High-Quality Visual Correspondence Estimation with Simple, Generic and Efficient Techniques

Jiangbo Lu

https://publish.illinois.edu/**visual-modeling-and-analytics/**

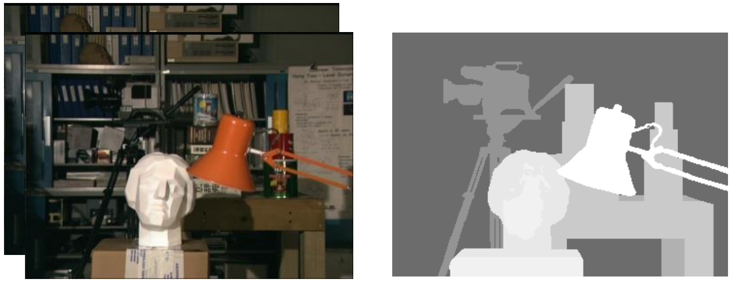
Joint works with Yu Li (ADSC), Dongbo Min (now CNU),
Wen-Yan Lin (ADSC), Hongsheng Yang (now Google), Minh N. Do (UIUC)

ADSC Illinois at Singapore Pte Ltd

ILLINOIS UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Everywhere Visual Correspondences

Across views



[Scharstein & Szeliskii, IJCV'02]

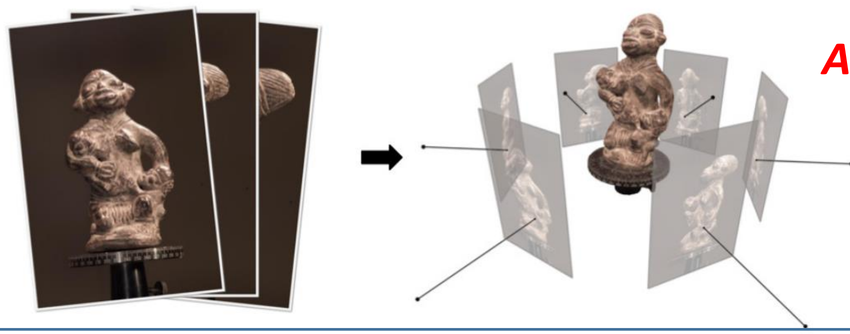
Across time

[Baker et al., IJCV'11]

Wide baseline

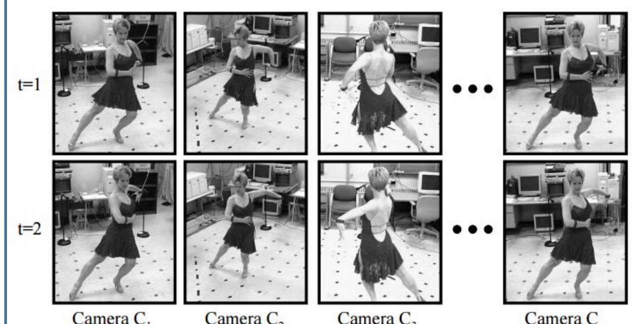
[Goedeme et al., CVPR'04]

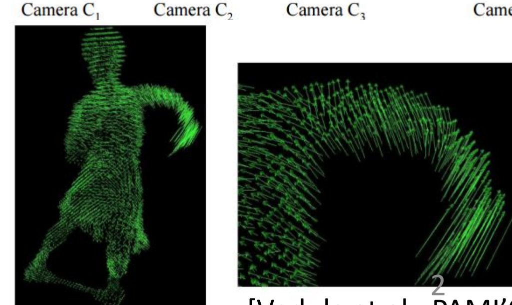
Across multiple views



[Furukawa & Ponce, PAMI'09]

Across view & time

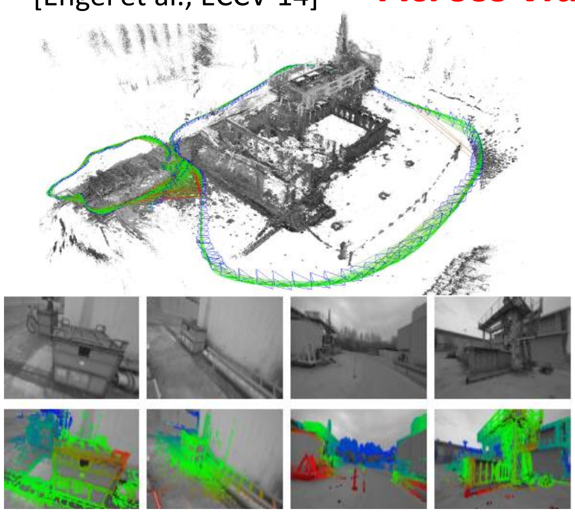




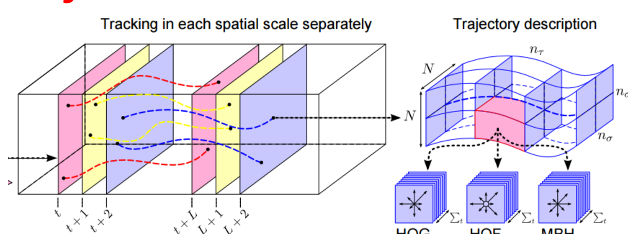
[Vedula et al., PAMI'05]

Across video frames

[Engel et al., ECCV'14]

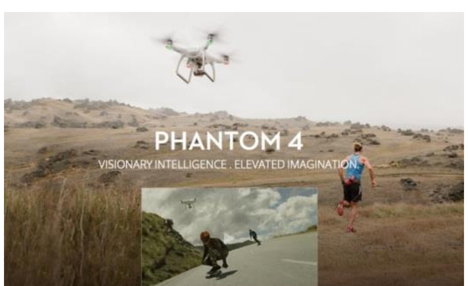


[Wang et al., IJCV'13]



Key to Many Applications

- Structure from motion, 3D reconstruction
- Robot navigation, video odometry
- Scene labeling and semantic understanding
- Depth transfer and synthesis
- Computational photography, image processing
- Image stitching, view synthesis
- Visual place recognition and object localization
- ....



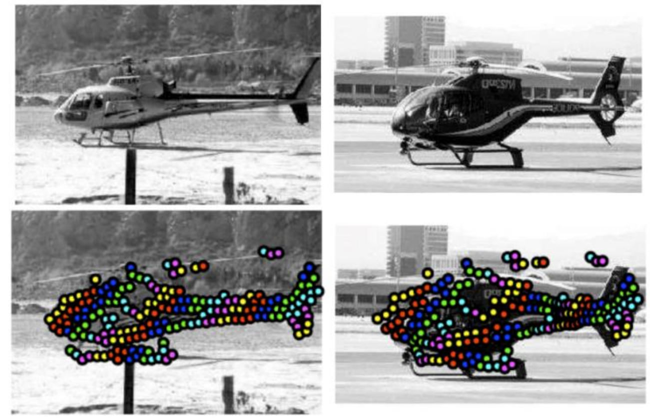


Yet Another Class: Correspondence across Scenes

Correspondence at pixel level

[Scharstein et al. *IJCV*’02]

Correspondence at object level



[Berg et al. *CVPR*’05]

Correspondence at scene level

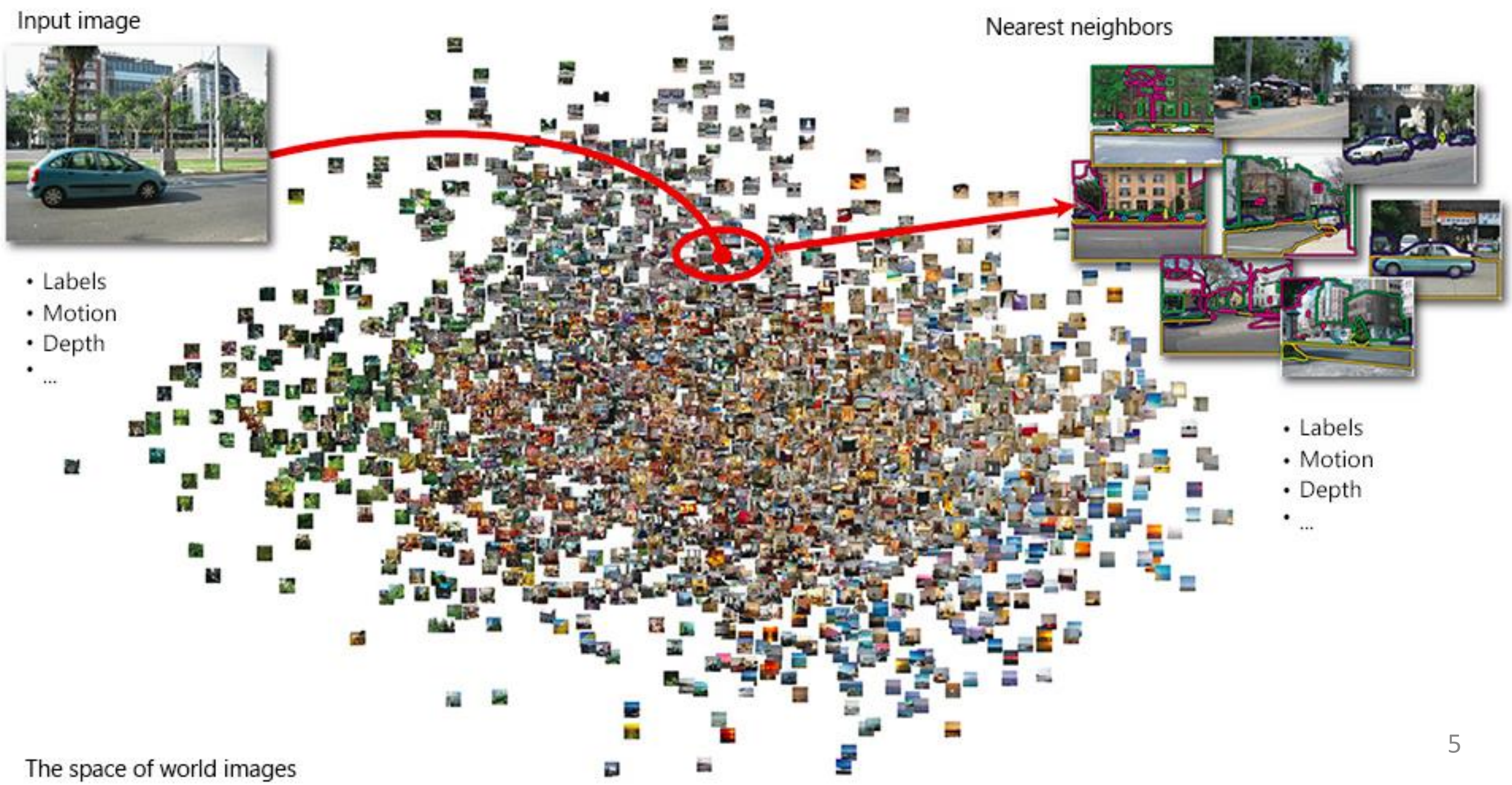
Dense correspondence;
Scene context preserved
[Liu et al., PAMI’11]

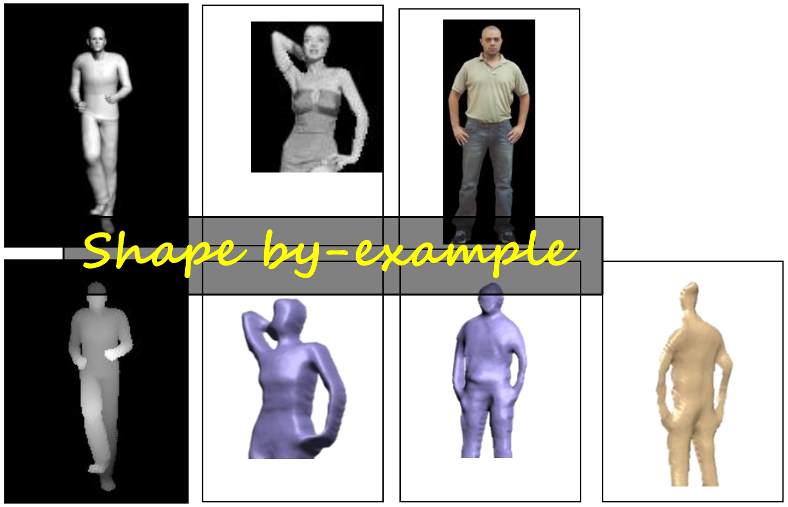
Applications of Dense (Semantic) Correspondences

CVPR 2014 Tutorial

Dense Image Correspondences for Computer Vision

Ce Liu[1] Michael Rubinstein[1] Jaechul Kim[2] Zhuowen Tu[3]
[1]Microsoft Research [2]Amazon [3]UCSD





Shape by-example

[Hassner&Basri ’06a, ‘06b,’13]

New view synthesis

[Hassner ‘13]

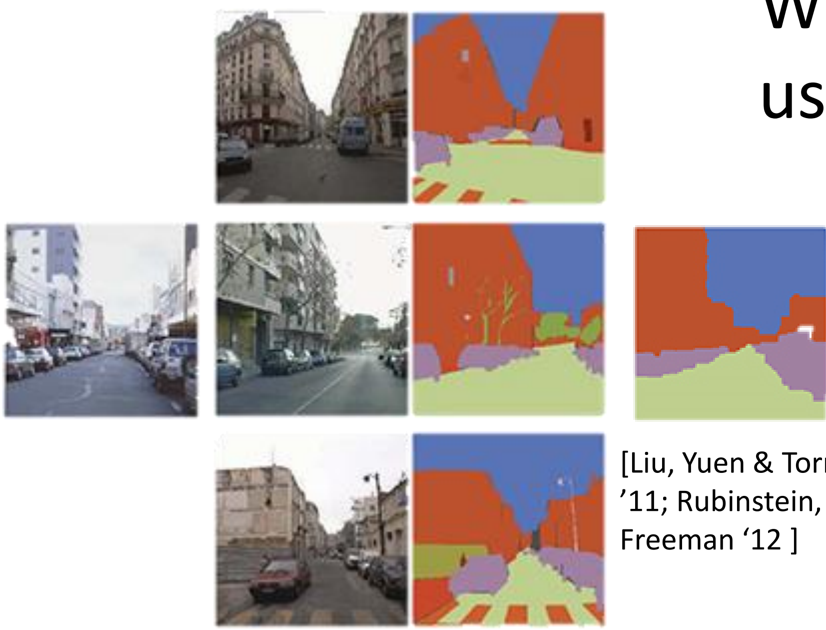
Face recognition

[Liu, Yuen & Torralba ’11]

Fingerprint recognition

[Hassner, Saban & Wolf]

Why is this useful?



[Liu, Yuen & Torralba ’11; Rubinstein, Liu & Freeman ‘12]

Label transfer / scene parsing

Depth transfer

[Karsch, Liu & Kang ‘12]

Slide courtesy T. Hassner

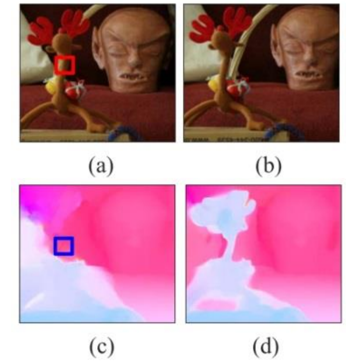
[1]Indeed, one of the oft-told stories is that when a student asked Takeo Kanade what are the three most important problems in computer vision, his reply was: "Alignment, alignment, alignment!". [Aubry et al., CVPR'14]

Correspondence, correspondence, correspondence!

- Image alignment
- Image registration
- Optical flow
- Stereo reconstruction
- Feature matching
- …

A Number of Challenges

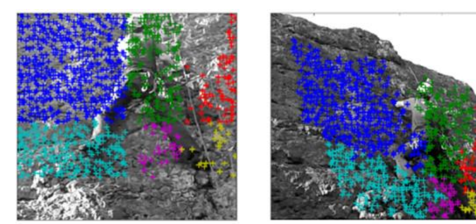
- Large displacement
- Non-rigid motion
- Independent object motion
- Small objects



- Photometric differences (exposure, tone, sharpness)
- Weakly textured regions

- Matching across different scene contents

- Motion coherence vs. boundary/detail preserving
- Precision vs. recall, density, spatial coverage/distribution

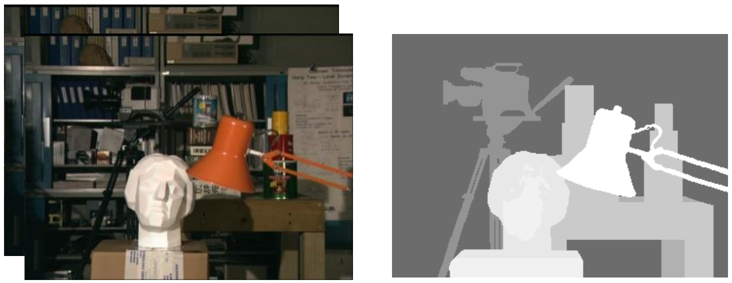


- Computational load
- Memory cost
- Large hypothesis space

Robust, dense, fast

Focus on Two-Frame *Dense* Correspondences

Across views

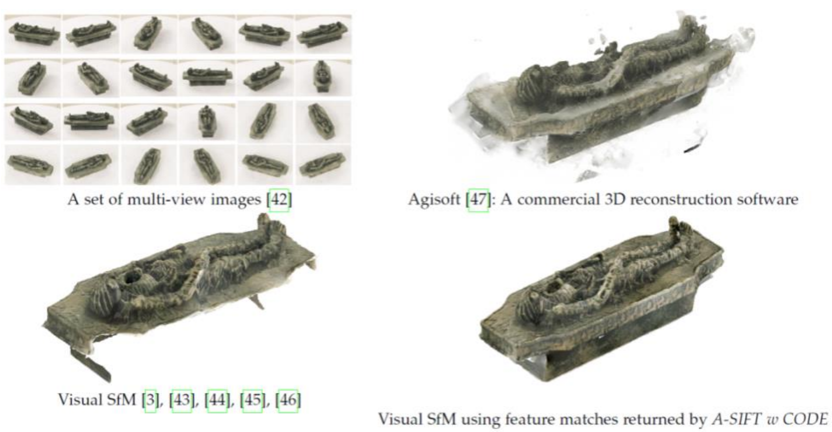


[Scharstein & Szeliskii, IJCV'02]

Across time

[Baker et al., IJCV'11]

Wide baseline & across multiple views



Across scenes

Desired Properties? How to *compute*?

- **General & versatile**
  - Applicable for different tasks & methods
  - Vs. task-specific, state-of-art methods
- **Simple & competitive**
  - Elegant and principled
  - e.g. w/o *complex energy terms, descriptor matching, coarse-to-fine, pre-processing*…
- **Efficient**
  - Fast speed and implementation advantages
  - Friendly for parallelization, embedded system

Desired Properties? How to *compute*?

- **Simple models** are easy to justify, understand, apply and accelerate

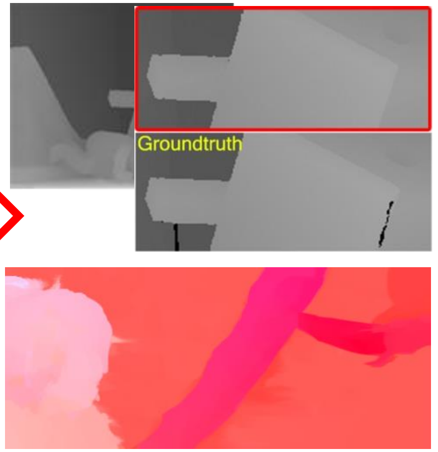
- **Computational efficiency** enables aggressive exploration, bringing quality gain and simple models

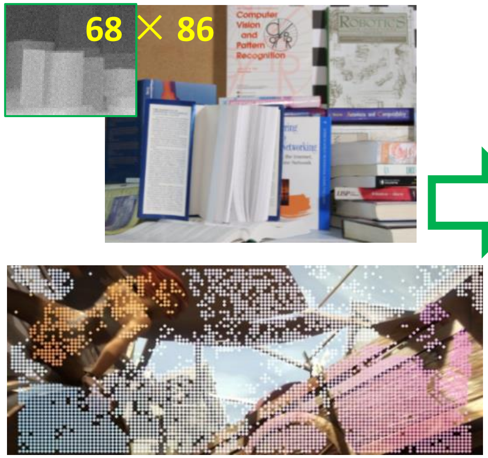
- General & versatile
  - Applicable for different tasks & methods
- Simple & competitive
  - Elegant and principled
  - *e.g. w/o complex energy terms, descriptor matching, coarse-to-fine, pre-processing…*
- Efficient
  - Fast speed and implementation advantages
  - Friendly for parallelization, embedded syst.

1: SPM-BP [ICCV'15]: Discrete Labeling Optimization for MRFs

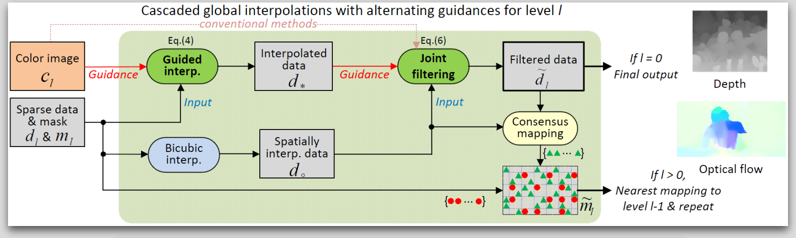
$$E = \sum_p E_p(l_p; W) + \sum_p \sum_{q\in\mathcal{N}_p} E_{pq}(l_p, l_q)$$

p: pixel, N_p: 4 neighbors

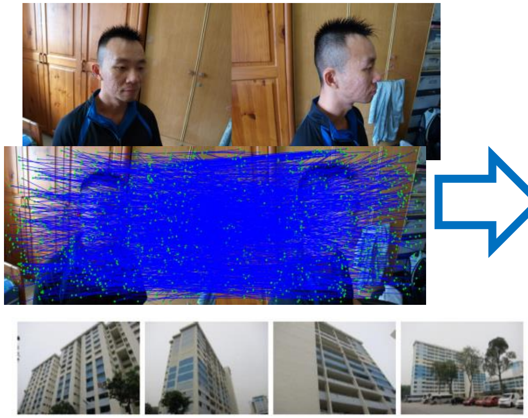




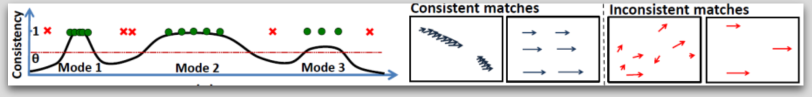
2: Fast Guided Global Interpolation [ECCV'16a]: from Sparse to Dense



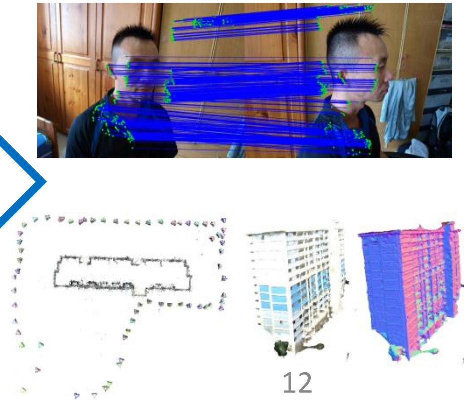
$$\mathcal{E}(\mathbf{u}) = (\mathbf{u}-\mathbf{f})^\top(\mathbf{u}-\mathbf{f}) + \lambda \mathbf{u}^\top \mathbf{A}\mathbf{u}$$



3: Coherence-Based Regression for Feature Matching [ECCV'14, '16b]



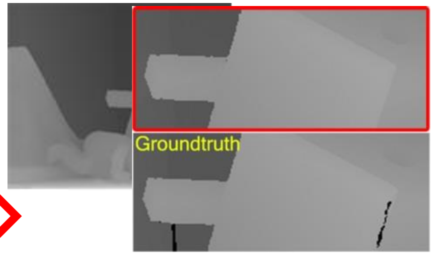
$$\arg\min_{\mathbf{w}} \sum_{i=1}^{N} C(1-f(\mathbf{m}_i)) + \lambda \mathbf{w}^T G \mathbf{w}$$



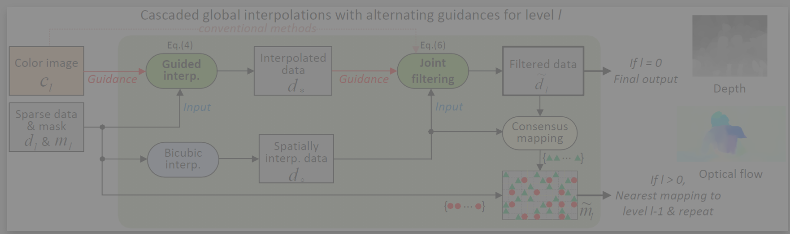
1: SPM-BP [ICCV’15]: Discrete Labeling Optimization for MRFs

$$E = \sum_{p} E_p(l_p; W) + \sum_{p} \sum_{q \in \mathcal{N}_p} E_{pq}(l_p, l_q)$$

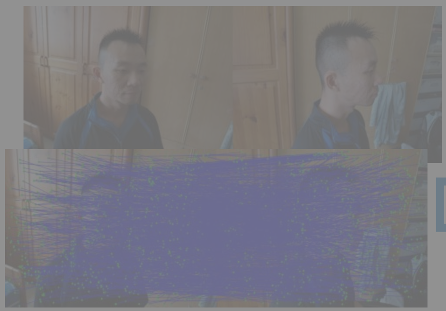
p: pixel, N_p: 4 neighbors

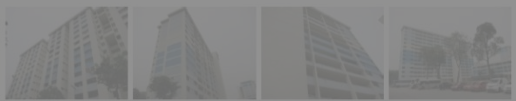


2: Fast Guided Global Interpolation [ECCV’16a]: from Sparse to Dense



$$\mathcal{E}(\mathbf{u}) = (\mathbf{u} - \mathbf{f})^{\top}(\mathbf{u} - \mathbf{f}) + \lambda \mathbf{u}^{\top} \mathbf{A} \mathbf{u}$$





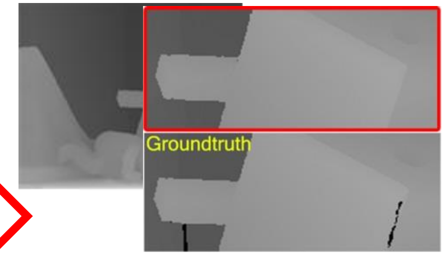
3: Coherence-Based Regression for Feature Matching [ECCV’14, ’16b]

$$\arg\min_{\mathbf{w}} \sum_{i=1}^{N} C(1 - f(\mathbf{m}_i)) + \lambda \mathbf{w}^T G \mathbf{w}$$

1: SPM-BP [ICCV’15]: Discrete Labeling Optimization for MRFs

$$E = \sum_{p} E_p(l_p; W) + \sum_{p} \sum_{q \in \mathcal{N}_p} E_{pq}(l_p, l_q)$$

p: pixel, N_p: 4 neighbors

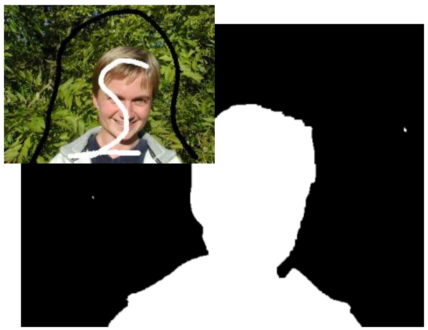


Part 1: MRF-Based Dense Correspondence Field via Efficient Inference

- **Y. Li, D. Min, M. Brown, M. N. Do, and J. Lu, “SPM-BP: Sped-up PatchMatch Belief Propagation for Continuous MRFs,” ICCV 2015. (Oral)**
- J. Lu, H. Yang, D. Min, and M. N. Do, “PatchMatch Filter: Efficient Edge-Aware Filtering Meets Randomized Search for Fast Correspondence Field Estimation,” CVPR 2013. (**Oral**)
- J. Lu, Y. Li, H. Yang, D. Min, W. Eng, and M. N. Do, “PatchMatch Filter: Edge-Aware Filtering Meets Randomized Search for Correspondence Field Estimation,” TPAMI 2016.
- H. Yang, W.-Y. Lin, and J. Lu, “Daisy Filter Flow: A Generalized Discrete Approach to Dense Correspondences,” CVPR 2014.

