Semantic Structure from Motion

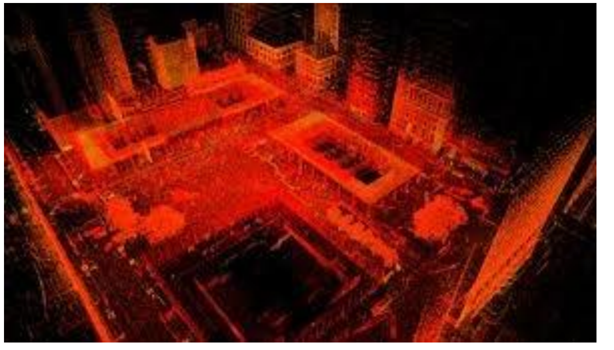
Paper by: Sid Yingze Bao and Silvio Savarese

Presentation by: Ian Lenz

Inferring 3D

With special hardware:

- Range sensor
- Stereo camera



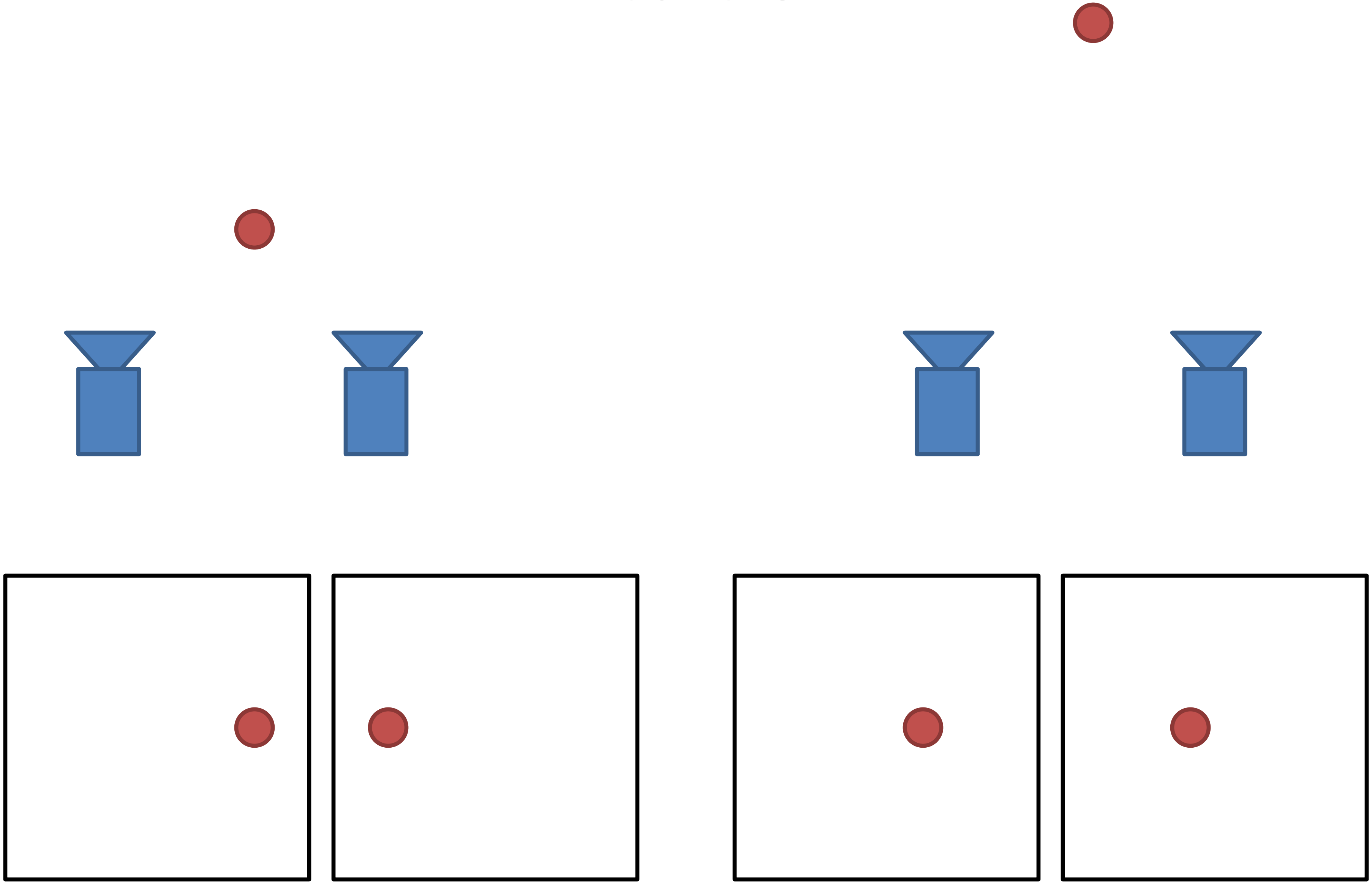
Without special hardware:

