Dense correspondence recovery involving images and 3D models

Sudipta N. Sinha

Microsoft Research

University of Utah,
March 27, 2018

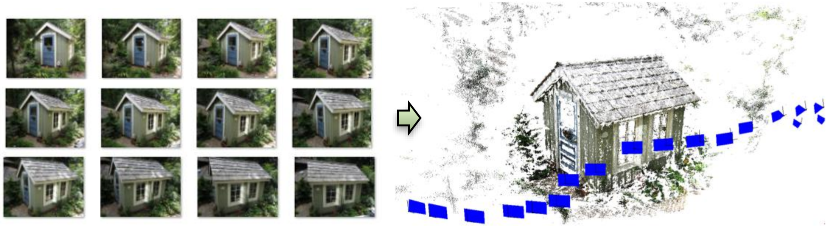
Introduction

- Estimate correspondences and align multiple entities
  - Image to image (stereo, optical flow, scene flow …)
  - Image to 3D model (object recognition, pose estimation …)
- Applications:
  - Vision: image stitching, structure from motion, visual odometry, SLAM, camera localization, 3D mapping, 4D reconstruction, …
  - Augmented Reality: object recognition, 6D pose recovery, tracking
  - Robotics: localization, avoiding obstacles, object grasping & moving

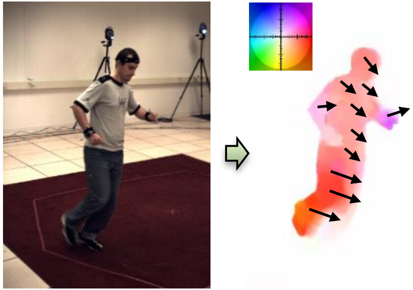
Dense Image Correspondence

Binocular stereo

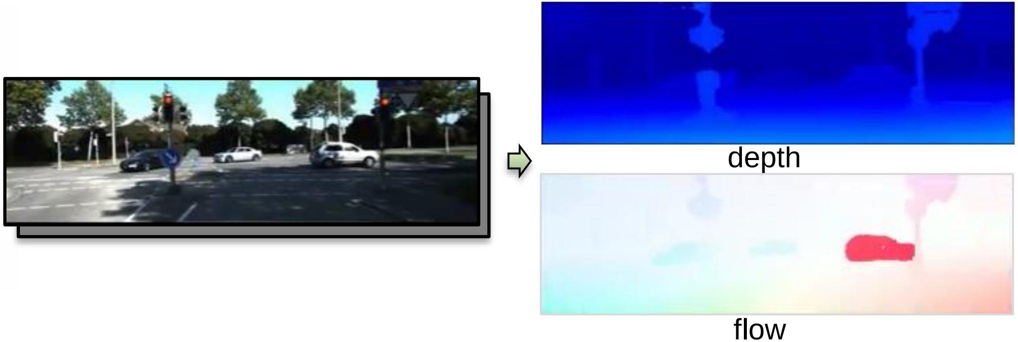
Multi-view stereo



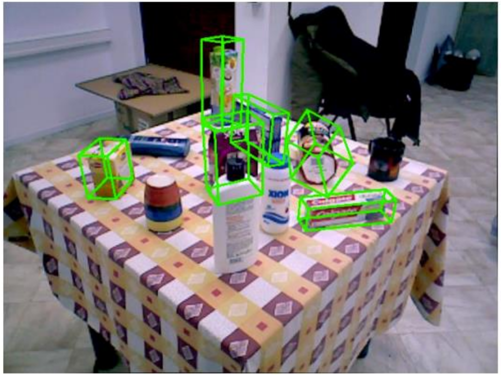
Optical flow



Scene Flow



Dense Image 3D Model Correspondence




- Task: recognize object instances in an image, find pose of associated 3D models; project 3D model to get dense alignment.
- Need training data (images, models, annotation); real images vs. CG
- Challenges: scene clutter, low texture, difficult lighting, low resolution

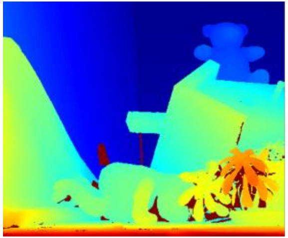
Outline

- Global Stereo Matching with piecewise-planar priors
- Semi-global Matching (SGM)
  - Local plane-sweep stereo
  - SGM with surface orientation priors
- Stereoscopic Scene Flow
- Deep Single-shot 6D Object Instance Detection

Stereo Matching

Left

Right



Left Disparity Map

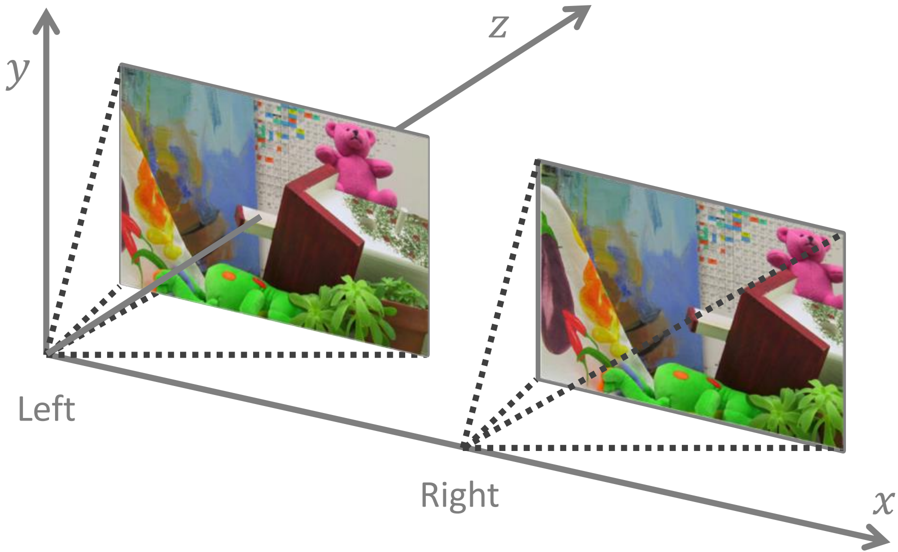
- Dense pixel correspondence in rectified pairs
- Disparity Map: $D(x, y)$

$$x' = x + D(x, y), \quad y' = y$$

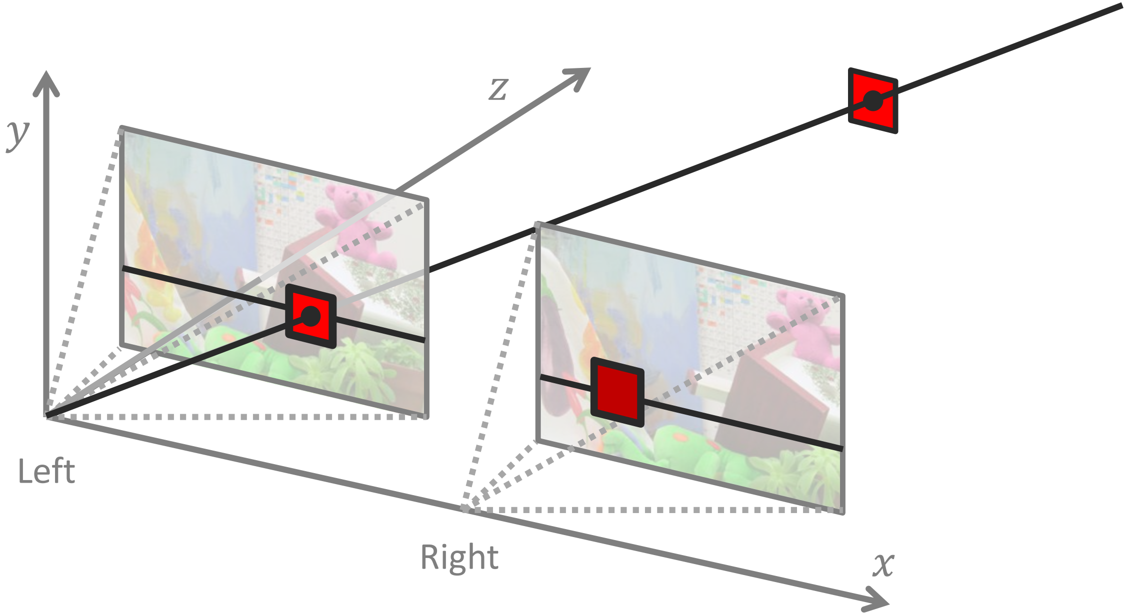
- Depth Map: $Z(x, y) = \dfrac{bf}{D(x,y)}$

Depth Map

Stereo Matching



Binocular Stereo Matching



Local Optimization

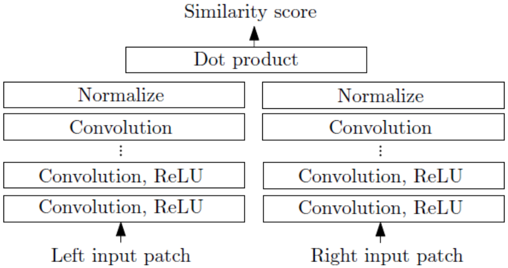
- Minimize matching cost at each pixel <u>independently</u>
- Winner-take-all (WTA)

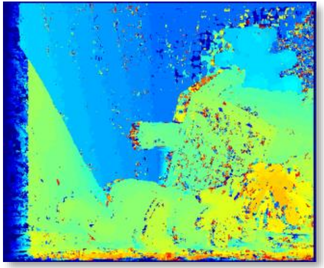
$$C_{SAD}(\mathbf{p},\mathbf{d}) = \sum_{\mathbf{q}\in N_\mathbf{p}} |I_L(\mathbf{q}) - I_R(\mathbf{q}-\mathbf{d})|$$

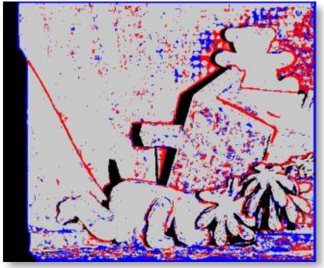
$$C_{ZNCC}(\mathbf{p},\mathbf{d}) = \frac{\sum_{\mathbf{q}\in N_\mathbf{p}}(I_L(\mathbf{q}) - \bar{I}_L(\mathbf{p}))(I_R(\mathbf{q}-\mathbf{d}) - \bar{I}_R(\mathbf{p}-\mathbf{d}))}{\sqrt{\sum_{\mathbf{q}\in N_\mathbf{p}}(I_L(\mathbf{q}) - \bar{I}_L(\mathbf{p}))^2 \sum_{\mathbf{q}\in N_\mathbf{p}}(I_R(\mathbf{q}-\mathbf{d}) - \bar{I}_R(\mathbf{p}-\mathbf{d}))^2}}$$

- Convolutional Neural Nets

  [Zbontar and Lecun 2015]