Discrete Pixel-Labeling Optimization on MRF

- Many computer vision tasks can be formulated as a pixel-labeling problem on Markov Random Field (MRF)

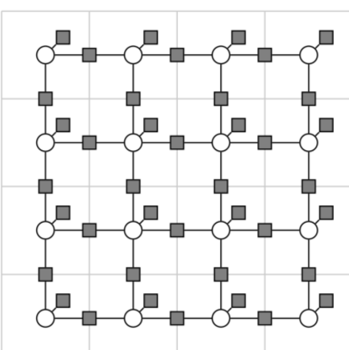


Segmentation
$l=\{B,G\}$

Denoising
l = *intensity*

Stereo
$l = d$

Optical flow
$l = (u,v)$

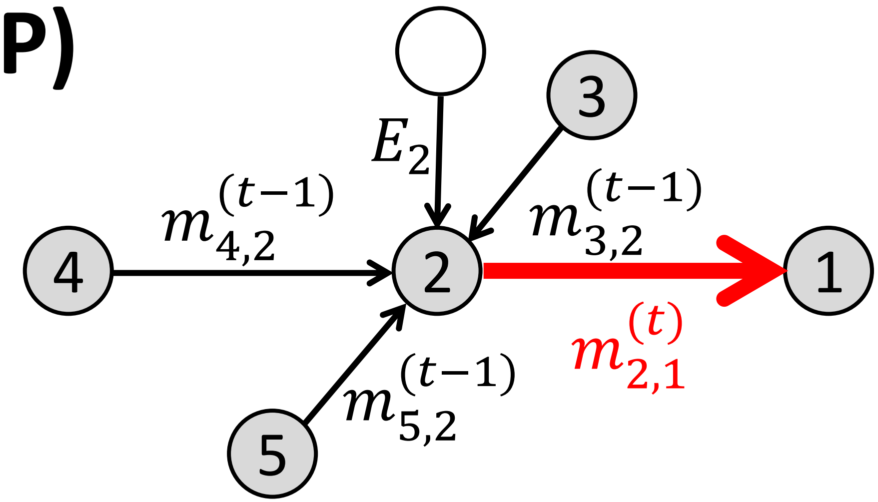


$$E = \sum_p E_p(l_p; W) + \sum_p \sum_{q \in \mathcal{N}_p} E_{pq}(l_p, l_q)$$

p: pixel, N_p: 4 neighbors

- **Simple & elegant**: data term + smoothness term, MAP
- **Effective**: labeling coherence, discontinuity handling
- **Flexible**: support a broad range of energy functions
- **Optimization**: Graph Cut, Belief Propagation, etc

Belief Propagation (BP)



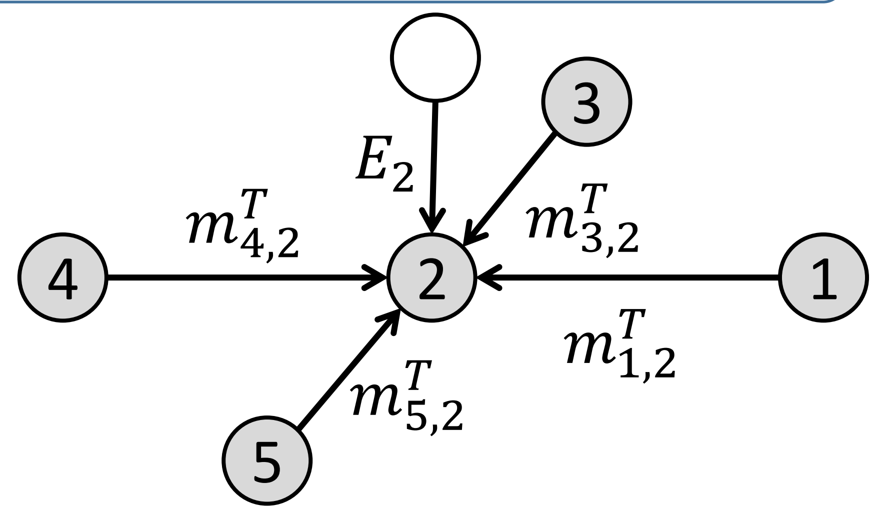
Iterative process in which neighbouring nodes “talk” to each other:

- Update message between neighboring pixels

$$m_{qp}^{(t)}(l_p) = \min_{l_q \in \mathcal{L}}(E_{pq}(l_p, l_q) + E_q(l_q) + \sum_{s \in \mathcal{N}_q \backslash p} m_{sq}^{(t-1)}(l_q))$$

- Stop after T iterations, decide the final label by picking the smallest dis-belief

$$B_p(l_p) = E_p(l_p) + \sum_{q \in \mathcal{N}_p} m_{qp}^{(T)}(l_p)$$



- **Challenge**:

When the label set L is huge or densely sampled, BP faces prohibitively high computational challenges.

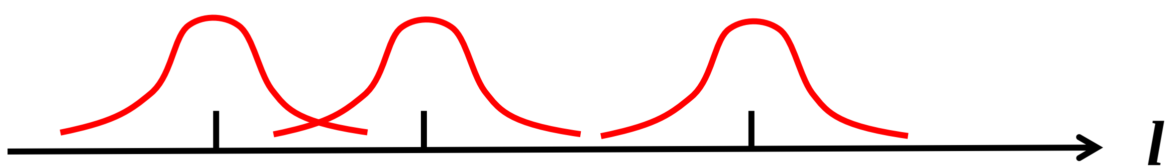
Particle Belief Propagation (PBP)

[Ihler and McAllester, "Particle Belief Propagation," *AISTATS*'09]

- **Solution**:

  (1) only store messages for *K* labels (particles)

  (2) generate new label particles with the MCMC sampling using a Gaussian proposal distribution



- **Challenge**:

MCMC sampling is still inefficient and slow for continuous label spaces (e.g. stereo with slanted surfaces).

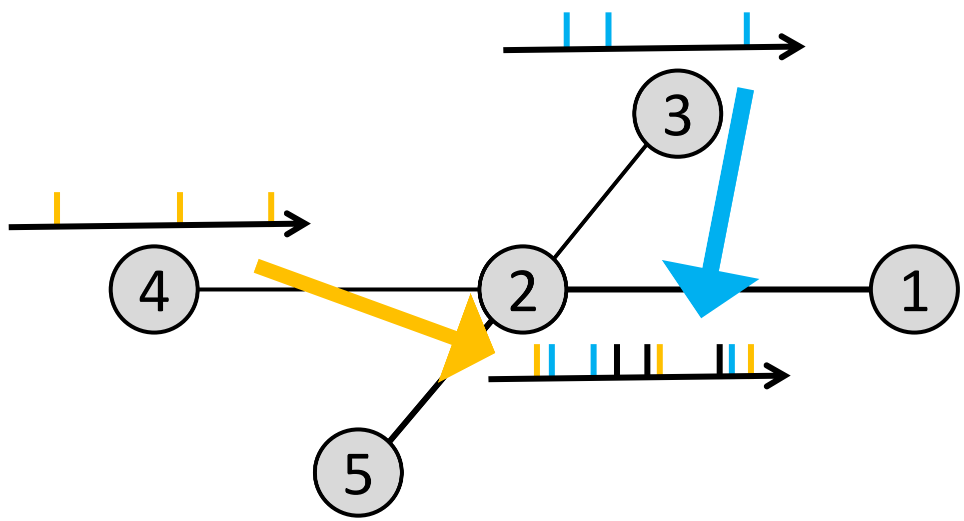
Patch Match Belief Propagation (PMBP)

[Besse et al, "PMBP: PatchMatch Belief Propagation for Correspondence Field Estimation," *IJCV* 2014]

- **Solution**:

Use Patch Match[Barnes et al., Siggraph'09]'s sampling algorithm – augment PBP with label samples from the neighbours as proposals

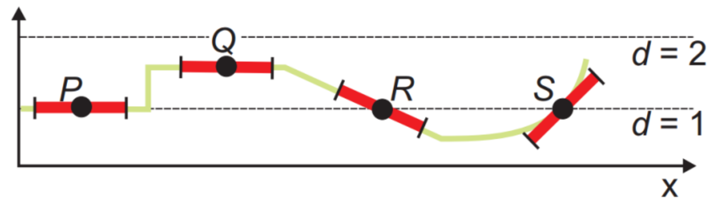
- **Orders of magnitude faster** than PBP



Patch Match Belief Propagation (PMBP)

- Effectively handles large label spaces in message passing
- Successfully applied to stereo with slanted surface modeling [Bleyer et al., BMVC'11]

Label: 3D plane normal $l = (a_p, b_p, c_p)$



$l = d\ (integer)$ | $l = (a_p, b_p, c_p)$

Left image | Disparity map | 3D reconstruction | Disparity map | 3D reconstruction

Image courtesy of [Bleyer et al., BMVC'11]

- Also successfully applied to optical flow [Hornáček et al., ECCV'14]

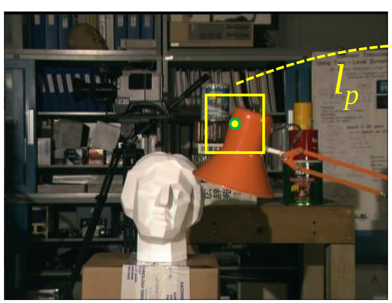
Problem of PMBP

- However, it suffers from a heavy computational load on the data cost computation

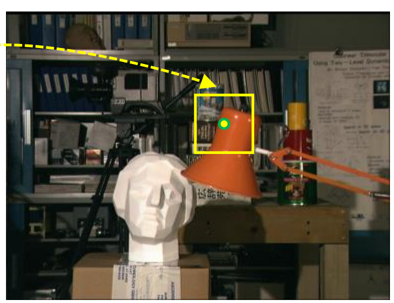
$$E = \boxed{\sum_{p} E_p(l_p; W)} + \sum_{p} \sum_{q \in \mathcal{N}_p} E_{pq}(l_p, l_q)$$

- Many works strongly suggest to gather stronger evidence from a local window for the data term

$$E_p(l_p; W) = \sum_{r \in W} \omega_{pr} C_r(l_p)$$



Left view



Right view

Weight

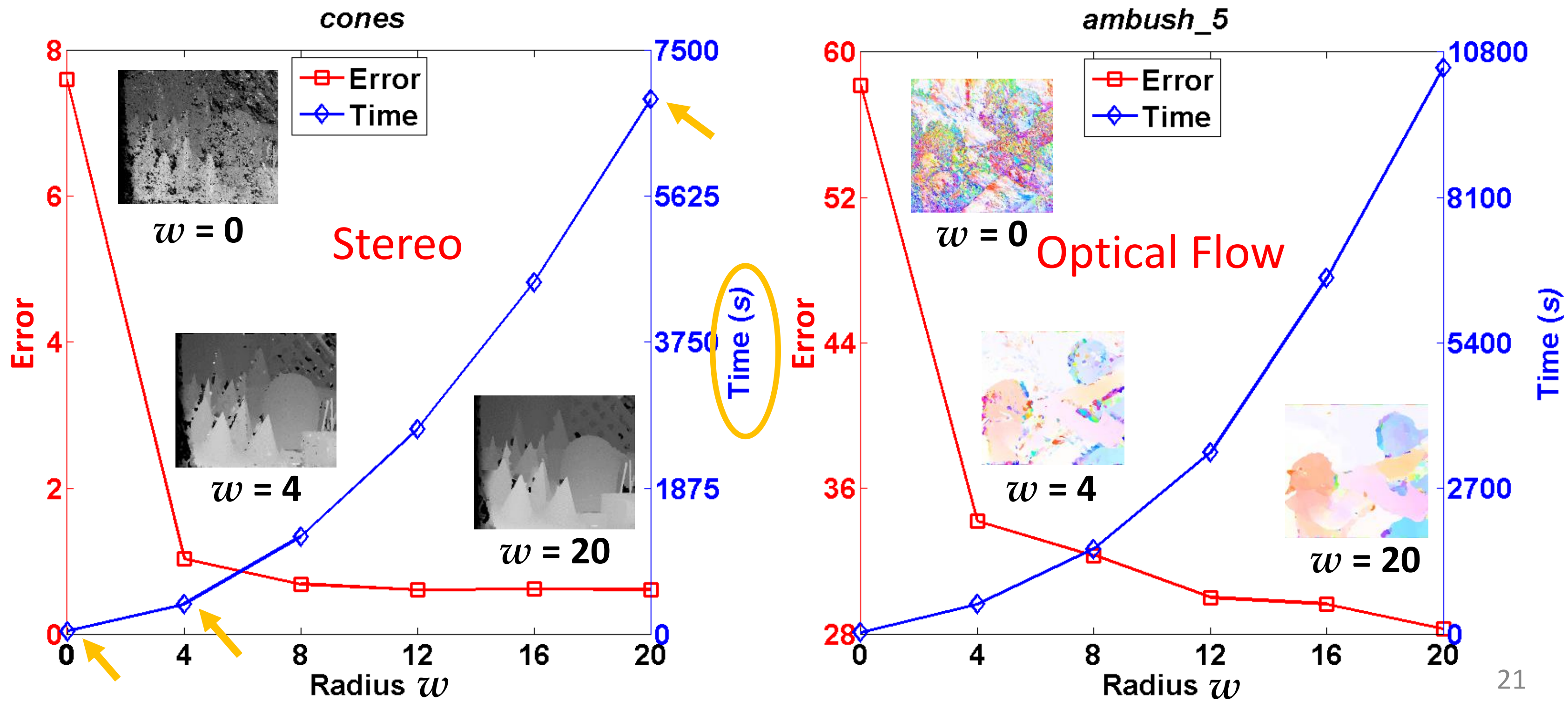
×

Raw matching cost

Data term is important!

- Better results with larger window sizes (2w+1)^2, but more computational cost!
  - e.g. **|*W*| = 40x40 in PMBP** **[IJCV'14]**, **|*W*| = 41x41 in PM-PM** **[TIP'15]**

$$\boxed{E_p(l_p; W) = \sum_{r \in W} \omega_{pr} C_r(l_p)}$$

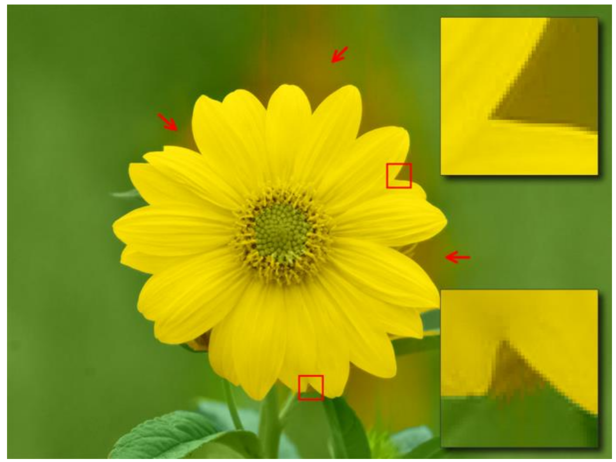


Aggregated Data Term also Justified in e.g.

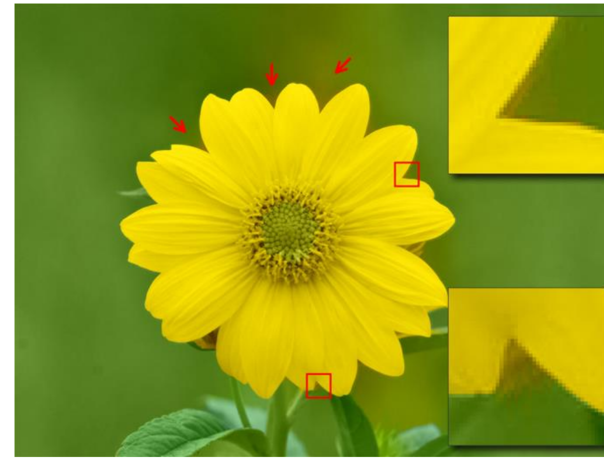
[Min et al., TIP'14]

Without aggregated data term

Input



After 3 iter.



After 10 iter.

With aggregated data term

No color bleeding artifacts!

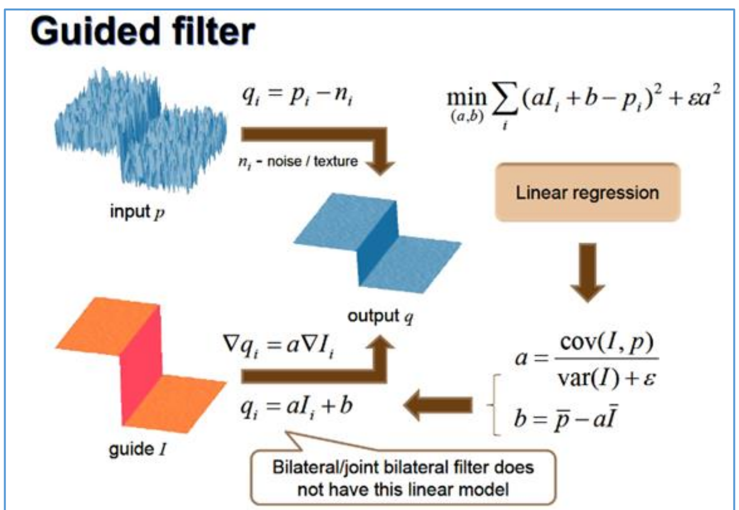
Aggregated Data Cost Computation

- Cross/joint bilateral filtering principles

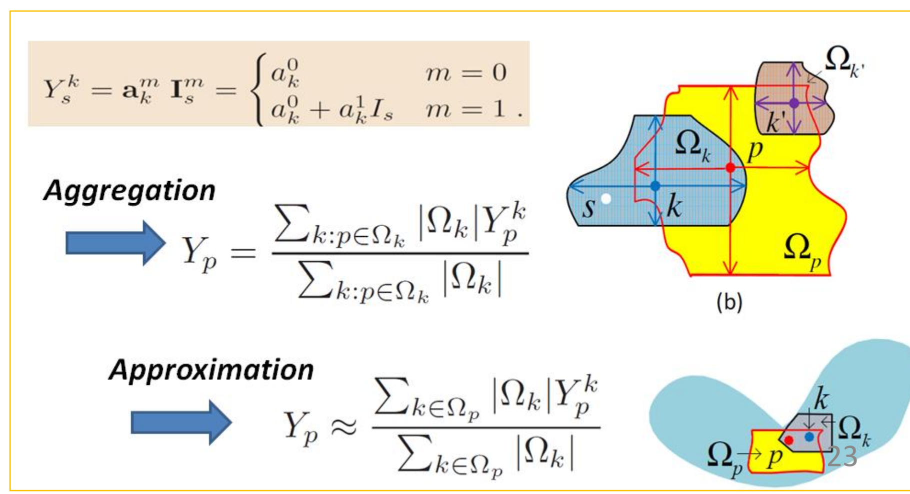
$$E_p(l_p; W) = \sum_{r \in W} \omega_{pr} C_r(l_p)$$

- **Local discrete labeling approaches** have often used O(1)-time edge-aware filtering (EAF) methods [Rhemann et al., CVPR'11].
  - O(1)-time: No dependency on window size used in EAF

Guided Filter [He et al., *ECCV* 2010]



Cross-based Local Multipoint Filtering (CLMF) [Lu et al., *CVPR* 2012]



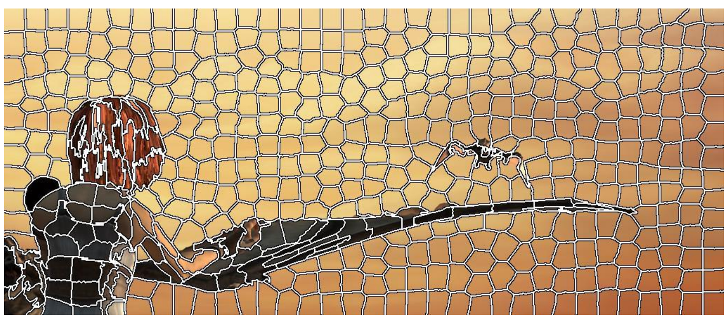
Why does PMBP NOT use O(1) time EAF?

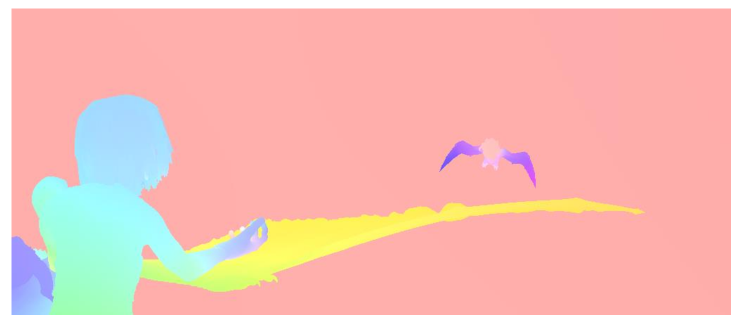
- Particle sampling and data cost computation are performed *independently* for each pixel

  ➔ Incompatible with EAF, essentially exploiting redundancy

- **Observation**

Labeling is often spatially smooth away from edges. This allows for shared label proposal and data cost computation for spatially neighboring pixels.





- **Our solution**

A superpixel based particle sampling belief propagation method, leveraging efficient filter-based cost aggregation

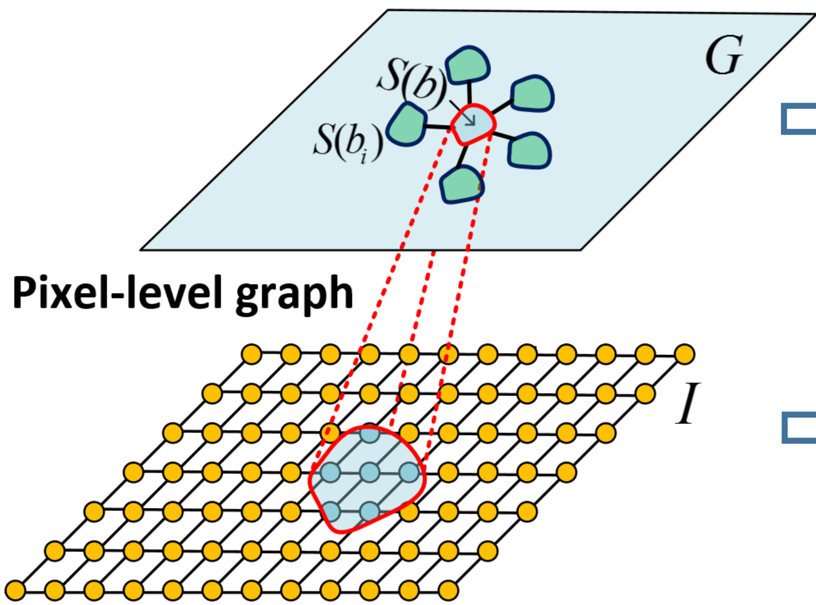
Sped-up Patch Match Belief Propagation (SPM-BP)

Sped-up Patch Match Belief Propagation

- **Two-layer graph structures in SPM-BP**

Superpixel-level graph

Pixel-level graph



1. Shared particle generation
2. Shared data cost computation

$$E = \boxed{\sum_p E_p(l_p; W)} + \sum_p \sum_{q \in \mathcal{N}_p} E_{pq}(l_p, l_q)$$

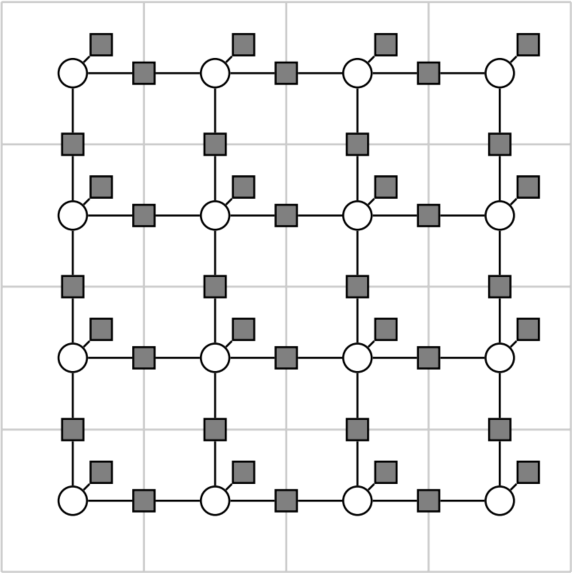
1. Message passing
2. Particle selection

$$E = \sum_p E_p(l_p; W) + \boxed{\sum_p \sum_{q \in \mathcal{N}_p} E_{pq}(l_p, l_q)}$$

- **Scan superpixels and perform :**
  - *Neighbourhood propagation*
  - *Random search (a.k.a. resampling based on a proposal distribution)*

Related works

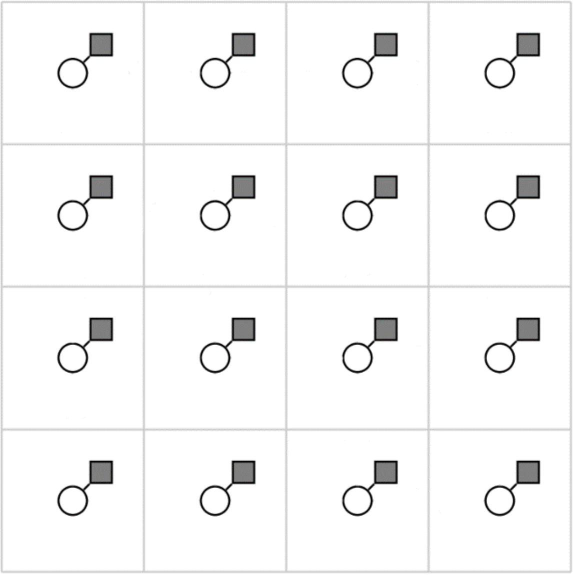
Pixel based MRF



Local methods

[Rhemann et al., CVPR'11]
[Lu et al., CVPR'13]

Only rely on data term

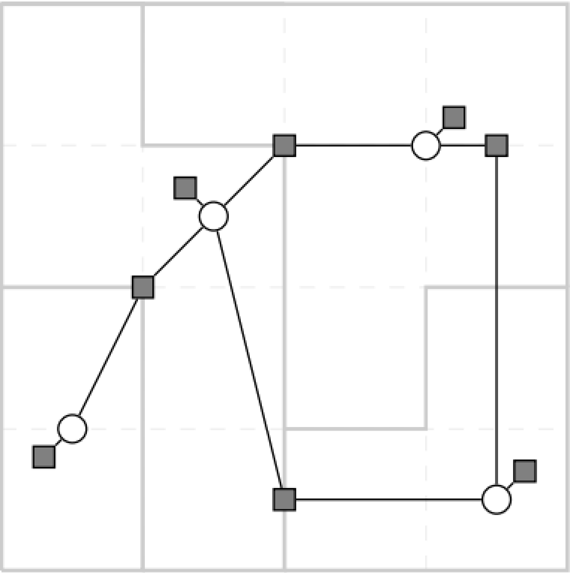


Superpixel based MRF

[Kappes et al., IJCV'15]
[Güney & Geiger, CVPR'15]

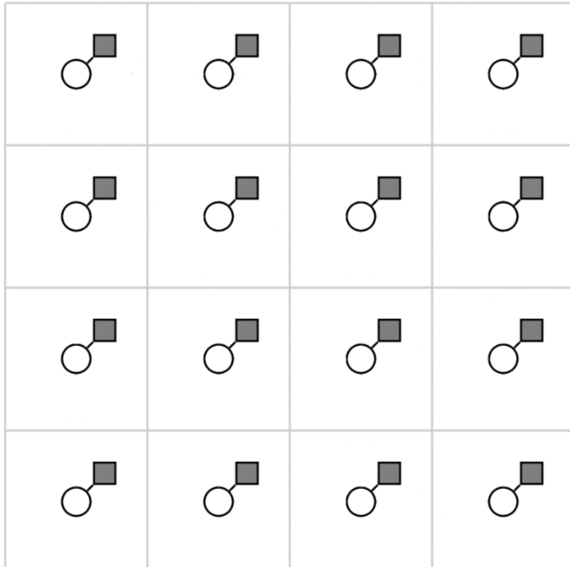
Superpixels as graph nodes

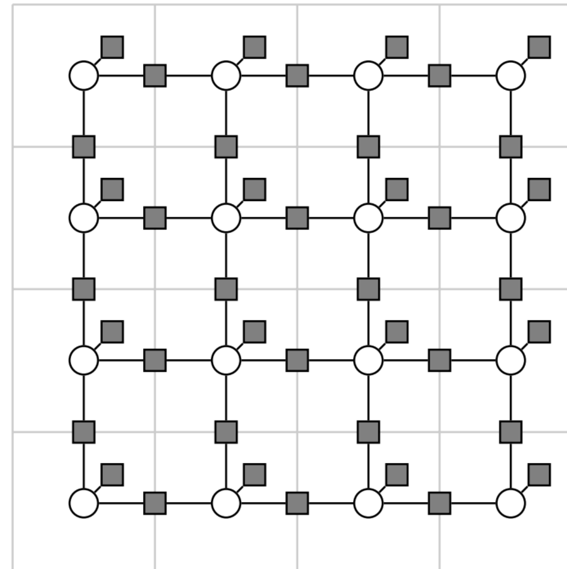
Image courtesy of [Kappes et al., IJCV'15]



Superpixel-based MRF: each superpixel is a node in the graph and **all pixels of the superpixel are constrained to have the same label**.

Our two-layer graph: superpixels are employed only for particle generation and data cost computation, the **labeling is performed for each pixel independently**.





