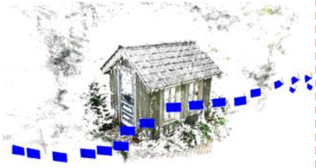
Efficient and Accurate 3D Scene Reconstruction and Object Pose Prediction

Sudipta Sinha

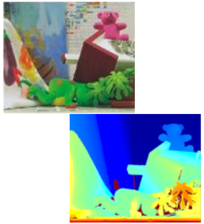
Microsoft Research

University of Illinois at Urbana-Champaign
May 9, 2018

Overview



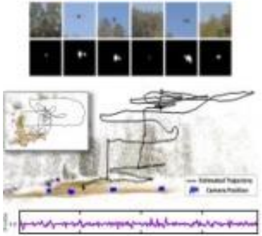
Structure from Motion



Dense Stereo

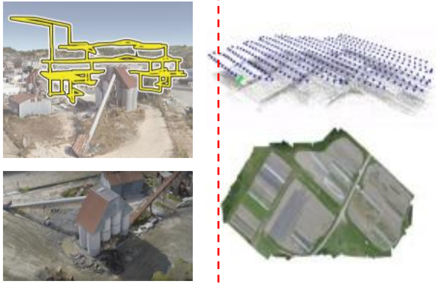
Photometric Stereo

Scene Flow



Multiview Tracking

Image-based Rendering



Aerial Mapping with UAVs

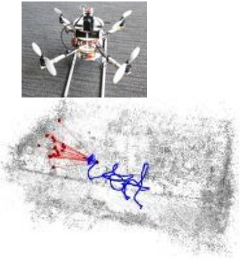


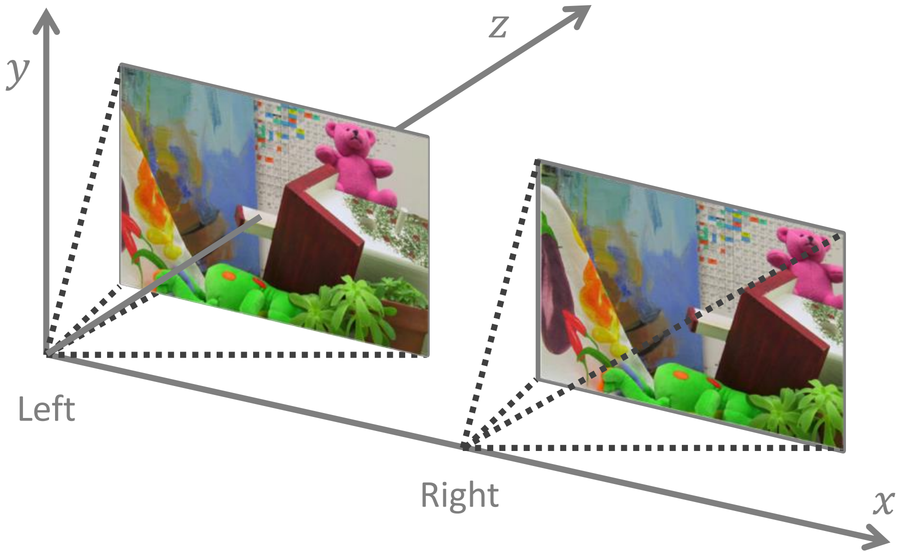
Image-based
Camera Localization

Object 6D Pose
Augmented Reality

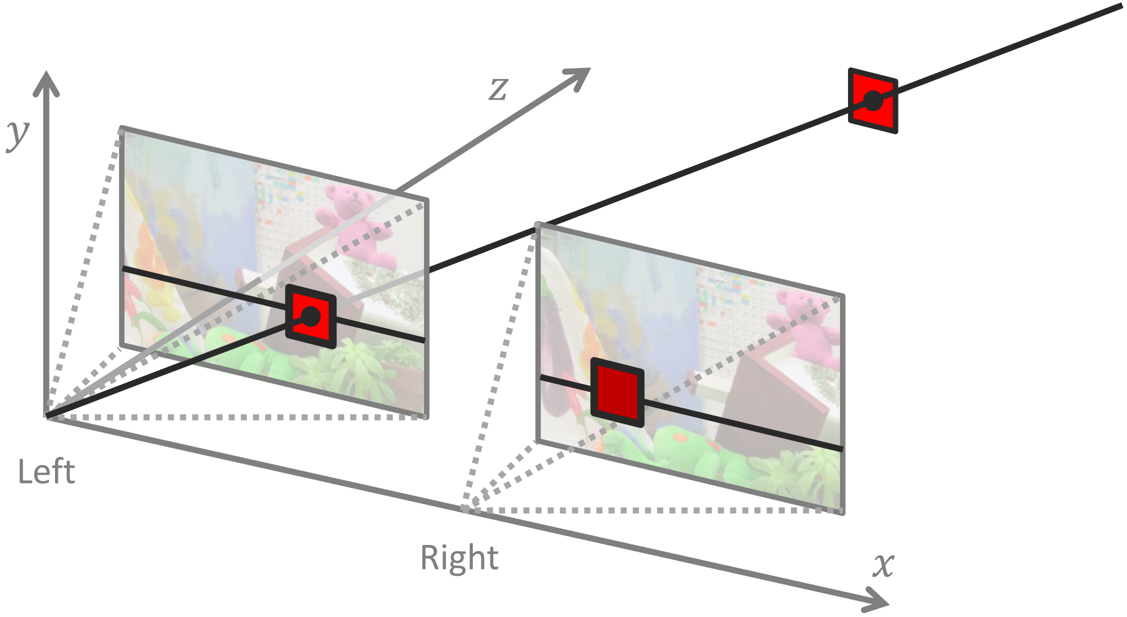
Outline

- Stereo Matching: New Trends
- Semi-Global Stereo Matching (SGM)
  - SGM with Surface Orientation Priors
- Stereo Scene Flow with Motion Segmentation
- Trajectory Planning for Aerial Multi View Stereo
- Deep 6D Object Pose Prediction

Stereo Matching



Stereo Matching



Still Challenging

Fore-shortening

Specular

Transparency, reflections

Different lighting

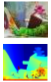
Dynamic range

Untextured slanted surfaces

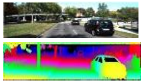
- State of the art methods are still …
  - Inaccurate in many corner cases
  - Too slow for real-time, resource constrained systems

Stereo benchmarks

Middlebury (2005)



KITTI (2012—15)



Middlebury (2014)

(5 — 6 MPixels)

Kim+ 2013

(10—20 Mpixels)

ETH3D (2017)

Classical Methods (MRF inference)

- Find a per-pixel label (disparity map) D, by minimizing energy:

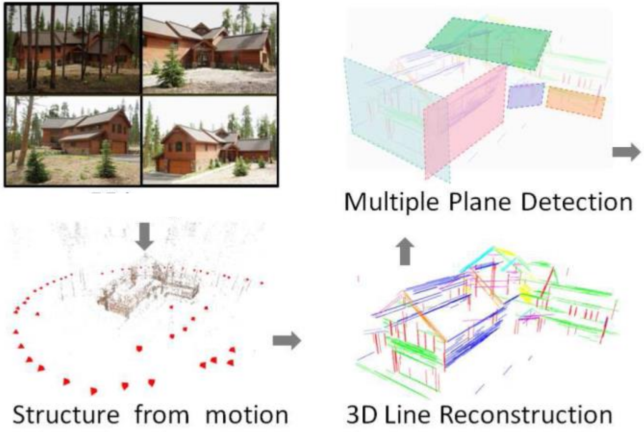
$$E(D) = \mathrm{E}_{\mathrm{data}}(D) + \mathrm{E}_{\mathrm{smooth}}(D)$$

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

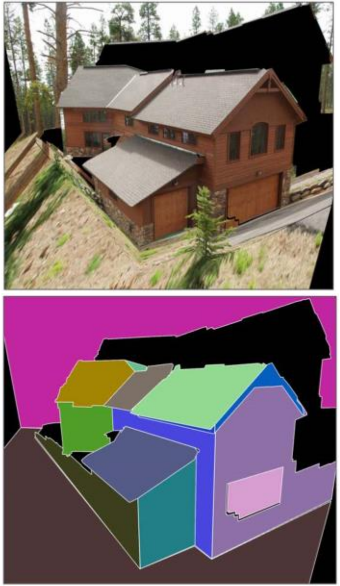
  - Data (cost) term encodes matching costs
  - Smoothness (cost) term encodes prior
- Discrete vs. Continuous labels
- Inference: Graph Cuts, Belief Propagation, PatchMatch-style optim.

Piecewise Planar Stereo

[Sinha, Steedly, Szeliski 2009]



MRF energy minimization
via graph cuts

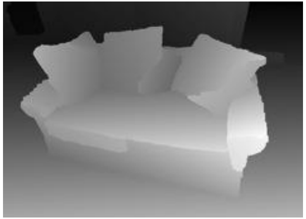


Novel View

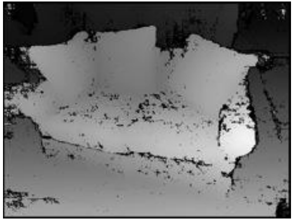
Piecewise Planar Stereo ***Revisited***

[Kowdle, Sinha, Szeliski 2012]

- Local plane fitting (more flexible)
- CRF models photo-consistency (stereo cue) and color segmentation (monocular cue)
- Learn color models per-surface
- Alternate between graph cuts and learning