Stereo benchmarks

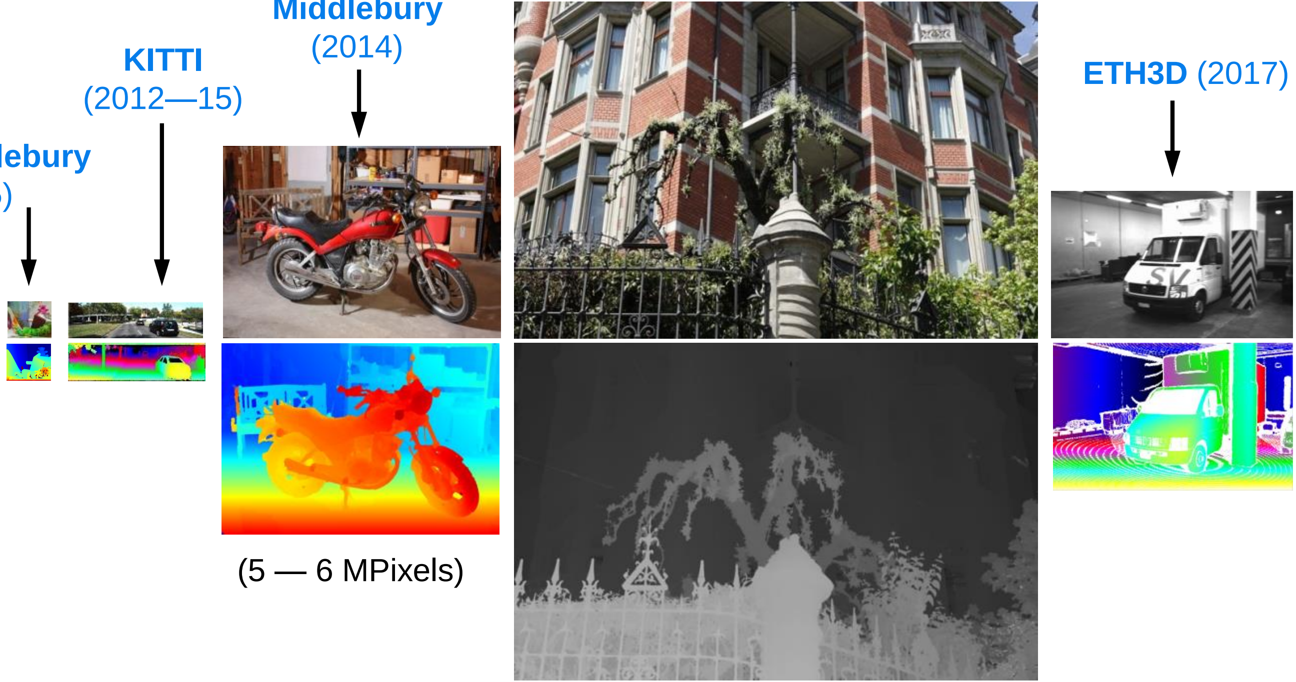
Kim+ 2013

Middlebury (2014)

KITTI (2012—15)

Middlebury (2005)

ETH3D (2017)



(5 — 6 MPixels)

(10—20 Mpixels)

Still challenging …

Fore-shortening

Specular

Transparency, reflections

Dynamic range

Untextured slanted surfaces

- Corner cases:
  - Challenging geometry
  - Complex appearance
- High resolution imagery
- Real-time platforms, resource-constraints

Different lighting

Priors for Stereo Matching

- Stereo matching is an ill-posed problem
- Priors provides robustness to ambiguity and noise, e.g.
  - Smoothness prior (1^{st} –order , 2^{nd} –order ...)
  - Discontinuities favored at image edges
  - Soft color segmentation cues (superpixels …)
- Priors explicitly added to optimization objective
- Priors terms in objective can be learned from training data

Outline

- Global Stereo Matching with piecewise-planar priors
- Semi-global Matching (SGM)
  - Local plane-sweep stereo
  - SGM with surface orientation priors
- Stereoscopic Scene Flow
- Deep Single-shot 6D Object Instance Detection

Global Optimization

- Find a per-pixel label map (D) (find all disparities jointly)
- Labels are discrete (ordered), $d \in L_D$

$$L_{\mathrm{D}} = [d_{min}, d_{max}]$$

- Optimize:

$$E(D) = \mathrm{E}_{\mathrm{data}}(D) + \mathrm{E}_{\mathrm{smooth}}(D)$$

- Data term encodes matching costs
- Smoothness term encodes prior/regularization
  - Example: neighboring pixels favored to take similar labels

Global Optimization

- Inference on Markov Random Fields (MRF)
- Minimize objective (energy):

$$E(D) = \mathrm{E}_{\mathrm{data}}(D) + \mathrm{E}_{\mathrm{smooth}}(L)$$
$$= \sum_{p \in I} C_p\left(d_p\right) + \sum_{(p,q) \in N} V_{pq}(d_p, d_q)$$

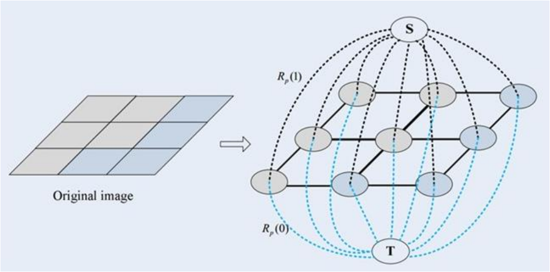
$C_p\left(d_p\right)$: matching cost term (*tabular representation*)

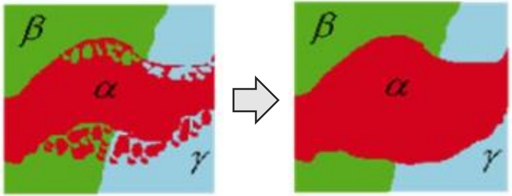
$V_{pq}(d, d')$: pairwise term (Potts, truncated linear or quadratic ...)

contrast sensitive Potts model favors discontinuity at image edges

Global Optimization

- Binary MRFs:
  - Efficient, exact methods known
  - Submodular $V(*,*)$: s-t mincut problem
- Multi-label MRFs:
  - NP-Hard, for useful choice of $V_{pq}(*,*)$
    - *Discontinuity-preserving* Potts model.
  - Approximation algorithms
    - Move-making (via binary graph cuts)

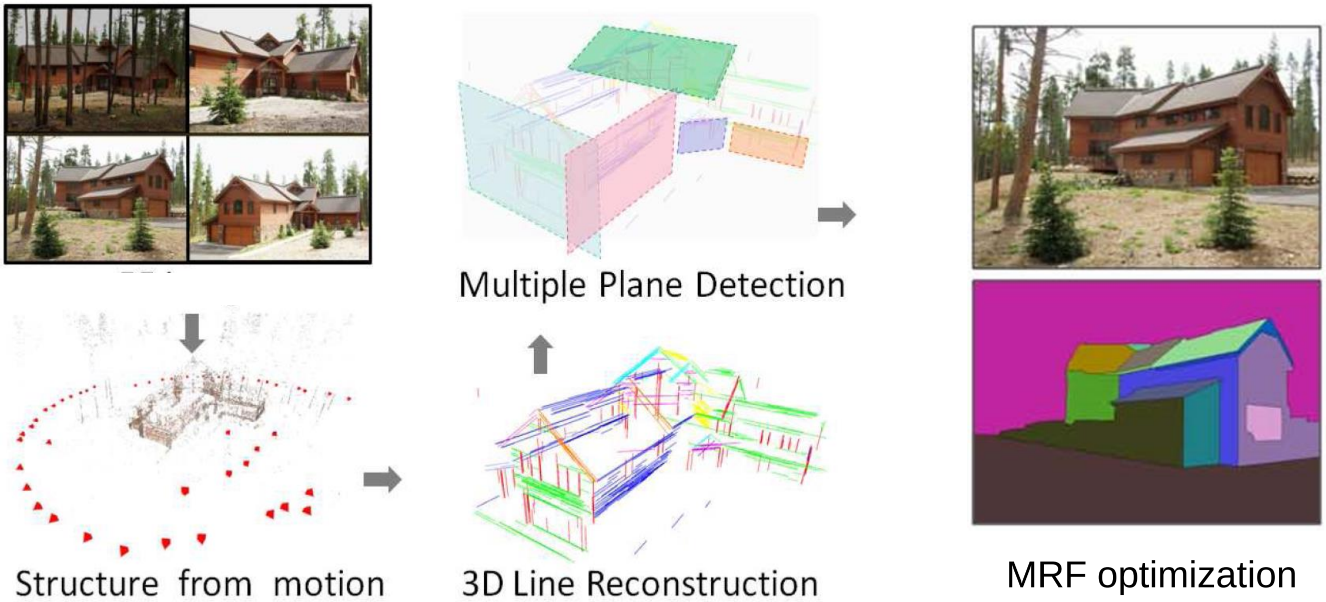




α- expansion move

Stereo matching with planar priors

[Sinha, Steedly, Szeliski 2009]