PMF: **Filter-based**
local **inference**
[Lu et al., CVPR’13]

SPM-BP: **BP-based**
global **inference**
[Li et al., ICCV’15]

$$\boxed{E = \sum_p E_p(l_p; W)} + \sum_p \sum_{q \in \mathcal{N}_p} E_{pq}(l_p, l_q)$$

p: pixel, N_p: 4 neighbors

Comparison of Existing Labeling Optimizers

Local labeling approaches		Data cost computation	
		w/o EAF: **O(|*W*|)**	w/ EAF: **O(1)**
Label space handling	w/o PatchMatch: **O(|*L*|)**	Adaptive Weighting [PAMI'06]	Cost Filtering [CVPR'11]
	w/ PatchMatch: **O(log|*L*|)**	PM Stereo [BMVC'11]	**PMF** [CVPR'13]

Global labeling approaches		Data cost computation	
		w/o EAF: **O(|*W*|)**	w/ EAF: **O(1)**
Label space handling	w/o PatchMatch: **O(|*L*|)**	BP [IJCV'06]	Fully-connected CRFs [NIPS'11]
	w/ PatchMatch: **O(log|*L*|)**	**PMBP** [IJCV'14]	**?**

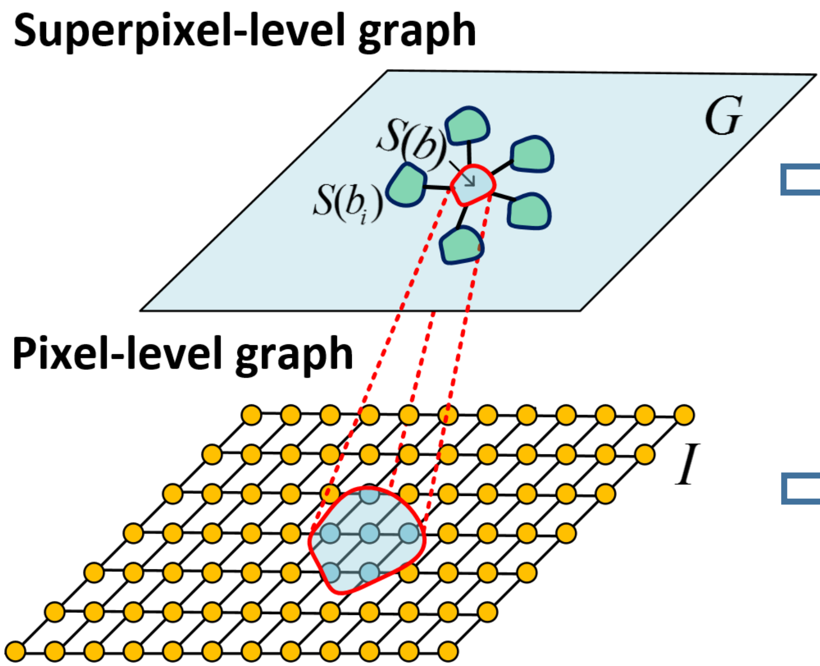
Comparison of Existing Labeling Optimizers

Local labeling approaches		Data cost computation	
		w/o EAF: **O(|*W*|)**	w/ EAF: **O(1)**
Label space handling	w/o PatchMatch: **O(|*L*|)**	Adaptive Weighting [PAMI'06]	Cost Filtering [CVPR'11]
	w/ PatchMatch: **O(log|*L*|)**	PM Stereo [BMVC'11]	**PMF** [CVPR'13]

Global labeling approaches		Data cost computation	
		w/o EAF: **O(|*W*|)**	w/ EAF: **O(1)**
Label space handling	w/o PatchMatch: **O(|*L*|)**	BP [IJCV'06]	Fully-connected CRFs [NIPS'11]
	w/ PatchMatch: **O(log|*L*|)**	**PMBP** [IJCV'14]	**SPM-BP** **[This paper]**

Sped-up Patch Match Belief Propagation

- **Two-layer graph structures in SPM-BP**



1. Shared particle generation
2. Shared data cost computation

$$E = \boxed{\sum_p E_p(l_p; W)} + \sum_p \sum_{q \in \mathcal{N}_p} E_{pq}(l_p, l_q)$$

1. Message passing
2. Particle selection

$$E = \sum_p E_p(l_p; W) + \boxed{\sum_p \sum_{q \in \mathcal{N}_p} E_{pq}(l_p, l_q)}$$

- **Scan superpixels and perform :**
  - *Neighbourhood propagation*
  - *Random search (a.k.a. resampling based on a proposal distribution)*

SPM-BP: Neighbourhood Propagation

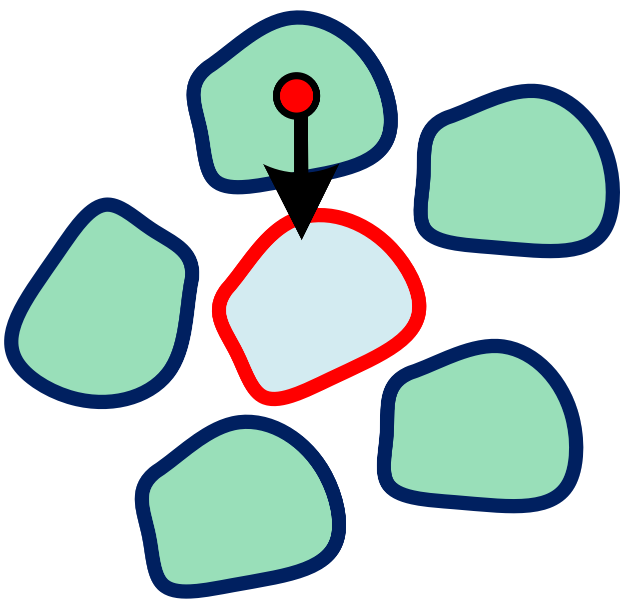
✓Step 1. Particle propagation

✓Step 2. Data cost computation

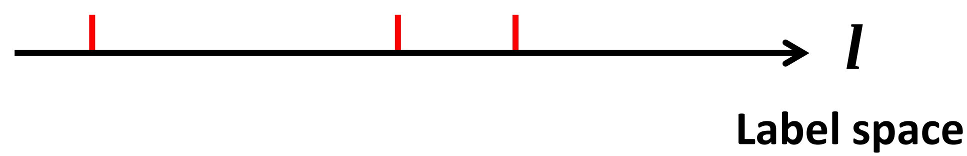
✓Step 3. Message update

1-1) Randomly select one pixel from each neighbouring superpixel

1-2) Add the particles at these pixels into the proposal set



K=3



SPM-BP: Neighbourhood Propagation

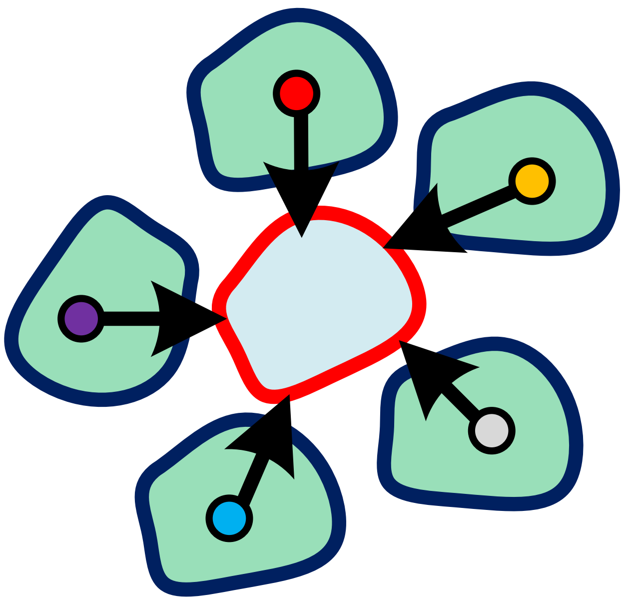
✓Step 1. Particle propagation

✓Step 2. Data cost computation

✓Step 3. Message update

1-1) Randomly select one pixel from each neighbouring superpixel

1-2) Add the particles at these pixels into the proposal set



K=3

SPM-BP: Neighbourhood Propagation

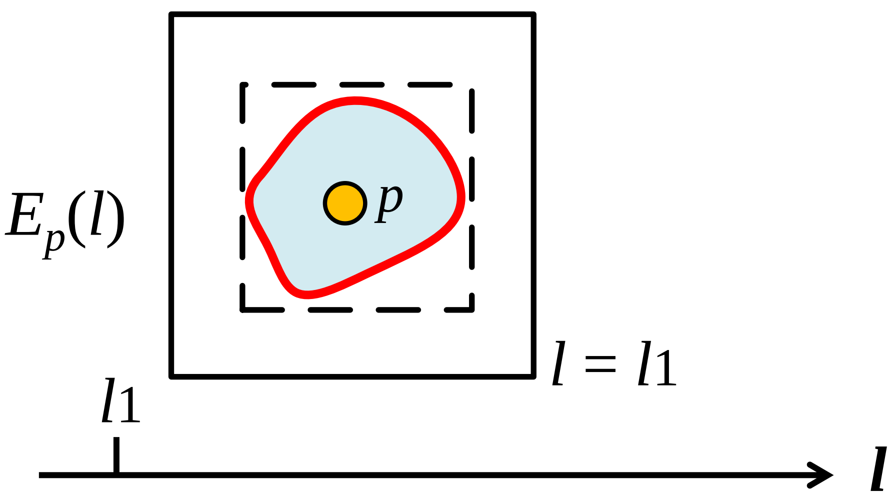
✓Step 1. Particle propagation

✓Step 2. Data cost computation

✓Step 3. Message update

2-1) Compute the raw matching data cost of these labels in a slightly enlarged region

2-2) Compute the aggregated data cost for each label by performing EAF on the raw matching cost



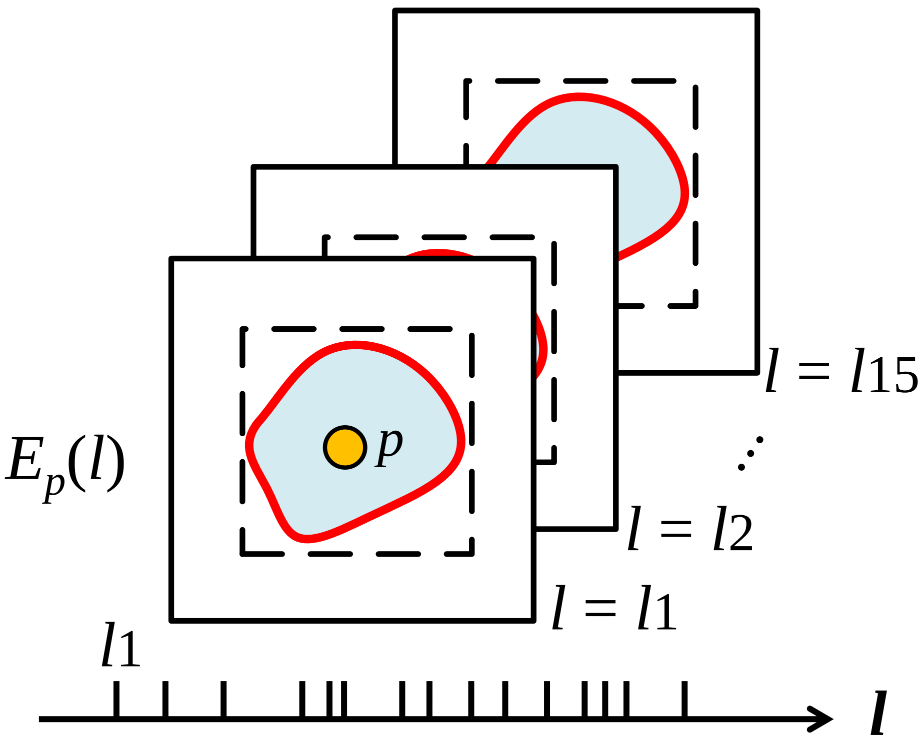
$$E_p(l_p; W) = \sum_{r \in W} \omega_{pr} C_r(l_p)$$

SPM-BP: Neighbourhood Propagation

✓Step 1. Particle propagation

✓Step 2. Data cost computation

✓Step 3. Message update

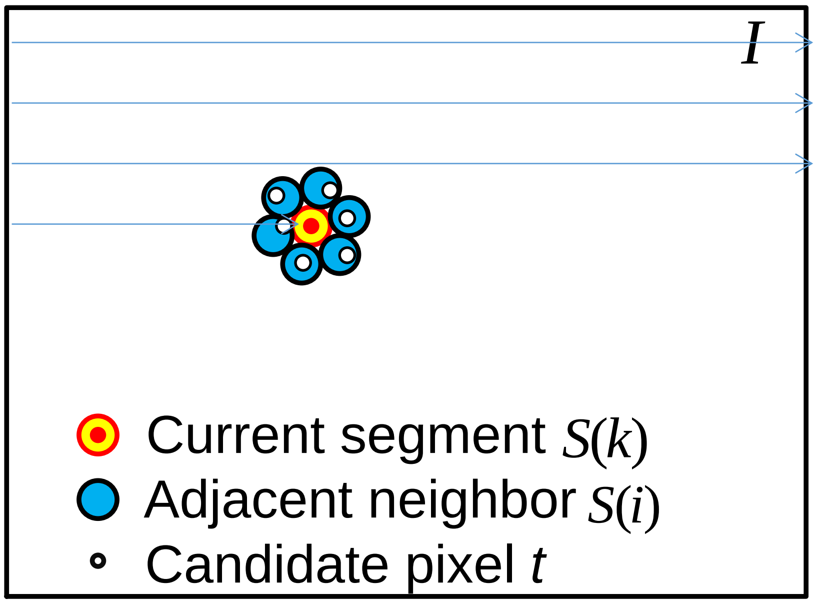


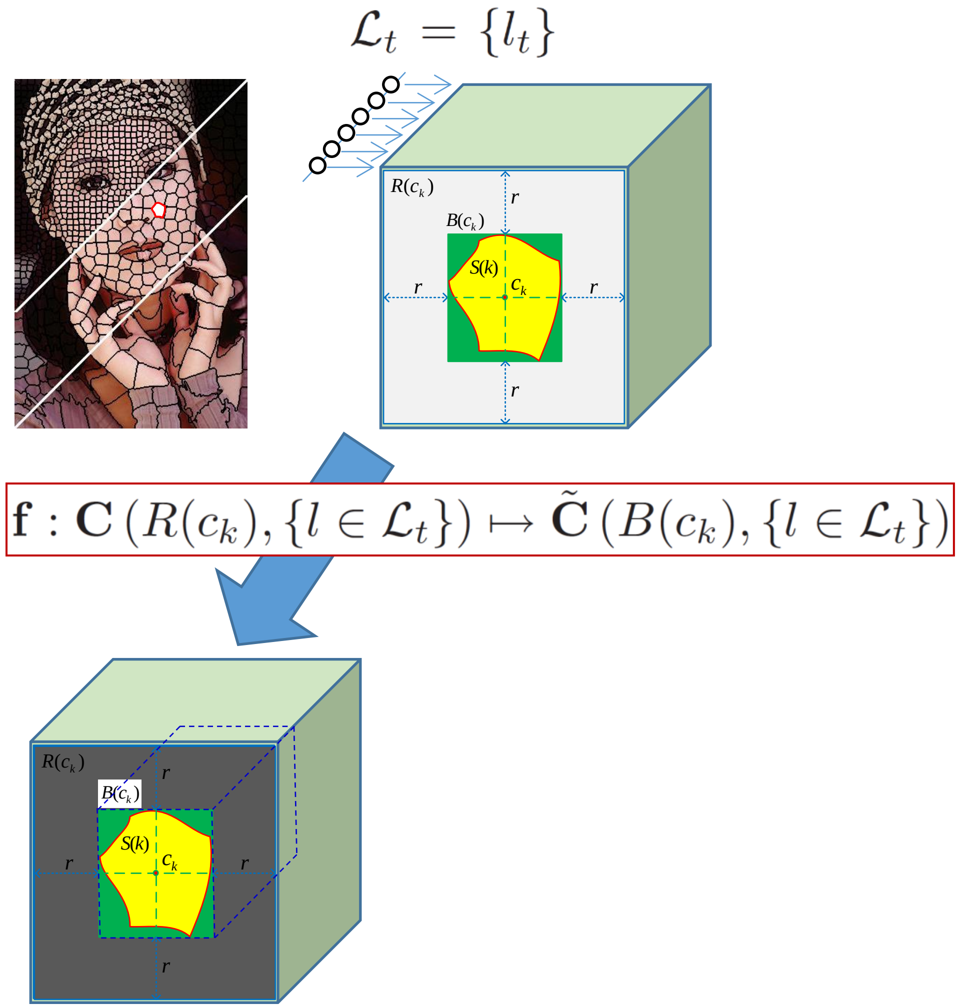
2-1) Compute the raw matching data cost of these labels in a slightly enlarged region

2-2) Compute the aggregated data cost for each label by performing EAF on the raw matching cost

$$E_p(l_p; W) = \sum_{r \in W} \omega_{pr} C_r(l_p)$$

O(1)-time Cost Aggregation for Subimages

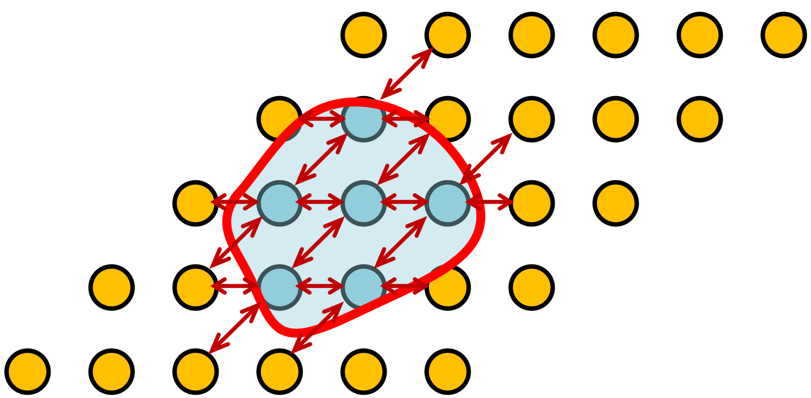




SPM-BP: Neighbourhood Propagation

- ✓Step 1. Particle propagation
- ✓Step 2. Data cost computation
- ✓Step 3. Message update

3-1) Perform message passing for pixels within the superpixel.



$$B_p(l_p) = E_p(l_p) + \sum_{q \in \mathcal{N}_p} m_{qp}^{(T)}(l_p)$$

$$l_p \in \mathcal{R}'_p$$

SPM-BP: Neighbourhood Propagation

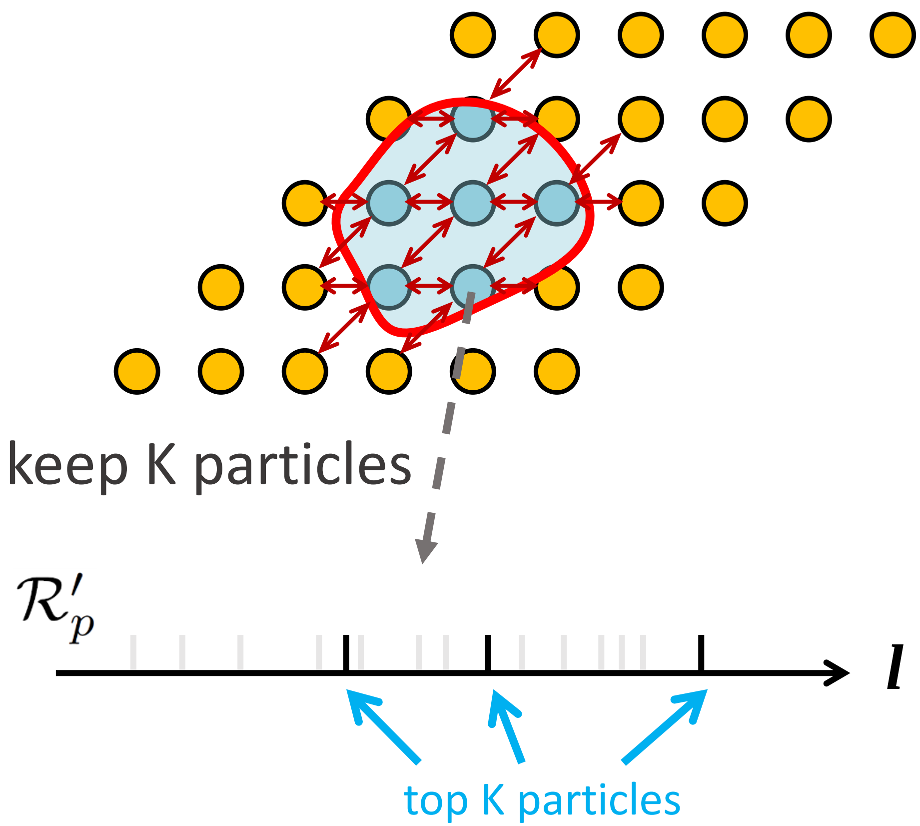
✓Step 1. Particle propagation

✓Step 2. Data cost computation

✓Step 3. Message update

3-1) Perform message passing for pixels within the superpixel.

3-2) Keep *K* particles with the smallest disbeliefs at each pixel.



$$\operatorname{argminK}_{l \in \mathcal{R}'_p} B_p(l)$$

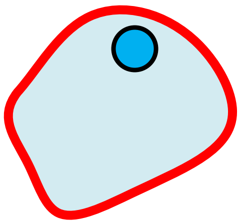
$$B_p(l_p) = E_p(l_p) + \sum_{q \in \mathcal{N}_p} m_{qp}^{(T)}(l_p)$$

$$l_p \in \mathcal{R}'_p$$

SPM-BP: Random Search

- ✓Step 1. Particle propagation
- ✓Step 2. Data cost computation
- ✓Step 3. Message update

1-1) Randomly select one pixel in the visiting superpixel



SPM-BP: Random Search

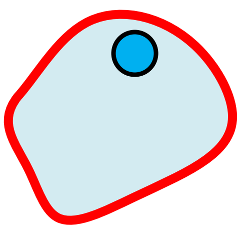
✓Step 1. Particle propagation

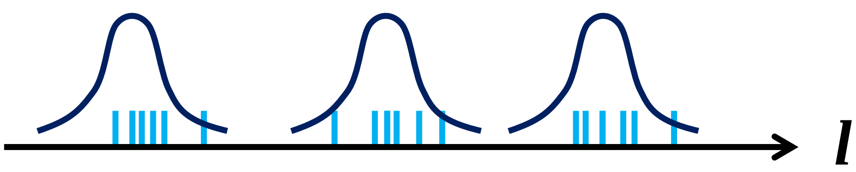
✓Step 2. Data cost computation

✓Step 3. Message update

1-1) Randomly select one pixel in the visiting superpixel

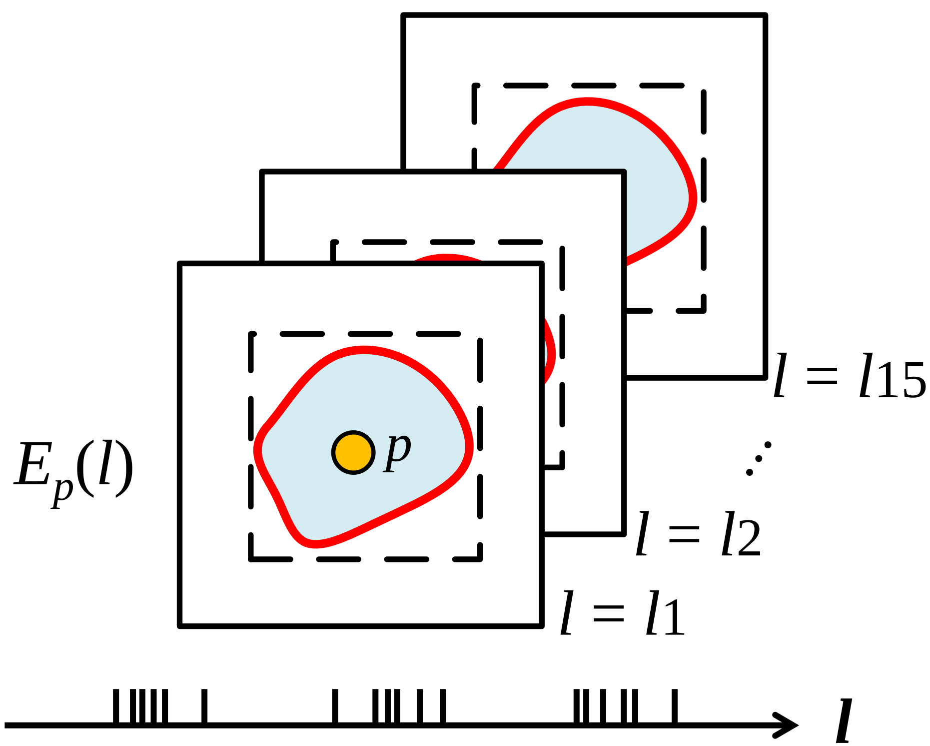
1-2) Generate new proposals around the sampled particles





SPM-BP: Random Search

- ✓Step 1. Particle propagation
- ✓Step 2. Data cost computation
- ✓Step 3. Message update



2-1) Compute the raw matching data cost of these labels in a slightly enlarged region

2-2) Compute the aggregated data cost for each label by performing EAF on the raw matching cost

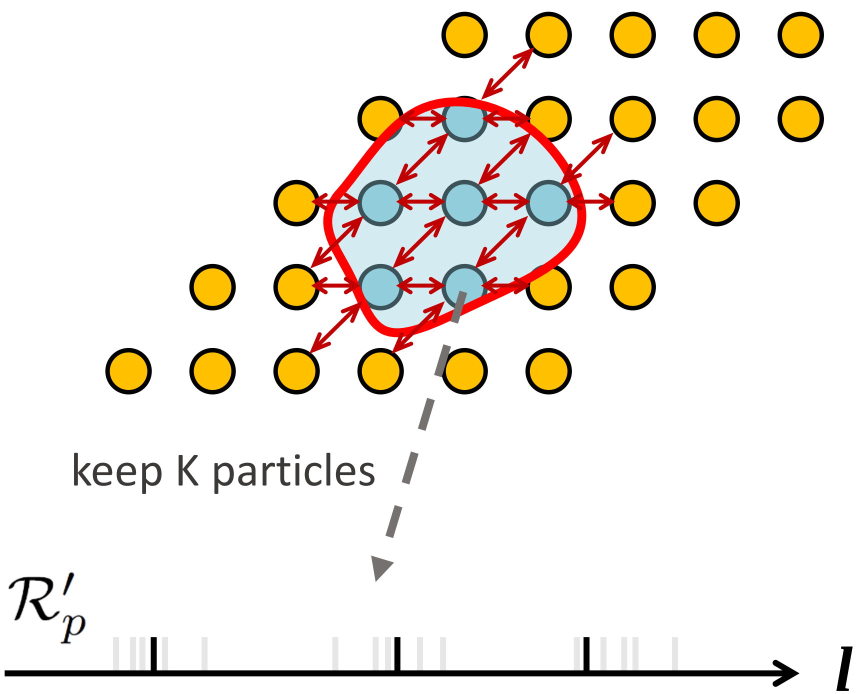
$$E_p(l_p; W) = \sum_{r \in W} \omega_{pr} C_r(l_p)$$

SPM-BP: Random Search

✓Step 1. Particle propagation

✓Step 2. Data cost computation

✓Step 3. Message update



3-1) Perform message passing for pixels within the superpixel.

3-2) Keep K particles with the smallest disbeliefs at each pixel.

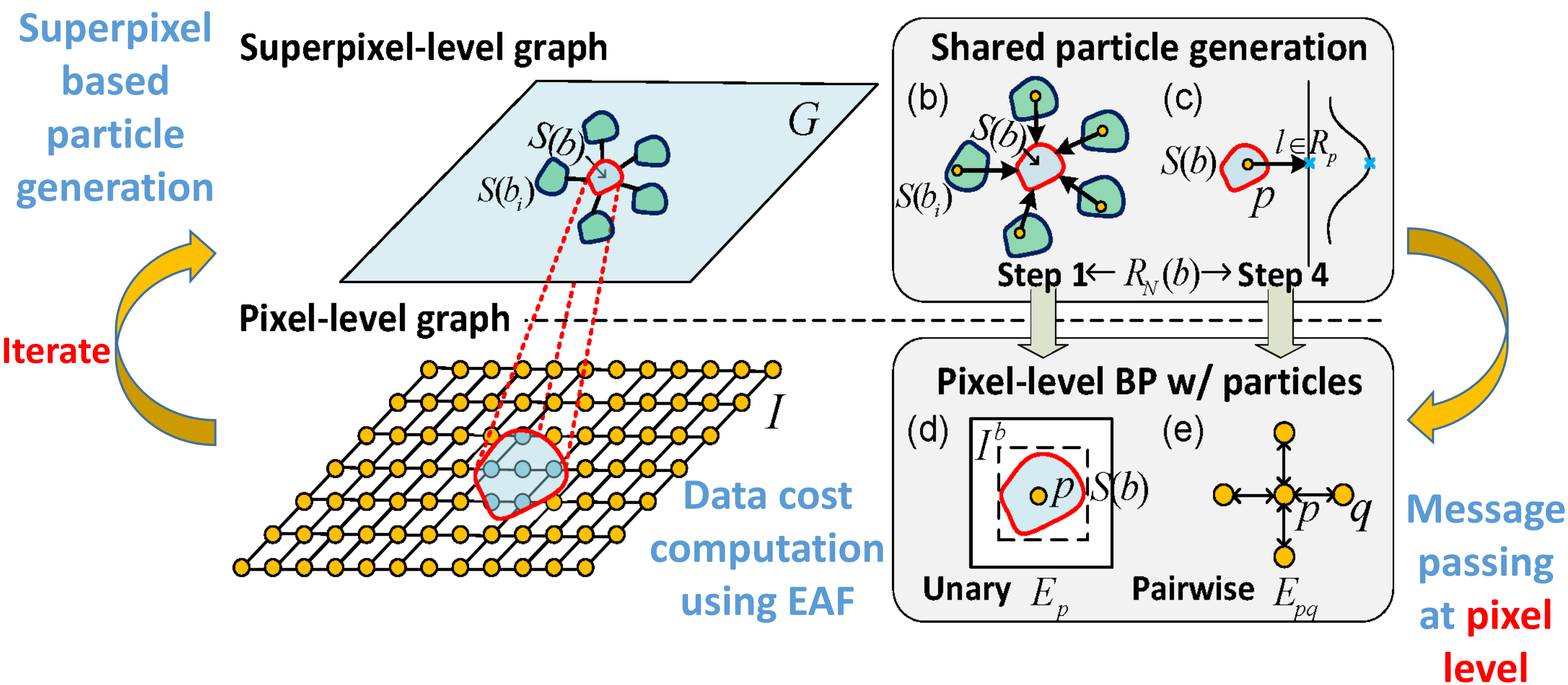
$$\mathrm{argminK}_{l \in \mathcal{R}'_p} B_p(l)$$

$$B_p(l_p) = E_p(l_p) + \sum_{q \in \mathcal{N}_p} m_{qp}^{(T)}(l_p)$$

$$l_p \in \mathcal{R}'_p$$

SPM-BP: Recap

Random Initialization



Final labels

Complexity Comparison

	PMF* [32]	PMBP [8]	**SPM-BP**
Data Cost	$O(N \log L)$	$O(\lvert W\rvert K N \log L)$	$O(K N \log L)$
Message Passing	-	$O(K^2 N \log L)$	$O(K^2 N \log L)$

- $\lvert W\rvert$ – local window size (e.g. 31x31 for stereo)
- K – number of particles used (small constant)
- N – number of pixels
- L – label space size (e.g. over 10 million for flow)

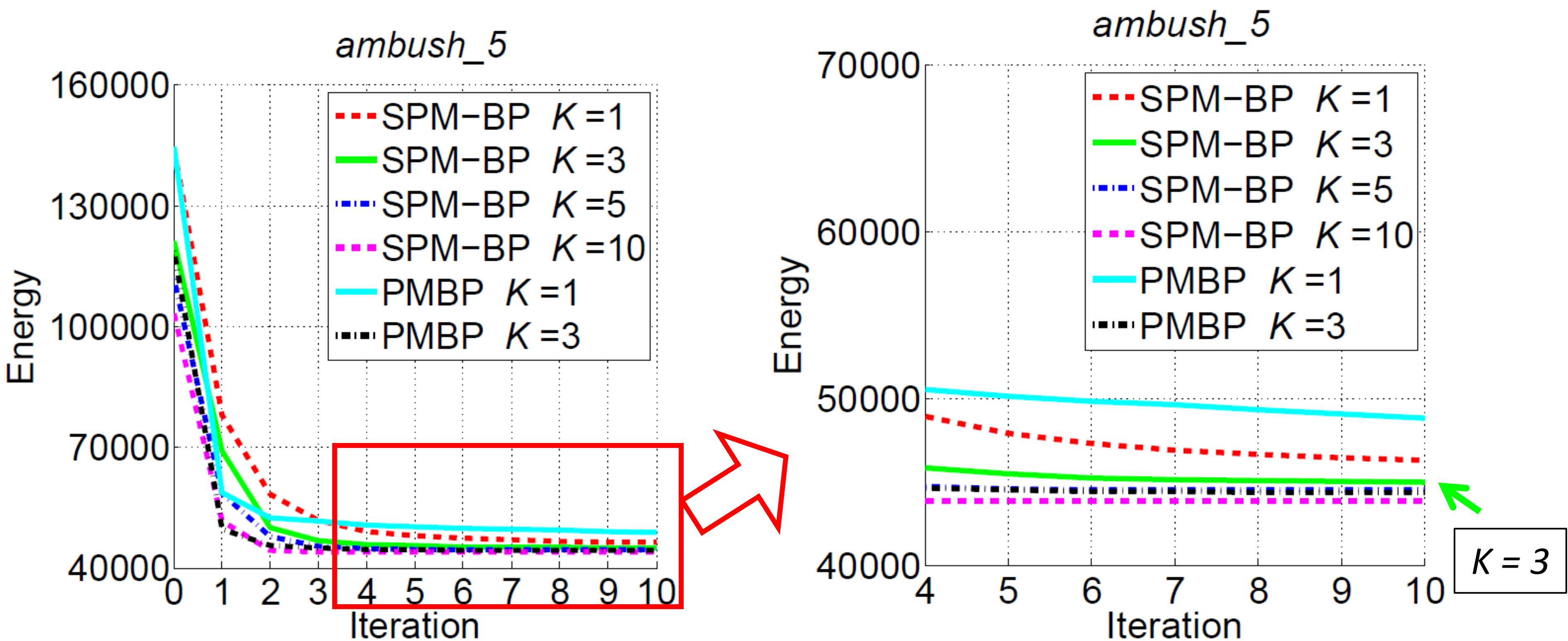
*PMF stores only one best particle (K = 1) per pixel node, thus requiring more iterations than the other two methods.

