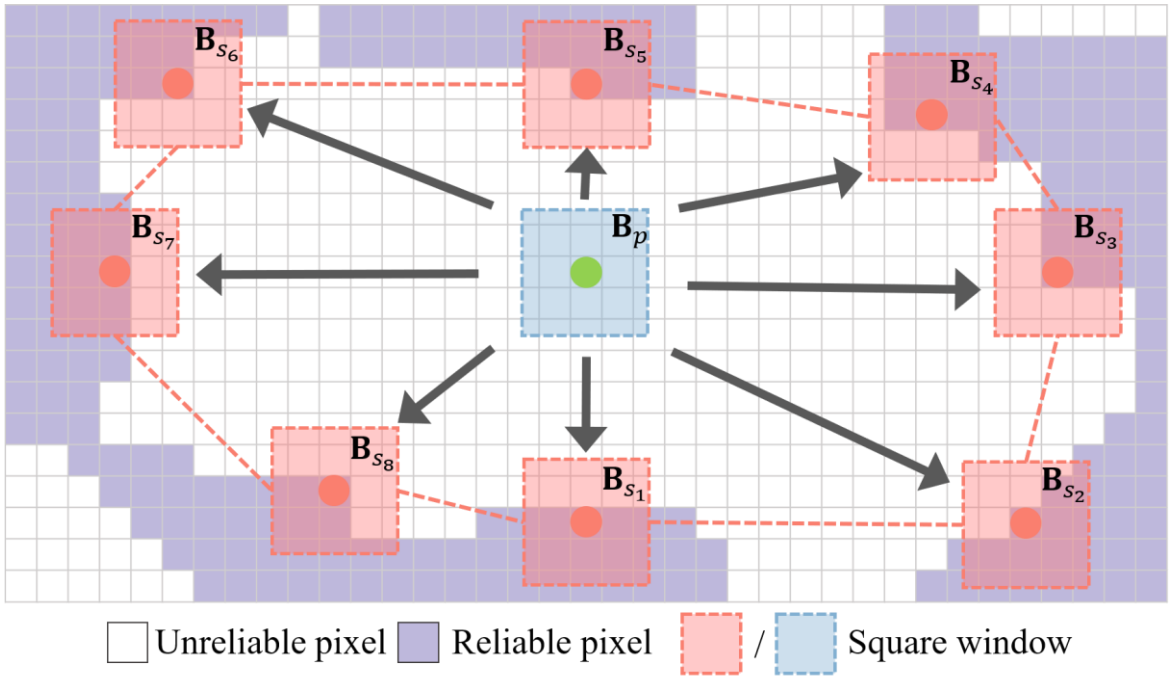


$Sector_i$



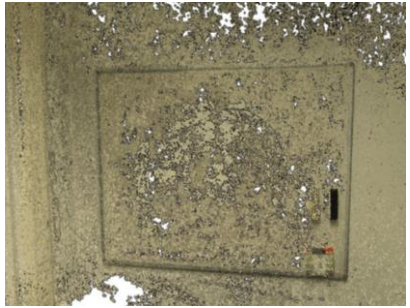
Center pixel p Related reliable pixel Unrelated reliable pixel

Compute Matching Cost:



$$m_D(\bullet, \mathbf{f}_p, \mathrm{S}) = \lambda * m(\bullet, \mathbf{f}_p, \square) + \cdot(1-\lambda) * \frac{1}{8} * \sum_{i=1}^{8} m(\bullet, \mathbf{f}_p, s_i)$$

IterMVS-LS
(CVPR2022)



ACMMP
(PAMI2022)

Ref. Image

Ours

Methods		GPU Mem.(GB)		
		Res. (8.04%)	Res. (50%)	Res. (100%)
Learning-based	GBi-Net	3.6	20.7	\
	PatchmatchNet	3.5	18.6	\
	IterMVS-LS	2.5	11.2	22.0
PatchMatch-based	ACMMP	1.4	4.5	7.9
	Ours	**1.4**	**3.7**	**6.6**