Stereo matching with planar priors

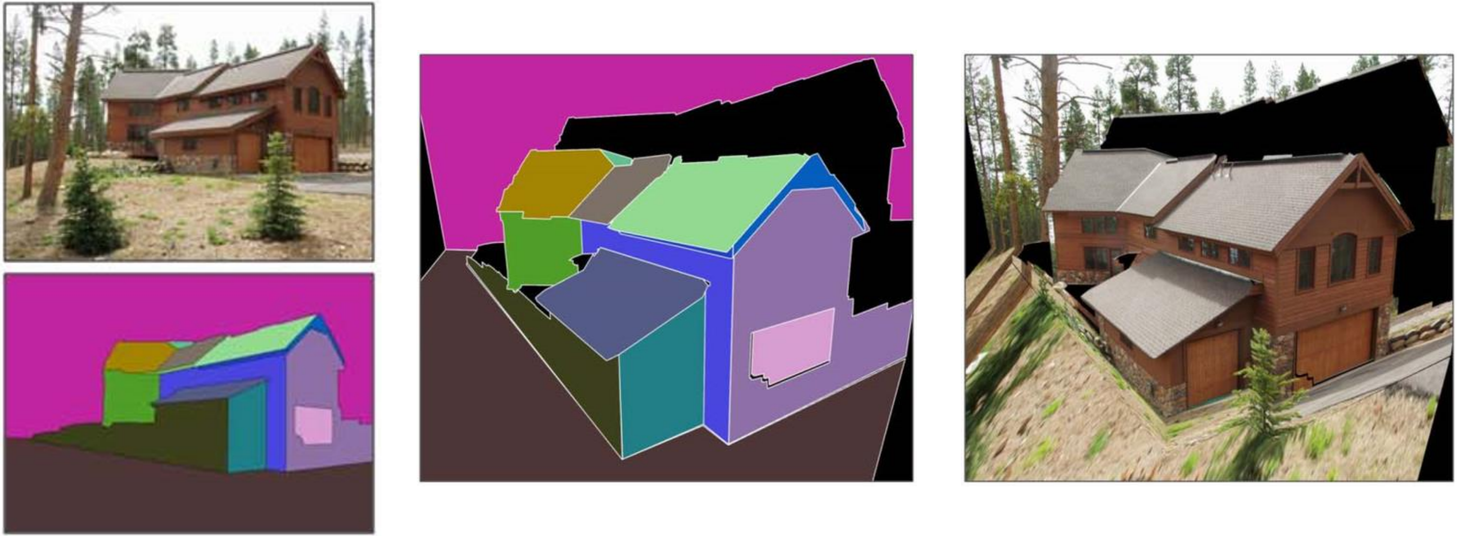
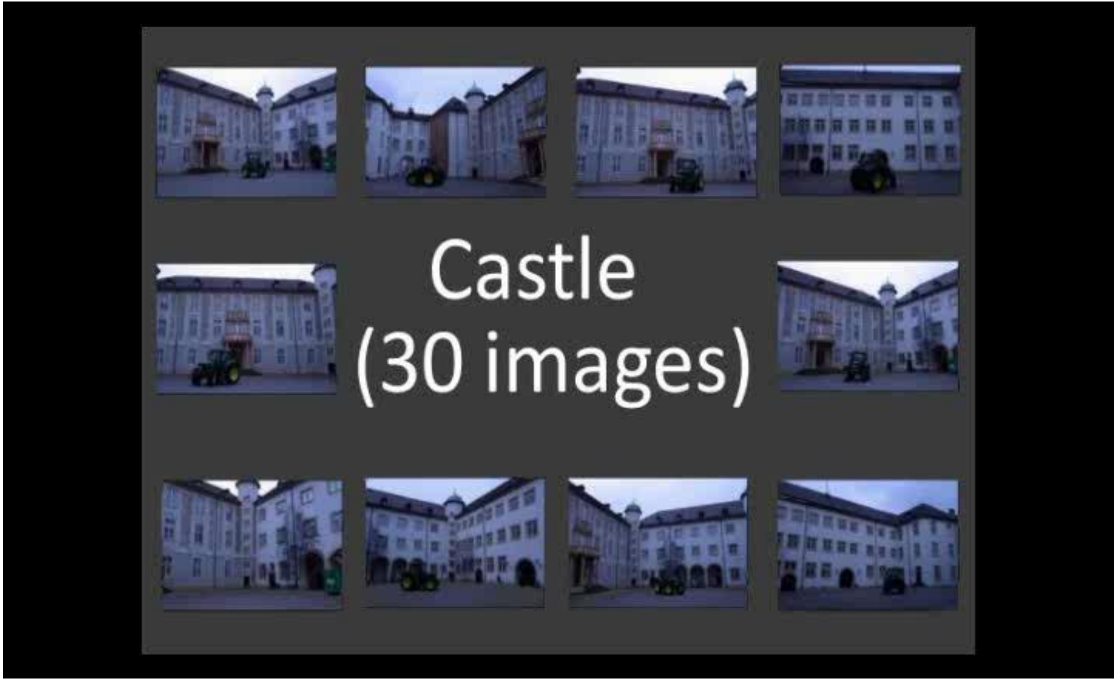


Image-based Rendering

Image-based Rendering

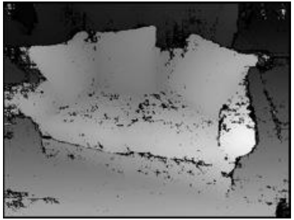


Stereo matching with planar priors

[Kowdle, Sinha, Szeliski 2012]

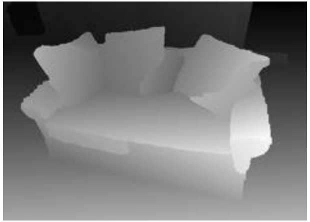
- Tackle *more* general scenes
- Plane hypotheses generated via local fitting
- Now, alternate between
  - Learning surface color models (online)
  - Graph cut optimization

Plane label map



Semi-global stereo (SGM)

Find planes



Depth map

Stereo matching with planar priors

[Kowdle, Sinha, Szeliski 2012]

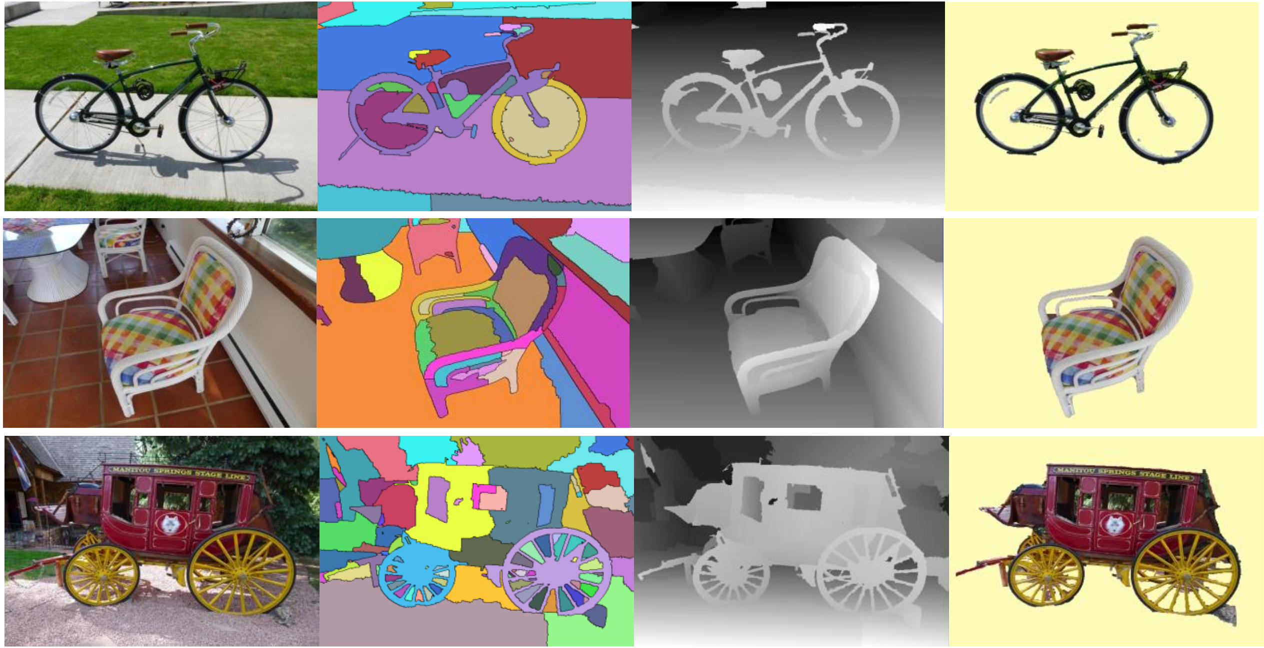
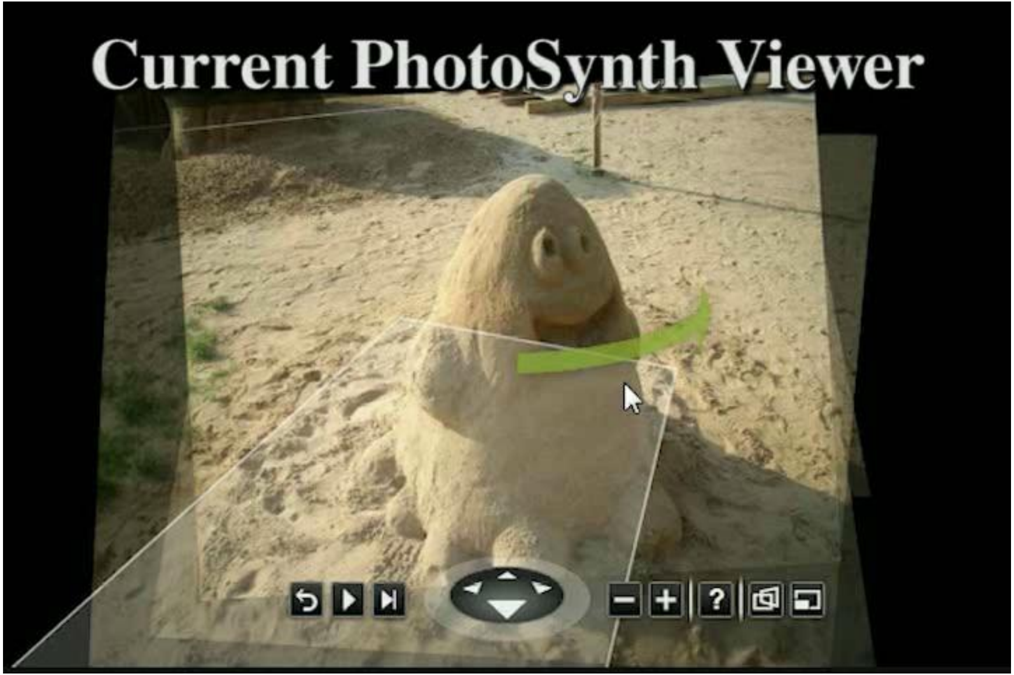


Image-based Rendering



Review: Stereo with planar priors

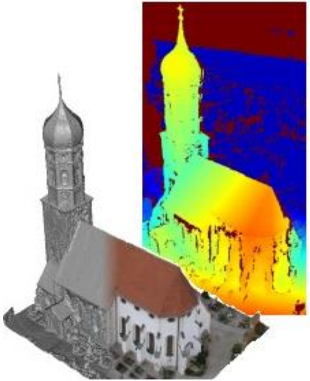
- MRF labels: planes (surfaces), NOT disparities.
- Estimated depth maps often approximate
  - ✓ accurate recovery of occlusion boundaries, surface normals
  - ✓ effective 2.5D proxies for novel view synthesis
- Limitations:
  - × Planarity prior too strong for general scenes
  - × Plane proposal generation is key; often imperfect

Outline

- Global Stereo Matching with piecewise-planar priors
- Semi-global Matching (SGM)
  - Local plane-sweep stereo
  - SGM with surface orientation priors
- Stereoscopic Scene Flow
- Deep Single-shot 6D Object Instance Detection

Semi Global Matching [Hirschmüller 2005]

- MRF inference (graph cuts, BP, ..) too slow
- SGM: Approximate even more; use heuristics
  - Parallelizable; practical on FPGA / GPUs
  - Widely used for assisted driving, robotics, aerial mapping …



Scanline Optimization (1D)