Depth map



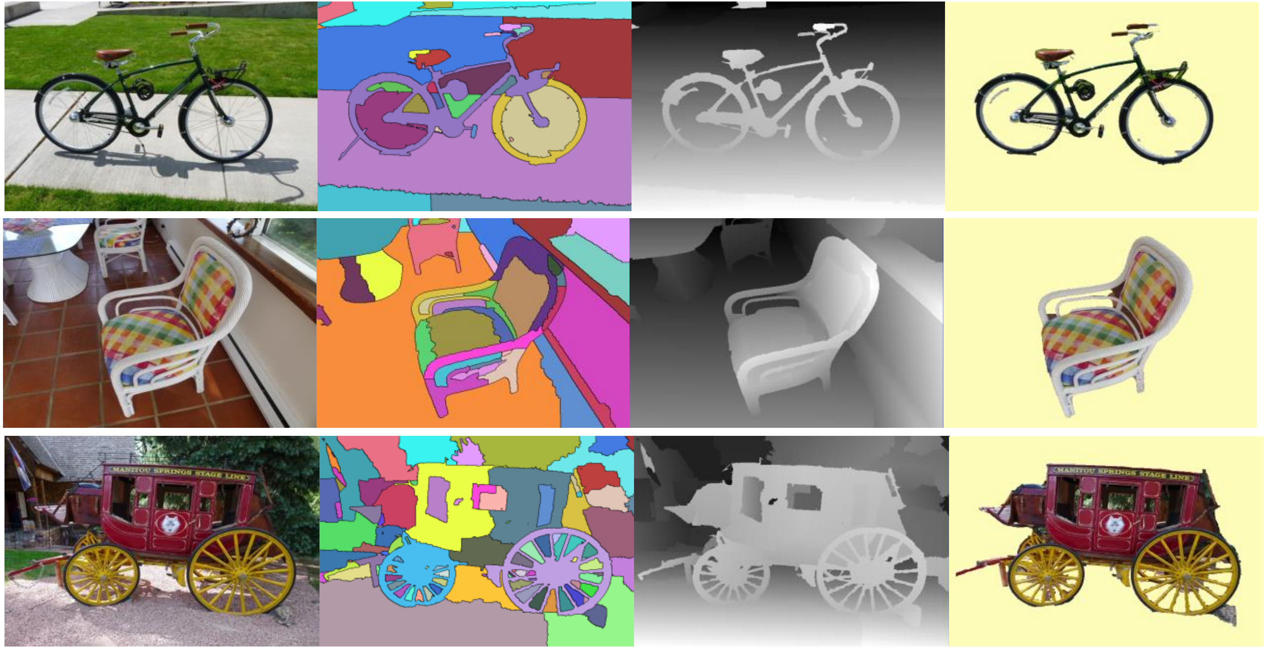
Semi-global stereo (SGM)

Find planes

Label map

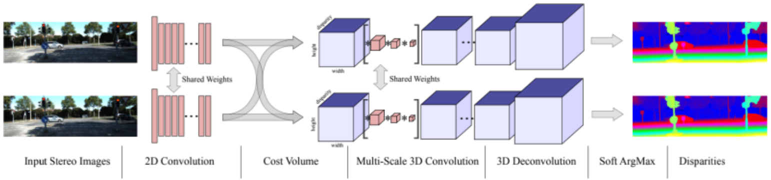
Piecewise Planar Stereo *Revisited*

[Kowdle, Sinha, Szeliski 2012]

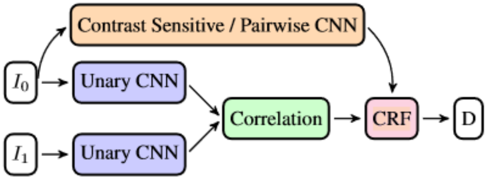


New Trends

- Learning the matching cost:
  - MC-CNN [Zbontar + Lecun 2015], Chen+ 2015, Luo+ 2016
- Continuous MRFs: [Taniai+ 2017] (Rank 1 on Middlebury 2014!)
- Deep stereo regression (end to end training)
  - FlowNet [Dosovitskiy+ 2015], DispNet [Mayer+ 2016]
- Return of "Correlation"
  - DispNetCorr [Mayer+ 2016]
  - GC-Net [Kendall+ 2017]

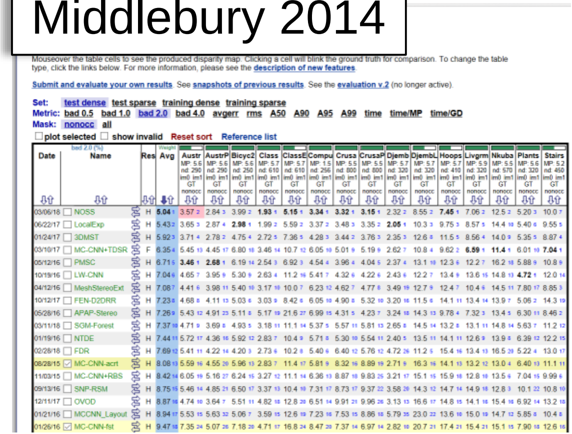


- Return of "CRFs" (Hybrid CNN-CRF models)
  - Seki and Pollefeys 2017, Knobelreiter+ 2017



Stereo Benchmark Rankings

Middlebury 2014



Mouseover the table cells to see the produced disparity map. Clicking a cell will blink the ground truth for comparison. To change the table type, click the links below. For more information, please see the description of new features.

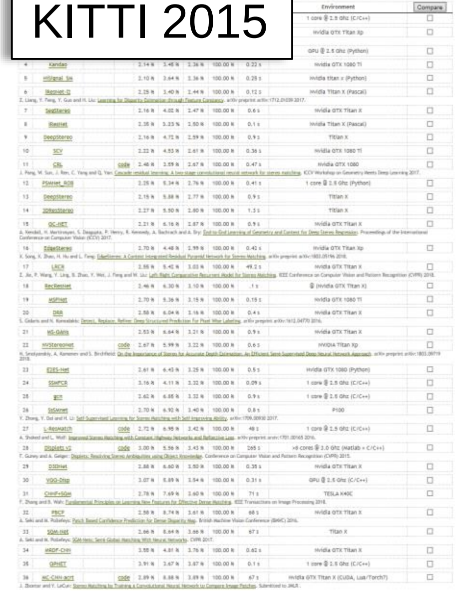
Submit and evaluate your own results. See snapshots of previous results. See the evaluation v.2 (no longer active).

Set: test dense test sparse training dense training sparse
Metric: bad 0.5 bad 1.0 bad 2.0 bad 4.0 avgerr rms A50 A90 A95 A99 time time/MP time/GD
Mask: nonocc all

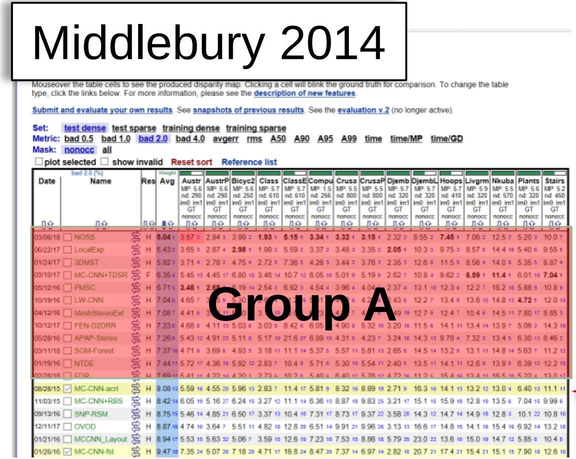
plot selected show invalid Reset sort Reference list

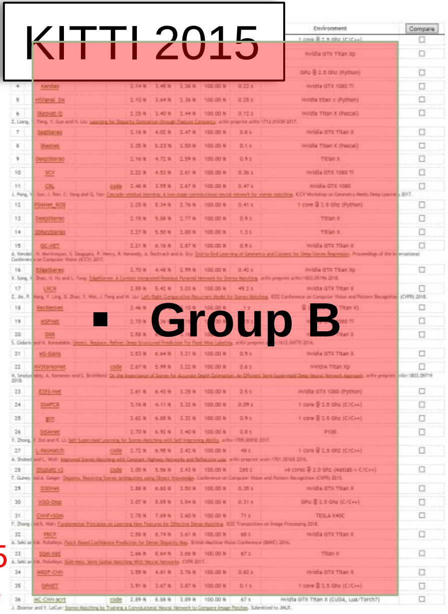
Date	Name (bad 2.0 (%))	Res	Avg (Weight)	Austr MP: 5.6 nd: 290	AustrP MP: 5.6 nd: 290	Bicyc2 MP: 5.6 nd: 250	Class MP: 5.7 nd: 610	ClassE MP: 5.7 nd: 610	Compu MP: 1.5 nd: 256	Crusa MP: 5.5 nd: 800	CrusaP MP: 5.5 nd: 800	Djemb MP: 5.7 nd: 320	DjembL MP: 5.7 nd: 320	Hoops MP: 5.7 nd: 410	Livgrm MP: 5.9 nd: 320	Nkuba MP: 5.5 nd: 570	Plants MP: 5.6 nd: 320	Stairs MP: 5.2 nd: 450
03/06/18	NOSS	H	5.04 1	3.57 2	2.84 3	3.99 2	1.93 1	5.15 1	3.34 1	3.32 1	3.15 1	2.32 2	8.55 2	7.45 1	7.06 2	12.5 2	5.20 3	10.0 7
06/22/17	LocalExp	H	5.43 2	3.65 3	2.87 4	2.98 1	1.99 2	5.59 2	3.37 2	3.48 3	3.35 2	2.05 1	10.3 3	9.75 3	8.57 5	14.4 10	5.40 6	9.55 5
01/24/17	3DMST	H	5.92 3	3.71 4	2.78 2	4.75 4	2.72 5	7.36 5	4.28 3	3.44 2	3.76 3	2.35 3	12.6 8	11.5 5	8.56 4	14.0 9	5.35 5	8.87 4
03/10/17	MC-CNN+TDSR	F	6.35 4	5.45 13	4.45 17	6.80 18	3.46 14	10.7 12	6.05 10	5.01 9	5.19 9	2.62 7	10.8 4	9.62 2	6.59 1	11.4 1	6.01 10	7.04 1
05/12/16	PMSC	H	6.71 5	3.46 1	2.68 1	6.19 14	2.54 3	6.92 3	4.54 4	3.96 4	4.04 5	2.37 4	13.1 10	12.3 6	12.2 7	16.2 18	5.88 9	10.8 9
10/19/16	LW-CNN	H	7.04 6	4.65 7	3.95 9	5.30 9	2.63 4	11.2 16	5.41 7	4.32 6	4.22 6	2.43 6	12.2 7	13.4 9	13.6 15	14.8 13	4.72 1	12.0 14
04/12/16	MeshStereoExt	H	7.08 7	4.41 6	3.98 11	5.40 10	3.17 10	10.0 7	6.23 12	4.62 7	4.77 8	3.49 19	12.7 9	12.4 7	10.4 6	14.5 11	7.80 17	8.85 3
10/12/17	FEN-D2DRR	H	7.23 8	4.68 8	4.11 13	5.03 6	3.03 9	8.42 6	6.05 10	4.90 8	5.32 10	3.20 16	11.5 6	14.1 11	13.4 14	13.9 7	5.06 2	14.3 19
05/28/16	APAP-Stereo	H	7.26 9	5.43 12	4.91 23	5.11 8	5.17 19	21.6 27	6.99 15	4.31 5	4.23 7	3.24 18	14.3 13	9.78 4	7.32 3	13.4 5	6.30 11	8.46 2
03/11/18	SGM-Forest	H	7.37 10	4.71 9	3.69 8	4.93 5	3.18 11	11.1 14	5.37 5	5.57 11	5.81 13	2.65 8	14.5 14	13.2 8	13.1 11	14.8 14	5.63 7	11.2 12
01/19/16	NTDE	H	7.44 11	5.72 17	4.36 16	5.92 12	2.83 7	10.4 9	5.71 8	5.30 10	5.54 11	2.40 5	13.5 11	14.1 11	12.6 9	13.9 8	6.39 12	12.2 15
02/28/18	FDR	H	7.69 12	5.41 11	4.22 14	4.20 3	2.73 6	10.2 8	5.40 6	6.40 12	5.76 12	4.72 26	11.2 5	15.4 16	13.4 13	16.5 20	5.22 4	13.0 17
08/28/15	MC-CNN-acrt	H	8.08 13	5.59 16	4.55 20	5.96 13	2.83 7	11.4 17	5.81 9	8.32 16	8.89 19	2.71 9	16.3 16	14.1 13	13.2 12	13.0 4	6.40 13	11.1 11
11/03/15	MC-CNN+RBS	H	8.42 14	6.05 19	5.16 27	6.24 15	3.27 12	11.1 14	6.36 13	8.87 18	9.83 25	3.21 17	15.1 15	15.9 18	12.8 10	13.5 6	7.04 15	9.99 6
09/13/16	SNP-RSM	H	8.75 15	5.46 14	4.85 21	6.50 17	3.37 13	10.4 10	7.31 17	8.73 17	9.37 22	3.58 20	14.3 12	14.7 14	14.9 18	12.8 3	10.1 22	10.8 10
12/11/17	OVOD	H	8.87 16	4.74 10	3.64 7	5.51 11	4.82 18	12.8 20	6.51 14	9.91 21	9.96 26	3.13 13	16.6 17	14.8 15	14.1 16	15.4 16	6.92 14	13.2 18
01/21/16	MCCNN_Layout	H	8.94 17	5.53 15	5.63 32	5.06 7	3.59 15	12.6 19	7.23 16	7.53 15	8.86 18	5.79 35	23.0 22	13.6 10	15.0 19	14.7 12	5.85 8	10.4 8
01/26/16	MC-CNN-fst	H	9.47 18	7.35 24	5.07 26	7.18 20	4.71 17	16.8 24	8.47 20	7.37 14	6.97 14	2.82 10	20.7 21	17.4 21	15.4 21	15.1 15	7.90 18	12.6 16