Example Applications of Our SPM-BP

- **Stereo with slanted surface supports**
  - **label**: 3D plane normal $l_p = (a_p, b_p, c_p)$
  - **Matching features**: color + gradient
  - **Smoothness term**: deviation between two local planes
  - **Cross checking + post processing for occlusion**

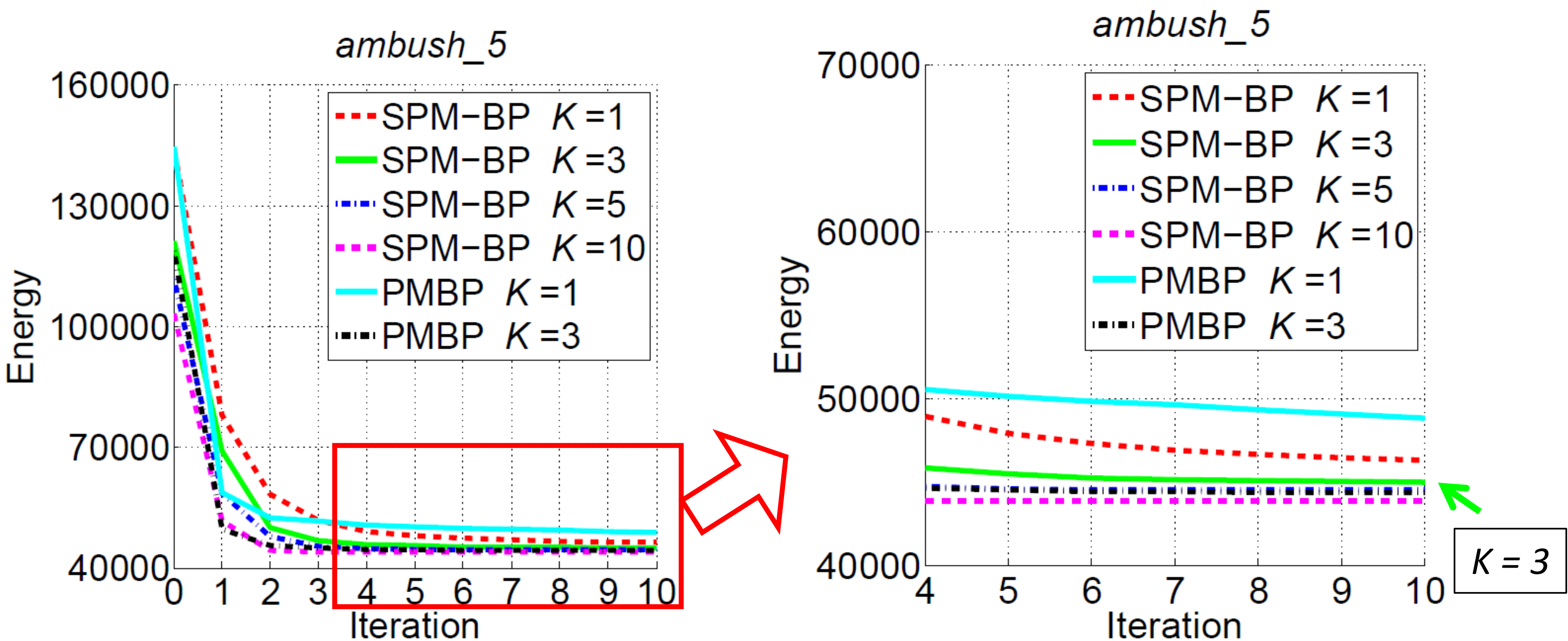
- **Large-displacement optical flow**
  - **label**: 2D displacement vector $l_p = (u, v)$
  - **Matching features**: color + Census transform
  - **Smoothness term**: truncated L_2 distance
  - **Cross checking + post processing for occlusion**

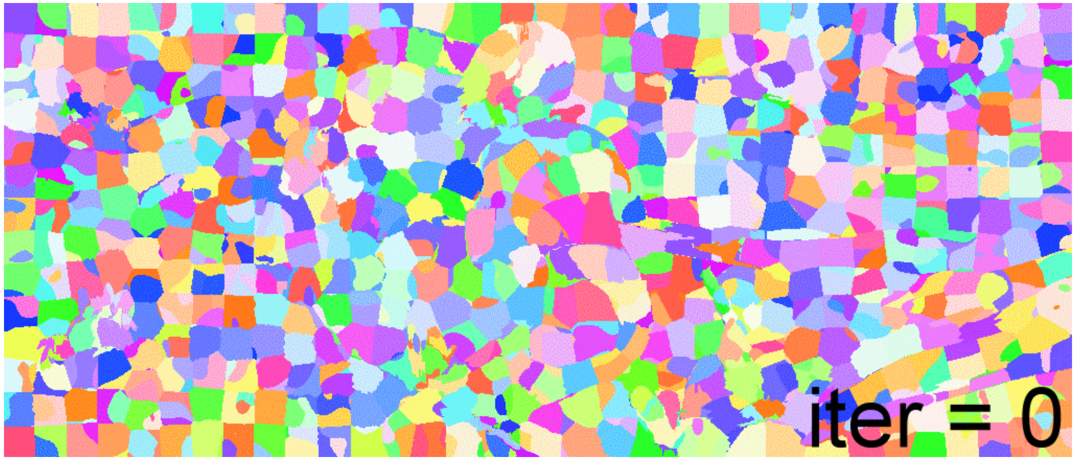
Convergence



#iteration = 5, K = 3

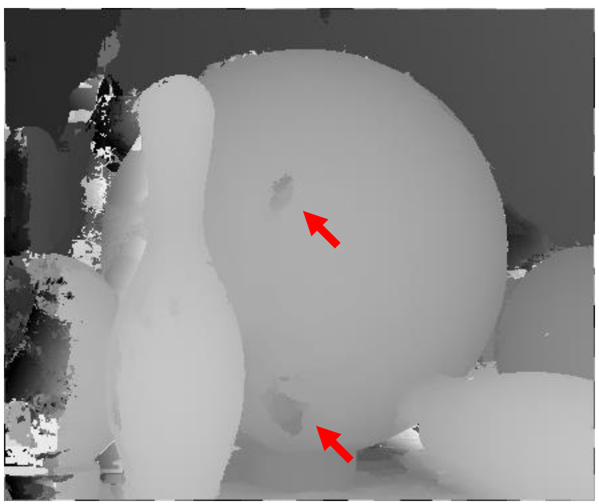
Convergence



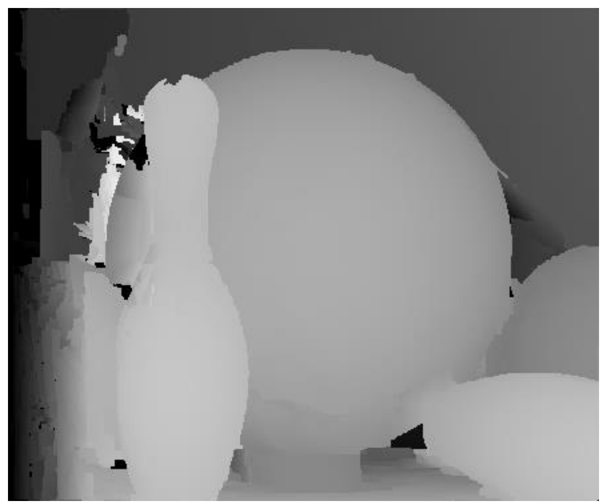


Stereo Results

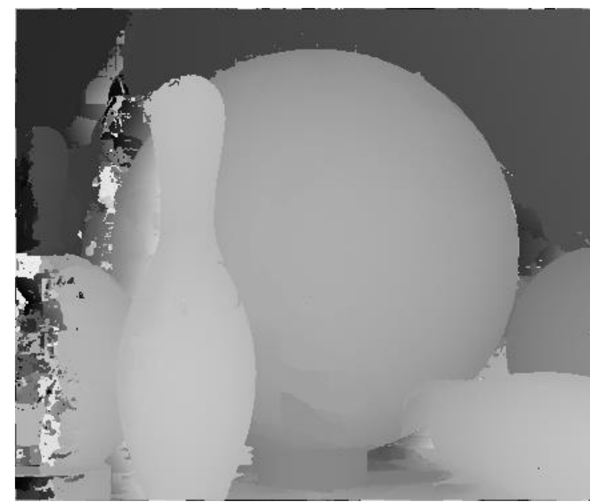
Stereo input



PMF
20 sec.



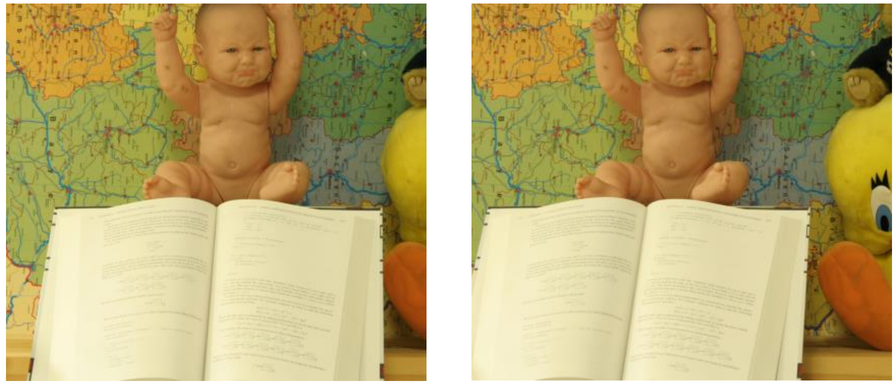
PMBP
3100 sec.



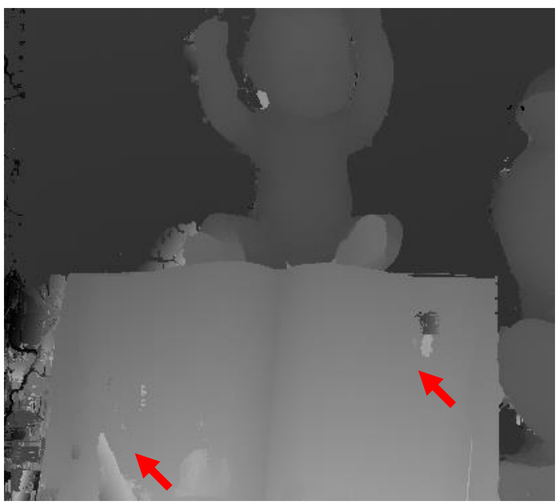
SPM-BP (ours)
30 sec.

Much faster than PMBP, and much better than PMF for textureless regions

Stereo Results



Stereo input



PMF
20 sec.

PMBP
3100 sec.

SPM-BP (ours)
30 sec.

Optical Flow Results

Optical flow input

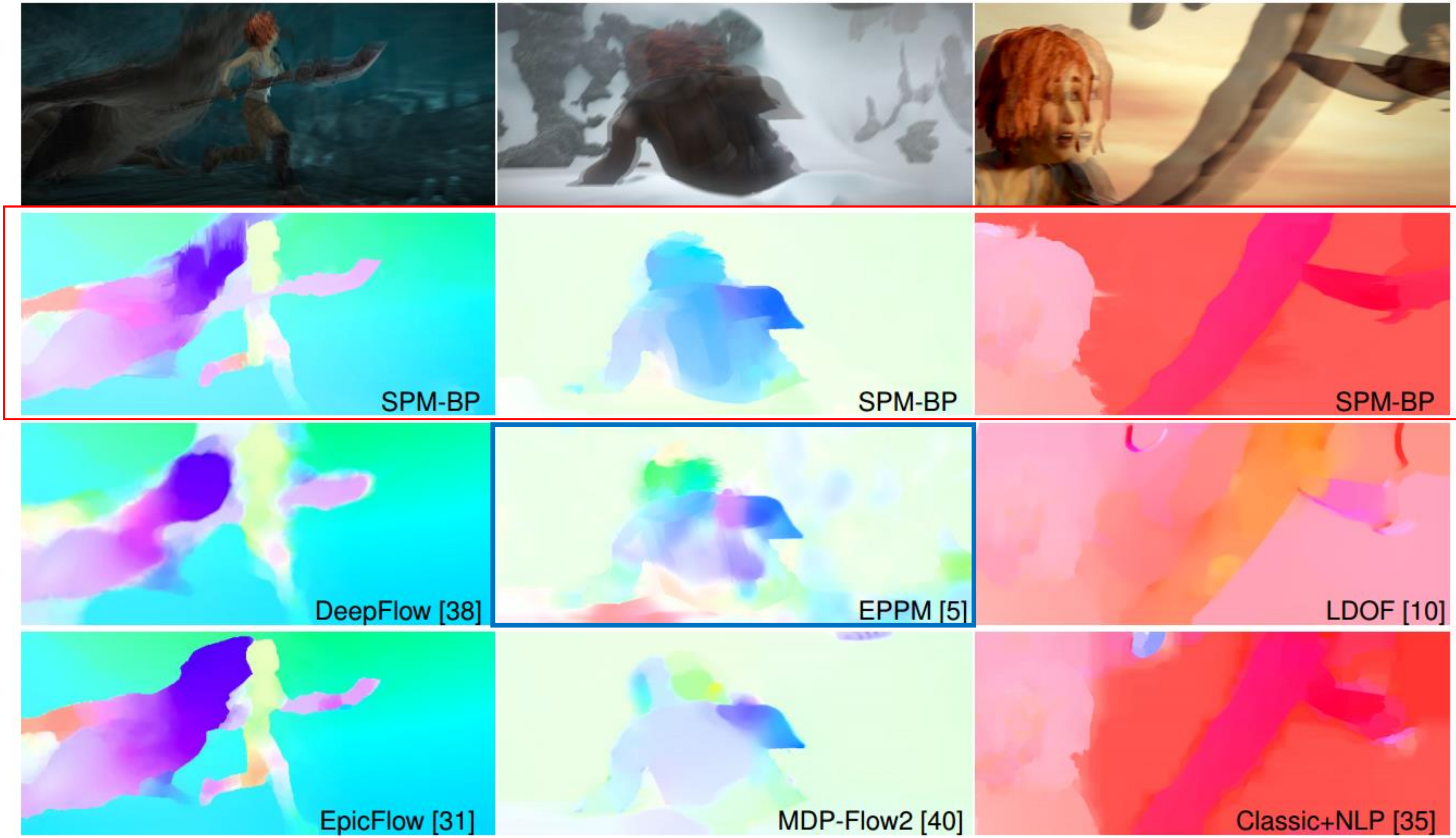
PMBP
2103 sec.

PMF
27 sec.

SPM-BP (ours)
42 sec.

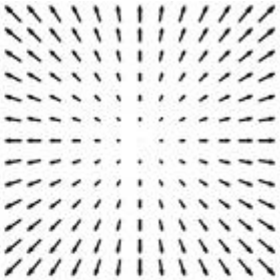
Much faster than PMBP, and much better than PMF for textureless regions

Optical Flow Results



SPM-BP Code is available online:
https://publish.illinois.edu/visual-modeling-and-analytics/efficient-inference-for-continuous-mrfs/

SPM-BP (ours)



Quantitative Performance Evaluation

Middlebury Stereo Performance (among over 160 listed methods)

Method	Avg. Rank	Avg. Error	Runtime(s)
PM-PM [39]	8.2	7.58	34 (GPU)
PM-Huber [17]	8.4	7.33	52 (GPU)
SPM-BP	12.1	7.71	30
PMF [24]	12.3	7.69	20
PMBP [7]	19.8	8.77	3100

Middlebury Stereo 2006 Performance

Dataset	PMF [25]	PMBP [7]	**SPM-BP**
Baby2	15.34	16.85	**12.82**
Books	**22.15**	27.57	22.52
Bowling2	15.95	15.20	**14.35**
Flowerpots	**24.59**	27.97	24.80
Lampshade1	25.02	30.22	**23.39**
Laundry	**26.77**	33.90	27.32
Moebius	21.47	25.09	**21.09**
Reindeer	**15.04**	21.57	16.02
Mean	20.79	24.79	**20.29**

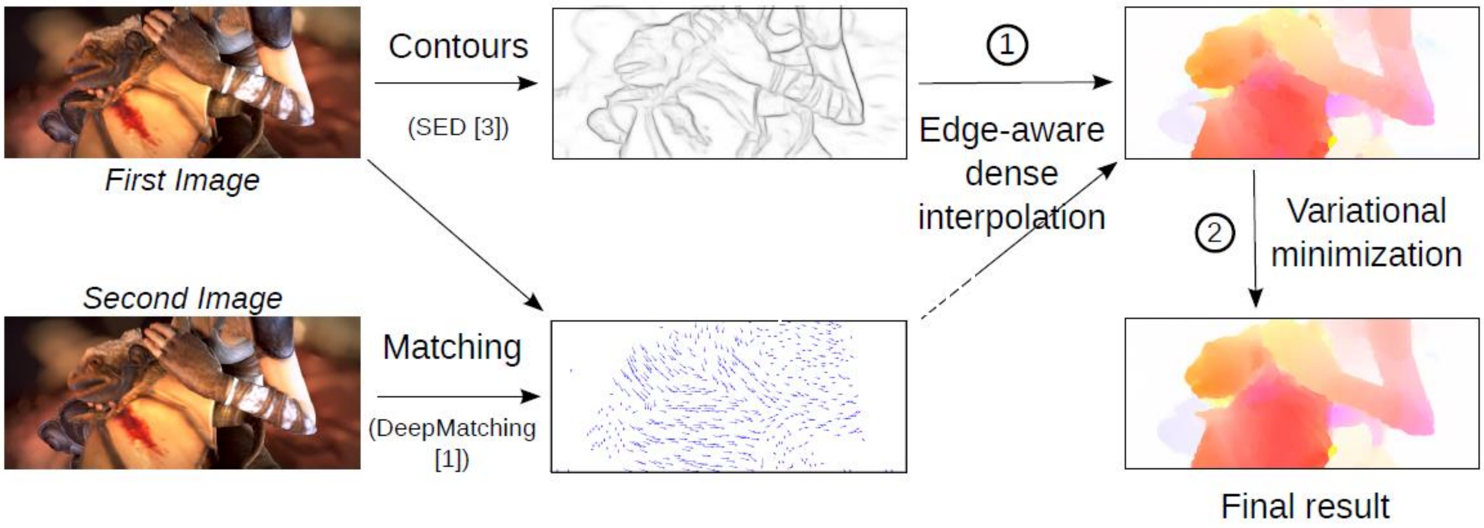
Optical Flow Performance on MPI Sintel Benchmark
(captured on 16/04/2015)

Method	EPE all Clean	Rank	EPE all Final	Rank	Runtime (Sec)
EpicFlow [30]	4.115	1	6.285	1	17
PH-Flow [41]	4.388	2	7.423	8	800
SPM-BP	5.202	5	7.325	6	42
DeepFlow [36]	5.377	7	7.212	4	19
LocalLayering [33]	5.820	13	8.043	13	-
MDP-Flow2 [38]	5.837	14	8.445	21	754
EPPM [5]	6.494	18	8.377	20	0.95*
S2D-Matching [21]	6.510	19	7.872	10	2000
Classic+NLP [34]	6.731	21	8.291	19	688
Channel-Flow [32]	7.023	24	8.835	26	>10000
LDOF [10]	7.563	25	9.116	28	30

Remarks

- A simple formulation, without
  - *complex* energy terms
  - a separate *initialization*
- Achieved top-tier performance
  - even when compared to *task-specific* techniques
- Applied on the full pixel grid
  - *w/o coarse-to-fine* steps

Comparison with [EpicFlow, CVPR'15]



Comparison with [EpicFlow, CVPR'15]

First Image

Second Image

$$E = \sum_p E_p(l_p; W) + \sum_p \sum_{q \in \mathcal{N}_p} E_{pq}(l_p, l_q)$$

p: pixel, N_p: 4 neighbors

Our SPM-BP

Final result

Plenary Speakers

Stephen P. Boyd - Stanford University

Convex Optimization with Abstract Linear Operators

Tuesday 22 March 2016, 10.20 - 11.10 (Grand Ballroom II & III)

Abstract:

Domain specific languages (DSLs) for convex optimization, such as CVX and YALMIP and the more recent CVXPY and Convex.jl, are very widely used to rapidly develop, prototype, and solve convex optimization problems of modest size, say, tens of thousands of variables, with linear operators described as sparse matrices. These systems allow a user to specify a convex optimization problem in a very succinct and natural way, and then solve the problem with great reliability, with no algorithm parameter tuning, and a reasonable performance loss compared to a custom solver hand designed and tuned for the problem. In this talk we describe recent progress toward the goal of extending these DSLs to handle large-scale problems that involve linear operators given as abstract operators with fast transforms, such as those arising in image processing and vision, medical

Learning: What are desired?

- *General-purpose solvers*
- *No tuning, no babysitting, just work with reliability*
- *With a reasonable loss to a custom solver*
- *Important to allow fast prototyping*

Comparison with [Full Flow, CVPR'16]

- Both use *no* descriptor matching
- Both optimize a simple global objective
- Both apply message passing techniques to solve MRF

- [Full Flow] has a linear complexity on the label space size *L*, i.e., O(*L*), in contrast to our O(log*L*)
- [Full Flow] is much slower, esp. cost volume computation
  - Multi-core parallelization w hyper-threading used in [Full Flow]
- SPM-BP has proven SoA results on slanted stereo matching

SPM-BP ver 2.0 – Among Top-Performing Methods

- Performance (*EPE all*) on MPI Sintel test benchmark

	Clean	Final	Runtime
SPM-BP v2 (w/ simple twists over SPM-BP)	**3.515**	**5.812**	**7 sec**
FullFlow	3.601	5.895	~4 min
EpicFlow	4.115	6.285	17 sec
SPM-BP	5.202	7.325	42 sec

MPI Sintel Dataset About Downloads Results FAQ Contact Signup Log

Final Clean

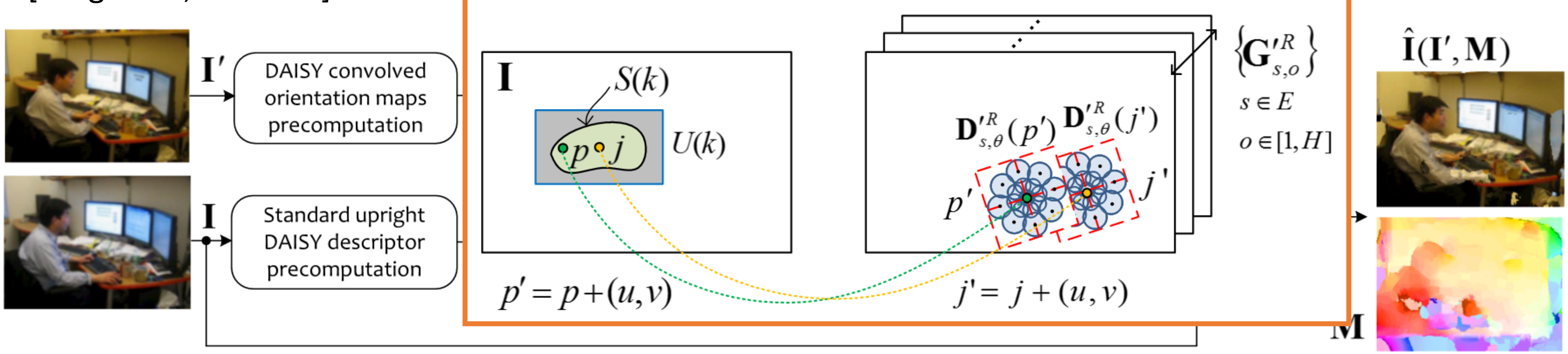
	EPE all	EPE matched	EPE unmatched	d0-10	d10-60	d60-140	s0-10	s10-40	s40+	
GroundTruth [1]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	Visualize Results
PGM-C [2]	5.591	2.672	29.389	4.975	2.340	1.791	1.057	3.421	33.339	Visualize Results
RicFlow [3]	5.620	2.765	28.907	5.146	2.366	1.679	1.088	3.364	33.573	Visualize Results
FlowFields+ [4]	5.707	2.684	30.356	4.691	2.117	1.793	1.131	3.330	34.167	Visualize Results
DeepDiscreteFlow [5]	5.728	2.623	31.042	5.347	2.478	1.590	0.959	3.072	35.819	Visualize Results
SBFlow [6]	5.791	2.280	34.405	4.409	2.024	1.471	0.975	3.475	35.270	Visualize Results
FlowFields [7]	5.810	2.621	31.799	4.851	2.232	1.682	1.157	3.739	33.890	Visualize Results
SPMBPv2 [8]	5.812	2.754	30.743	4.736	2.255	1.933	1.048	3.468	35.118	Visualize Results
FullFlow [9]	5.895	2.838	30.793	4.905	2.506	1.913	1.136	3.373	35.592	Visualize Results
CPM-Flow [10]	5.960	2.990	30.177	5.038	2.419	2.143	1.155	3.755	35.136	Visualize Results
GlobalPatchCollider [11]	6.040	2.938	31.309	5.310	2.624	1.824	1.102	3.589	36.455	Visualize Results

Future Work (1/2)

- **Applications:** robust cross-scene matching, annotation transfer, scene labeling [Liu et al., PAMI'11]

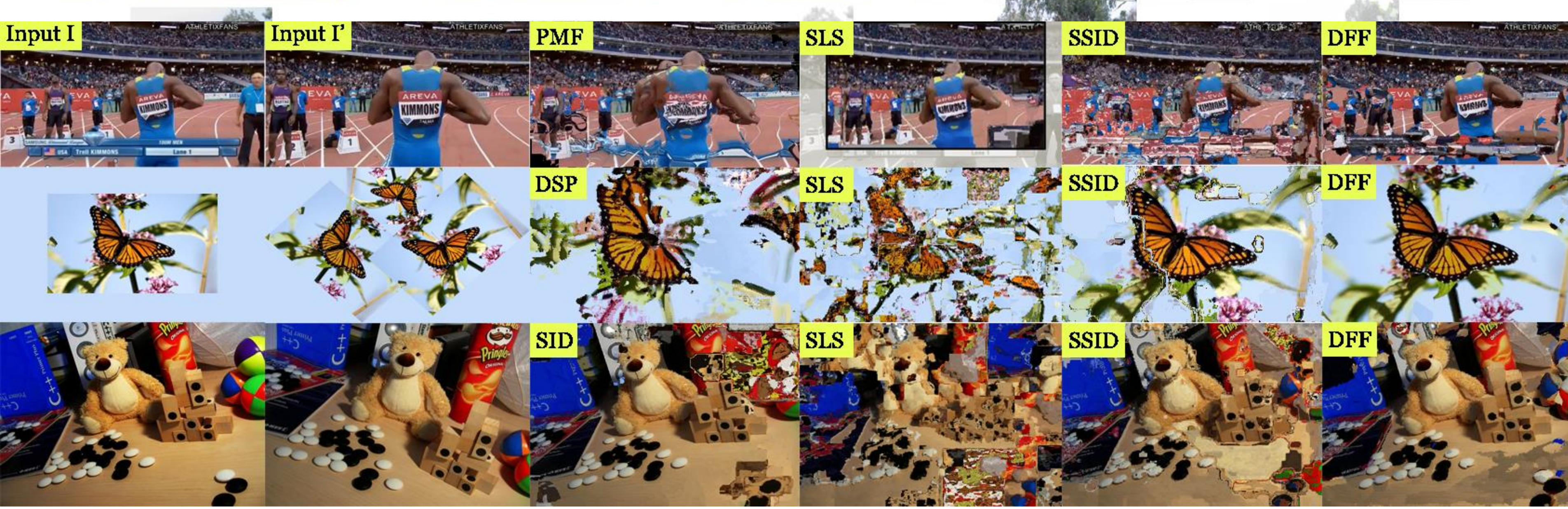
Augmented label space: $l(p) = (u, v, s, \vartheta)$ for each pixel p

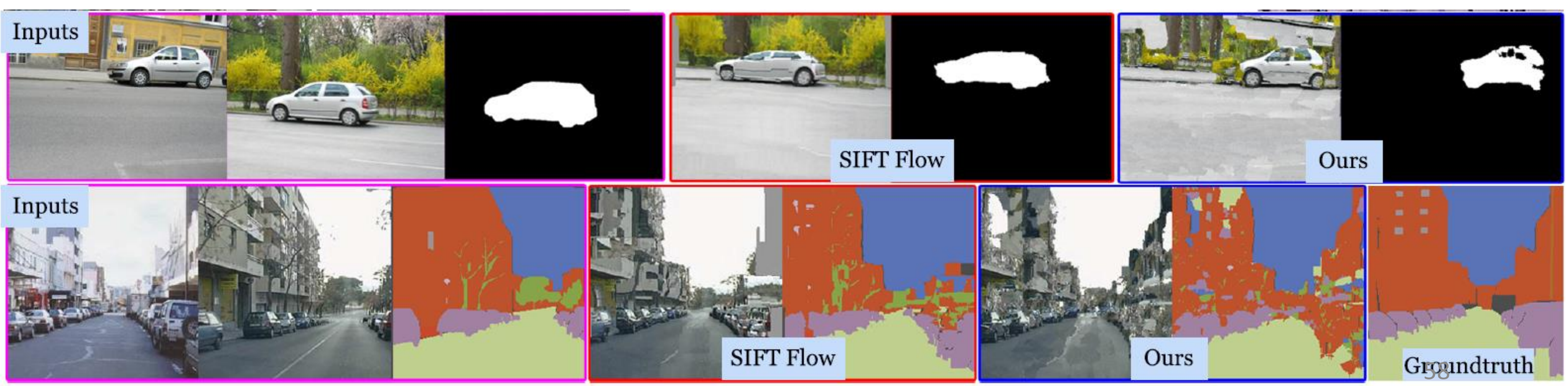
[Yang et al., CVPR'14]



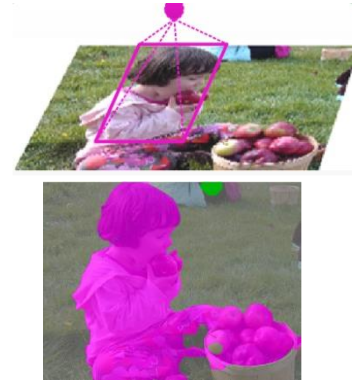
* H. Yang et al., "Daisy Filter Flow: A Generalized Discrete Approach to Dense Correspondences," CVPR 2014.