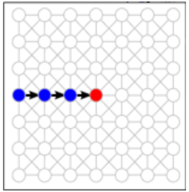
Minimize:

$$E(D) = \sum_{p \in I} C_p(d_p) + \sum_{(p,q) \in N} V_{pq}(d_p, d_q)$$

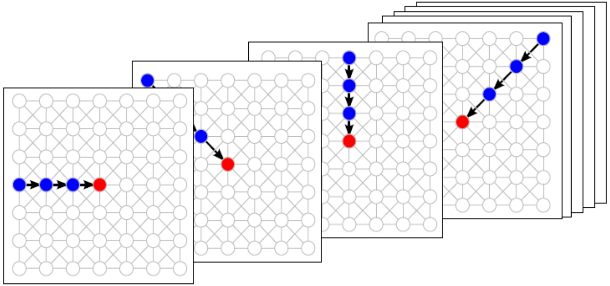
- Consider the above problem on a 1D scanline.
- Compute an aggregated matching cost

$$L_{\mathbf{r}}(\mathbf{p}, d) = C_{\mathbf{p}}(d) + \min_{d' \in \mathcal{D}} (L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d') + V(d, d')).$$

  - $\mathbf{r} = (1, 0)$: start at leftmost pixel, scan left



Semi Global Matching (SGM)



- For 8 directions
  - calculate aggregated costs

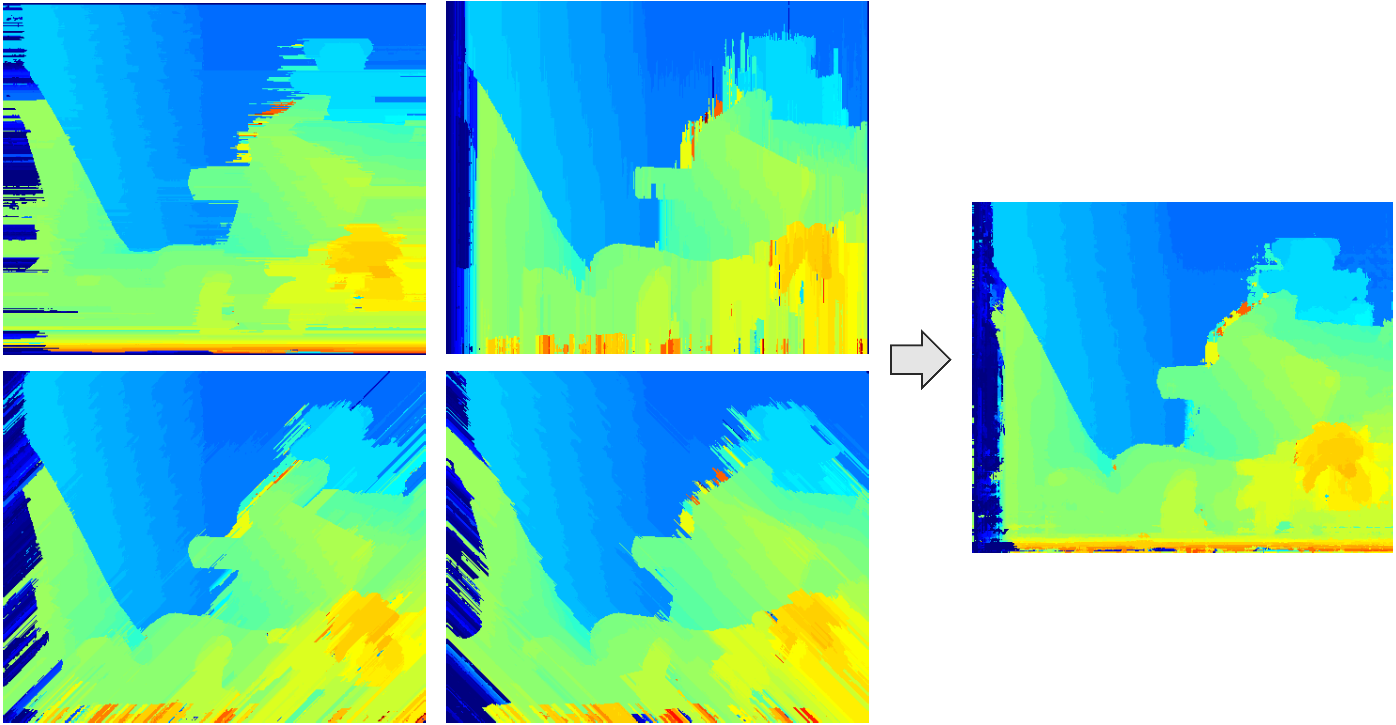
$$L_{\mathbf{r}}(\mathbf{p}, d) = C_{\mathbf{p}}(d) + \min_{d' \in \mathcal{D}} \left(L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d') + V(d, d') \right).$$

- Finally, sum the costs and select per-pixel minima.

$$S(\mathbf{p}, d) = \sum_{\mathbf{r}} L_{\mathbf{r}}(\mathbf{p}, d)$$

$$D_{\mathbf{p}} = \arg\min_{d} S(\mathbf{p}, d).$$

Semi Global Matching (SGM)



Semi Global Matching [Hirschmüller 2005]

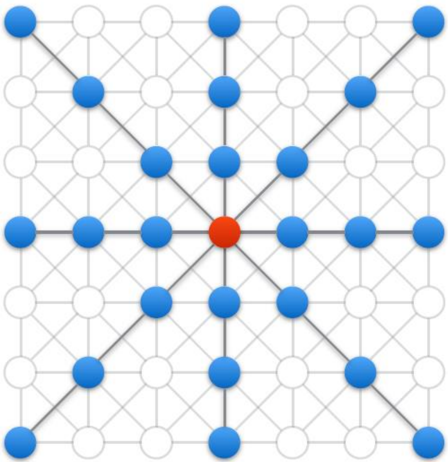
Approximates 2D MRF using 1D optimization

along 8 cardinal directions

$$E(D) = \sum_{\mathbf{p}} C_{\mathbf{p}}(d_{\mathbf{p}}) + \sum_{\mathbf{p},\mathbf{q} \in \mathcal{N}} V(d_{\mathbf{p}}, d_{\mathbf{q}})$$

- related to Belief Propagation

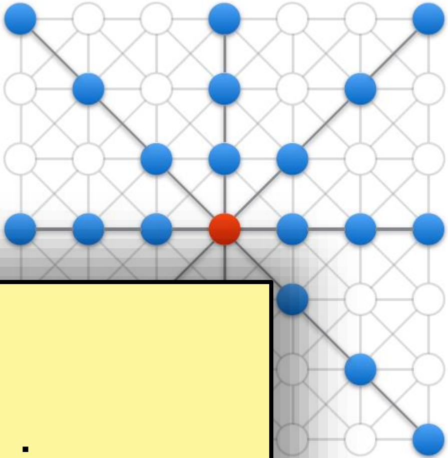
  [Drory et al. 2014]



Semi Global Matching [Hirschmüller 2005]

Approximates 2D MRF using 1D optimization

along 8 cardinal directions



$$E(D) = \sum_{\mathbf{p}} C_{\mathbf{p}}(d_{\mathbf{p}}) + \sum_{\mathbf{p},\mathbf{q}\in\mathcal{N}} V(d_{\mathbf{p}}, d_{\mathbf{q}})$$

- rela

- Evaluates the whole DSI
- Inefficient for high-resolution images

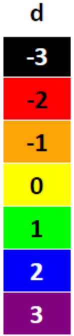
Outline

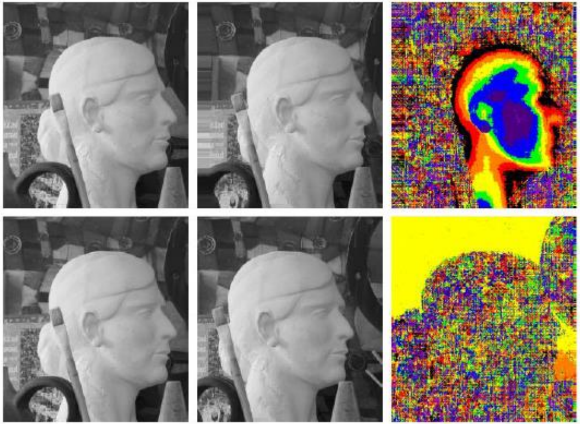
- Global Stereo Matching with piecewise-planar priors
- Semi-global Matching (SGM)
  - Local plane-sweep stereo
  - SGM with surface orientation priors
- Stereoscopic Scene Flow
- Deep Single-shot 6D Object Instance Detection

Local Plane Sweep (LPS) Stereo

[Sinha, Scharstein, Szeliski 2014]

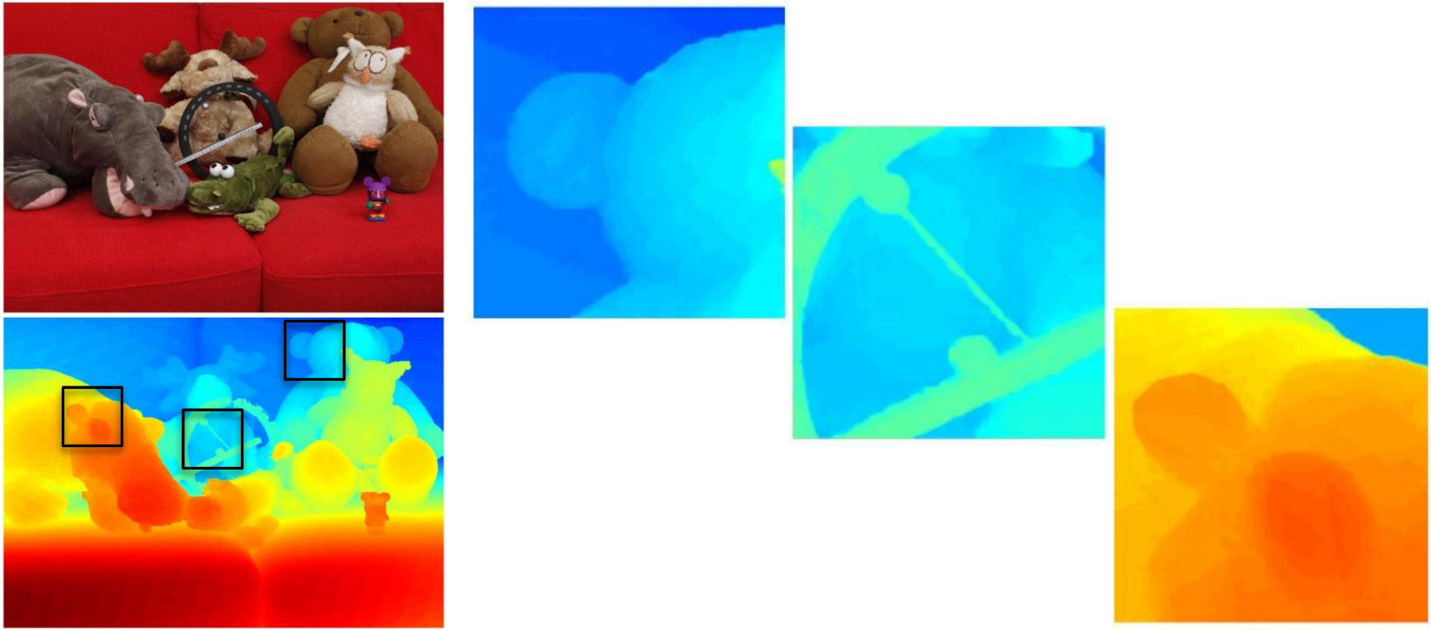
- Solve many local plane sweep stereo (LPS) problems
- Generates *surface* proposals; fuse them into a disparity map