KITTI 2015



Stereo Benchmark Rankings





- Group A and B have no methods in common!
- Group A entries all use MC-CNN acrt but no other “deep learning” technique!
- Group B methods do NOT use MC-CNN acrt; they use ResNet, 3D convolutions, 3D deconvolutions, U-shaped Nets, RNNs; End to end learning is very popular!

Conclusions

Must train **one** model on combined training set and submit to all benchmarks!

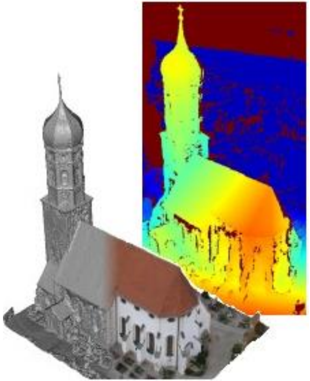
	Stereo	MVS	Flow	Depth	Semantic	Instance
Middlebury	✗	✗	✗			
KITTI	✗		✗	✗	✗	✗
MPI Sintel			✗			
ETH3D	✗	✗				
HD1K			✗			
ScanNet				✗	✗	✗
Cityscapes					✗	✗
WildDash					✗	✗

Outline

- Stereo Matching: New Trends
- **Semi-Global Stereo Matching (SGM)**
  - SGM with Surface Orientation Priors
- Stereo Scene Flow with Motion Segmentation
- Trajectory Planning for Aerial Multi View Stereo
- Deep 6D Object Pose Prediction

Semi Global Matching [Hirschmüller 2005]

- MRF inference (Graph Cuts, BP, ..) too slow
- SGM: Approximate even more; use heuristics
  - Widely used: assisted driving, robotics, aerial mapping …
  - Runs in real-time on FPGAs, GPUs …

Scanline Optimization (1D)

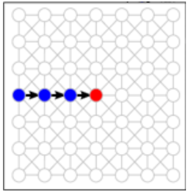
Minimize:

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

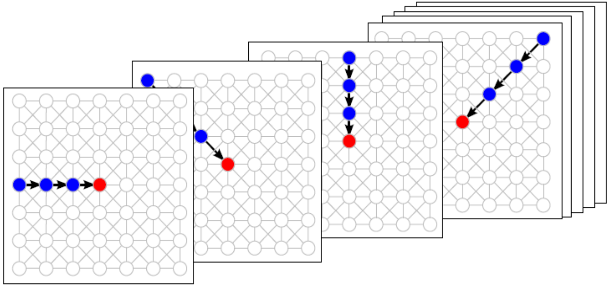
- Consider the above problem on a 1D scanline.
- Compute an aggregated matching cost

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

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



Semi Global Matching (SGM)



- For 8 directions
  - calculate aggregated costs

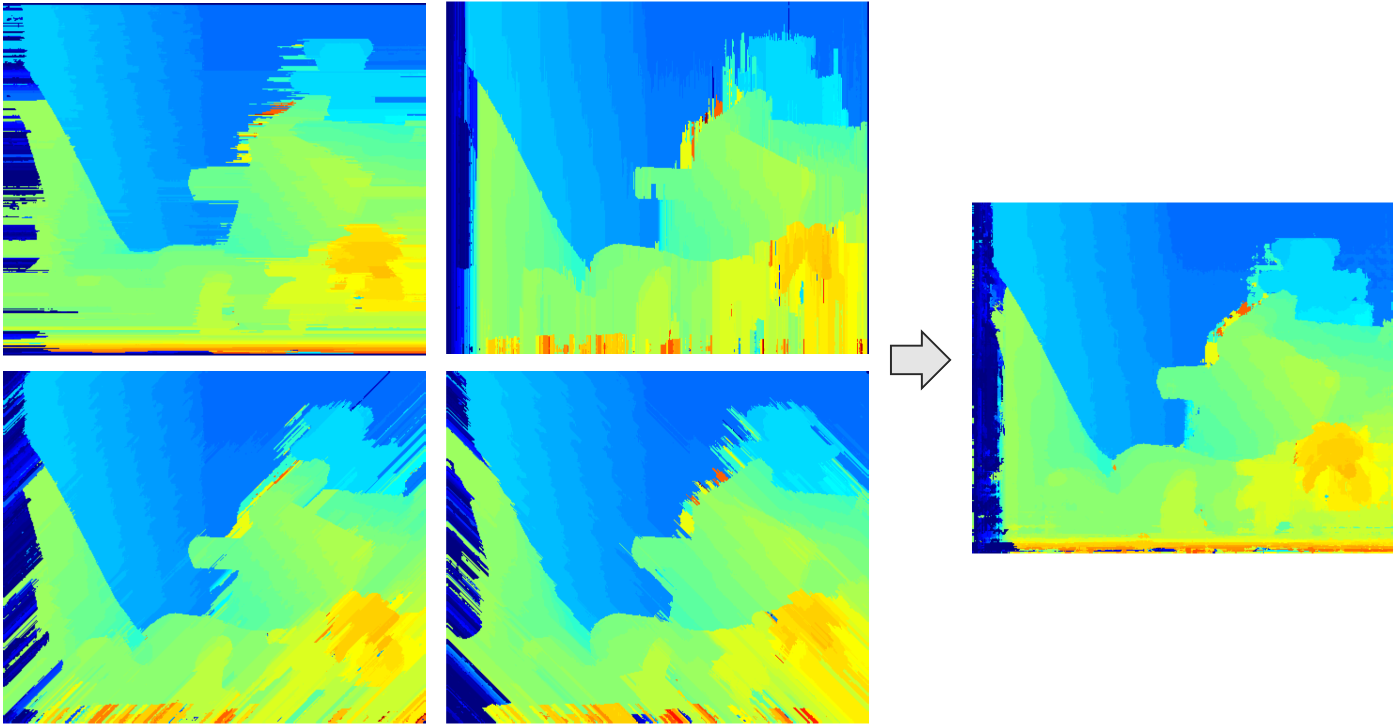
$$L_{\mathbf{r}}(\mathbf{p}, d) = C_{\mathbf{p}}(d) + \min_{d' \in \mathcal{D}} (L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d') + V(d, d')).$$

- 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)

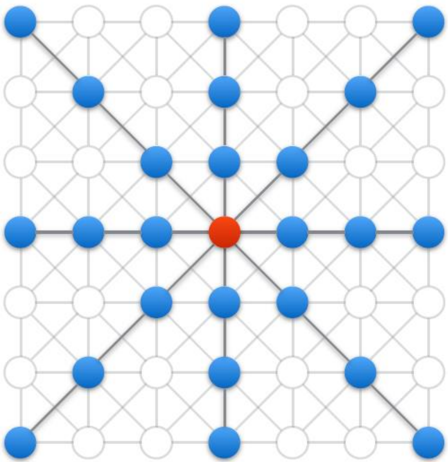


Semi Global Matching [Hirschmüller 2005]

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, TRW-S

  [Drory et al. 2014]