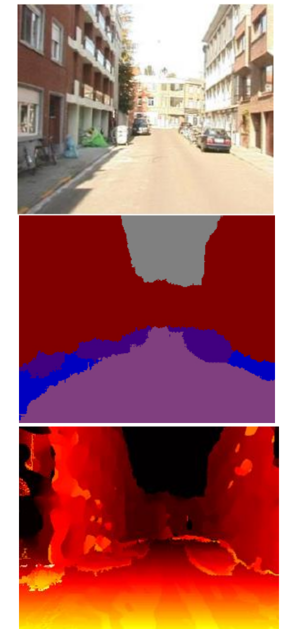

Input I
Input I'
NRDC
SIFT Flow
Ours


Input I
Input I'
PMF
SLS
SSID
DFF
DSP
SLS
SSID
DFF
SID
SLS
SSID
DFF

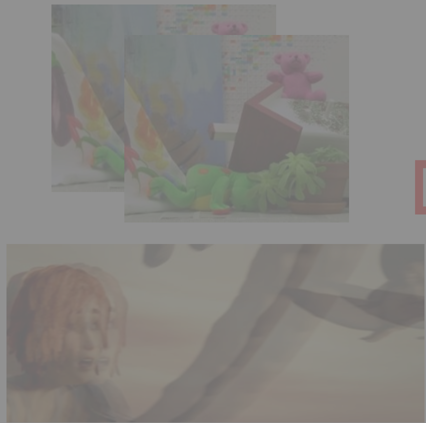

Inputs
SIFT Flow
Ours
Inputs
SIFT Flow
Ours
Groundtruth

Future Work (2/2)





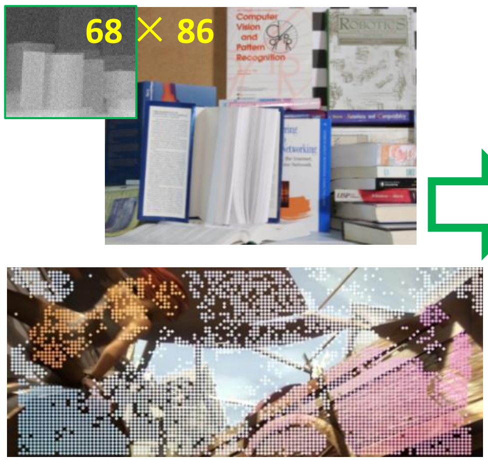
- **Modeling:**
  - More expressive MRF/CRF models with higher-order terms [ECCV'12, IJCV'15, CVPR'15, TMM'16]
  - Joint labeling tasks in product label-spaces [ECCV'12]
- **Inference:**
  - Efficient approaches to deal with complex models
  - More theoretic study on convergence [ICML'15]
- **Deep learning:**
  - Learned matching similarity from CNN models to deal with illumination, shadow, or transparent objects, e.g.
    - ✓ Jure Žbontar, and Yann LeCun, "Computing the Stereo Matching Cost With a Convolutional Neural Network" [CVPR 2015]
    - ✓ ….
  - Research issues including efficiency & generalization
  - "Geometry still lies at the heart of CV" – Stéphane Mallat
    - ✓ Pose, viewpoint, invariance, …



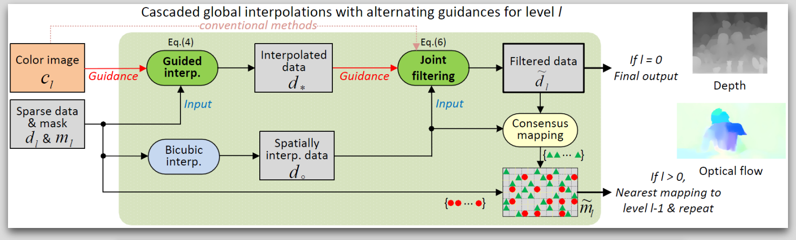
1: SPM-BP [ICCV'15]: Discrete Labeling Optimization for MRFs

$$E = \sum_{p} E_p(l_p; W) + \sum_{p} \sum_{q \in \mathcal{N}_p} E_{pq}(l_p, l_q)$$

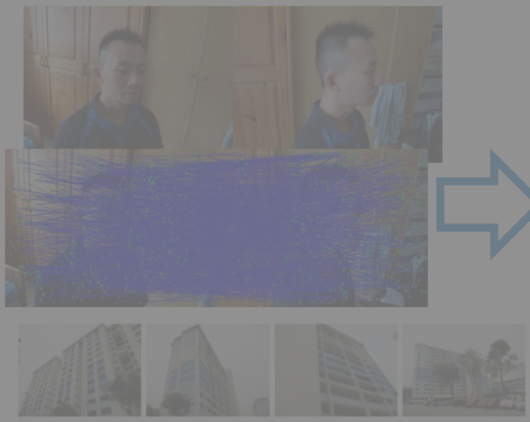
p: pixel, N_p: 4 neighbors



2: Fast Guided Global Interpolation [ECCV'16a]: from Sparse to Dense

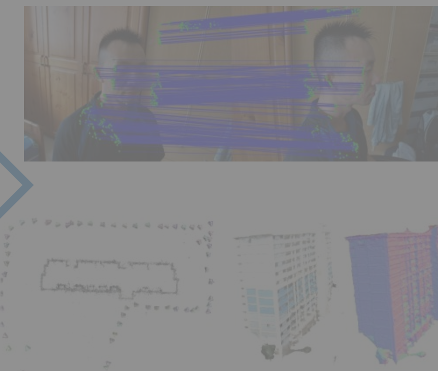


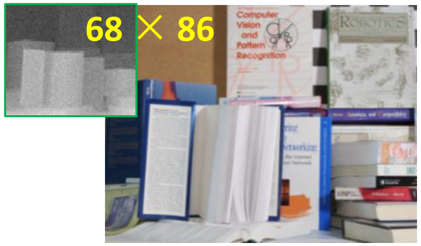
$$\mathcal{E}(\mathbf{u}) = (\mathbf{u}-\mathbf{f})^\top(\mathbf{u}-\mathbf{f}) + \lambda \mathbf{u}^\top \mathbf{A}\mathbf{u}$$



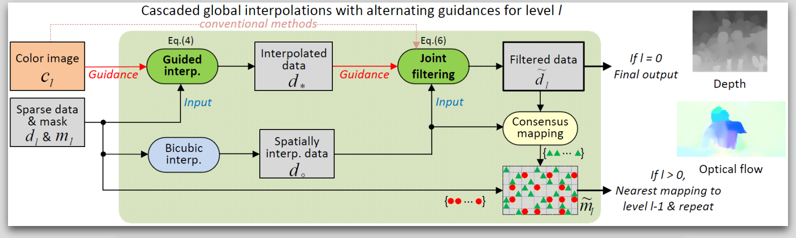
3: Coherence-Based Regression for Feature Matching [ECCV'14, '16b]

$$\arg\min_{\mathbf{w}} \sum_{i=1}^{N} C(1 - f(\mathbf{m}_i)) + \lambda \mathbf{w}^T G \mathbf{w}$$

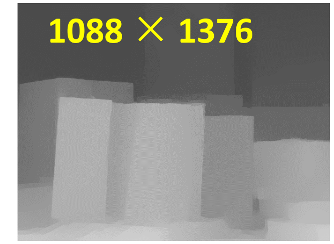




2: Fast Guided Global Interpolation [ECCV'16a]: from Sparse to Dense



$$\mathcal{E}(\mathbf{u}) = (\mathbf{u}-\mathbf{f})^\top(\mathbf{u}-\mathbf{f}) + \lambda\,\mathbf{u}^\top \mathbf{A}\mathbf{u}$$



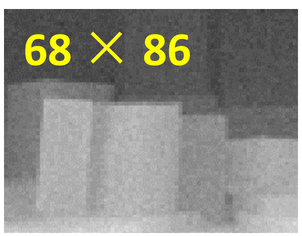
Part 2: Fast Guided Global Interpolation for Sparse Input Data

- **Y. Li, D. Min, M. N. Do, and J. Lu, "Fast Guided Global Interpolation for Depth and Motion," ECCV 2016. (Spotlight)**
- D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, and M. N. Do, "Fast Global Image Smoothing Based on Weighted Least Squares," TIP 2014. (Included in official OpenCV 3.1 release)

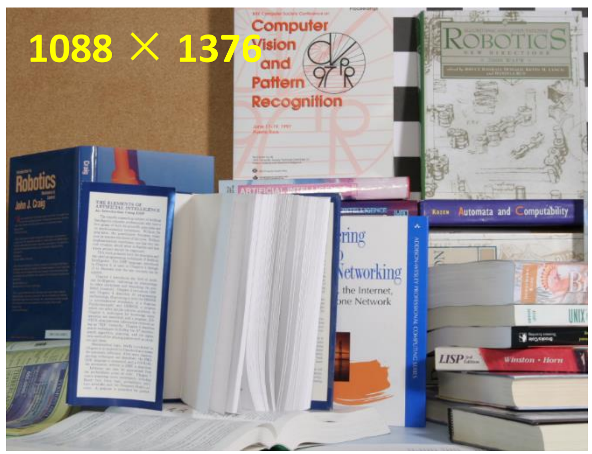
Introduction

Depth upsampling and ***motion interpolation*** are often required to generate a dense, high-quality, and high resolution depth map or optical flow field.

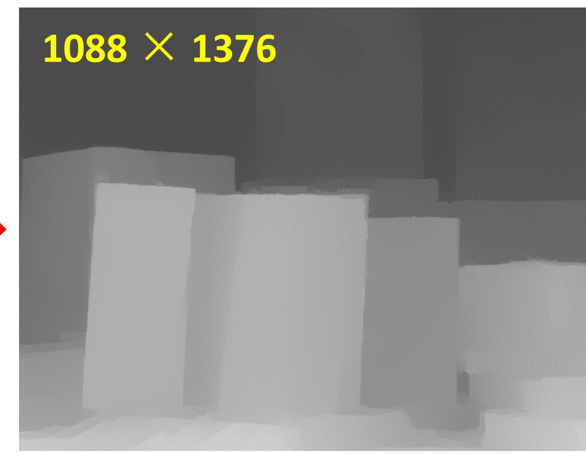
Low-res & noisy depth (ToF)



High-res. color guidance



High-res. depth



Depth upsampling (color guided)

Input TOF depth: noisy, low resolution, regularly distributed

Introduction

Depth upsampling and ***motion interpolation*** are often required to generate a dense, high-quality, and high resolution depth map or optical flow field.

Color frame and sparse matches from DM

DM data density < 1%

Dense optical flow field

Motion interpolation

Input matches: typically reliable, but highly scattered, varying density

*[**DM**: Weinzaepfel et al., "DeepFlow: Large displacement optical flow with deep matching, ICCV 2013.]*

** General to use other* low-res. /semi-dense matches as input : [FlowFields, ICCV’15], [FullFlow, CVPR’16], etc

Motivation

Existing methods are often tailored to one *specific* task:

Depth upsampling	JBF [Kopf et al. 2007], MRF+nlm [Park et al. 2011], TGV [Ferstl et al. 2013], JGU [Liu et al. 2013], AR [Yang et al. 2014], Data-driven [Kwon et al. 2015], etc
Motion interpolation	EpicFlow [Revaud et al. 2015], [Drayer and Brox 2015] , [Leordeanu et al. 2013], etc

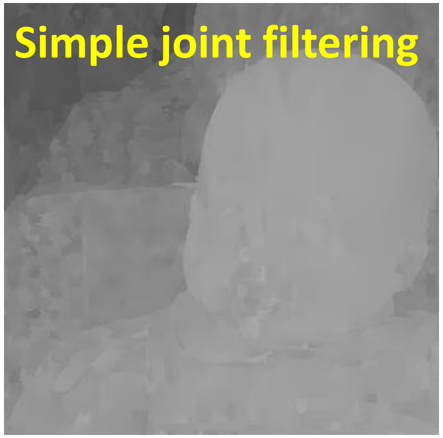
The common objective for both tasks is to ***densify*** a set of ***sparse data points***, either regularly distributed or scattered, to a full image grid through a 2D ***guided interpolation*** process.

***Our approach*: Fast Guided Global Interpolation (FGI)**

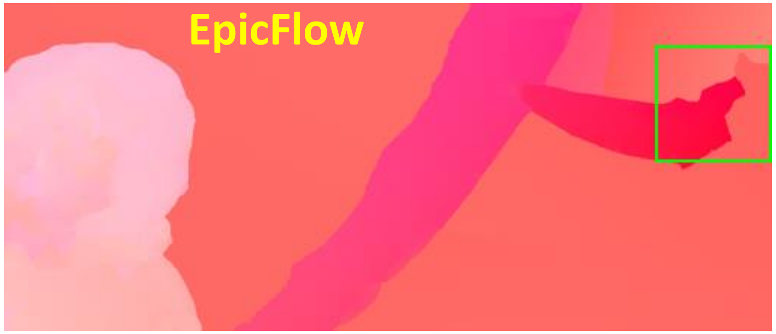
A unified approach that casts the guided interpolation problem into a hierarchical, global optimization framework.

Several Challenges, such as

- Texture-copy artifacts due to inconsistent structures



- Large occlusions, long-range propagation and extrapolation



- Loss of thin structures, missing motion boundaries
- Complex algorithms and often time-consuming

Weighted Least Square (WLS)-based Minimization

- Optimization on 2D signal (image)

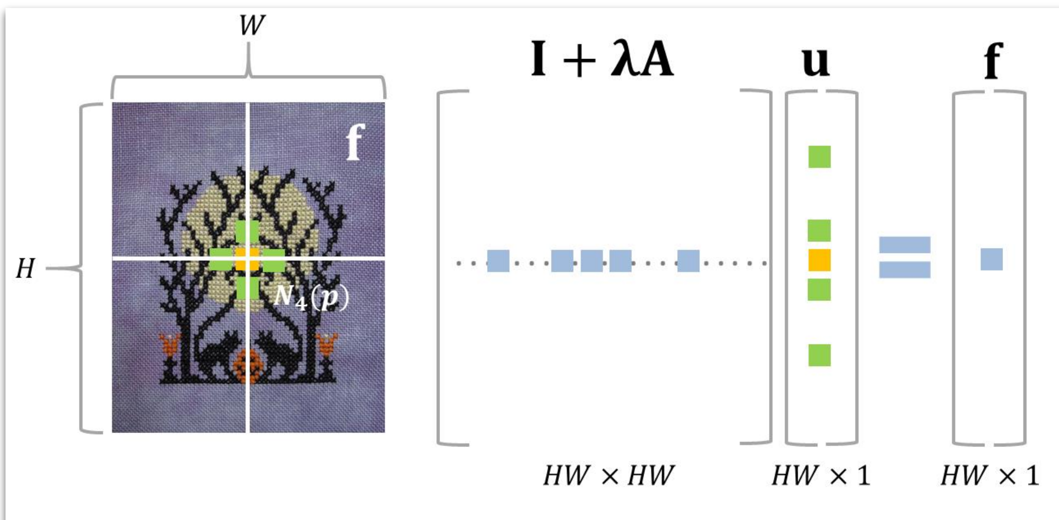
Given an input image f and a guidance image g, a desired output u **minimizes** the following

✔ No artifacts e.g. halo
✔ Various applications
✘ Not efficient

$$J(u) = \sum_{p} \left((u_p - f_p)^2 + \lambda \sum_{q \in \mathcal{N}(p)} w_{p,q}(g)(u_p - u_q)^2 \right)$$

$$(\mathbf{I} + \lambda \mathbf{A})\mathbf{u} = \mathbf{f}$$

$\mathbf{I}$: Identity matrix
$\mathbf{A}$: Spatially-varying Laplacian matrix



[Farbman et al., "Edge-preserving decompositions for multi-scale tone and detail manipulation," SIGGRAPH 2008]

Sparse data interpolation

$$(\mathbf{H} + \lambda \mathbf{A})\mathbf{u} = \mathbf{H}\mathbf{f}$$

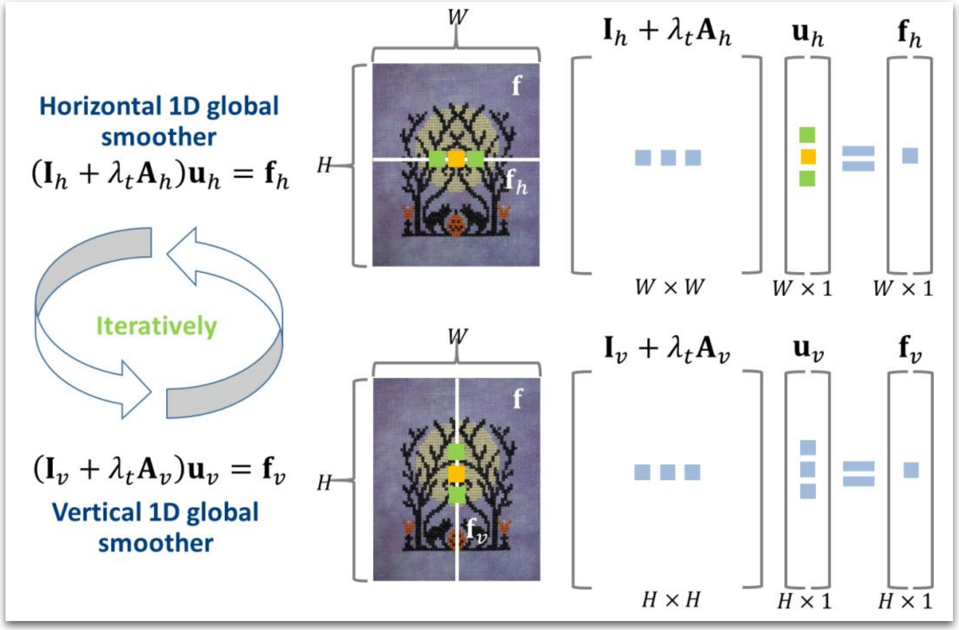
[Levin et al., SIGGRAPH 2004]

Fast Global Smoother (FGS) [TIP'14, OpenCV 3.1]

Quality & Applicability of Optimization + Efficiency of Local filters

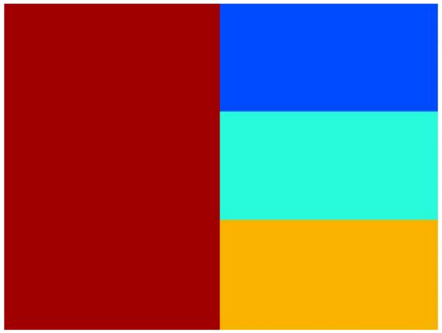
- Fast **O(*N*)** edge-preserving smoothing by approximating the solution of a large linear system with *two linear sub-systems*
- Efficient to solve a *tridiagonal matrix* with Gaussian elimination

Properties	**GF** [8]	**DT** [9]	**WLS** [15]	**Ours**
Runtime efficiency [Sec. V.C]	0.15s	0.05s	3.3s	0.10s
Smoothing quality [Sec. V.A]	halo	halo	no halo	no halo
L_γ norm smoothing ($0 < \gamma < 2$) [Sec. IV.D]	N.A.	N.A	N.A.	Yes
Using aggregated data [Sec. IV.E]	N.A.	N.A.	N.A.	Yes

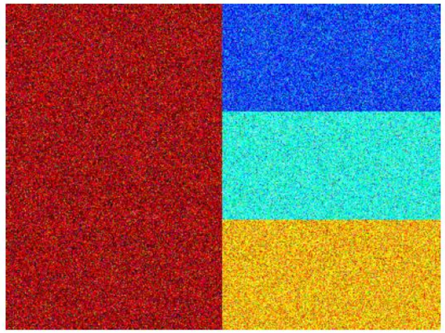


- High quality smoothing,
  but **10-30x** faster than SOA
- Support L_γ *smoothing* with IRLS (iterative re-weighted LS)
- Support *an aggregated (robust) data term*

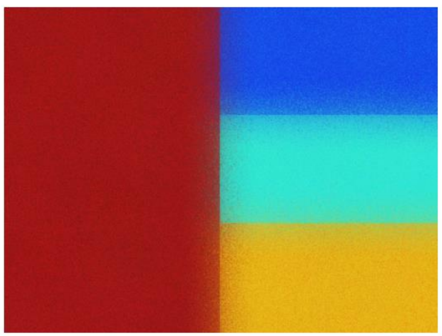
FGS Results https://sites.google.com/site/globalsmoothing/



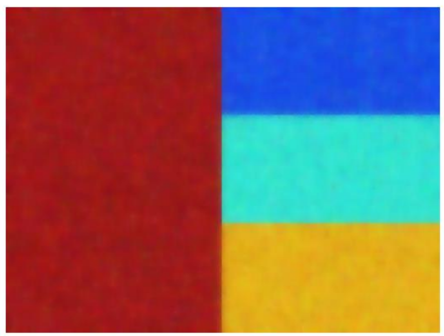
(a) Original



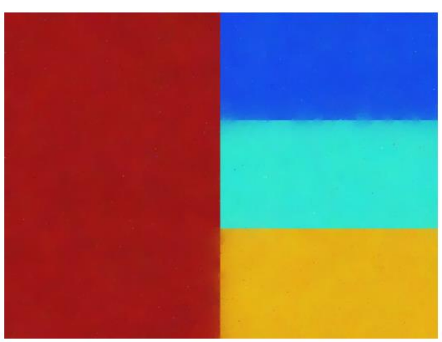
(b) Noisy input



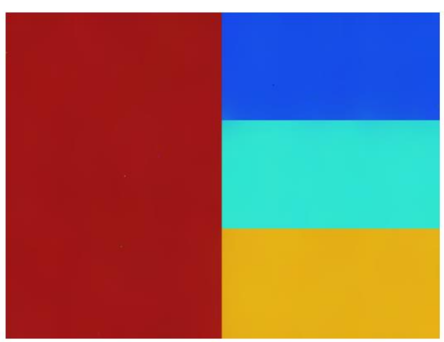
(c) Guided Filter [8]



(d) RF of DT [9]



(e) Original WLS [15]



(f) Proposed method

Structure extraction from texture w/ IRLS

Input

[Xu et al. SIGGRAPH'12]

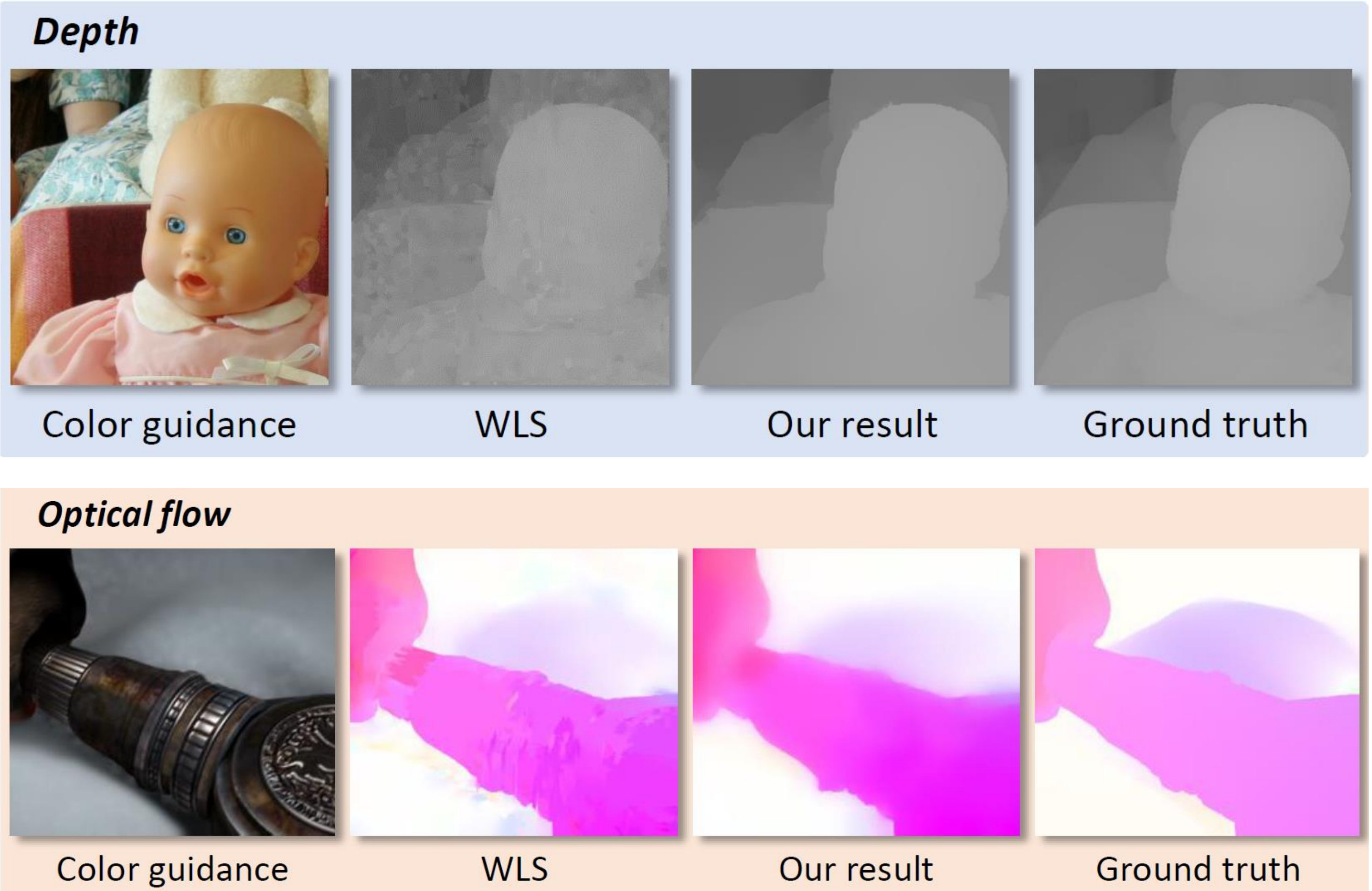
3.7s

FGS

0.6s

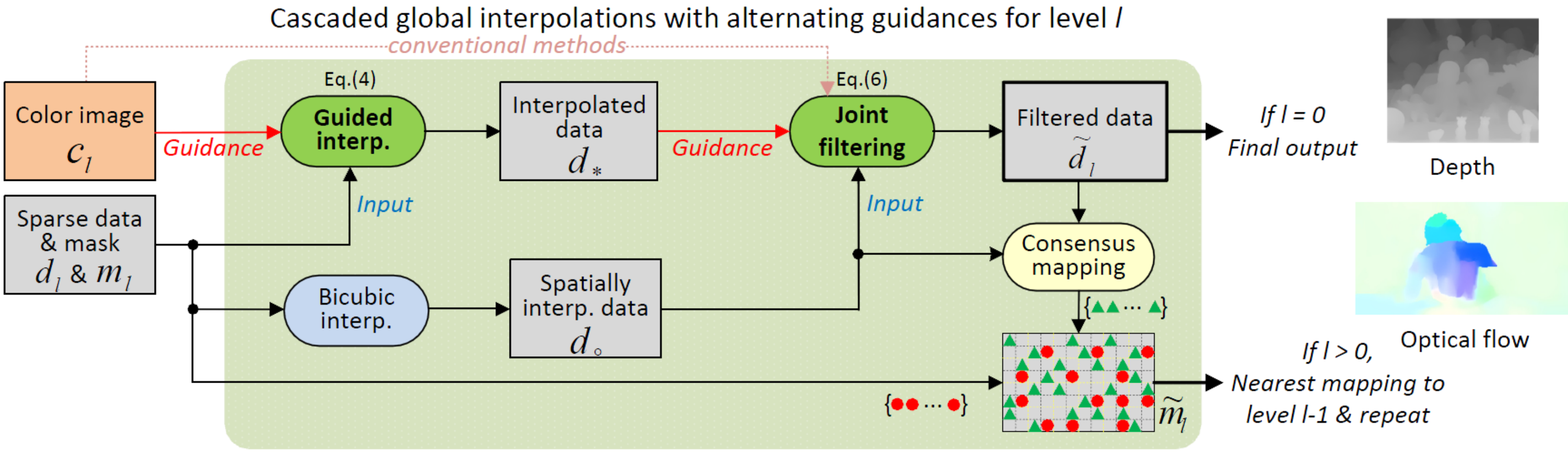
Handling imprecise sparse input

Single-Scale WLS Methods Vs. Our FGI Method



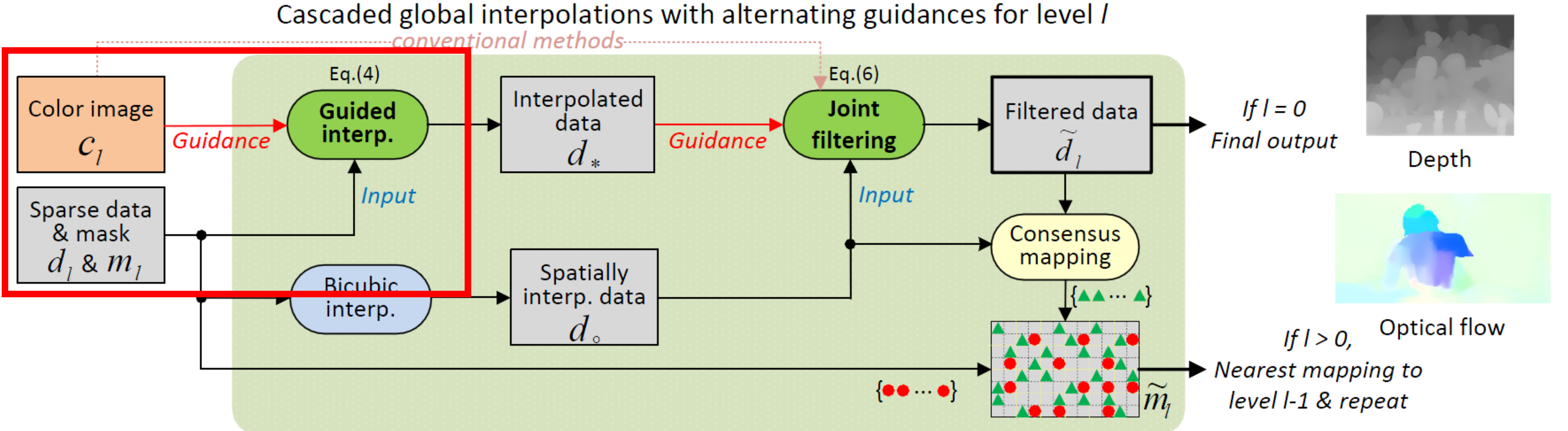
[**WLS**: *Farbman et al., “Edge-preserving decompositions for multi-scale tone and detail manipulation,” SIGGRAPH 2008*]
[**FGS**: *Min et al., "Fast global image smoothing based on weighted least squares," TIP 2014.*]