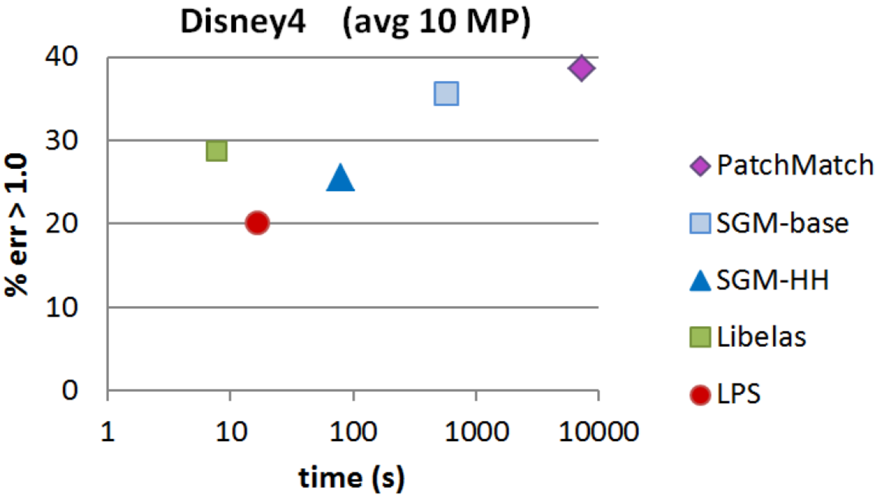
Local Plane Sweep (LPS) Stereo

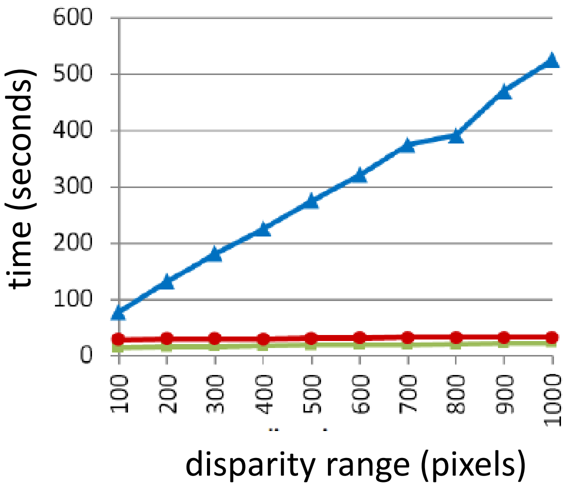
[Sinha, Scharstein, Szeliski 2014]



Local Plane Sweep (LPS) Stereo

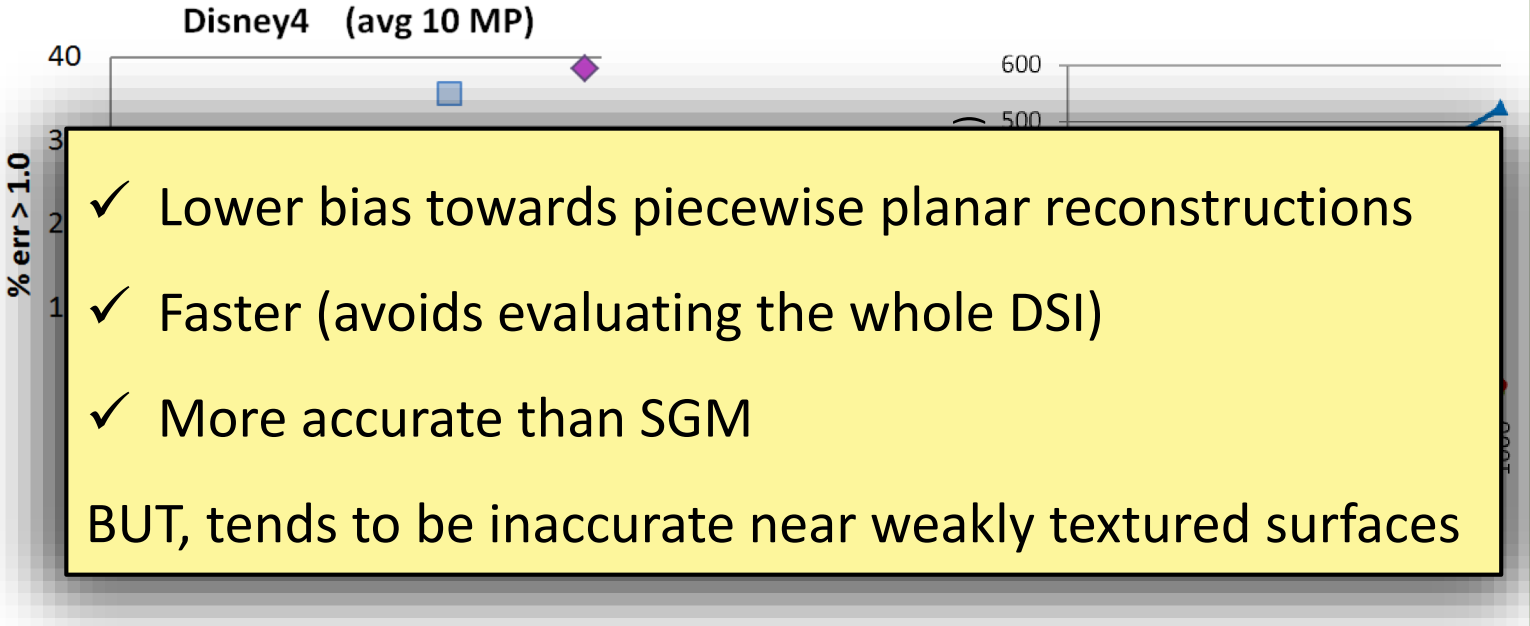
[Sinha, Scharstein, Szeliski 2014]





Local Plane Sweep (LPS) Stereo

[Sinha, Scharstein, Szeliski 2014]



- ✓ Lower bias towards piecewise planar reconstructions
- ✓ Faster (avoids evaluating the whole DSI)
- ✓ More accurate than SGM

BUT, tends to be inaccurate near weakly textured surfaces

Outline

- Global Stereo Matching with piecewise-planar priors
- Semi-global Matching (SGM)
  - Local plane-sweep stereo
  - SGM with surface orientation priors
- Stereoscopic Scene Flow
- Deep Single-shot 6D Object Instance Detection

Semi Global Matching [Hirschmüller 2005]

Approximates 2D MRF using 1D optimization along 8 cardinal directions

$$E(D) = \sum_{\mathbf{p}} C_{\mathbf{p}}(d_{\mathbf{p}}) + \sum_{\mathbf{p},\mathbf{q} \in \mathcal{N}} V(d_{\mathbf{p}}, d_{\mathbf{q}})$$

$$V(d, d') = \begin{cases} 0 & \text{if } d = d' \\ P_1 & \text{if } |d - d'| = 1 \\ P_2 & \text{if } |d - d'| \geq 2 \end{cases}$$

Semi Global Matching [Hirschmüller 2005]

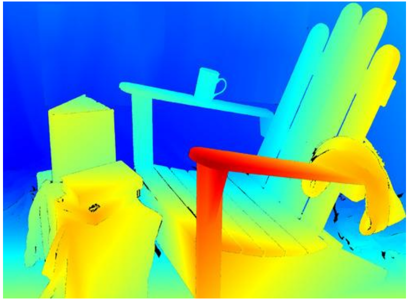
Approximates 2D MRF using 1D optimization along 8 cardinal directions

$$E(D) = \sum_{\mathbf{p}} C_{\mathbf{p}}(d_{\mathbf{p}}) + \sum_{\mathbf{p},\mathbf{q}\in\mathcal{N}} V(d_{\mathbf{p}}, d_{\mathbf{q}})$$

$$\begin{cases} 0 & \text{if } d = d' \end{cases}$$

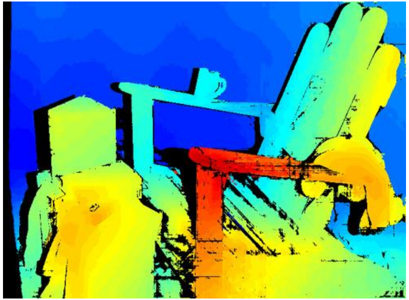
- Fronto parallel bias
- Inaccurate on slanted untextured surfaces

Left image

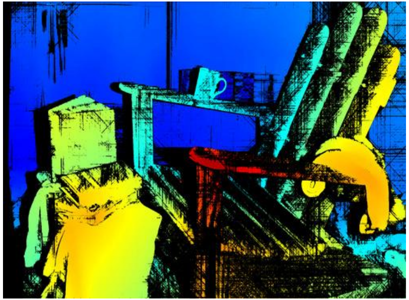


GT disparities

SGM @ quarter resolution



SGM @ full resolution (6 MP)



SGM-P: SGM with orientation priors

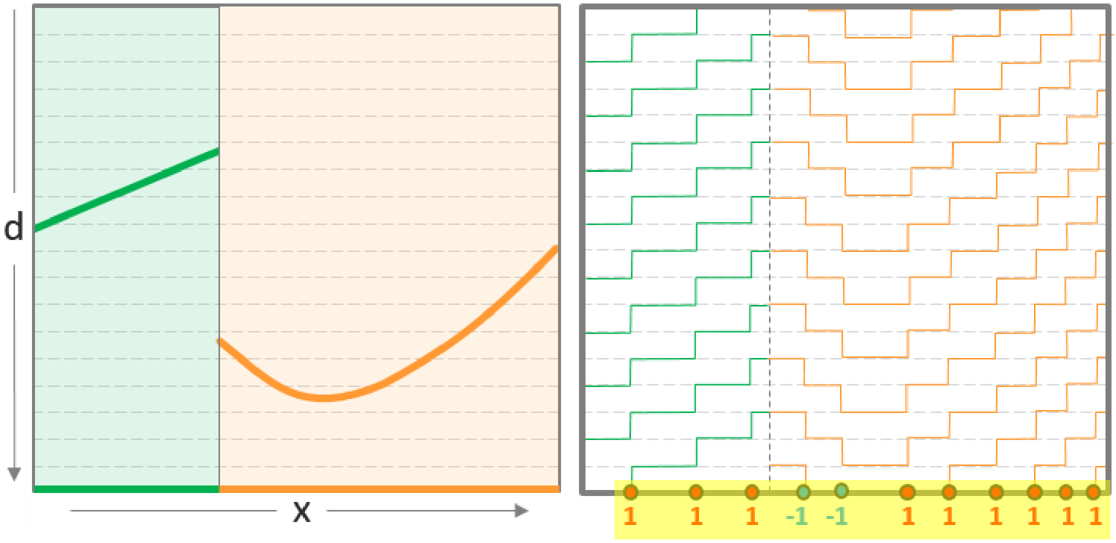
[Scharstein, Taniai, Sinha, 3DV 2017]