Semi Global Stereo Matching with Surface Orientation Priors

3DV 2017

Daniel Scharstein
Middlebury College

Tatsunori Taniai
RIKEN, Tokyo

Sudipta Sinha
Microsoft Research

Semi Global Matching [Hirschmüller 2005]

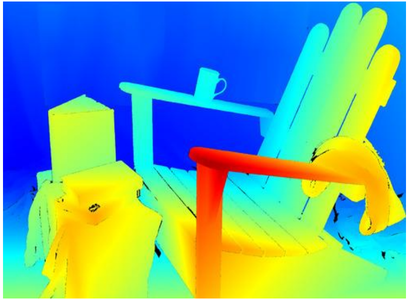
Approximates 2D MRF using 1D optimization along 8 cardinal directions

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

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

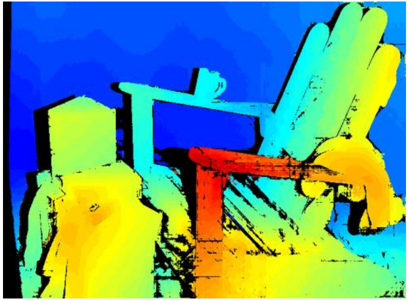
- Fronto parallel bias
- Inaccurate on slanted untextured surfaces

Left Image

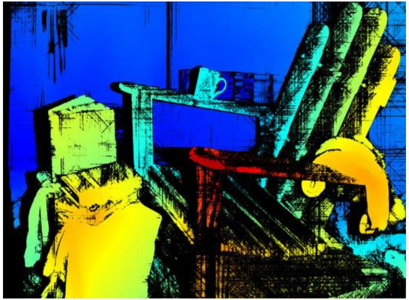


Ground Truth

SGM* (quarter resolution)



SGM* (full resolution (6 MP))



* Confidence measure used to prune uncertain pixels (black holes)

SGM-P: SGM with orientation priors

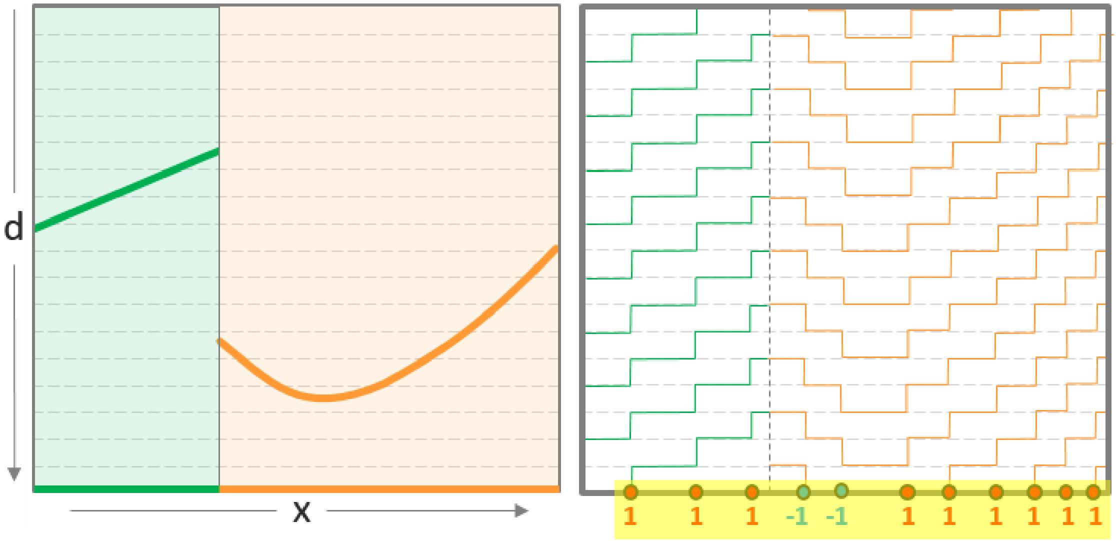
[Scharstein, Taniai, Sinha, 3DV 2017]

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

Idea:

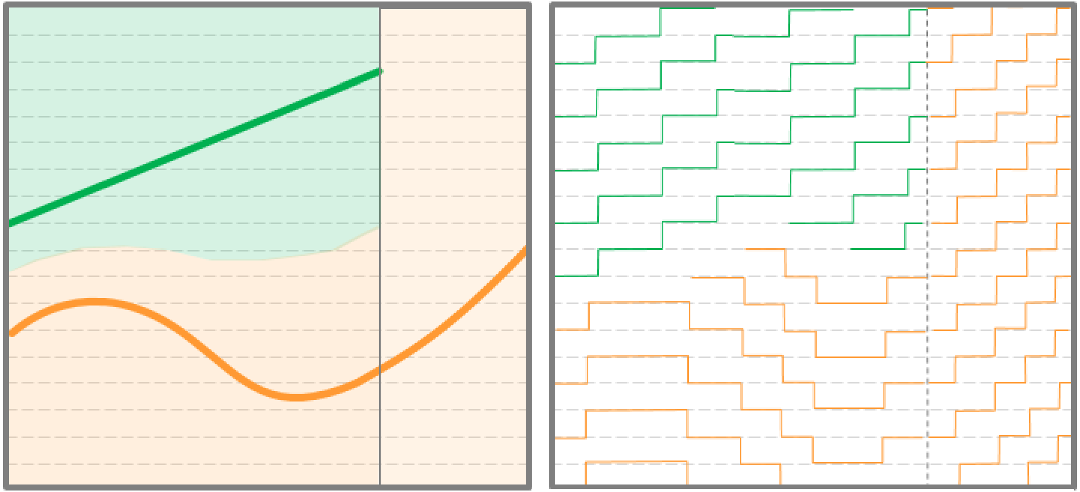
- *Rasterize* disparity surface prior (at arbitrary depth)
- Adjust $V(d, d')$ to follow discrete disparity "steps"

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

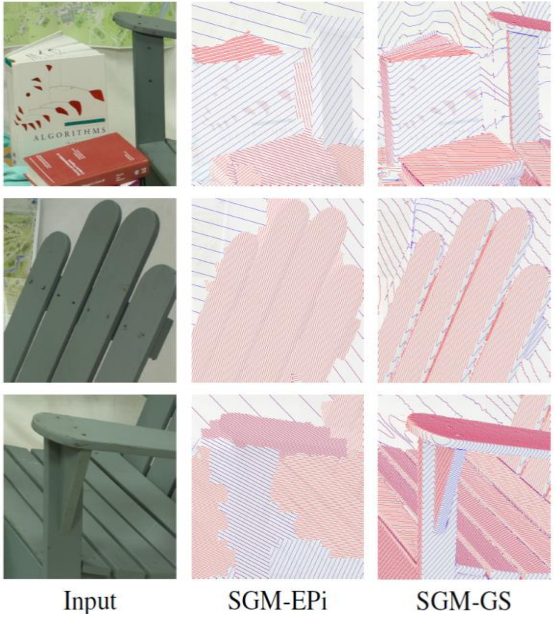


$$V_S(d_{\mathbf{p}}, d'_{\mathbf{p}}) = V(d_{\mathbf{p}} + j_{\mathbf{p}}(d_{\mathbf{p}}), d'_{\mathbf{p}})$$

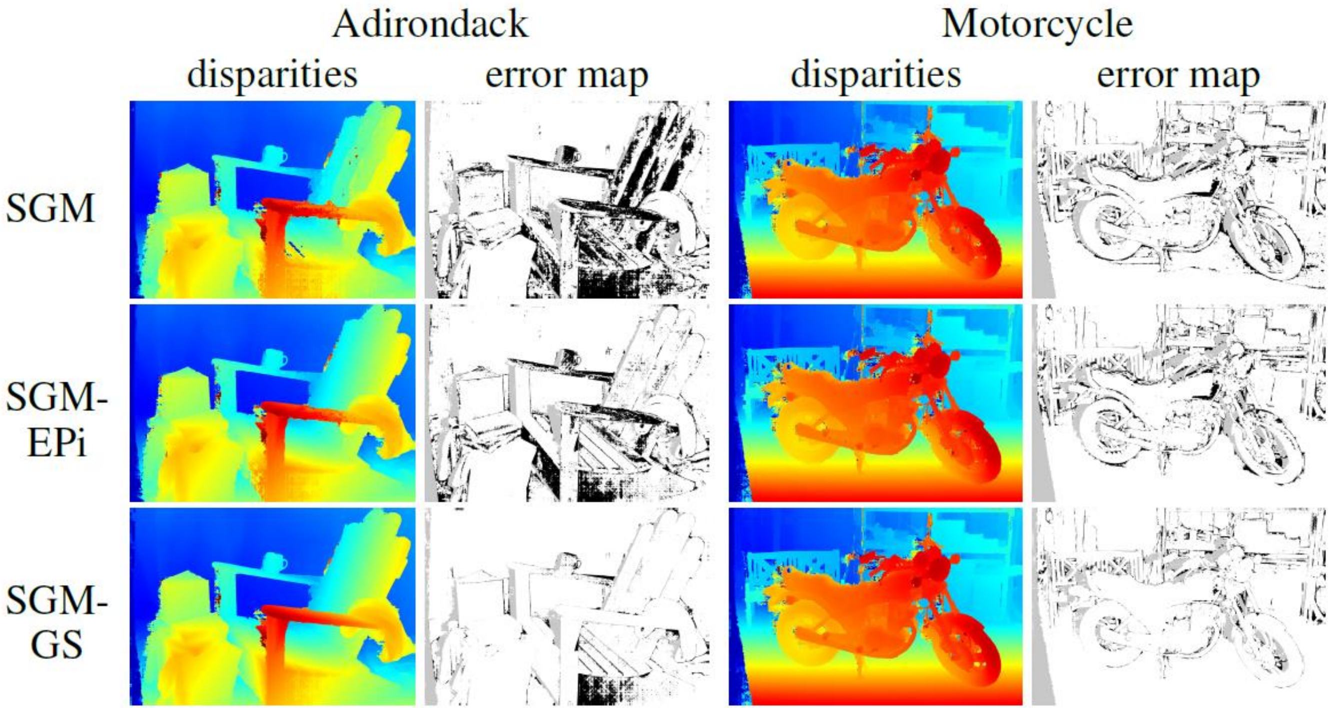
Jump locations
vary with disparity

SGM-P: Where do we get priors?