Our FGI Pipeline: Overview



- A *hierarchical* (*coarse-to-fine*), *multi-pass* guided interpolation framework
- Divide the problems into a sequence of interpolation tasks each with smaller scale factors
- Gradually fill the large gap between the sparse measurement and the dense data

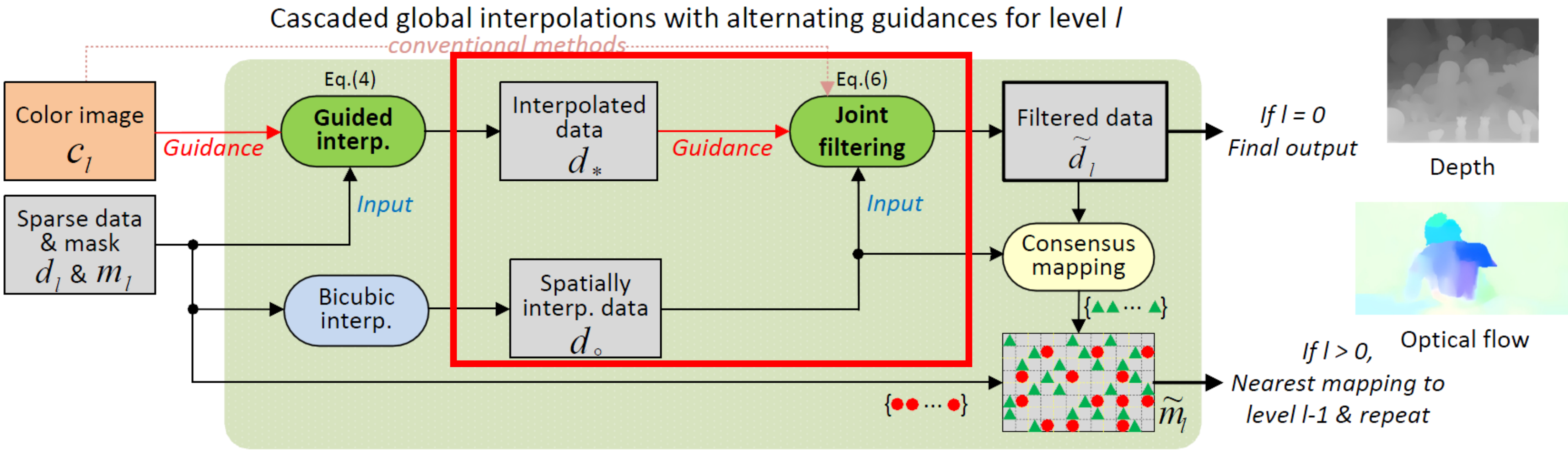
Our Pipeline: Filtering with Alternating Guidances



- From the coarse level *l*= *L*-1, we upsample the signal by a factor of 2 at each level by solving the following weighted least square (WLS) using the recent **FGS** solver.
- **Guided interp. :** $\mathcal{E}(\mathbf{d}_*) = (\mathbf{d}_* - \mathbf{d}_l)^\top \mathbf{M}_l (\mathbf{d}_* - \mathbf{d}_l) + \lambda_1 \mathbf{d}_*^\top \mathbf{A}_{c_l} \mathbf{d}_*$
  - The color image c_l as the guidance
  - $\mathbf{A}_{c_l}$ is the spatially varying Laplacian matrix defined by c_l

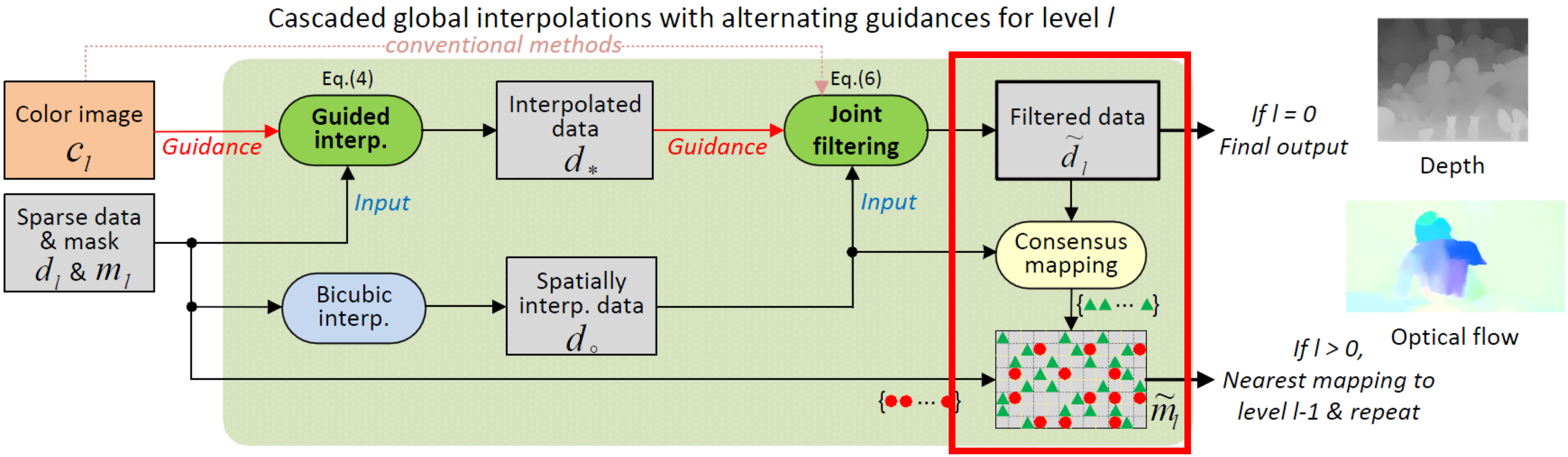
*[**FGS**: Min et al., "Fast global image smoothing based on weighted least squares," TIP 2014.]*
Why FGS? *100 ms for filteirng 1MPixels RGB images on 1 CPU core. More details in our paper.*

Our Pipeline: Filtering with Alternating Guidances



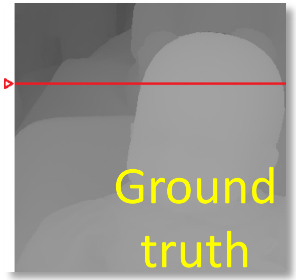
- Next, another WLS is solved with the output d_* as guidance and bicubic interpolated signal as input.
- **Joint filtering**: $\mathcal{E}(\tilde{\mathbf{d}}_l) = (\tilde{\mathbf{d}}_l - \mathbf{d}_\circ)^\top(\tilde{\mathbf{d}}_l - \mathbf{d}_\circ) + \lambda_2 \tilde{\mathbf{d}}_l^\top \mathbf{A}_{d_*} \tilde{\mathbf{d}}_l$
  - The intermediate interpolated map d_* as the guidance
  - $\mathbf{A}_{d_*}$ is the spatially varying Laplacian matrix defined by d_*

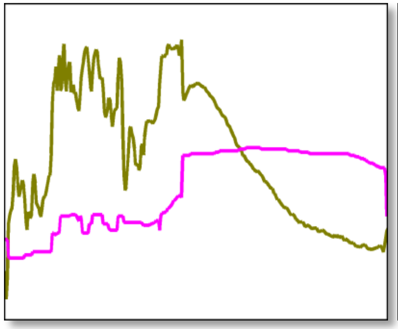
Our Pipeline: Consensus-Based Data Augmentation



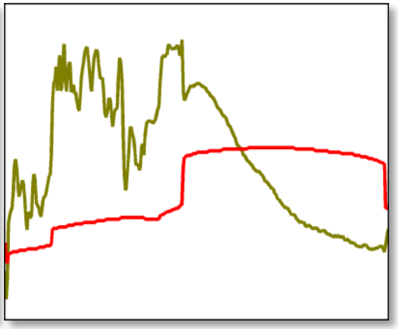
- Then, check the **consistency** between the output and the bicubic upsampled data, and pick the most consistent points to add to the data mask map $\tilde{m}_l$
  - The bibubic upsampled data is free from texture-copying
  - Proceed in a non-overlapping patch fashion (2x2 patches)
- The entire process is repeated until the finest level (l = 0) is reached.

1D Scanline Illustration



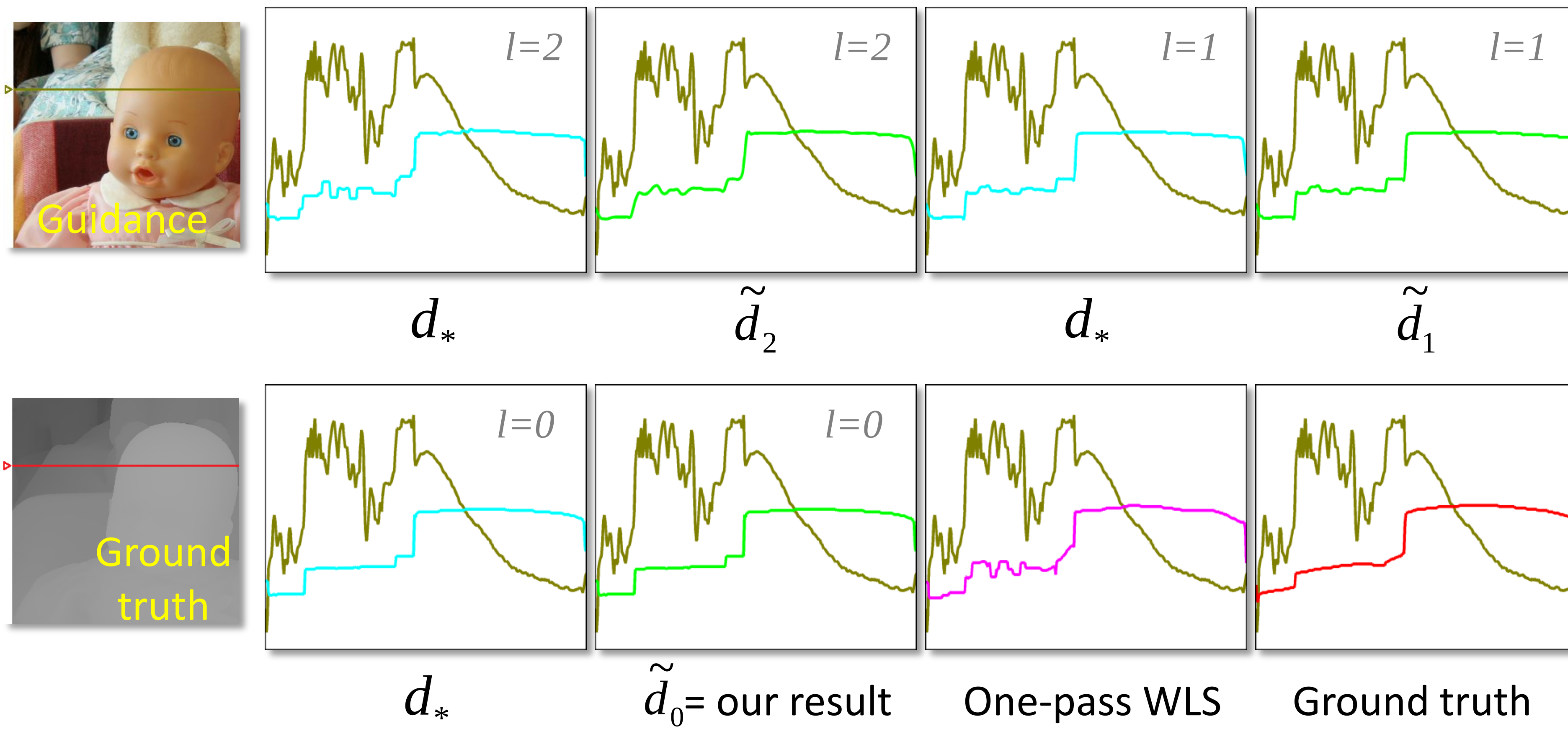


One-pass WLS

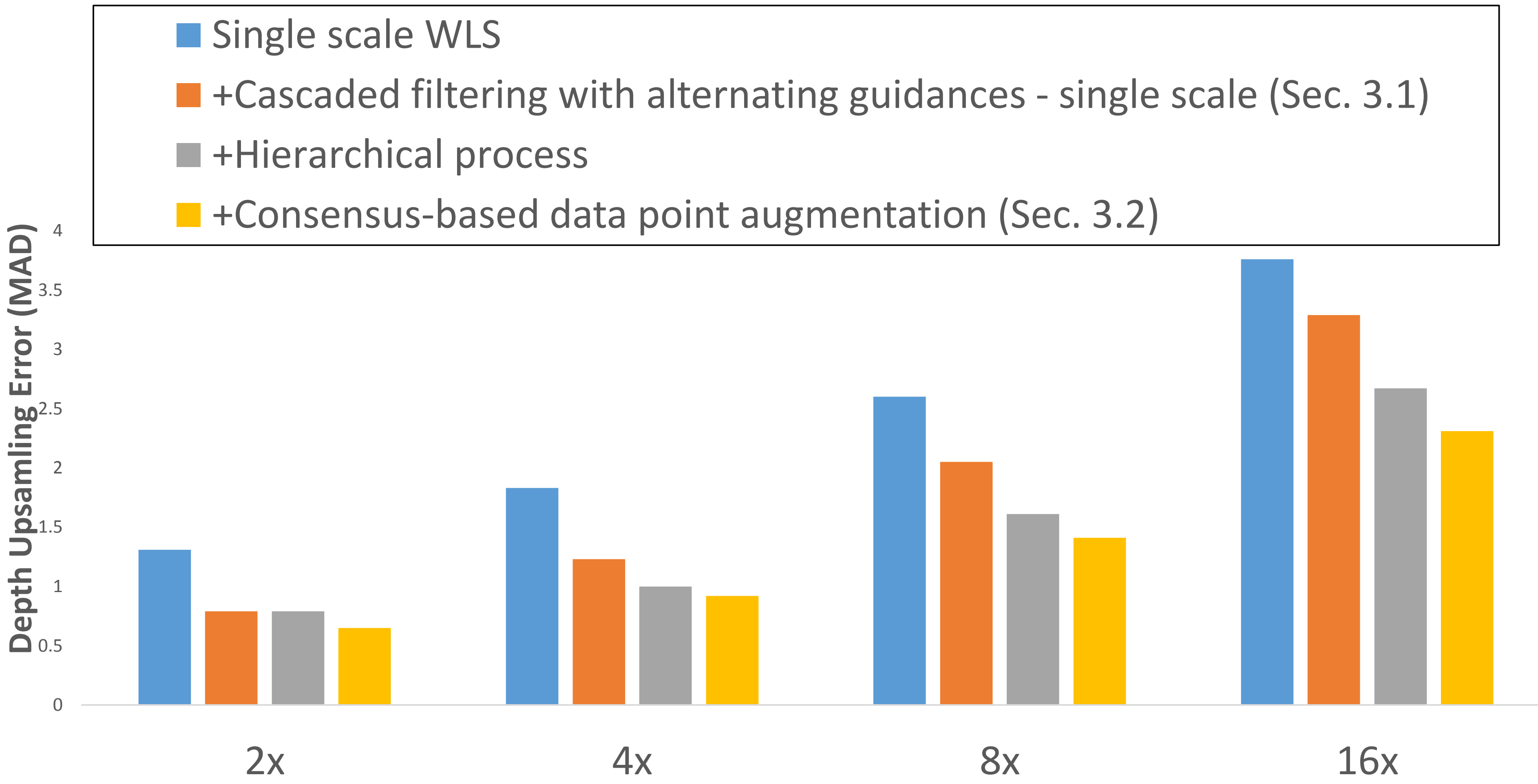


Ground truth

1D Scanline Illustration



Pipeline Validation on Depth Upsampling



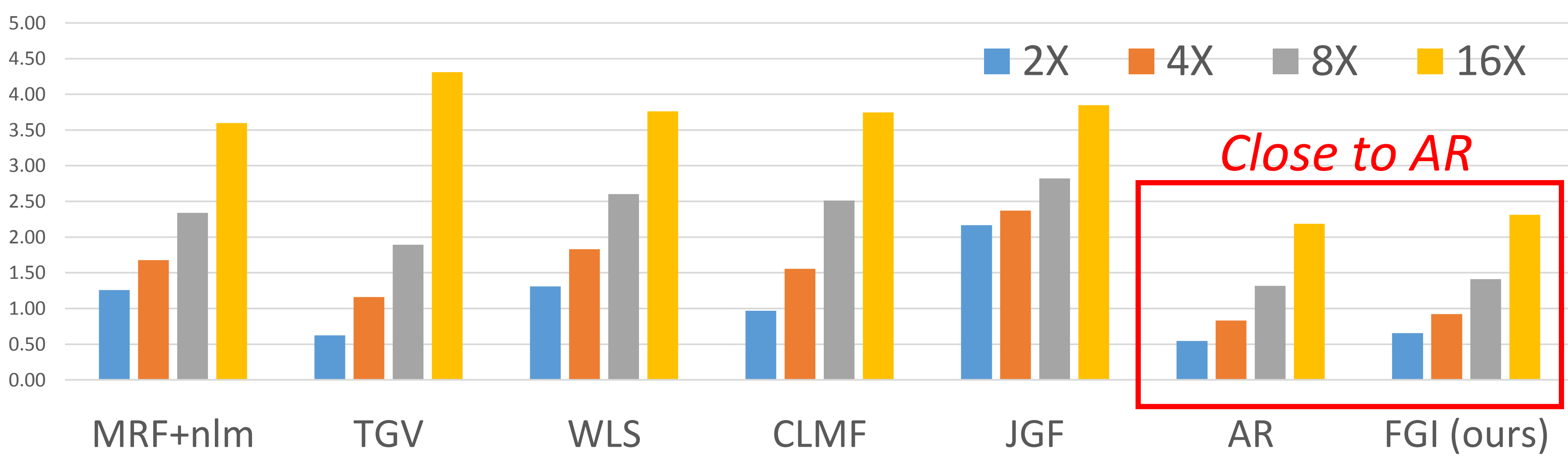
Depth Upsampling Results

Average runtime to upsample a 272 × 344 depth to 1088 × 1376 (in *seconds*)

MRF+nlm	TGV	AR	GF	CLMF	**FGI(ours)**
170	420	900	1.3	2.4	**0.6**

1000x faster than AR

Average Depth Upsampling Error on ToF Synthetic Dataset (6 cases)



Our framework also improves other edge-aware smoothing filters, e.g. the guided filter

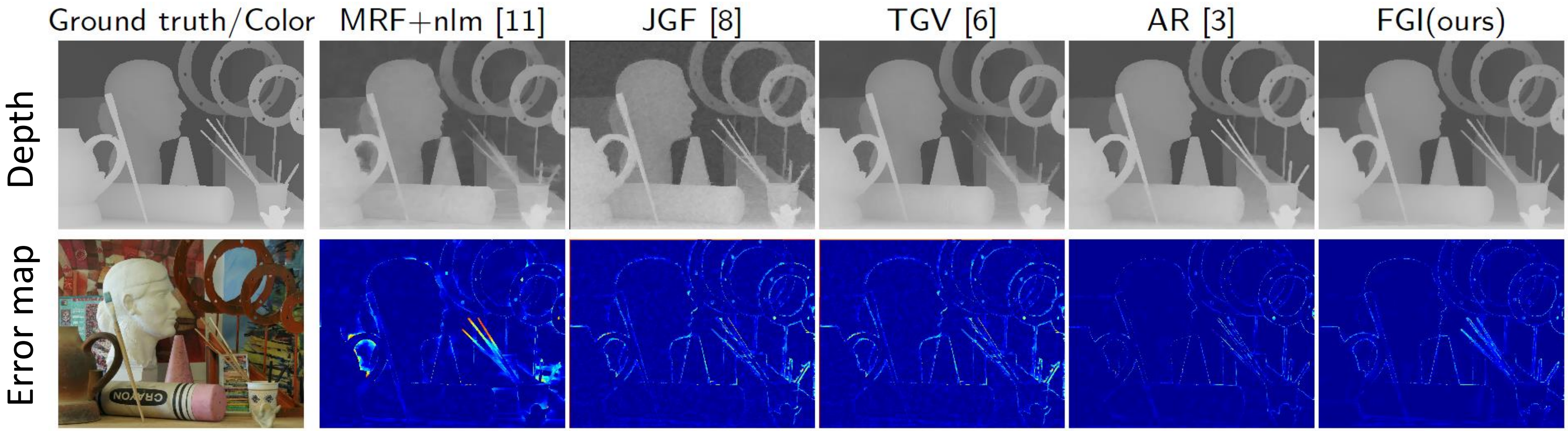
Depth avg. error	2x	4x	8x	16x
Single-pass **GF**	1.31	1.54	2.04	3.12
GF in our framework	**1.06**	**1.21**	**1.63**	**2.59**

*[**GF**: He et al., "Guided image filtering," ECCV 2010.]*

Depth Upsampling Results

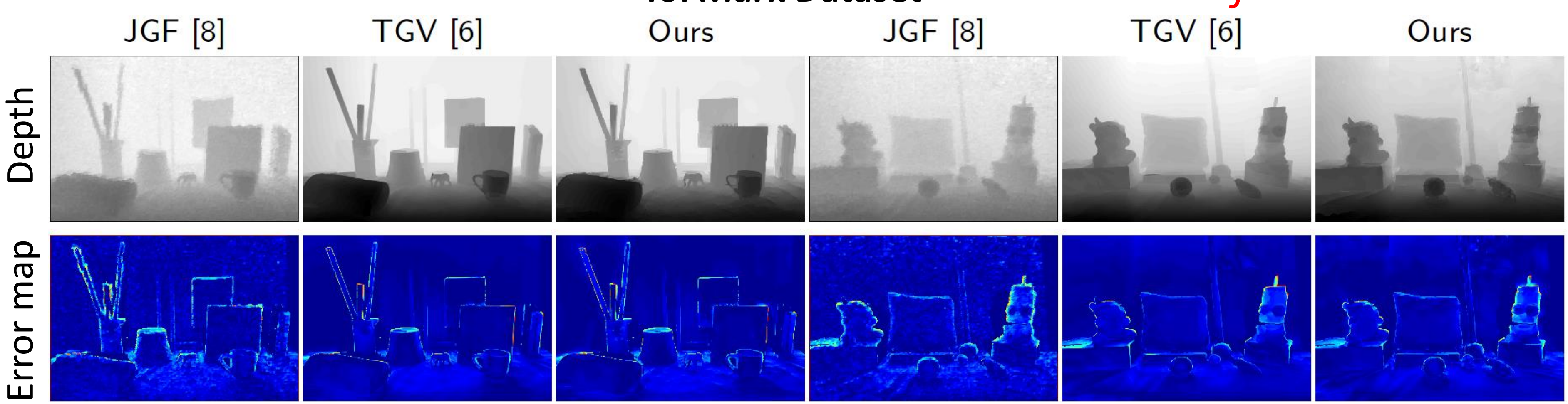
1000x faster than AR

ToF Synthetic Dataset



ToFMark Dataset

650x faster than TGV

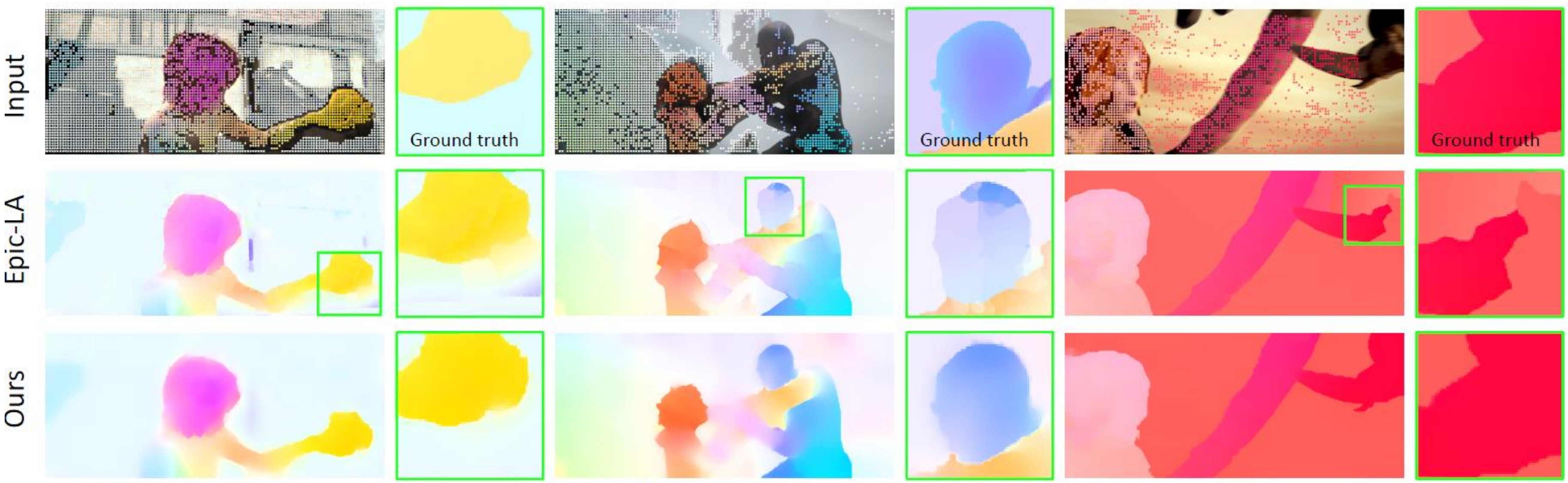


Motion Interpolation Results

Performance (EPE) on MPI Sintel training set

	WLS/FGS	EpicFlow-NW	EpicFlow-LA	FGI(ours)
Clean	3.23	3.17	2.65	2.75
Final	4.68	4.55	4.10	4.14
Runtime (sec)	0.21	0.80	0.94	0.39