- What if we knew the surface slant?
- Replace fronto-parallel bias with bias parallel to surface

Idea:

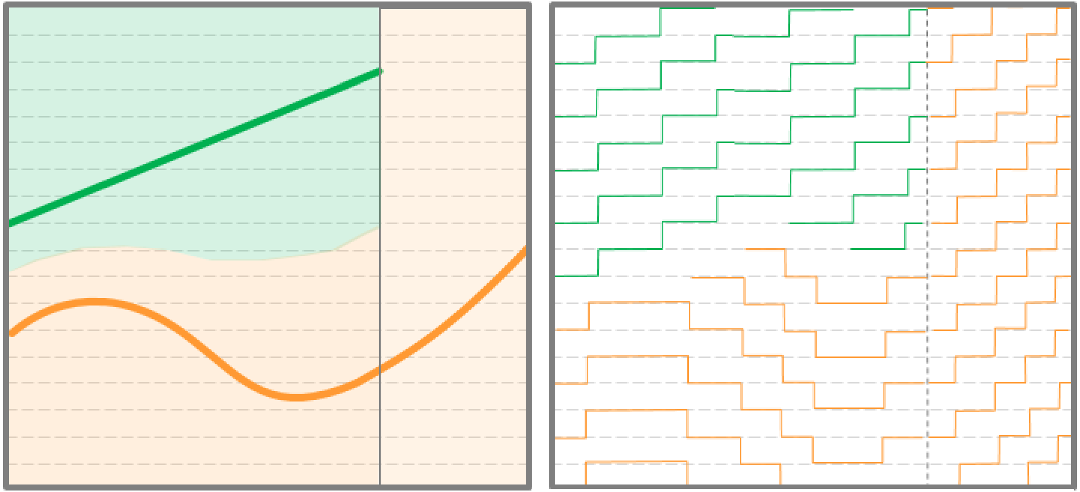
- *Rasterize* disparity surface prior (at arbitrary depth)
- Adjust $V(d, d')$ to follow discrete disparity “steps”

SGM-P: 2D orientation priors



$$V_S(d_{\mathbf{p}}, d'_{\mathbf{p}}) = V(d_{\mathbf{p}} + j_{\mathbf{p}}, d'_{\mathbf{p}})$$

SGM-P: 3D orientation priors



$$V_S(d_{\mathbf{p}}, d'_{\mathbf{p}}) = V(d_{\mathbf{p}} + j_{\mathbf{p}}(d_{\mathbf{p}}), d'_{\mathbf{p}})$$

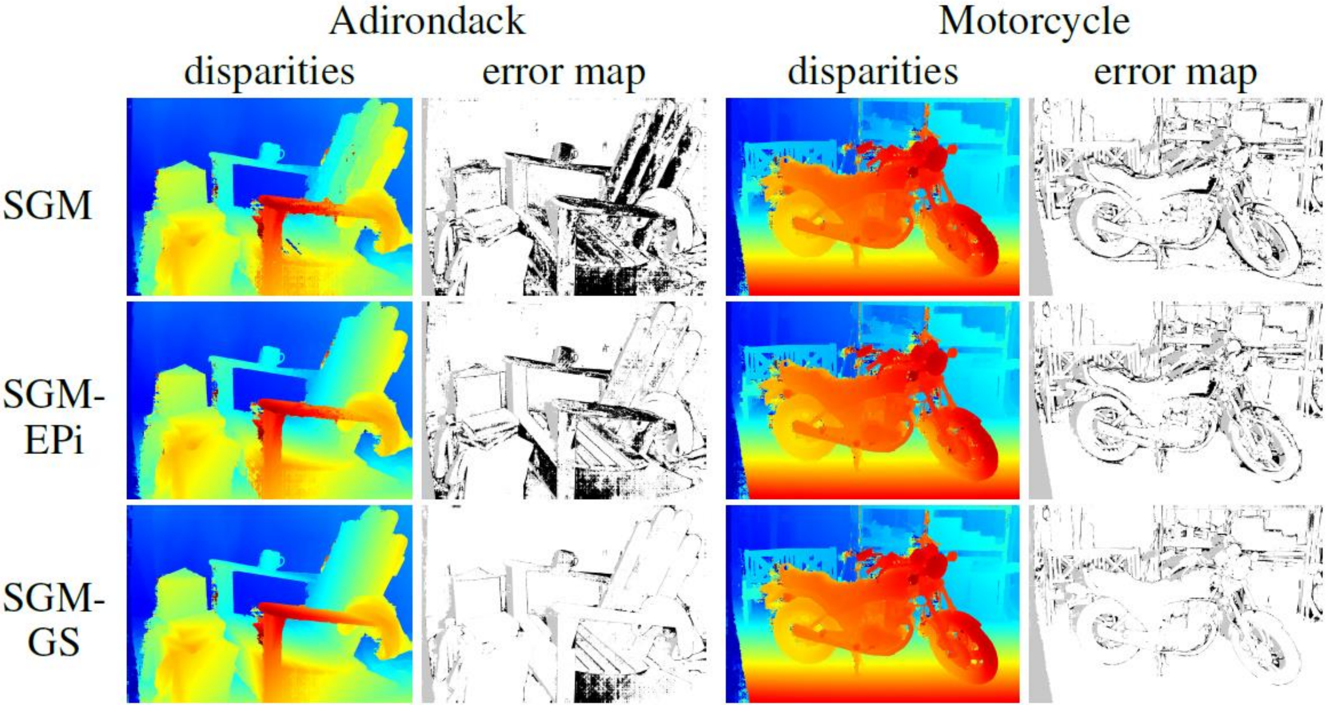
Jump locations vary with disparity

SGM-P: Where do we get priors?

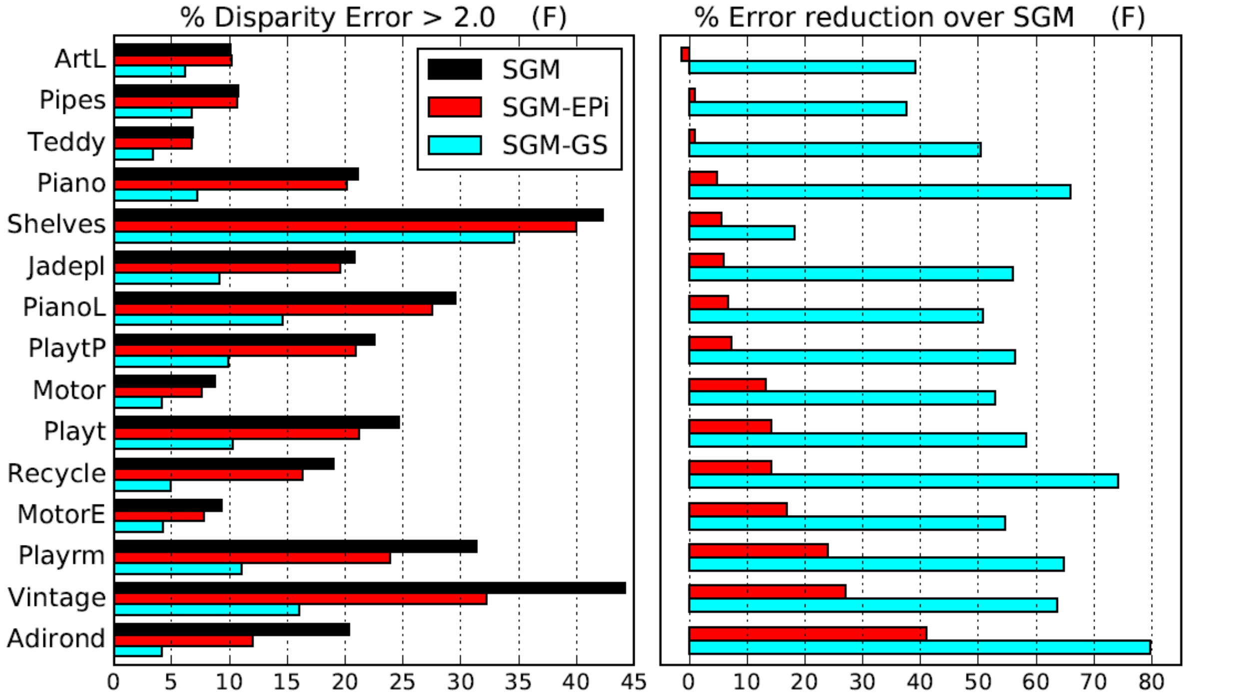
- Matched features + triangulation
- Matched features + plane fitting
- Low-res matching + plane fitting
- Ground truth oracle
- Semantic analysis
- Manhattan-world assumptions

Input SGM-EPi SGM-GS

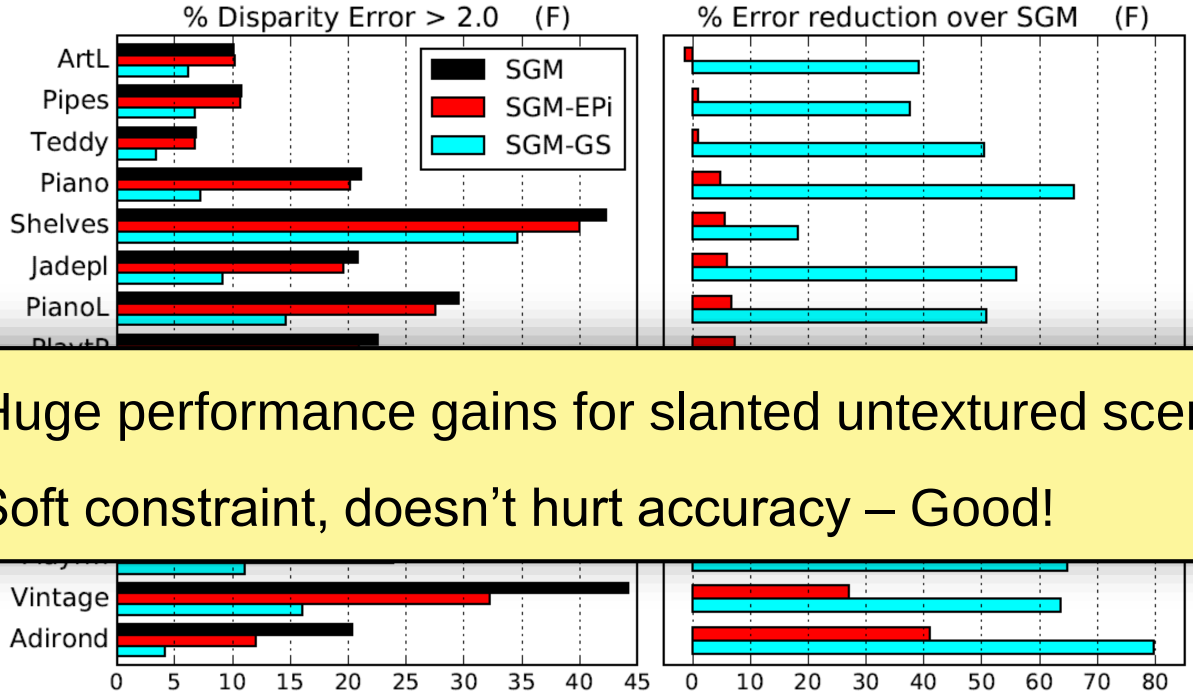
SGM-P: Results



SGM-P: Results



SGM-P: Results



- Huge performance gains for slanted untextured scenes
- Soft constraint, doesn’t hurt accuracy – Good!