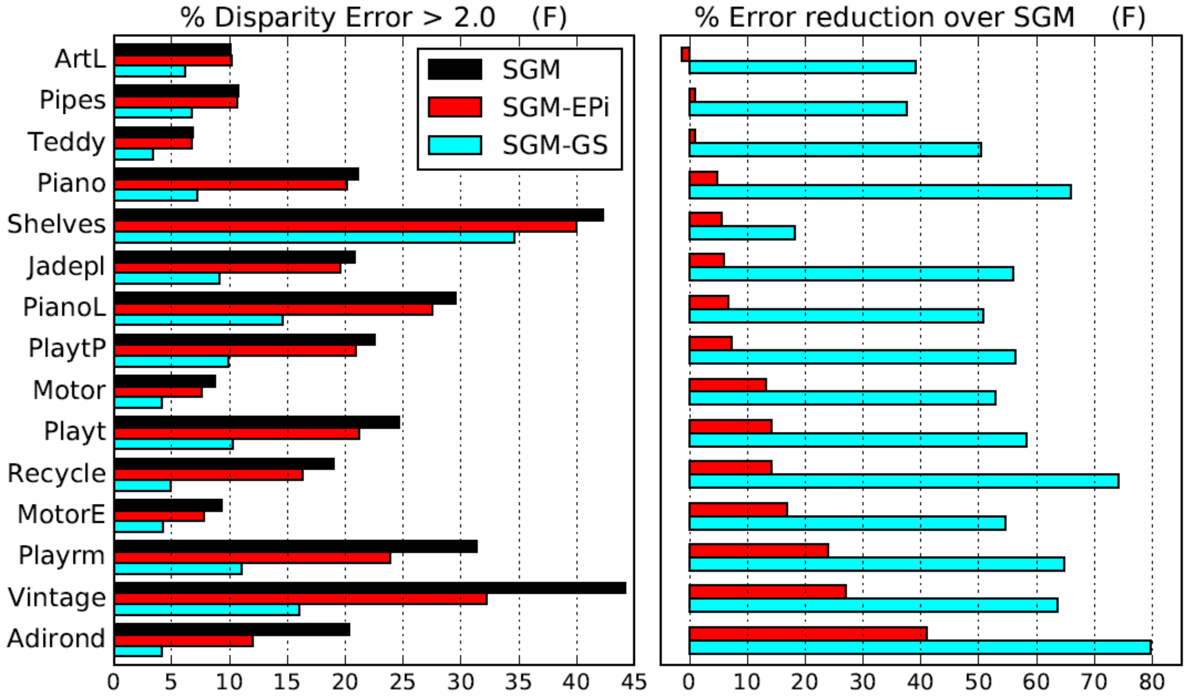
- Matched features + triangulation
- Matched features + plane fitting
- Low-res matching + plane fitting
- Ground truth oracle
- Semantic analysis
- Manhattan-world assumptions



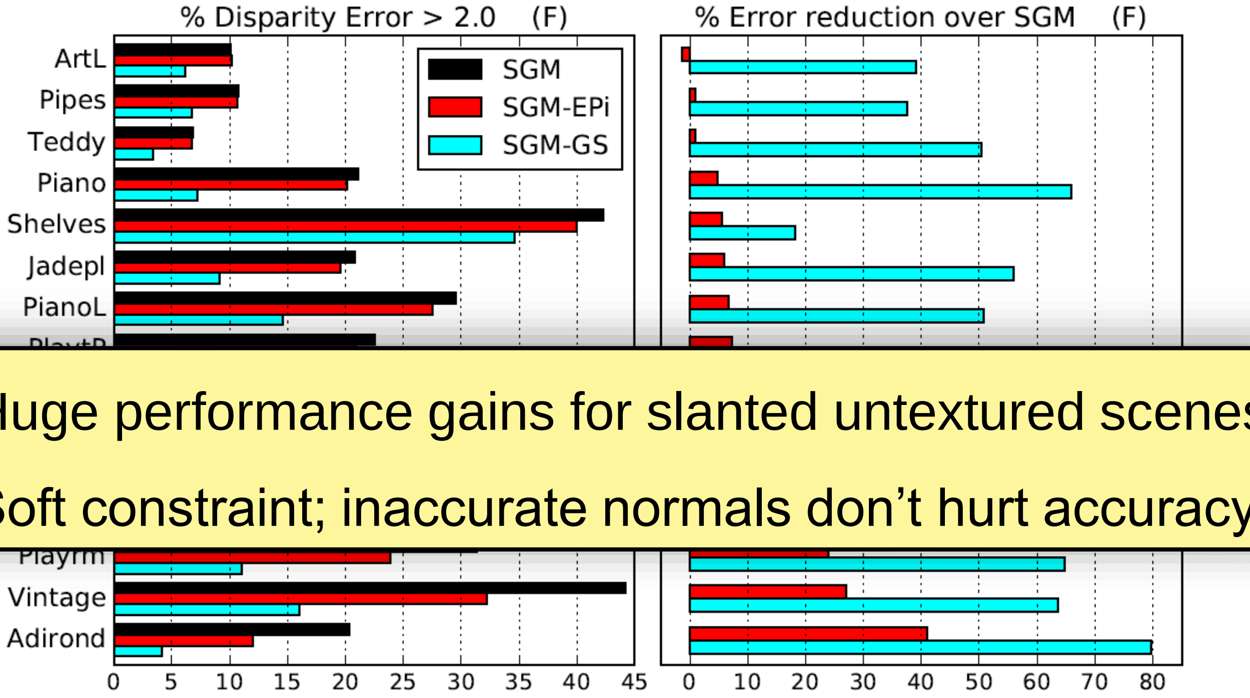
SGM-P: Results



SGM-P: Results



SGM-P: Results



- Huge performance gains for slanted untextured scenes
- Soft constraint; inaccurate normals don’t hurt accuracy

Outline

- Stereo Matching: New Trends
- Semi-Global Stereo Matching (SGM)
  - SGM with Surface Orientation Priors
- Stereo Scene Flow with Motion Segmentation
- Trajectory Planning for Aerial Multi View Stereo
- Deep 6D Object Pose Prediction

Fast Multi-frame Stereo Scene Flow with Motion Segmentation

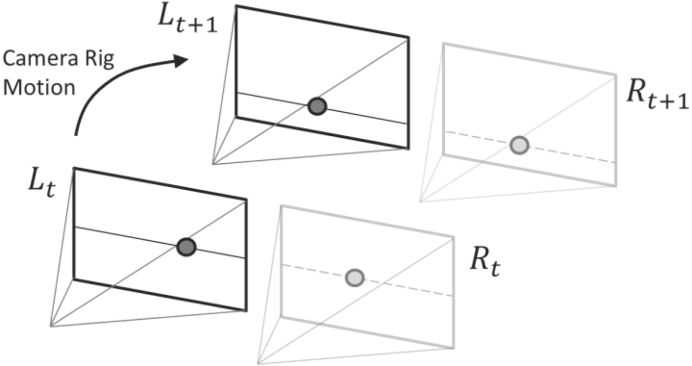
CVPR 2017

Tatsunori Taniai
RIKEN, Tokyo

Sudipta Sinha
Microsoft Research

Yoicho Sato
Univ. of Tokyo

Multi-frame Stereo Scene Flow



- Stereo video from moving stereo camera rig (calibrated)
- Scene Flow equivalent to stereo matching and optical flow estimation

Application

Object scene flow for autonomous vehicles

Menze and Geiger 2015

Action recognition by dense trajectories

Wang+ 2011

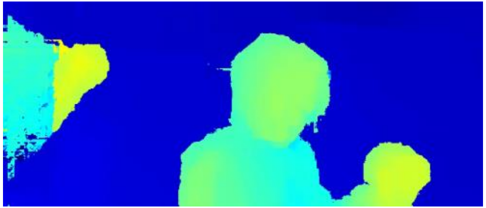
- Depth and flow sequences are useful in many applications

Motivation

Efficient, unified method for

- Stereo
- Optical Flow
- Moving object segmentation
- Visual Odometry (Camera ego-motion)