Close to EpicFlow, but over 2x faster



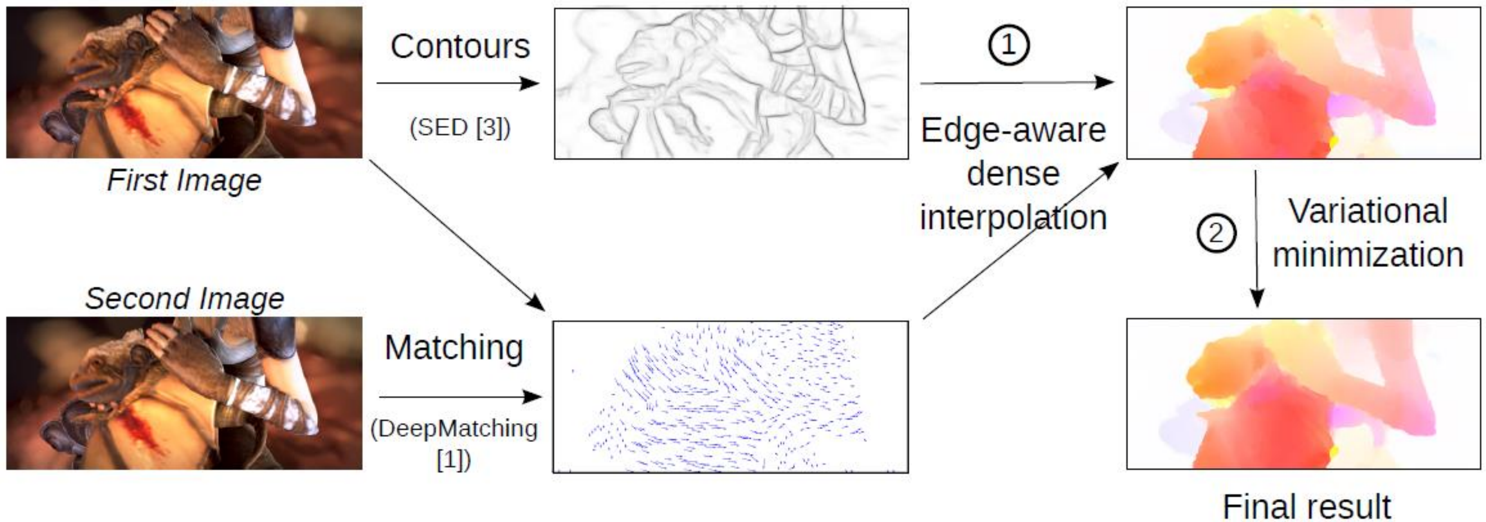
(a) Weak edge in color guidance (b) Sparsely scattered points (c) Extrapolation

Performance (EPE) on the MPI Sintel testing benchmark

	FlowFields[13]	EpicFlow[2]	PH-Flow[37]	**FGI (ours)**	Deep+R[15]	SPM-BP[38]	DeepFlow[14]	PCA-Layers[39]	MDP-Flow2[40]
Clean	3.748	4.115	4.388	4.664	5.041	5.202	5.377	5.730	5.837
Final	5.810	6.285	7.423	6.607	6.769	7.325	7.212	7.886	8.445

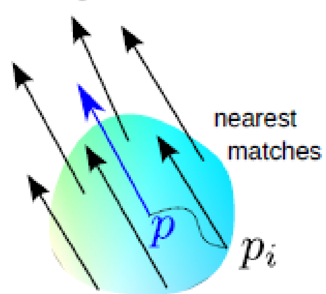
Comparison with [EpicFlow, CVPR’15]

EpicFlow [CVPR’15]



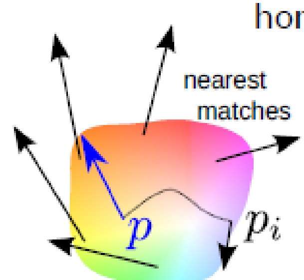
In EpicFlow, interpolation at pixel *p* using either:

- **Nadaraya-Watson**:



simple **average** of nearest matches weighted by $e^{-\text{dist\_geodesic}(p,p_i)}$

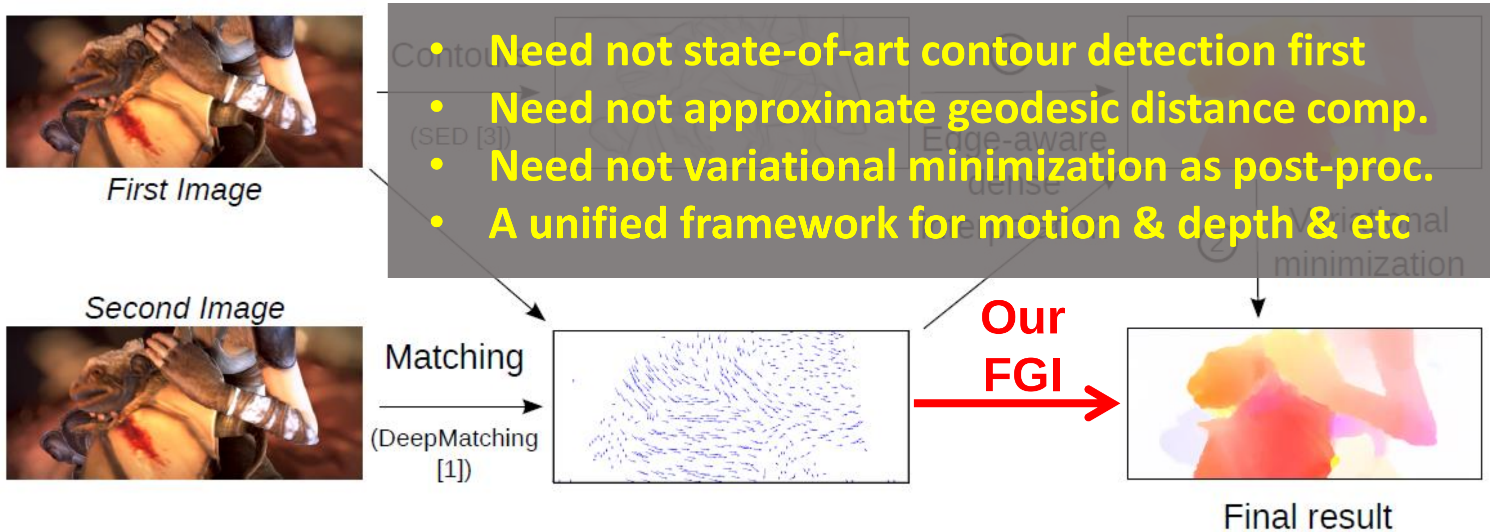
- **Local affine fit**: (best for rotations & homographies)

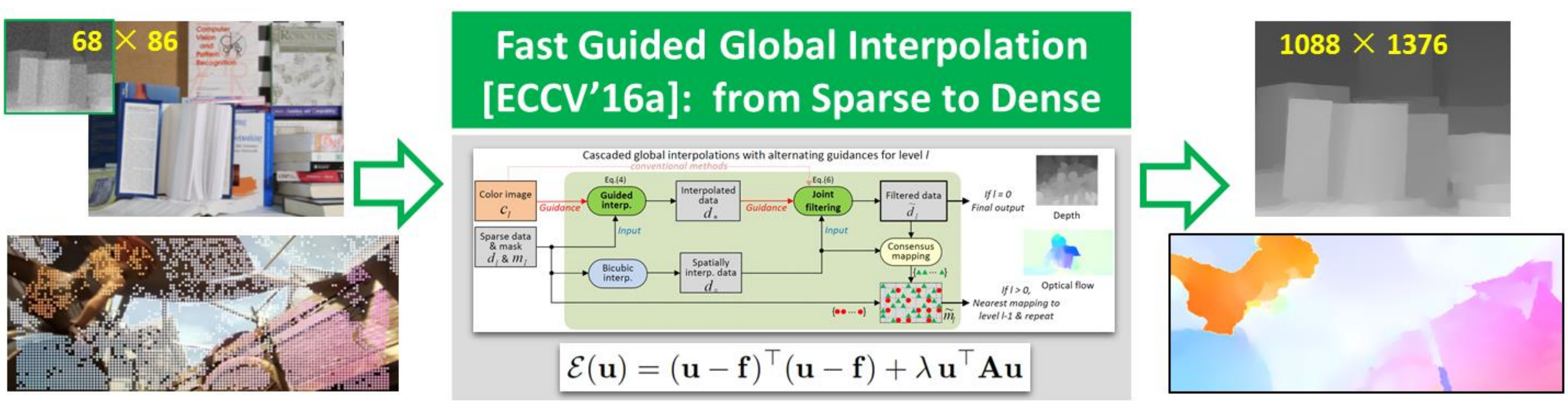


we solve a simple least-square problem (the match weight is the same than at left)

Comparison with [EpicFlow, CVPR'15]

- Need not state-of-art contour detection first
- Need not approximate geodesic distance comp.
- Need not variational minimization as post-proc.
- A unified framework for motion & depth & etc





Comparison with [The Bilateral Solver, ECCV'16]

- FGI does *not* need extra "domain transform to smooth out the blocky artifacts introduced by the simplified bilateral grid"
- Very close depth super-resolution results (below), but FGI also has proven strong performance for *optical flow interpolation*
- FGI (& FGS), using similar recursive computations as *Domain Transform* (@21ms, implemented in Halide) –> *easy to speed up*

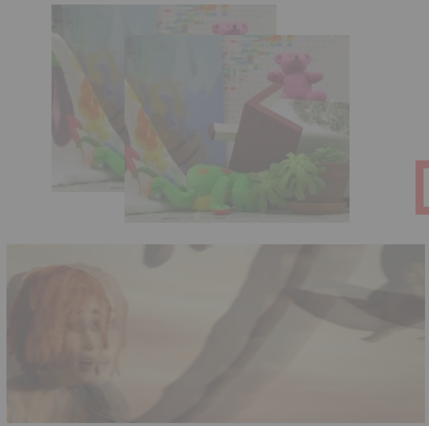
Method	Art				Books				Moebius				Avg.	Time (sec)
A) Nearest Neighbor	6.55	7.41	8.87	11.24	6.16	6.32	6.63	7.36	6.59	6.78	6.98	7.48	7.26	0.003
B) Bicubic	5.32	6.00	7.15	9.35	5.00	5.17	5.46	5.98	5.34	5.52	5.66	6.07	5.91	0.007
C)†Kiechle *et al.*[14]	2.82	5.10	6.83	10.80	3.83	5.10	6.12	8.43	4.50	5.73	6.64	8.96	5.86	450
D) Bilinear	4.57	5.53	6.99	9.45	3.94	4.31	4.71	5.38	4.19	4.55	4.83	5.37	5.16	0.004
E) Liu *et al.* [19]	4.10	5.43	7.69	11.36	3.08	3.87	4.82	6.46	3.18	4.04	5.11	6.62	5.10	16.60
F) Shen *et al.* [27]	3.49	4.62	6.13	8.68	2.86	3.48	4.43	5.57	2.29	3.07	4.22	5.43	4.24	31.48
G) Diebel & Thrun [6]	3.49	4.41	6.24	9.11	2.06	3.00	4.06	5.13	2.13	3.10	4.14	5.12	3.98	–
H) Chan *et al.*[4]	3.44	4.38	5.98	8.41	2.09	2.77	3.78	5.45	2.08	2.69	3.73	5.33	3.83	3.02
I) GuidedFilter[12, 8]	3.55	4.31	5.59	8.22	2.37	2.73	3.42	4.52	2.48	2.82	3.54	4.53	3.76	23.89
J) Min *et al.*[22]	3.65	4.08	5.09	7.91	2.85	2.77	2.97	3.81	3.46	3.25	3.20	3.86	3.74	0.383
K)†Lu & Forsyth[20]	4.30	5.05	6.33	7.94	2.17	2.71	3.30	4.29	2.16	2.50	3.15	4.10	3.69	20
L) Park *et al.*[24]	3.76	4.48	5.80	8.75	1.95	2.60	3.30	4.86	1.96	2.49	3.21	4.48	3.61	24.05
M) Domain Transform [9]	3.95	4.76	6.14	8.49	1.80	2.40	3.23	4.44	1.83	2.40	3.35	4.64	3.56	0.021
N) Ma *et al.* [21]	3.27	3.99	5.08	7.39	2.39	2.70	3.09	3.77	2.55	2.84	3.23	3.81	3.49	18
O) GuidedFilter(Matlab)[12]	3.60	4.25	5.49	7.99	2.39	2.52	2.89	3.89	2.50	2.57	2.90	3.61	3.47	0.434
P) Zhang *et al.* [33]	4.15	4.22	5.03	7.86	1.96	2.24	3.13	4.80	1.80	2.19	3.22	4.90	3.45	1.346
Q) FastGuidedFilter[11]	3.40	4.16	5.46	7.97	2.08	2.51	3.04	3.95	2.13	2.55	3.08	3.79	3.41	0.225
R) Yang 2015 [29]	3.27	4.15	5.46	7.93	2.00	2.38	3.00	4.04	2.25	2.57	3.13	4.00	3.41	0.304
S) Yang *et al.* 2007 [30]	3.01	3.92	4.85	7.57	1.87	2.38	2.86	4.26	1.92	2.41	2.96	4.37	3.25	–
T) Farbman *et al.* [7]	3.14	4.00	5.30	7.70	1.76	2.26	2.90	3.88	1.79	2.29	2.98	3.93	3.19	6.11
U) JBU [1, 15]	3.17	4.02	5.37	7.59	1.83	2.18	2.80	4.00	1.83	2.13	2.71	3.76	3.14	1.98
V) Ferstl *et al.*[8]	3.19	4.06	5.08	7.61	1.52	2.21	2.47	3.54	1.47	2.03	2.58	3.50	2.93	140
W)†Li *et al.*[18]	3.02	3.12	4.43	7.43	1.18	1.70	2.55	3.58	1.11	1.59	2.28	3.50	2.56	700
X)†Kwon *et al.*[16]	0.87	1.30	2.05	3.56	0.51	0.75	1.14	1.88	0.57	0.89	1.37	2.14	1.21	300
Y) BS (Ours)	2.93	3.79	4.95	7.13	1.39	1.84	2.38	3.29	1.38	1.80	2.38	3.23	2.70	0.234
FGI	2.77	3.74	5.12	7.91	1.44	1.86	2.42	3.39	1.52	1.91	2.48	3.28	2.78	

Conclusion

- **General & versatile technique**:
  - Tackle **both** depth and motion interpolation tasks, and potentially more
  - Generally applicable to other edge-aware smoothing filters, e.g. GF
- **Competitive results** while running **much faster** than task-specific state-of-the-art methods
- **Simple & effective**:
  - No *color edge detection* & *variational minimization* in [Revaud et al., CVPR'15]
  - No *domain transform filtering* for post-smoothing in [Barron & Poole, ECCV'16]
- **Further acceleration** on GPUs and FPGA, offering a common engine for guided interpolation

Project page (code is available):

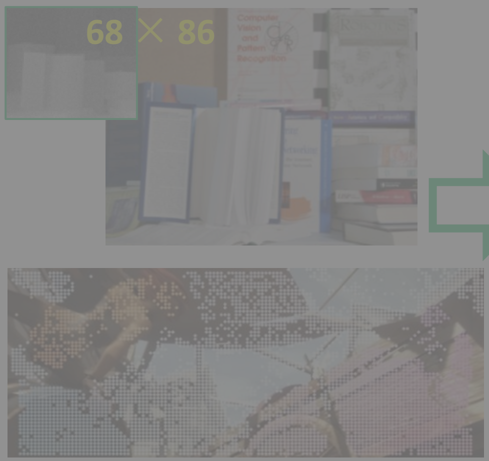
http://publish.illinois.edu/visual-modeling-and-analytics/fast-guided-global-interpolation/



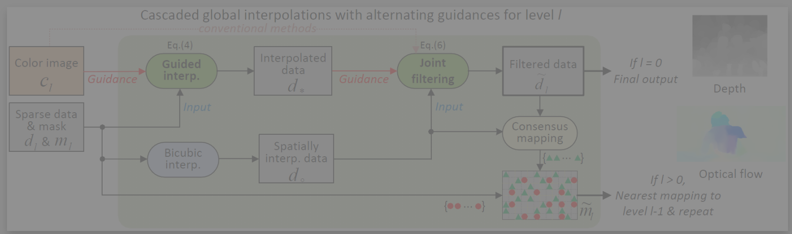
1: SPM-BP [ICCV'15]: Discrete Labeling Optimization for MRFs

$$E = \sum_p E_p(l_p; W) + \sum_p \sum_{q \in \mathcal{N}_p} E_{pq}(l_p, l_q)$$

p: pixel, N_p: 4 neighbors

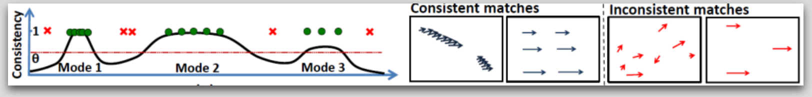


2: Fast Guided Global Interpolation [ECCV'16a]: from Sparse to Dense

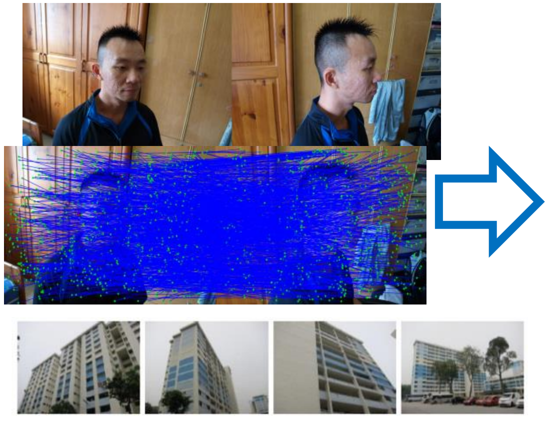


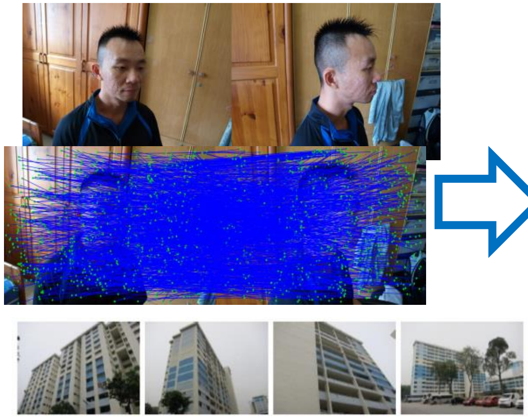
$$\mathcal{E}(\mathbf{u}) = (\mathbf{u} - \mathbf{f})^\top (\mathbf{u} - \mathbf{f}) + \lambda \mathbf{u}^\top \mathbf{A} \mathbf{u}$$

3: Coherence-Based Regression for Feature Matching [ECCV'14, '16b]

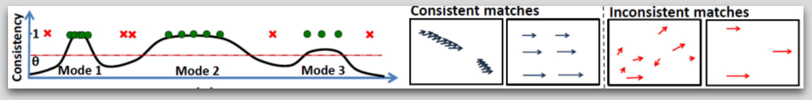


$$\arg\min_{\mathbf{w}} \sum_{i=1}^{N} C(1 - f(\mathbf{m}_i)) + \lambda \mathbf{w}^T G \mathbf{w}$$

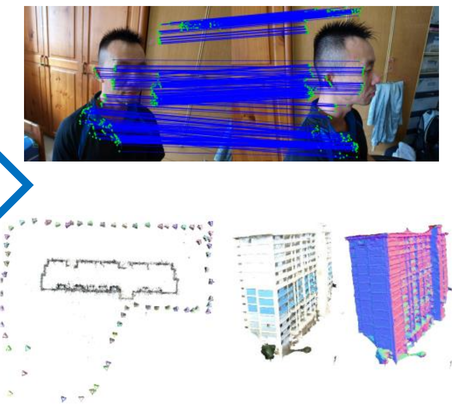




3: Coherence-Based Regression for Feature Matching [ECCV'14, '16b]



$$\arg\min_{\mathbf{w}} \sum_{i=1}^{N} C(1 - f(\mathbf{m}_i)) + \lambda \mathbf{w}^T G \mathbf{w}$$

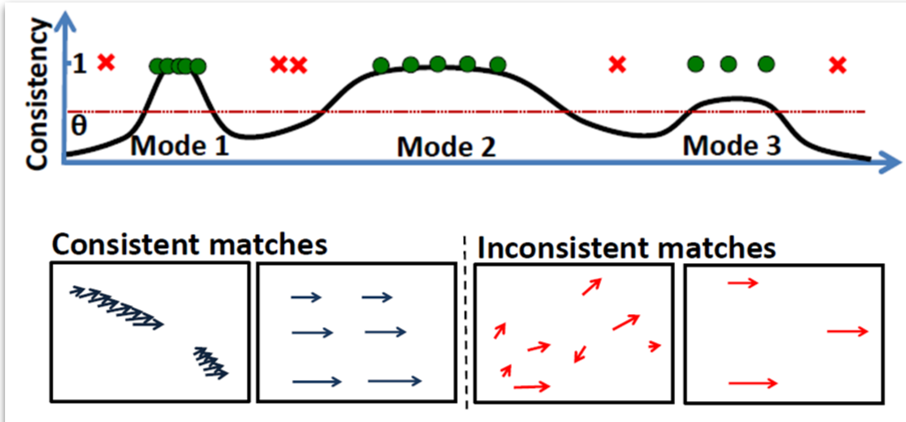


Part 3: Bilateral Motion Coherence Modeling for Robust Feature Matching

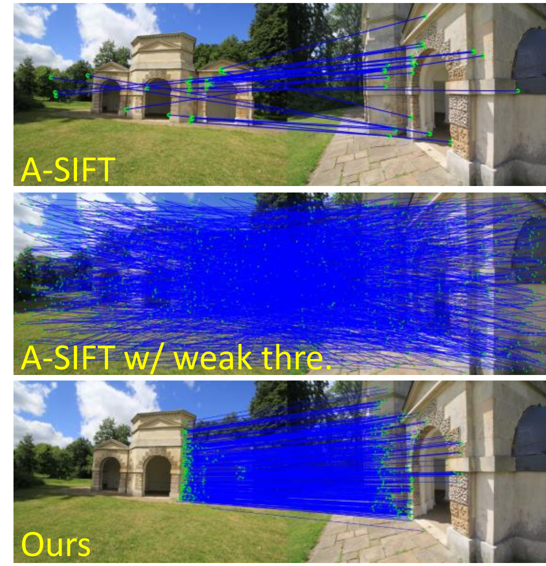
- **W.-Y. Lin, M. Cheng, J. Lu, H. Yang, M. N. Do, and P. H. S. Torr, "Bilateral Functions for Global Motion Modeling," ECCV 2014.**
- W.-Y. Lin, F. Wang, M. Cheng, S.-K. Yeung, P. Torr, M. N. Do, and J. Lu, "CODE: Coherence Based Decision Boundaries for Feature Correspondence," TPAMI (under review). http://www.kind-of-works.com/CODE_matching.html
- W.-Y. Lin, S. Liu, N. Jiang, M. N. Do, P. Tan, and J. Lu, "RepMatch: Robust Feature Matching and Pose for Reconstructing Modern Cities," ECCV 2016.

Overview: Wide-Baseline Matching & 3D Mapping

A reliable **feature matcher** for **pose** and **3D reconstruction**

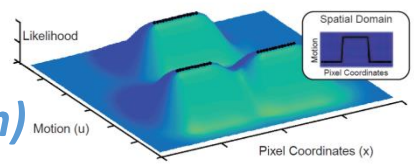


$$\arg\min_{\mathbf{w}} \sum_{i=1}^{N} C(1 - f(\mathbf{m}_i)) + \lambda \mathbf{w}^T G \mathbf{w}$$

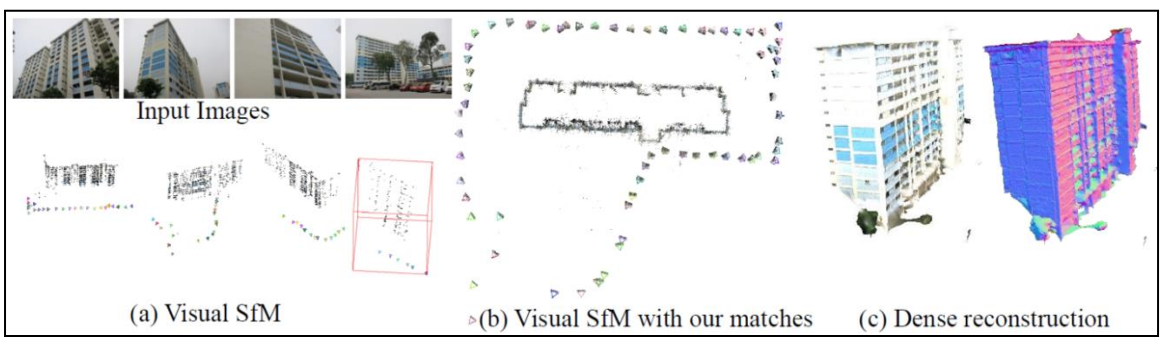


✓ ***Motion coherence modeling***

✓ ***Bilateral domain (spatial + motion)***



- Providing a number of matches, while having almost *no* outliers
- w/ RANSAC to handle *repetitive structures*
- Highly reliable 2-view pose for SfM, mapping



* W.-Y. Lin et al., “RepMatch: Robust Feature Matching and Pose for Reconstructing Modern Cities,” ECCV 2016

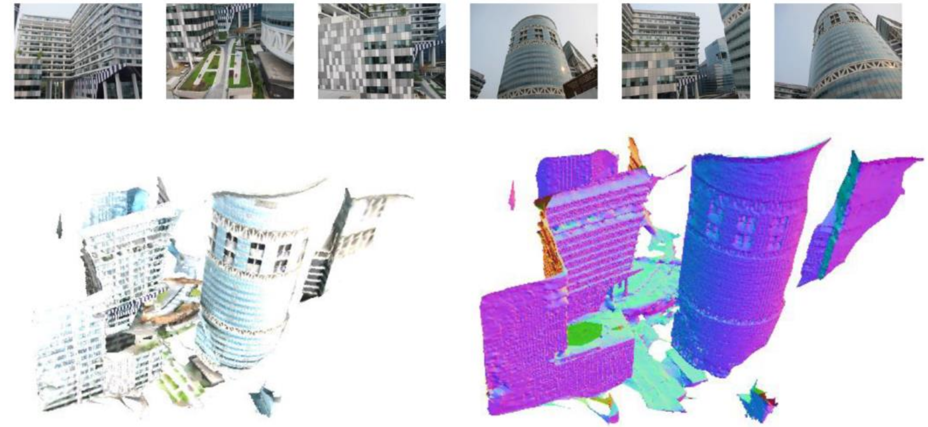
* W.-Y. Lin et al., “Bilateral functions for global motion modeling,” ECCV 2014

A set of multi-view images [42]

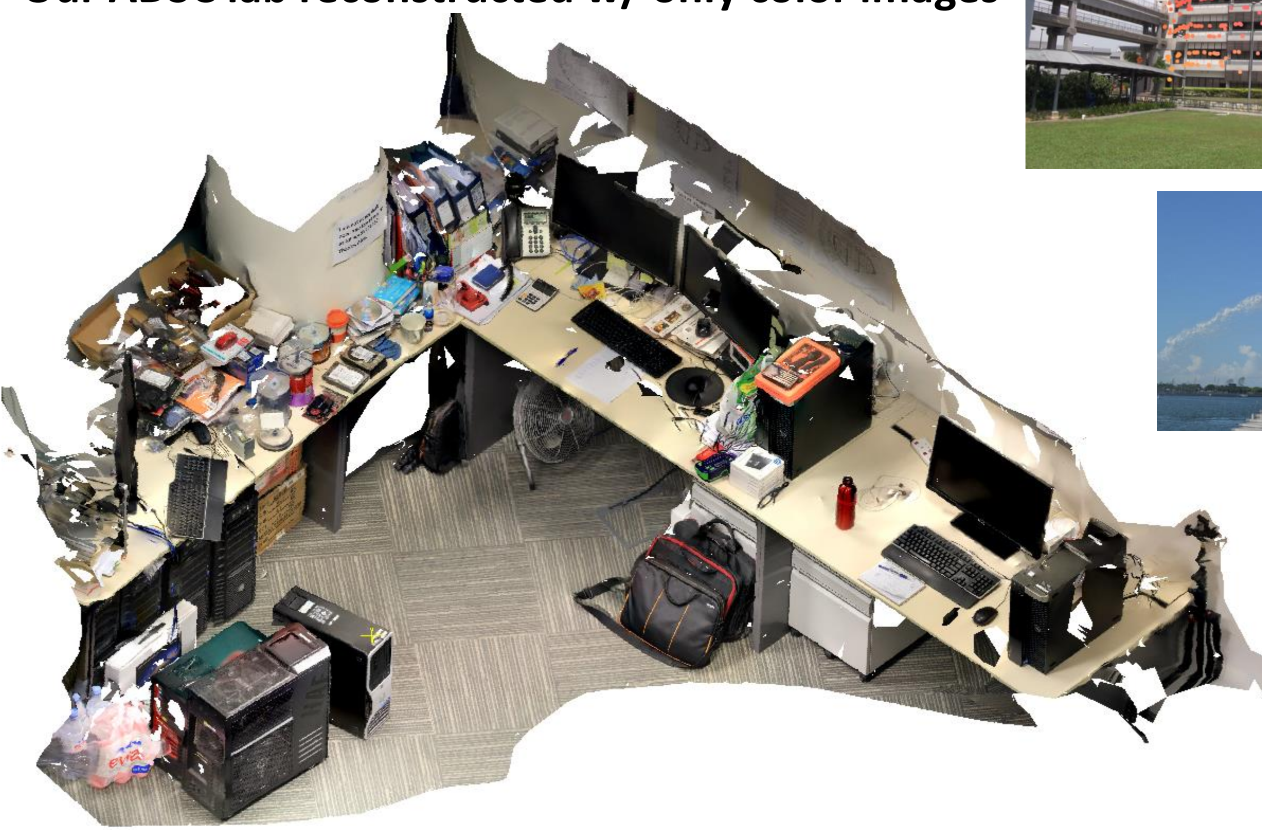
Agisoft [47]: A commercial 3D reconstruction software