Fast Multi-frame Stereo Scene Flow with Motion Segmentation

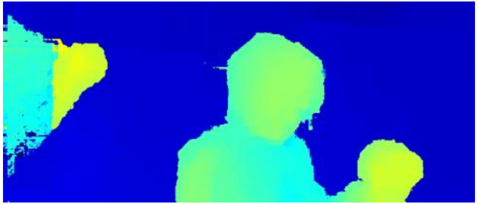
Taniai, Sinha, Sato 2017

Input:
Stereo Video

Left

Right

Output



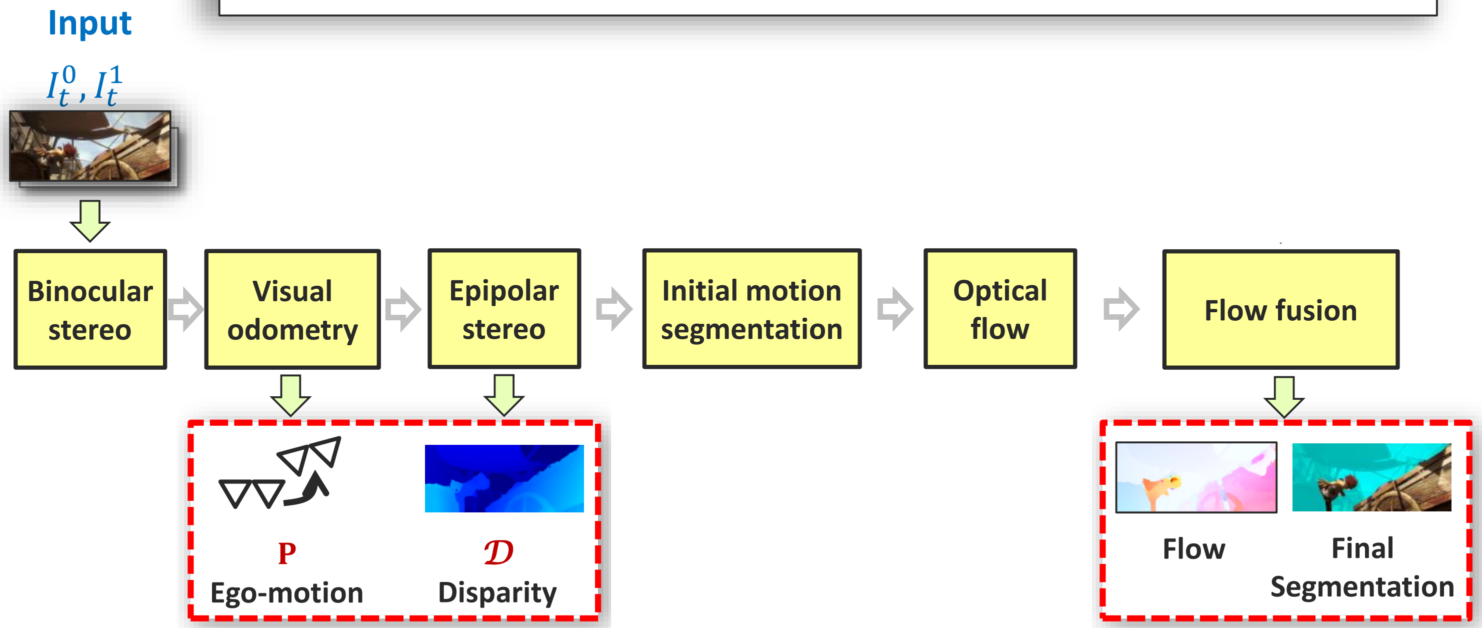
Disparity Map

Optical Flow

Moving object segmentation

Fast Multi-frame Stereo Scene Flow with Motion Segmentation

Taniai, Sinha, Sato 2017



Fast Multi-frame Stereo Scene Flow with Motion Segmentation

Taniai, Sinha, Sato 2017

KITTI 2015 Scene Flow Benchmark (Nov 2016)

Rank	Method	D1-bg	D1-fg	D1-all	D2-bg	D2-fg	D2-all	Fl-bg	Fl-fg	Fl-all	SF-bg	SF-fg	SF-all	Time
1	PRSM [43]	**3.02**	**10.52**	**4.27**	**5.13**	**15.11**	**6.79**	**5.33**	**17.02**	**7.28**	**6.61**	**23.60**	**9.44**	300 s
2	OSF [30]	4.54	12.03	5.79	5.45	19.41	7.77	5.62	22.17	8.37	7.01	28.76	10.63	50 min
3	**FSF+MS** (ours)	5.72	11.84	6.74	7.57	21.28	9.85	8.48	29.62	12.00	11.17	37.40	15.54	2.7 s
4	CSF [28]	4.57	13.04	5.98	7.92	20.76	10.06	10.40	30.33	13.71	12.21	36.97	16.33	80 s
5	PR-Sceneflow [42]	4.74	13.74	6.24	11.14	20.47	12.69	11.73	27.73	14.39	13.49	33.72	16.85	150 s
8	PCOF + ACTF [10]	6.31	19.24	8.46	19.15	36.27	22.00	14.89	62.42	22.80	25.77	69.35	33.02	0.08 s (GPU)
12	GCSF [8]	11.64	27.11	14.21	32.94	35.77	33.41	47.38	45.08	47.00	52.92	59.11	53.95	2.4 s

200 road scenes with multiple moving objects

Rank	SF-all	Time
1	**9.44**	300 s
2	10.63	50 min
3	15.54	2.7 s
4	16.33	80 s
5	16.85	150 s
8	33.02	0.08 s (GPU)
12	53.95	2.4 s

Outline

- Global Stereo Matching with piecewise-planar priors
- Semi-global Matching (SGM)
  - Local plane-sweep stereo
  - SGM with surface orientation priors
- Stereoscopic Scene Flow
- Deep Single-shot 6D Object Instance Detection

Object recognition + pose estimation

- Task: Given a RGB image (with known camera intrinsics), recognize the object instance and predict its 3D position and orientation.

Lowe 2001

Rothganger+ 2005

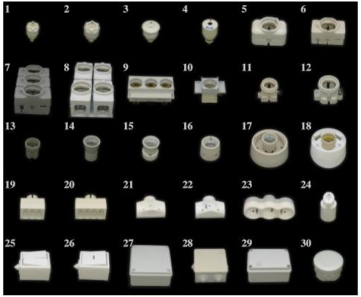
Hinterstoisser+ 2012
Brachmann+ 2014, 16
…
…

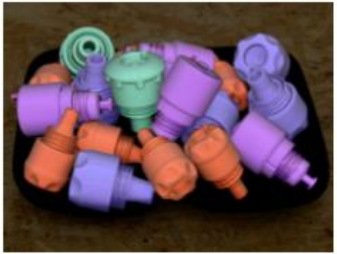
- Local features (SIFT, Affine invariance)
- Textured, distinctive objects
- Geometric verification
- A few training images are fine ..

CNN-based
Rad + Lepetit 2017,
Kehl+ 2017,
Xiang+ 2017

- Global deep features
- Geometry not used
- small, texture-less objects
- Huge training set needed

Texture-less Object 6D Pose Datasets





LINEMOD [2012]
15 objects

T-LESS [2017]
30 objects

YCB-VIDEO [2018]
21 objects

Deep 6D object pose estimation

- BB8 [Rad and Lepetit 2017]

image	→	CNN$_1$	→	cropped image	→	CNN$_2$	→	CNN$_3$	→	6D pose
		2d detector				*pose estimator*		*pose refinement*		

- SSD-6D [Kehl+ 2017]

image	→	CNN$_1$(~SSD)	→	2D to 6D	→	CNN$_2$	→	6D pose
		2d detector, viewpoint classifier		*heuristic*		*pose refinement*		

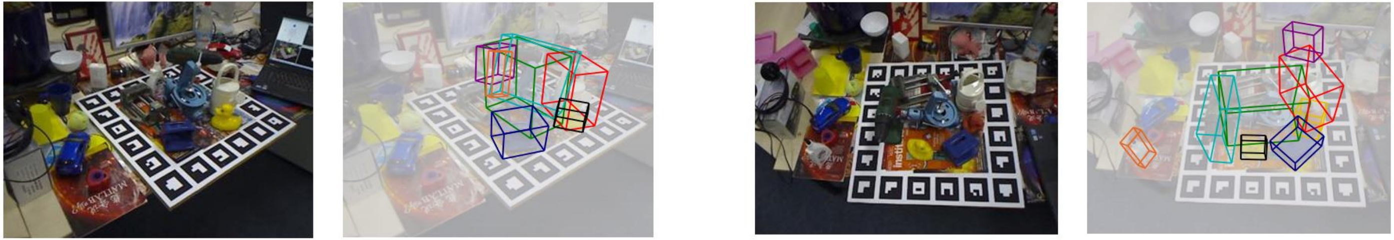
- Ours

image	→	CNN$_1$(~YOLO)	→	PnP	→	6D pose
		3D bounding box corner predictor		*pose solver*		

Real-Time Seamless Single Shot 6D Object Pose Prediction

Tekin, Sinha, Fua 2018 (in CVPR, to appear)

- Single-shot 2D object detection (YOLO, SSD)
- Our CNN predicts 2D projections of 3D bounding box vertices (+ centroid). We run PnP solver on 9 2D-3D correspondences.
- Accurate, fast (50-90 fps); detects multiple objects in one pass.