Disparity Map



Optical Flow

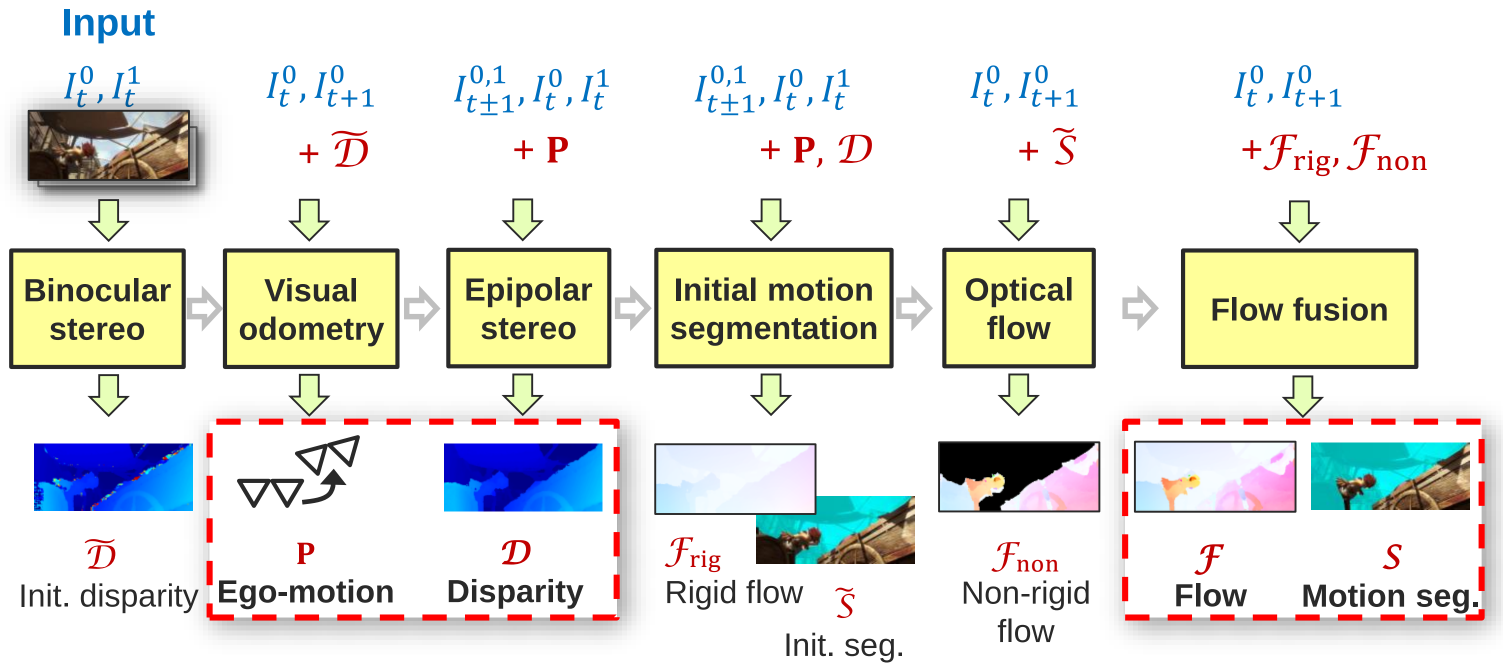
Moving Object Segmentation

Main Idea: Dominant Rigid Scene Assumption

- Most of the scene is rigid; hence, camera motion determines the *rigid optical flow*.
- Given *rigid flow map*, only find regions with moving objects and recompute their flow.

Proposed Approach



Results – KITTI 2015 Scene Flow Benchmark (Nov 2016)

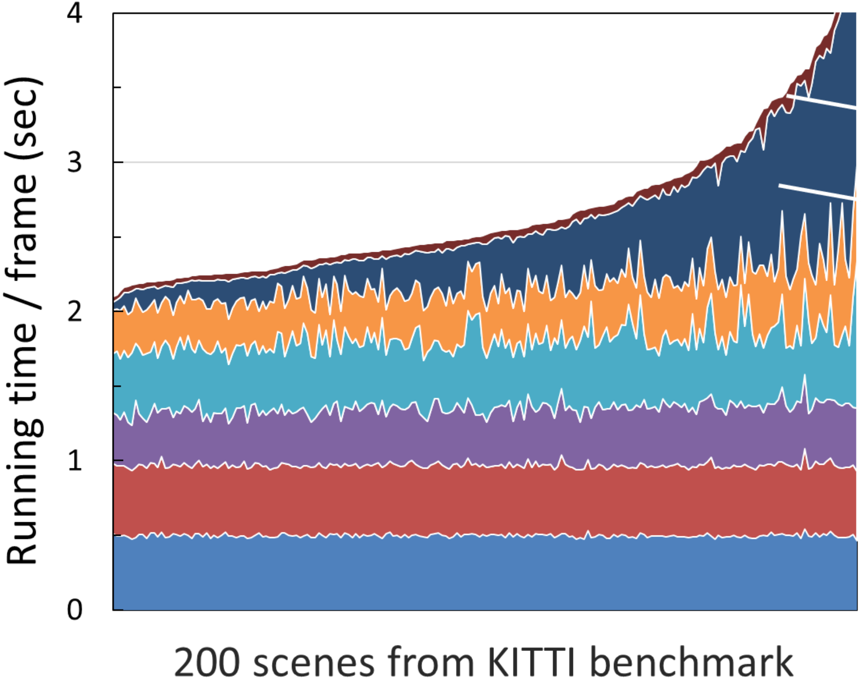
Rank	Method	D1-bg	D1-fg	D1-all	D2-bg	D2-fg	D2-all	Fl-bg	Fl-fg	Fl-all	SF-bg	SF-fg	SF-all	Time
1	PRSM [43]	**3.02**	**10.52**	**4.27**	**5.13**	**15.11**	**6.79**	**5.33**	**17.02**	**7.28**	**6.61**	**23.60**	**9.44**	300 s
2	OSF [30]	4.54	12.03	5.79	5.45	19.41	7.77	5.62	22.17	8.37	7.01	28.76	10.63	50 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	$\boxed{2.7\text{ s}}$
4	16.33	80 s
5	16.85	150 s
8	33.02	0.08 s (GPU)
12	53.95	2.4 s

Results – KITTI 2015 Scene Flow Benchmark (Nov 2016)

CPU: 3.5 GHz × 4 Cores
Image: (1242 × 375) × 0.65 scale



0.07 sec Flow fusion

0.48 sec Optical flow

0.36 sec Initial segmentation

0.47 sec Epipolar stereo

0.38 sec Visual odometry

0.47 sec Binocular stereo

0.72 sec Initialization

2.72 sec per frame

Outline

- Stereo Matching: New Trends
- Semi-Global Stereo Matching (SGM)
  - SGM with Surface Orientation Priors
- Stereo Scene Flow with Motion Segmentation
- Trajectory Planning for Aerial Multi View Stereo
- Deep 6D Object Pose Prediction

Submodular Trajectory Optimization for Aerial 3D Scanning

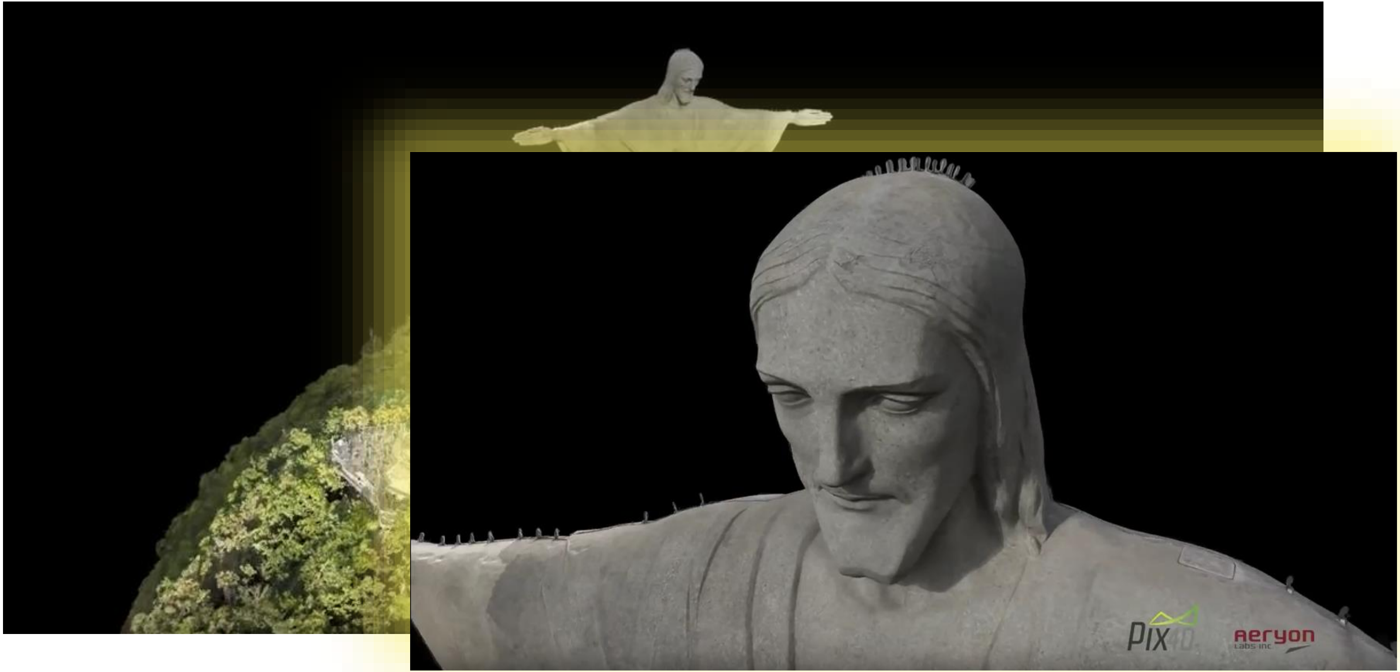
ICCV 2017

Mike Roberts[1,2] Debadeepta Dey[2] Anh Truong[3] Sudipta Sinha[2]
Shital Shah[2] Ashish Kapoor[2] Pat Hanrahan[1] Neel Joshi[2]

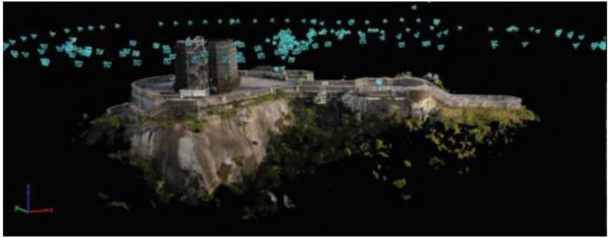
[1]Stanford University [2]Microsoft Research [3]Adobe Research

Acquiring imagery using drones

Multi-view Stereo Reconstruction



Manual Planning Prior to Capture



- Waypoints planned by human experts …
  - Several redundant flight trajectories were flown
- 3,500 images from 6 days with 19 ten-minute flights
- Projeto Redentor (Pix4D whitepaper, 2015)

Our Goal

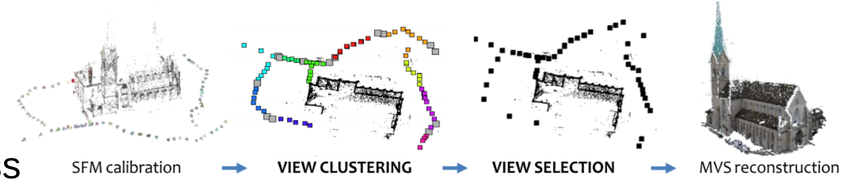
- Automatically generate optimized trajectories for 3D scanning using drones, such that

1. the acquired images will produce an accurate 3D model when processed using a Multi View Stereo (MVS) algorithm.
2. the UAV makes best use of its limited flight time budget.

- Processing happens post flight.
- Battery typically lasts 15—20 minutes.