Visual SfM [3], [43], [44], [45], [46]

Visual SfM using feature matches returned by *A-SIFT w CODE*



Our ADSC lab reconstructed w/ only color images



- Also useful for : place recognition, localization, multi-frame tracking…

CODE: Coherence-Based Decision Boundaries

Set t = 0.82 to generate many matches (but also a lot of outliers)

Coarse modeling

Matching likelihood modeling

$$E = \sum_{j=1}^{N} huber(1 - f(\mathbf{p}_j)) + \lambda\Psi$$

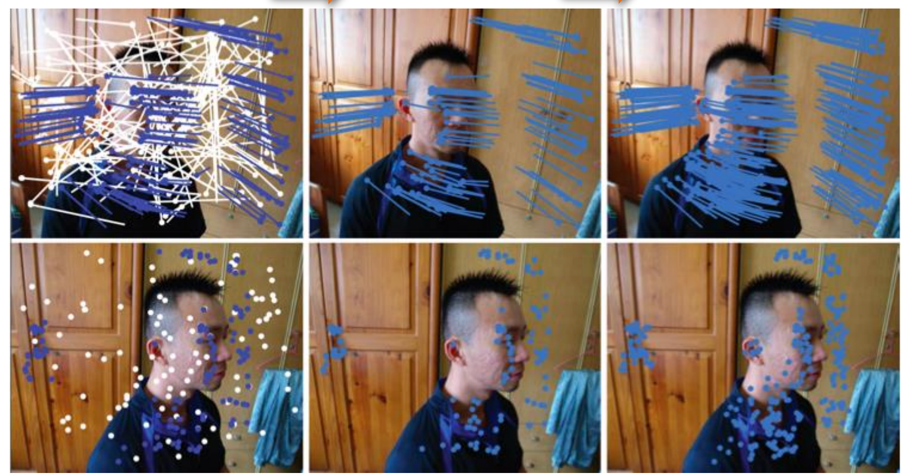
Fine modeling

Affine motion modeling

$$E = \sum_{j=1}^{N} C(\hat{q}_{xj} - q_x(\mathbf{p}_j)) + \lambda \sum_{k=1}^{3} \Psi_k$$

Set t = 1.0 to validate more matching hypotheses

Eliminate outliers → *Expand inliers*



Inlier acceptance

$$f(\mathbf{p}_j) > 0.5$$

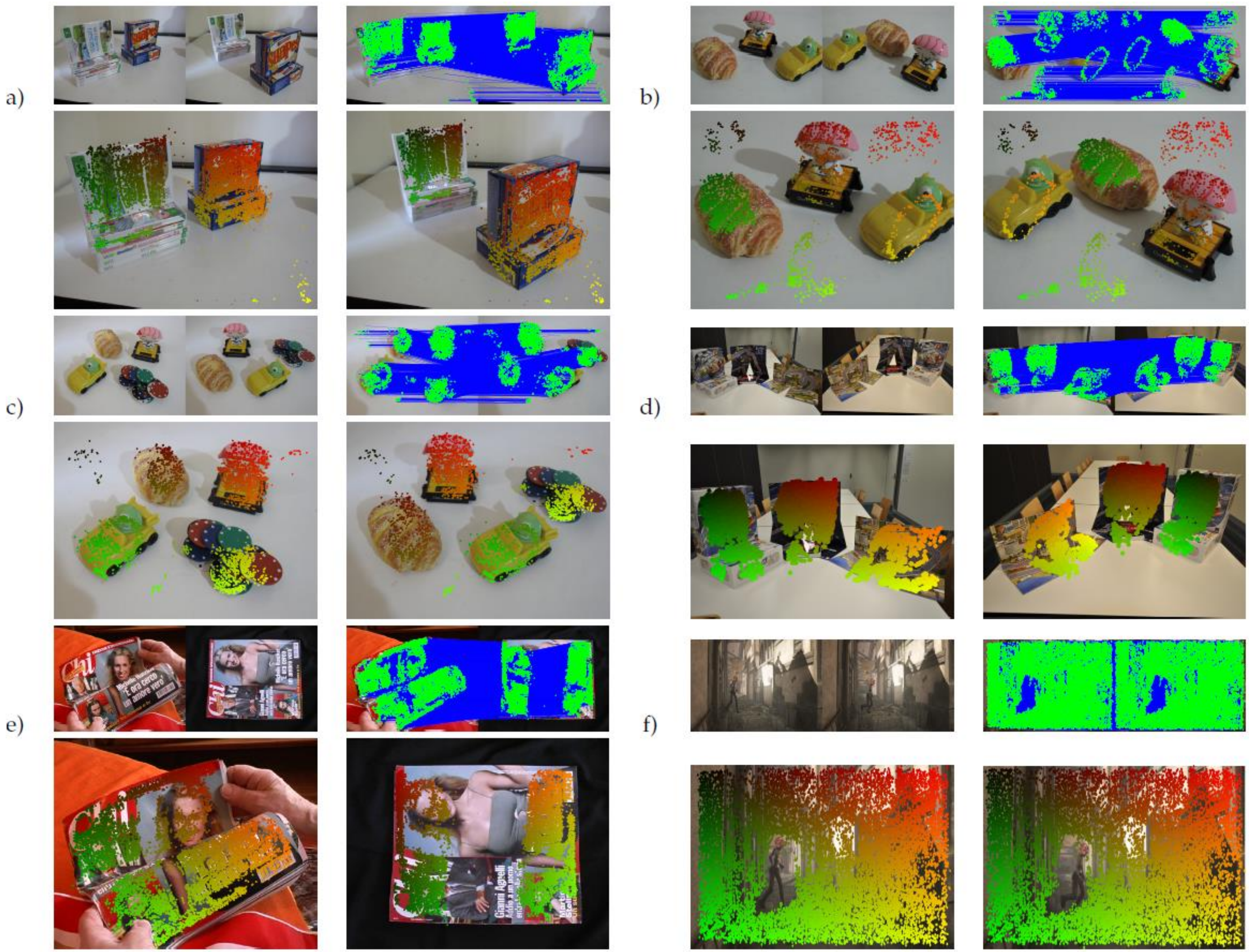
Fine validation

$$(q_x(\mathbf{p}) - (x + u))^2 + (q_y(\mathbf{p}) - (y + v))^2 < 0.01$$

* W.-Y. Lin et al., “CODE: Coherence Based Decision Boundaries for Feature Correspondence,” TPAMI (under review)

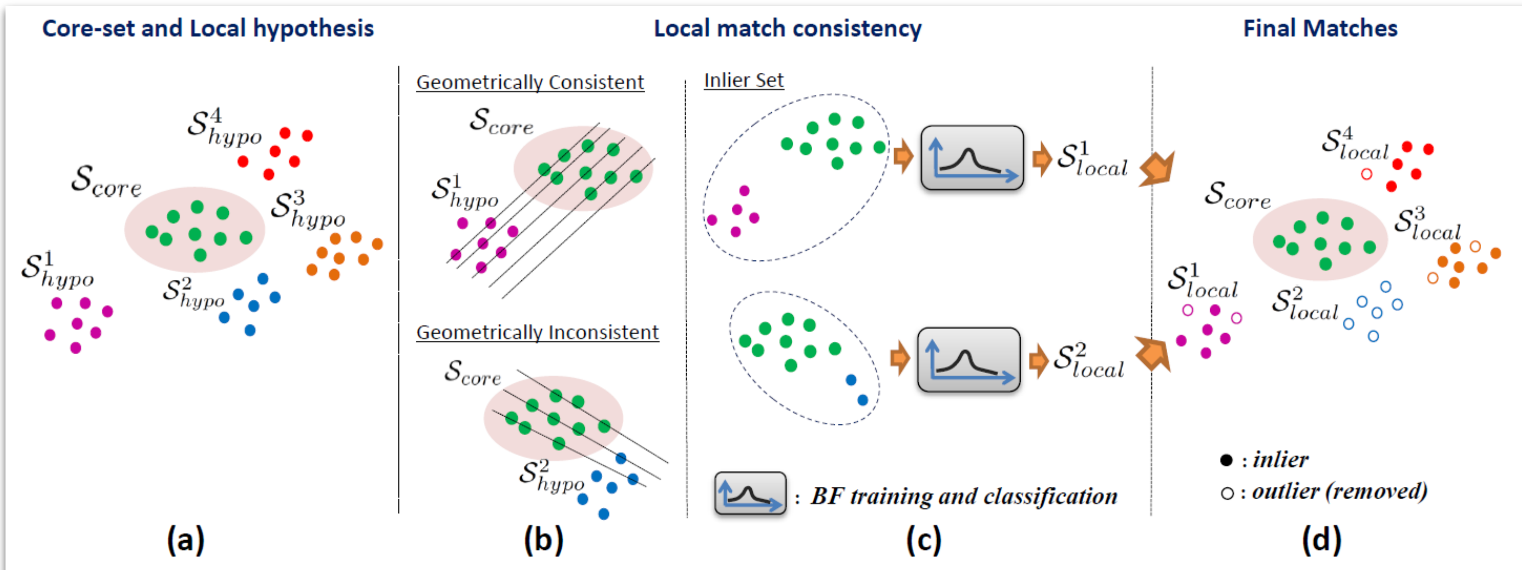
* W.-Y. Lin et al., “Bilateral functions for global motion modeling,” ECCV 2014

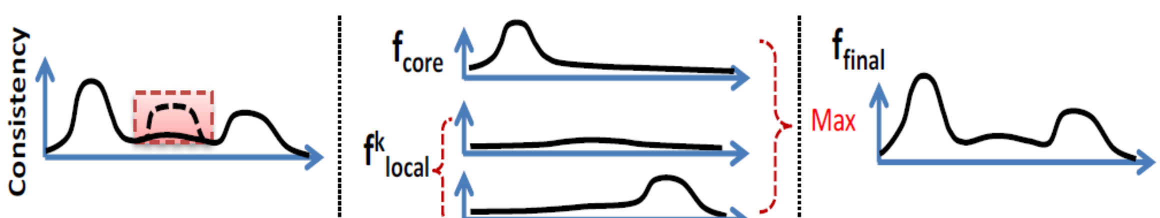
Correspondences on Non-rigid Scenes



RepMatch for Robust Matching & Pose

- Handles *wide baselines* & *repeated structures*
- CODE + Epipolar guided matching (via RANSAC)





Repeated structures induce false consistency modes. Only the core mode is reliable

Core mode with epipolar constraint can compute new local consistency curves.

Max of all consistency curves creates an epipolar consistency curve without the false modes

* W.-Y. Lin et al., "RepMatch: Robust Feature Matching and Pose for Reconstructing Modern Cities," ECCV 2016

Handling Repetitive Structures

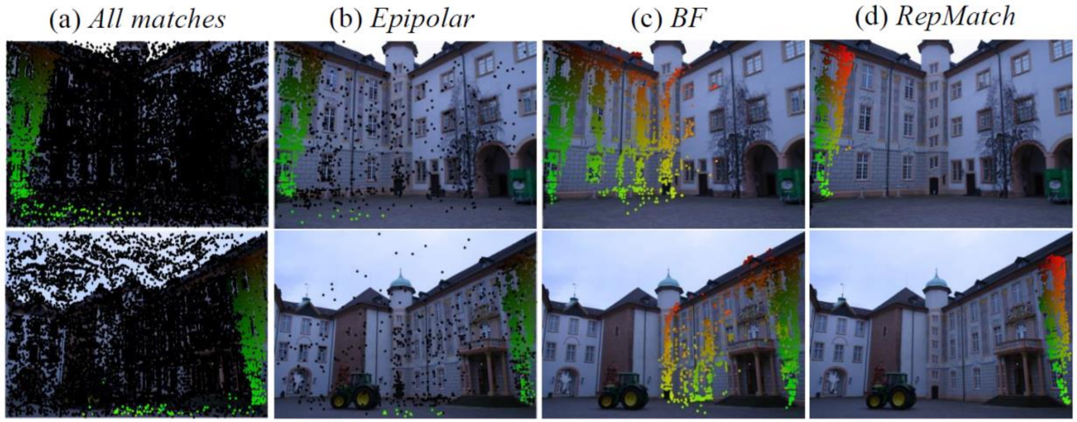
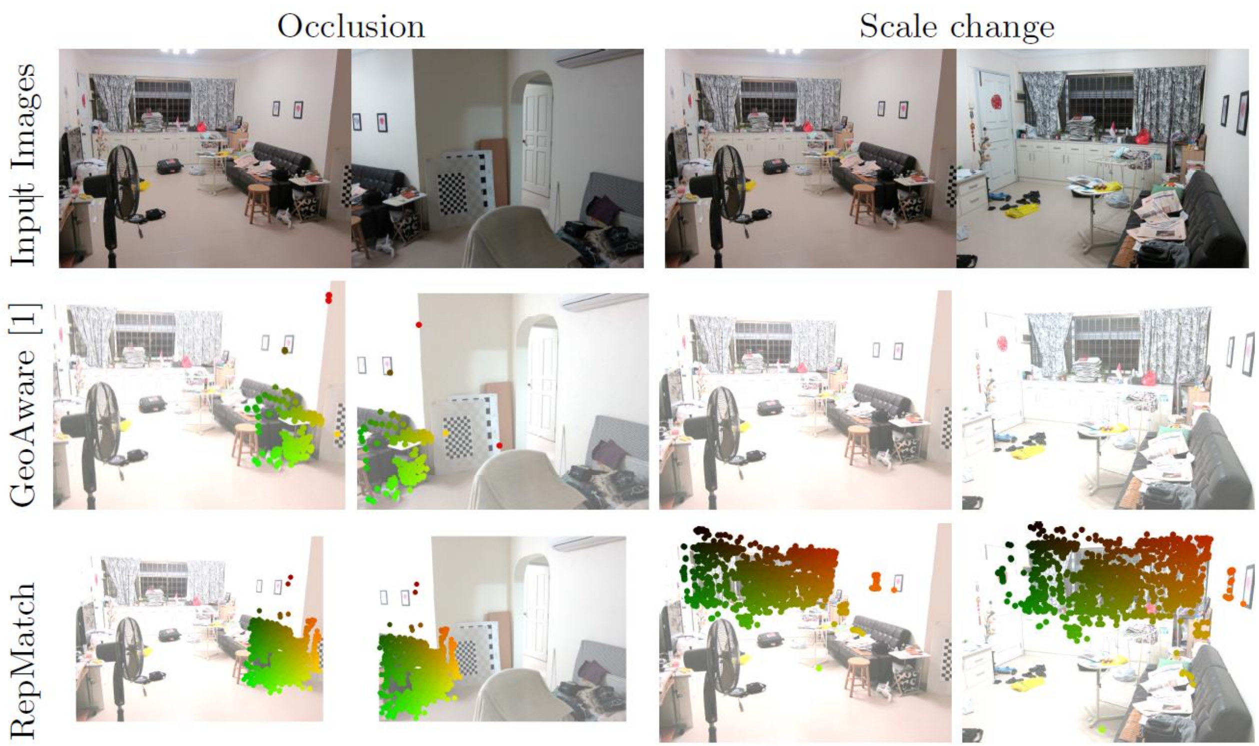


Illustration on real images. Black dots in (a) & (b) indicate wrong matches.
Note: Common central tower belong to physically different parts of the building.

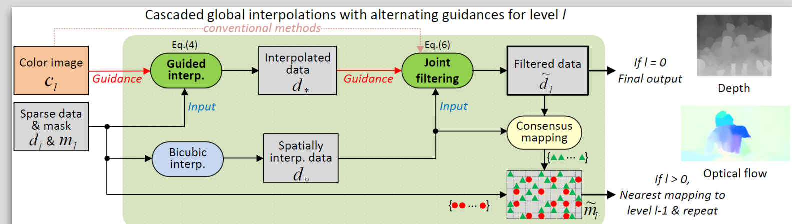


1: SPM-BP [ICCV'15]: Discrete Labeling Optimization for MRFs

$$E = \sum_{p} E_p(l_p; W) + \sum_{p} \sum_{q \in \mathcal{N}_p} E_{pq}(l_p, l_q)$$

p: pixel, N_p: 4 neighbors

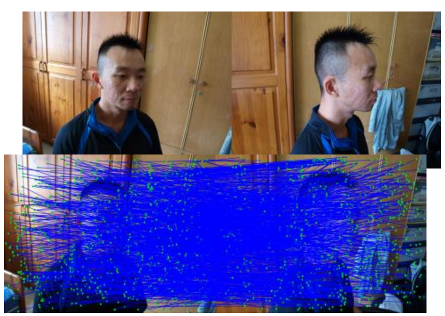
2: Fast Guided Global Interpolation [ECCV'16a]: from Sparse to Dense



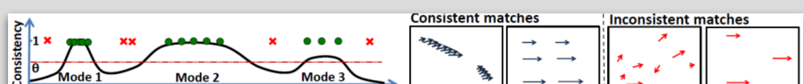
$$\mathcal{E}(\mathbf{u}) = (\mathbf{u} - \mathbf{f})^\top (\mathbf{u} - \mathbf{f}) + \lambda\, \mathbf{u}^\top \mathbf{A} \mathbf{u}$$

What if

- ***SPM-BP + FGI → ?***
- ***CODE + FGI → ?***
- ***SPM-BP & CODE → ?***



3: Coherence-Based Regression for Feature Matching [ECCV'14, '16b]



$$\arg\min_{\mathbf{w}} \sum_{i=1}^{N} C(1 - f(\mathbf{m}_i)) + \lambda \mathbf{w}^T G \mathbf{w}$$

Brief Intro

[https://publish.illinois.edu/
visual-modeling-and-analytics/](https://publish.illinois.edu/visual-modeling-and-analytics/)

Holistic (Computer/Robotic) Vision

Geometric Reconstruction meets Semantic Recognition for 3D holistic vision: Real-time, robust, geometry-centric vision

Where: geometry, location…

What: semantics, action…

Reconstruction

- Real-time camera pose localization
- 3D environment mapping
- Depth and motion estimation
- Large-scale urban reconstruction

Recognition

- Visual place/scene recognition
- Object recognition, localization
- Human re-identification
- Action recognition, tracking

How: model, solve, compute

- Multiple sensors
- Modern vehicles
- Moving robots
- Opportunistic scan

Key techniques

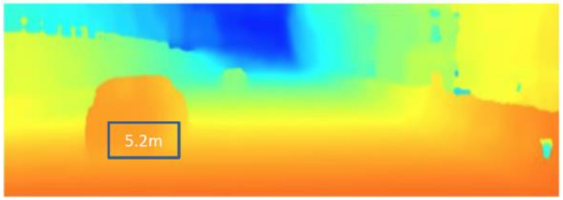
- Geometry-aware filtering
- Fast randomized algorithms
- Efficient inference models
- Deep learning innovation

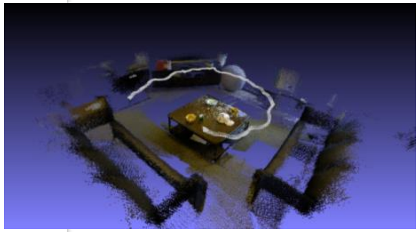
- Mobile cameras
- Big visual data
- Rich annotations
- Powerful machine

Key Themes and Areas for a Smart Nation

Objectives:

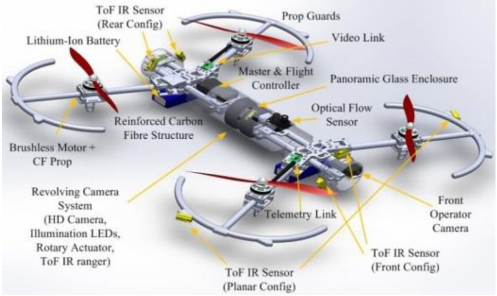
- ❑ Make sense of the *biggest* big data – video
- ❑ Address urban challenges by focusing on *visual modeling and analytics*
- ❑ Develop cutting-edge technologies and solutions for a *Smart Nation*





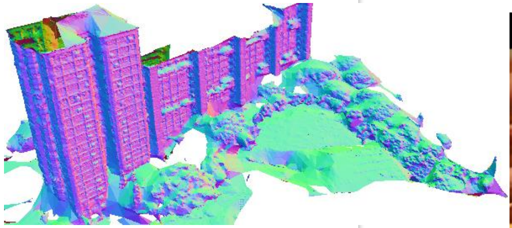
Key themes:

Real-time, robust localization and geometry-centric computer vision

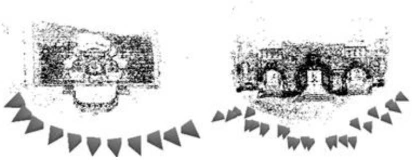


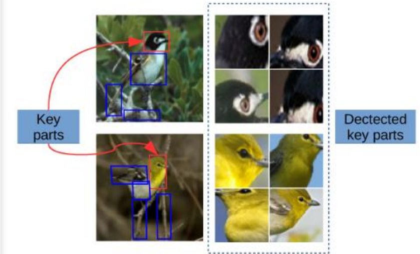
Key technologies:

- ❑ Localization for robots and AVs
- ❑ 3D environment mapping
- ❑ Visual analytics and understanding
- ❑ Computational imaging and augmented reality



Some example results and collaborations of our research





Scene structure

- Depth enhance.
- Slanted stereo
- Optical flow
- Live FG segment
- 3D city recons.

Comp. imaging

- Photo refocus
- Rain removal
- Image stitching
- Multi-scale dec.
- Structure extra.
- Image warping
- Colorization

3D vision & percept.

CODE [ECCV'14, '16]
Feature matching

DSE [CVPR'15]
Camera pose rec.

SLAM-O [TCSVT'16]
3D object prop.

Recogn. & segment.

Fine-grained recog. [TIP'16]

PISA [CVPR'13b]
Pixel-acc. saliency

MHIC [TMM'16]
Recog & Co-seg.

Edge-aware filtering

CLMF [CVPR'12]
Local EAF

FGS [TIP'14]
Global EAF

FGI [ECCV'16]
Sparse to dense

Efficient inference

PMF [CVPR'13]
Local optim.

DFF [CVPR'14]
Generalized

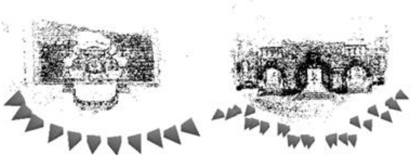
SPM-BP [ICCV'15]
Global optim.

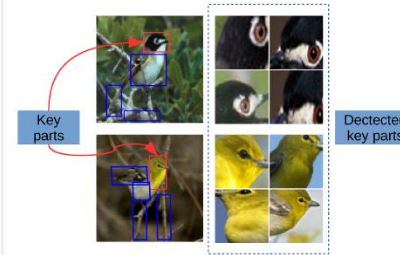
Current topics

- *Scene flow*
- *Robust SLAM*
- *Motion seg.*

- *Text detection*
- *Place recogn.*
- *Scene labeling*
- *Action recog.*

- *Deep learning*





Scene structure

- Depth enhance.
- Slanted stereo
- Optical flow
- Live FG segment
- 3D city recons.

Comp. imaging

- Photo refocus
- Rain removal
- Image stitching
- Multi-scale dec.
- Structure extra.
- Image warping
- Colorization

3D vision & percept.

CODE [ECCV'14, '16]
Feature matching

DSE [CVPR'15]
Camera pose rec.

SLAM-O [TCSVT'16]
3D object prop.

Recogn. & segment.

Fine-grained recog. [TIP'16]

PISA [CVPR'13b]
Pixel-acc. saliency

MHIC [TMM'16]
Recog & Co-seg.

Edge-aware filtering

CLMF [CVPR'12]
Local EAF

FGS [TIP'14]
Global EAF

FGI [ECCV'16]
Sparse to dense

Efficient inference

PMF [CVPR'13]
Local optim.

DFF [CVPR'14]
Generalized

SPM-BP [ICCV'15]
Global optim.

Current topics

- *Scene flow*
- *Robust SLAM*
- *Motion seg.*

- *Text detection*
- *Place recogn.*
- *Scene labeling*
- *Action recog.*

- *Deep learning*

https://publish.illinois.edu/visual-modeling-and-analytics/research/

ADSC
Illinois at Singapore Pte Ltd

ILLINOIS

Visual Modeling and Analytics of Dynamic Environments for the Mass

HOME | NEWS | RESEARCH | PUBLICATIONS | DEMOS | PEOPLE | CONTACT US

Research

This project will focus on addressing the following fundamental challenges key to a multitude of visual data analytics applications: 1) raw visual data cleaning; 2) visual data registration and fusion; and 3) visual data analytics and management. The existing efforts, either from the academia or the industry, are not capable of robustly and efficiently modeling, analyzing, and fusing continuous or discrete visual data captured by individuals or big companies.

Grounded on the recent novel and exciting developments described in this project, we plan to extend, generalize, and optimize them to address the aforementioned key challenges of visual modeling and analytics for the masses using the following research directions:

1. Localization – recovering the geometric locations of the user, the camera viewpoint, or the objects in the environment around the user;
2. Registration – aligning and modeling dynamically captured images and measurements of the scene over different time and viewpoints together;
3. Inference – estimating and analyzing the semantic information of the scene from the registered visual information and recovered geometric information.

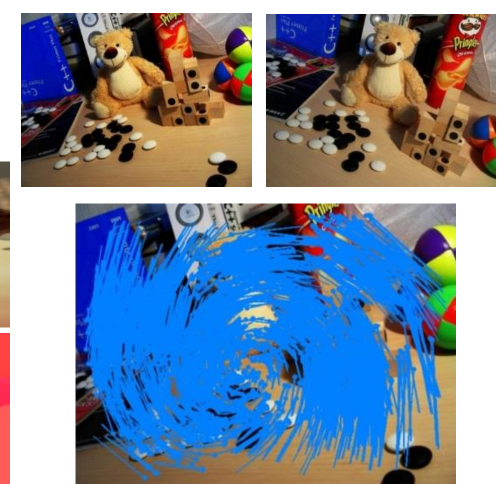
We aim at achieving both high robustness and accuracy for the above tasks at unprecedented processing speeds on commodity computing devices and mobile cameras, often producing more than one or two orders of magnitude of speedup over the existing state-of-the-art solutions.

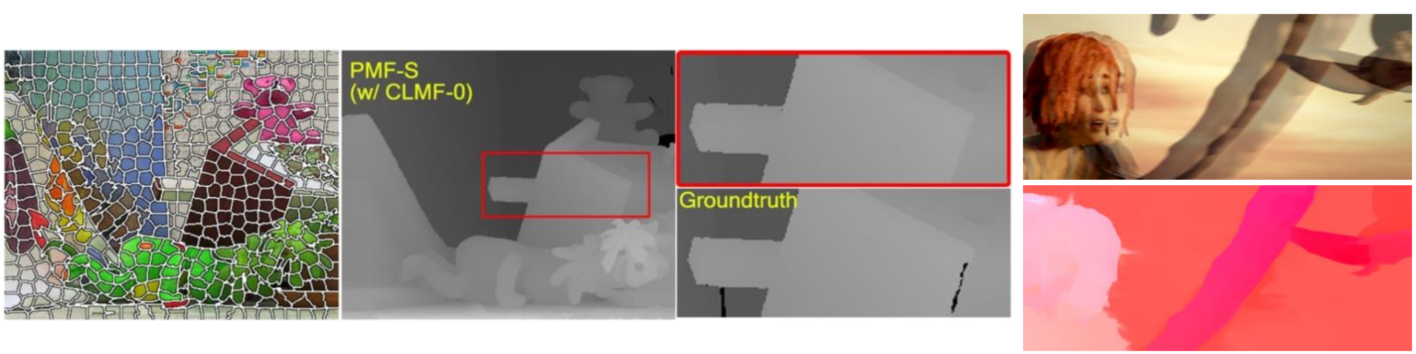
We have been working on the following clusters of research topics, and now are actively innovating in a broader scope.

- Edge-aware filtering and joint filtering
- Dense stereo, optical flow and view synthesis
- Dense correspondences across scenes
- Motion coherence and wide-baseline matching
- Structure from motion, 3D reconstruction
- Computational photography, image enhancement
- Efficient inference for continuous MRFs
- Fast guided global interpolation
- Saliency, recognition, cosegmentation
- Autonomous systems, robot vision
- Hash techniques

- ICIP’13 tutorial: ***Image Filtering 2.0: Efficient Edge-Aware Filtering and Their Applications***
  https://sites.google.com/site/filteringtutorial/

- ICME’15 tutorial: ***Visual Correspondences: Taxonomy, Modern Approaches and Ubiquitous Applications***
  https://sites.google.com/site/icme15tutorial/





Acknowledgement

- A*STAR HCCS and HSSP funding
- Main collaborators
  - UIUC: Minh N. Do
  - CNU: Dongbo Min
  - NTU: Jianfei Cai, Jianmin Zheng
  - NUS/York: Michael Brown
- Group members (current & former staff, visiting PhD students)
  - Yu Li
  - Hongsheng Yang
  - Hongyuan Zhu
  - Yu Zhang
  - Viet Anh Nguyen
  - Nianjuan Jiang
  - Wenyan, Daniel, Lin
  - Hongwei Ng
  - Johan Vu
  - Weiyong Eng
  - Raman S. Pahwa
  - Benjamin Chidester
  - ……
  - ……

- Since 01/2017, → **Shenzhen**
- Focus on **3D + AI for humans**
- We’re hiring **core researchers**
  - **Computer vision**
  - **Computer graphics**
  - **Machine learning**
  - **Image processing**

- jiangbo.lu AT gmail.com
  jiangbo AT cloudream.com

Thanks

通过人和商品数据化
打造一站式个人形象管理平台