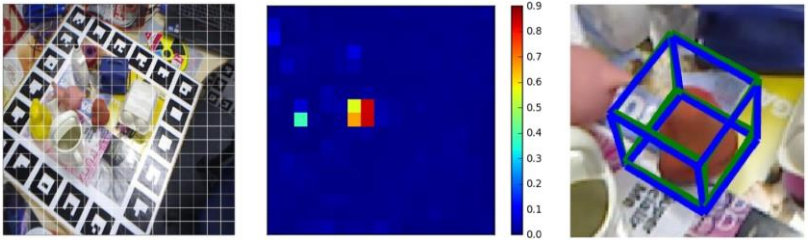


Real-Time Seamless Single Shot 6D Object Pose Prediction

Tekin, Sinha, Fua 2018 (in CVPR, to appear)

Training:

- ground truth 2D coordinates of the 9 control points are the targets
- modify YOLO loss function (for confidence estimation)
- data augmentation

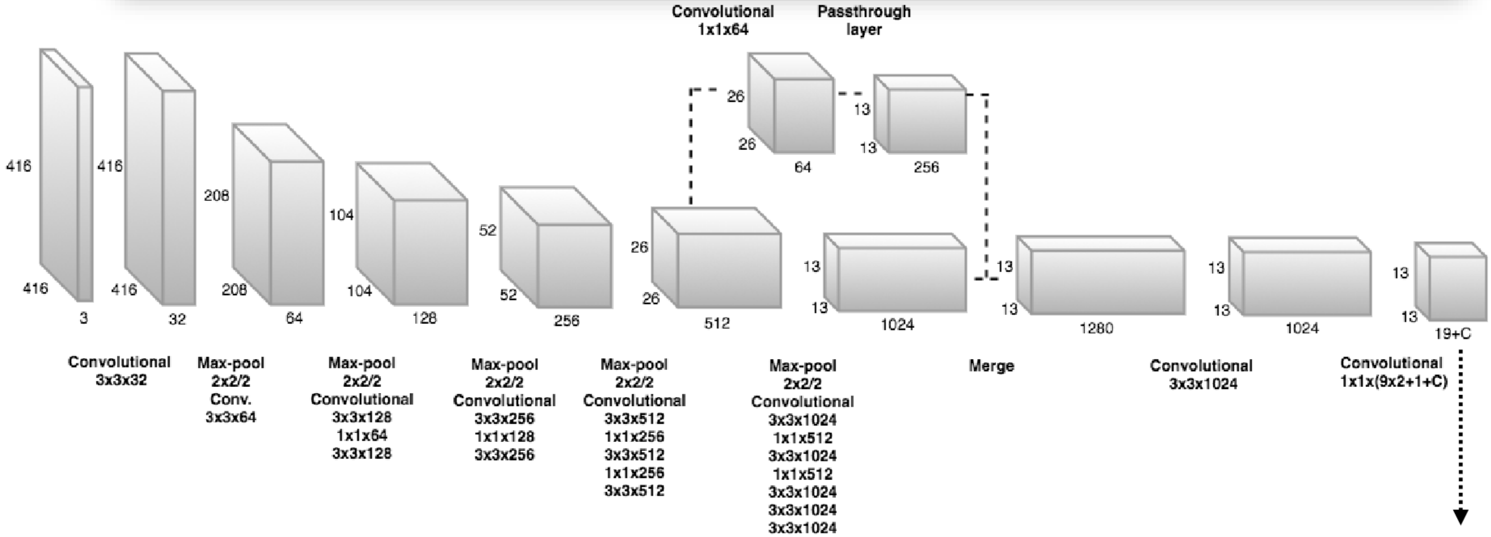


Testing:

- Subpixel refinement
- PnP (RANSAC, least squares)

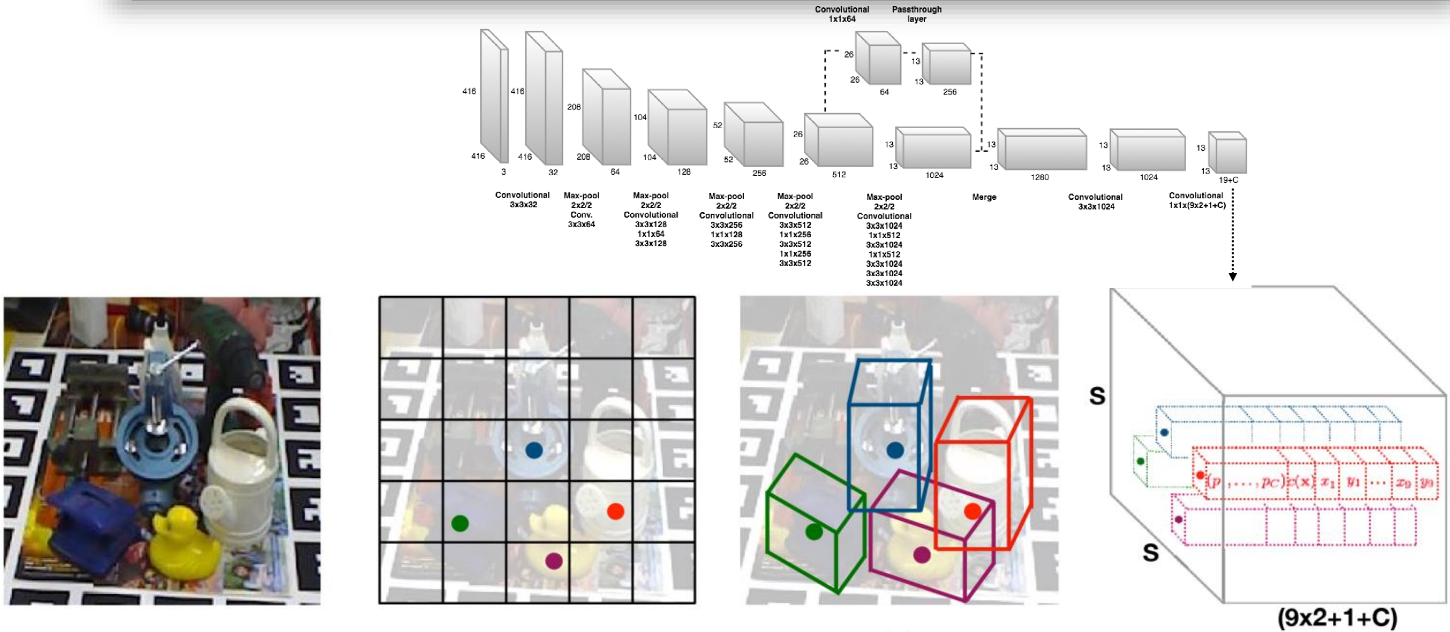
Real-Time Seamless Single Shot 6D Object Pose Prediction

Tekin, Sinha, Fua 2018 (in CVPR, to appear)



Real-Time Seamless Single Shot 6D Object Pose Prediction

Tekin, Sinha, Fua 2018 (in CVPR, to appear)



Real-Time Seamless Single Shot 6D Object Pose Prediction

Tekin, Sinha, Fua 2018 (in CVPR, to appear)

- Accuracy w.r.t. two metrics (2D projection, 3D overlap)
  - Percentage of test images where the error was within a threshold

2D metric

Method	w/o Refinement			w/ Refinement	
Object	Brachmann [2]	BB8 [25]	OURS	Brachmann [2]	BB8 [25]
Ape	-	***95.3***	92.10	85.2	**96.6**
Benchvise	-	80.0	**95.06**	67.9	90.1
Cam	-	80.9	**93.24**	58.7	86.0
Can	-	84.1	**97.44**	70.8	91.2
Cat	-	97.0	**97.41**	84.2	98.8
Driller	-	74.1	***79.41***	73.9	**80.9**
Duck	-	81.2	**94.65**	73.1	92.2
Eggbox	-	87.9	**90.33**	83.1	91.0
Glue	-	89.0	**96.53**	74.2	92.3
Holepuncher	-	90.5	**92.86**	78.9	95.3
Iron	-	78.9	***82.94***	83.6	**84.8**
Lamp	-	74.4	**76.87**	64.0	75.8
Phone	-	77.6	**86.07**	60.6	85.3
Average	69.5	83.9	**90.37**	73.7	89.3

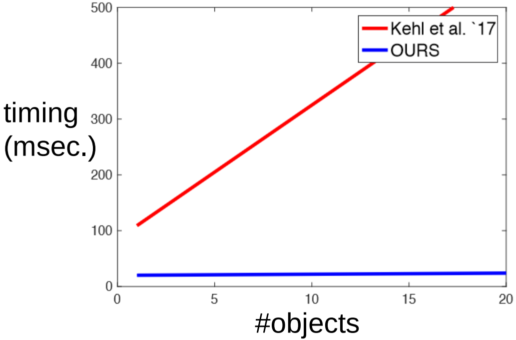
3D metric

Method	w/o Refinement				w/ Refinement		
Object	Brachmann [2]	BB8 [25]	SSD-6D [10]	OURS	Brachmann [2]	BB8 [25]	SSD-6D [10]
Ape	-	***27.9***	0	21.62	33.2	40.4	**65**
Benchvise	-	62.0	0.18	***81.80***	64.8	**91.8**	80
Cam	-	***40.1***	0.41	36.57	38.4	55.7	**78**
Can	-	48.1	1.35	***68.80***	62.9	64.1	**86**
Cat	-	***45.2***	0.51	41.82	42.7	62.6	**70**
Driller	-	58.6	2.58	***63.51***	61.9	**74.4**	73
Duck	-	***32.8***	0	27.23	30.2	44.3	**66**
Eggbox	-	40.0	8.9	***69.58***	49.9	57.8	**100**
Glue	-	27.0	0	***80.02***	31.2	41.2	**100**
Holepuncher	-	42.4	0.30	***42.63***	52.8	**67.2**	49
Iron	-	67.0	8.86	***74.97***	80.0	**84.7**	78
Lamp	-	39.9	8.20	***71.11***	67.0	**76.5**	73
Phone	-	35.2	0.18	***47.74***	38.1	54.0	**79**
Average	32.3	43.6	2.42	***55.95***	50.2	62.7	**79**

Real-Time Seamless Single Shot 6D Object Pose Prediction

Tekin, Sinha, Fua 2018 (in CVPR, to appear)

- Running Times:
  - On TitanX or similar GPU.
  - using cuDNN



Method	Overall speed for 1 object	Refinement runtime
Brachmann et al. [2]	2 fps	100 ms/object
Rad & Lepetit [25]	3 fps	21 ms/object
Kehl et al. [10]	10 fps	24 ms/object
OURS	50 fps	-

Method	2D projection metric	Speed
416×416	89.71	94 fps
480×480	90.00	67 fps
544×544	90.37	50 fps
688×688	90.65	43 fps

When input image is resized, our method remains accurate and runs much faster

Summary

- Image – Image Correspondence
  - Stereo Matching
    - Algorithmic improvements with different trade-offs
  - Unified Stereoscopic Scene flow estimation
    - Main insight: Solving *more* improves accuracy but *<u>also speed</u>*
- Image – 3D model alignment
  - 6D Object detection and Pose Estimation
    - Predict 2D *control point locations* in image; solve pose algebraically

Collaborators

Tatsunori Taniai
RIKEN, Tokyo

Daniel Scharstein
Middlebury College

Rick Szeliski
Facebook

Drew Steedly
Microsoft

Yoichi Sato
Univ. of Tokyo

Adarsh Kowdle
Google

Bugra Tekin
EPFL

Pascal Fua
EPFL