Related Work

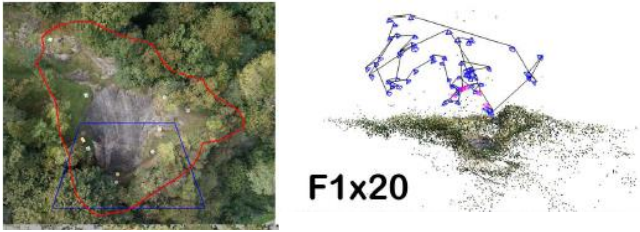
- View selection [Hornung+ 2011]
  - First, acquire dense imagery
  - Later, select subset & process



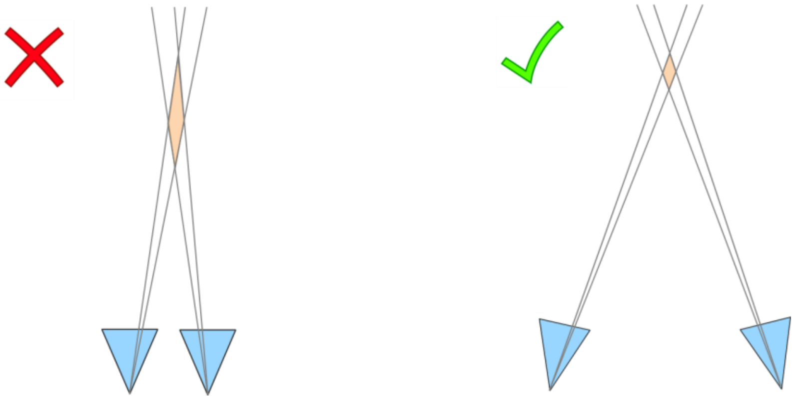
- Next-best-view planning
  - Information-gain maximization [Isler+ 2016]
  - Robotic RGB-D 3D scanning [Wu+ 2014]
  - No travel budget constraints

- Real-time Drone view planning

  [Mostegel+ 2016]
  - Greedy technique; heuristic-based



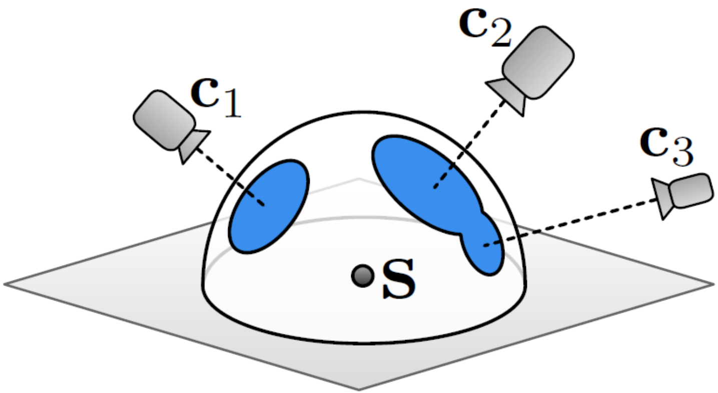
Diverse Viewpoints help Multi View Stereo



Preference for

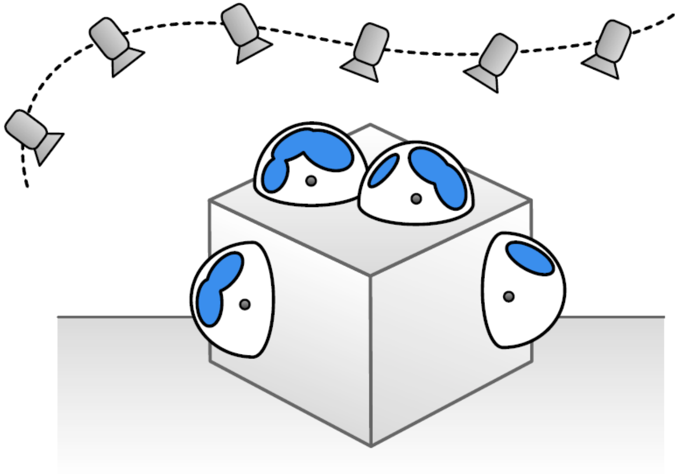
- diverging viewing angles
- close-up views
- fronto parallel views of surfaces

Coverage Measure



For a surface point S observed from multiple cameras, we define coverage as the area of the union of all the blue disks on a hemisphere.

Coverage Measure



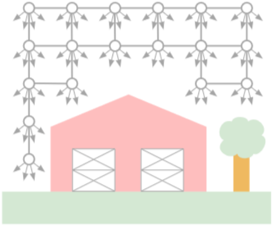
Similarly, we define coverage for multiple surface points observed from multiple camera viewpoints.

Method

- Evaluating coverage function requires knowledge of scene geometry
- Thus, we follow a two-staged procedure.

1. Fly an easy-to-generate trajectory;
2. Compute coarse reconstruction (SFM → MVS → meshing)
3. Plan *optimized* trajectory based on mesh from step 2.
4. Fly trajectory computed in step 3.
5. Run SFM + MVS on images from step 1 and 4.

Planning Optimized Trajectories



Graph of all possible camera location (and orientation); edge weights are Euclidean distances between locations.

Propose to solve the problem in two steps.

1. Solve optimal set of orientations; ignoring path constraints
2. Then, find the set of locations by solving a *graph orienteering problem*

Solving for Camera Orientations

- Coverage *set* function is submodular
  - Adding new elements to an existing set gives diminishing returns
- Cardinality and Mutual Exclusion Constraint
  - Select exactly one look-at vector at each position

- Constrained submodular maximization
  - Always, pick the next best element with the most marginal reward
  - Greedy algorithm; good theoretical approximation guarantee

Solving for Camera Positions

- Graph Orienteering Problem
  - NP-Hard; related to TSP and Knapsack
  - Find short paths that let you collect most rewards (at nodes).
- In standard orienteering, rewards are additive.
- But, our reward function is submodular, not additive!
- Hence, we must solve a *submodular orienteering problem.*

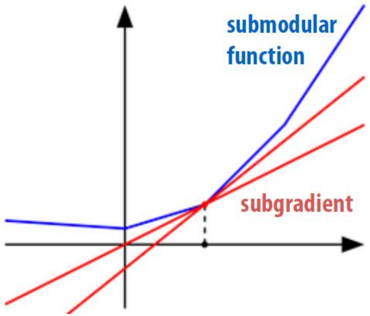
(c) Coarsened problem (d) Additive approximation to the coarsened problem (e) Solve for the optimal closed path on the additive approximation

Solving for Camera Positions

- Choose a *good* sub-gradient

  (additive approximation)

  for our submodular function



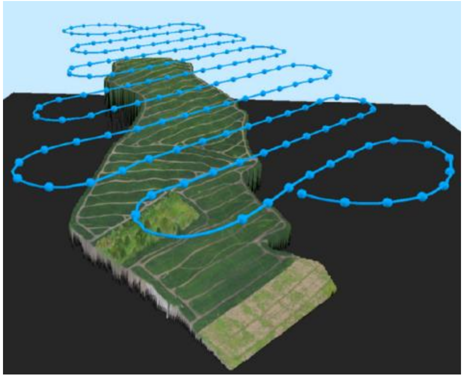
- Approximation yields an instance of the orienteering problem

$$\begin{aligned} &\text{maximize} \quad \sum c(s) \\ &\text{subject to} \quad T(S) < B \end{aligned}$$

- Solve as an integer linear program (ILP)

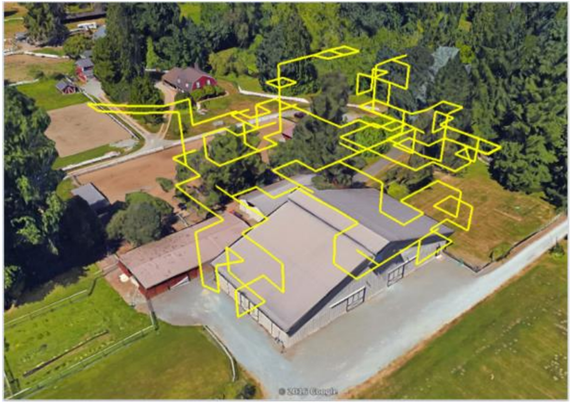
Results

- Pix4D for 3D reconstruction
  - Outputs texture-mapped 3D model
- Baselines:
  - **Overview:** Lawn-mower pattern
  - **Random:**
    - recover coarse 3d model; estimate free space.
    - select random points in free space.
    - compute TSP tour.

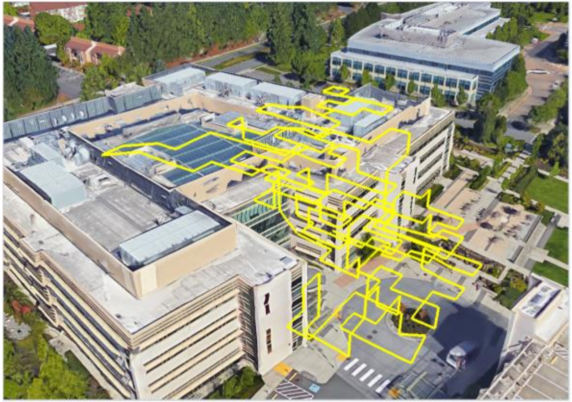


Results

Our computed trajectories visualized in Google Earth



Barn



MSR Redmond

Results

Insert video (ICCV supplementary video here)

Outline

- Stereo Matching: New Trends
- Semi-Global Stereo Matching (SGM)
  - SGM with Surface Orientation Priors
- Stereo Scene Flow with Motion Segmentation
- Trajectory Planning for Aerial Multi View Stereo
- Deep 6D Object Pose Prediction

Real-Time Seamless Single Shot 6D Object Pose Prediction

CVPR 2018

Bugra Tekin
EPFL

Sudipta Sinha
Microsoft Research

Pascal Fua
EPFL

3D Recognition, 2D-3D Model Alignment

Given a RGB image (with known intrinsics), recognize the objects and predict their 3D position and orientation within the scene.

<u>Classical methods:</u>

Lowe 2001

Rothganger+ 2005

Lepetit+ 2005

- Recognizing Image Patches
- Scale, Affine invariant features
- Geometric verification (rigid scenes)

- Worked for textured, distinctive objects
- Required a small # of training images

Object 6D Pose Estimation

Given a RGB image (with known intrinsics), recognize the objects and predict their 3D position and orientation within the scene.

RGB-D methods:

- Lai+ 2010
- Hinterstoisser+ 2012
- Brachmann+ 2014, 2016

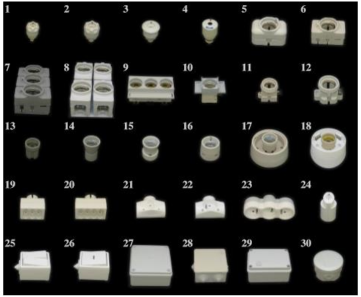
- Classical Object Recognition
- Fast Image Retrieval

CNN methods:

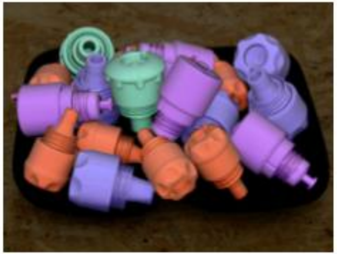
- Rad + Lepetit 2017
- Kehl+ 2017
- Xiang+ 2017
- Tekin+ 2018
- Oberweger+ 2018

- Global deep feature representations
- Not much use of geometry
- Promising for 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

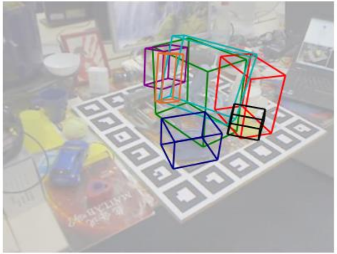
- BB8 [Rad and Lepetit 2017]

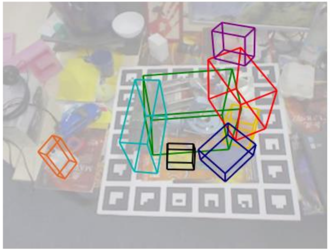
image ⇨ $\boxed{\text{CNN}_1}$ ⇨ cropped image ⇨ $\boxed{\text{CNN}_2}$ ⇨ $\boxed{\text{CNN}_3}$ ⇨ 6D pose

CNN_1	CNN_2	CNN_3
2d detector	*pose estimator*	*pose refinement*

Our Method

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

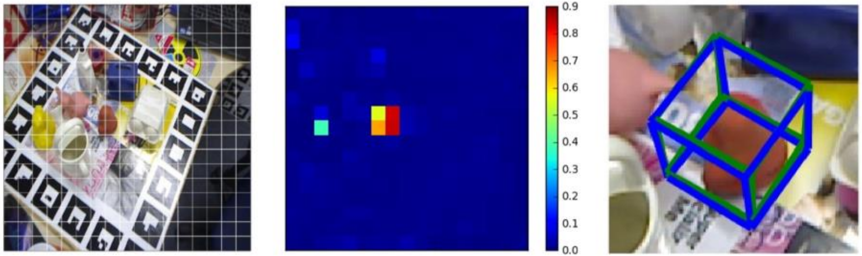





Our Method

Training:

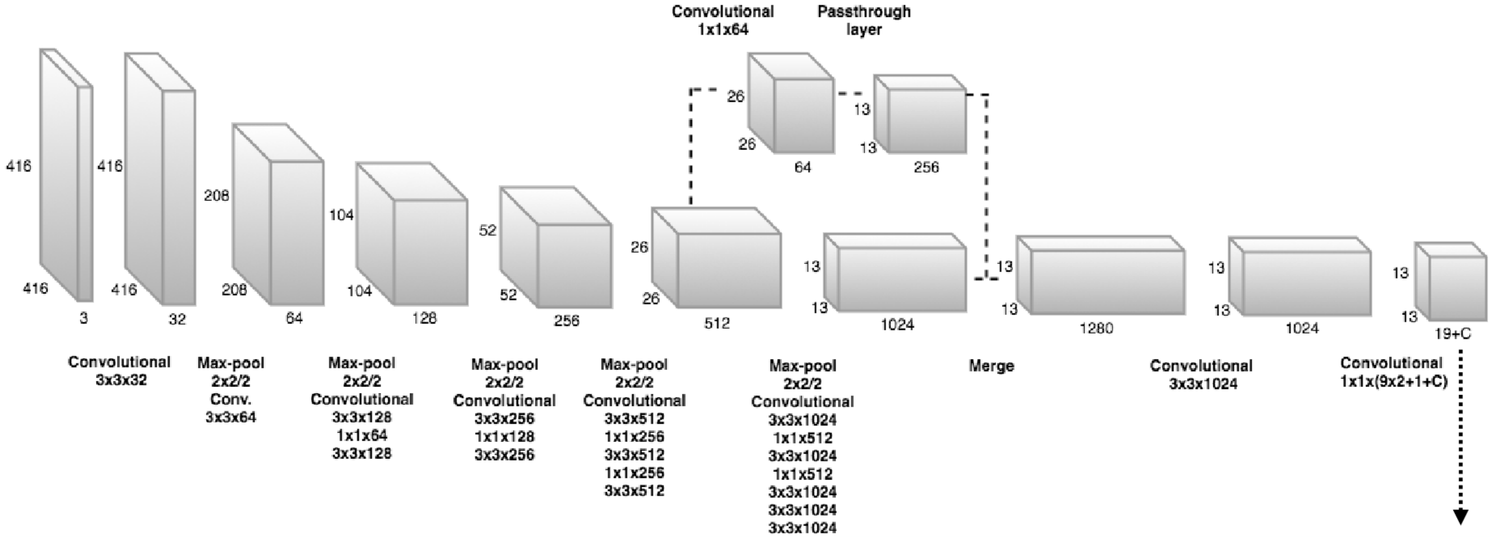
- ground truth 2D coordinates of the 9 control points are the targets
- modify YOLO loss function (for confidence estimation)
- data augmentation



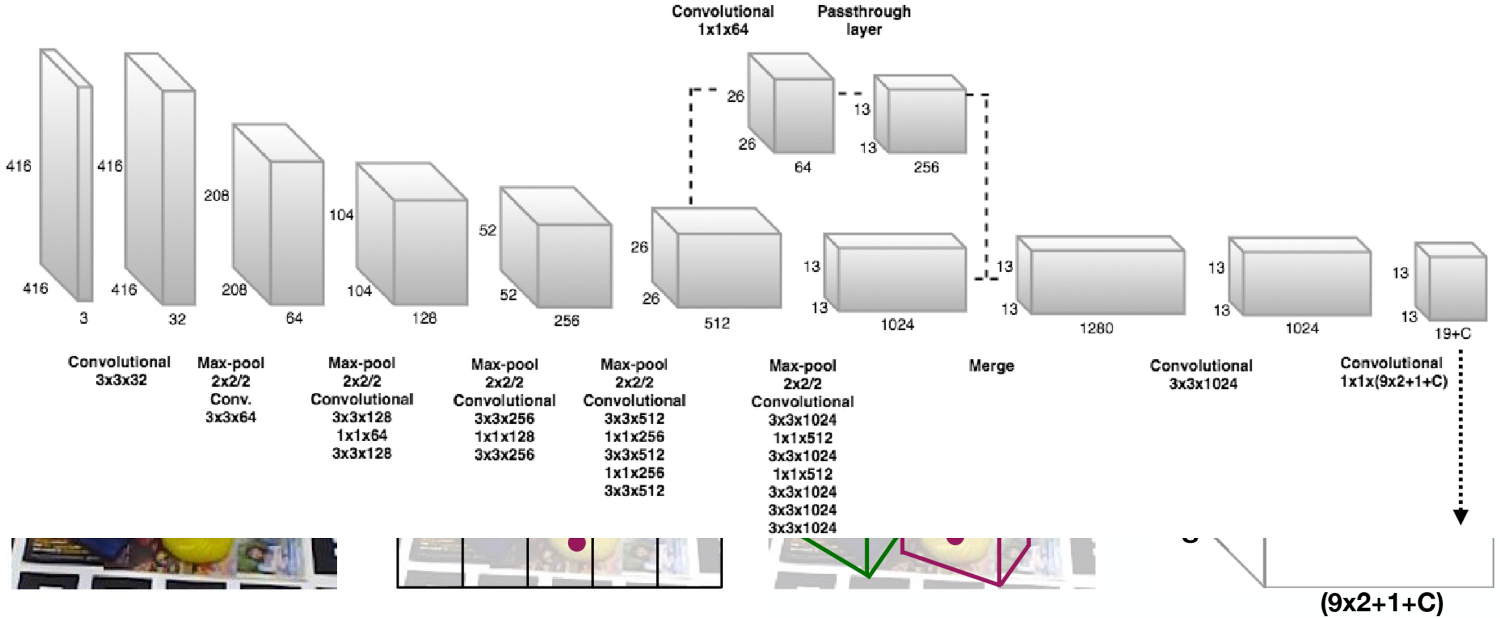
Testing:

- Subpixel refinement
- PnP (RANSAC, least squares)

CNN Architecture



CNN Architecture



Results on LineMOD dataset

- Two accuracy metrics (2D image projection, 3D model overlap).
- Percentage of test images where the error was lower than specified thresholds.

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

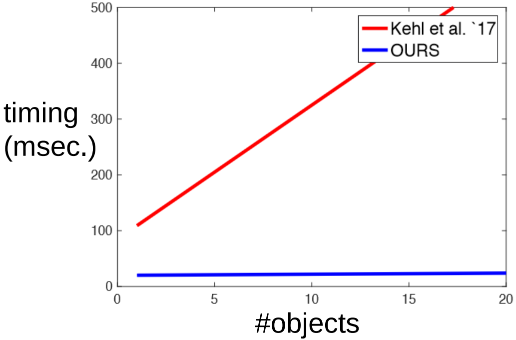
3D metric

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**

Results on LineMOD dataset

- 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

Conclusions

- State of the art in stereo matching; new challenges
- Improvements to Semi Global Matching
  - Incorporating soft surface orientation priors
- Fast scene flow with motion segmentation
- Camera path planning for improved multi-view stereo
- Deep single shot 6D object pose estimation
  - CNN architecture conceptually simpler (~YOLO architecture) and faster