- Local features/graphical models (Make3D, etc)
- Structure from motion

Structure from Motion

- Obtain 3D scene structure from multiple images from the same camera in different locations, poses
- Typically, camera location & pose treated as unknowns
- Track points across frames, infer camera pose & scene structure from correspondences

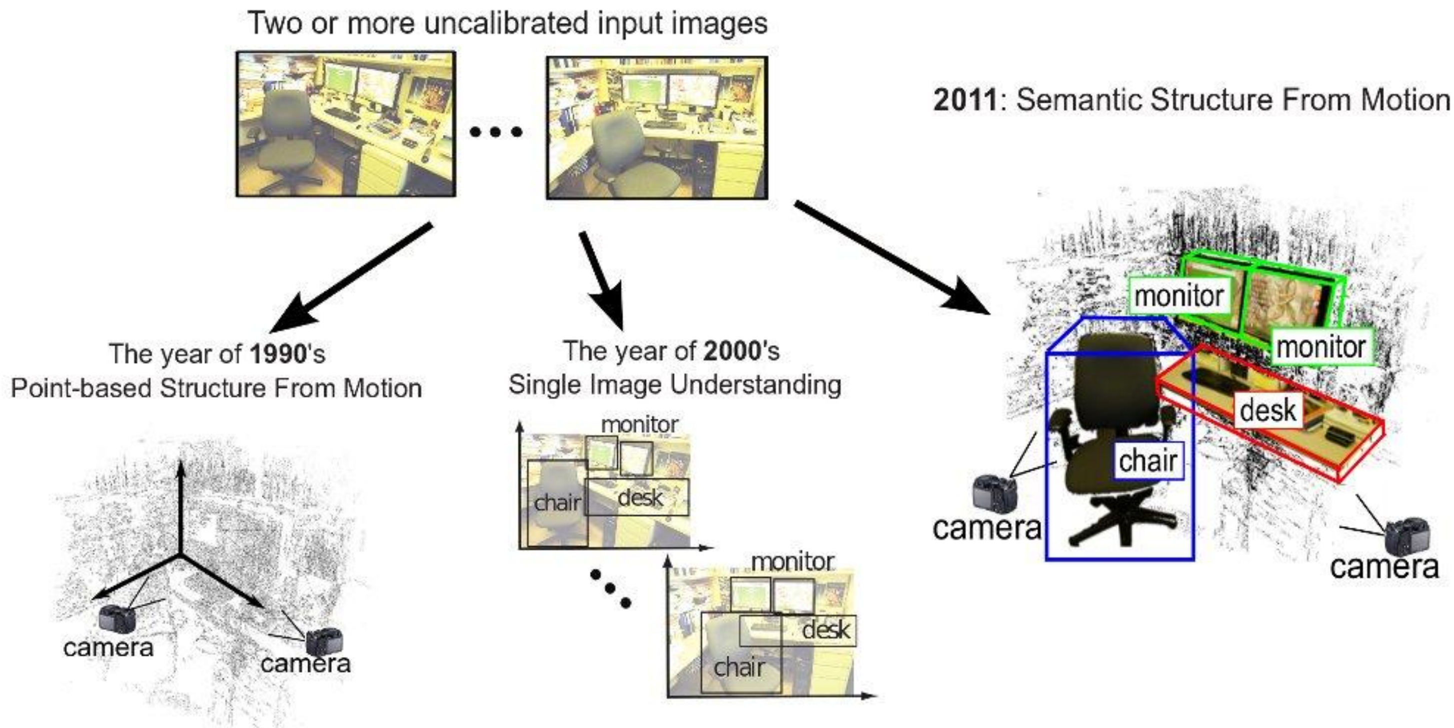
Intuition



Typical Approaches

- Fit model of 3D points + camera positions to 2D points
- Use point matches (e.g. SIFT, etc.)
- Use RANSAC or similar to fit models
- Often complicated pipeline
  - "Building Rome in a Day"

Semantic SfM



Semantic SfM

- Use semantic object labels to inform SfM
- Use SfM to inform semantic object labels
- Hopefully, improve results by modeling both together

High-level Approach

- Maximum likelihood estimation
- Given: object detection probabilities at various poses, 2D point correspondences
- Model probability of observed images given inferred parameters
- Use Markov Chain Monte Carlo to maximize

$$\{\mathbf{Q}, \mathbf{O}, \mathbf{C}\} = \arg\max_{Q,O,C} \Pr(\mathbf{q}, \mathbf{u}, \mathbf{o} | \mathbf{Q}, \mathbf{O}, \mathbf{C})$$

Model Parameters

C: camera parameters

C^k : parameters for camera k

$C^k = \{K^k, R^k, T^k\}$

K: internal camera parameters – **known**

R: camera rotation – **unknown**

T: camera translation - **unknown**

Model Parameters

q: 2D points

q^k_i : ith point in camera k

$q^k = \{x, y, a\}^k_i$

x, y : point location

a : visual descriptor (SIFT, etc.)

Known

Model Parameters

Q: 3D points

$Q_s = (X_s, Y_s, Z_s)$

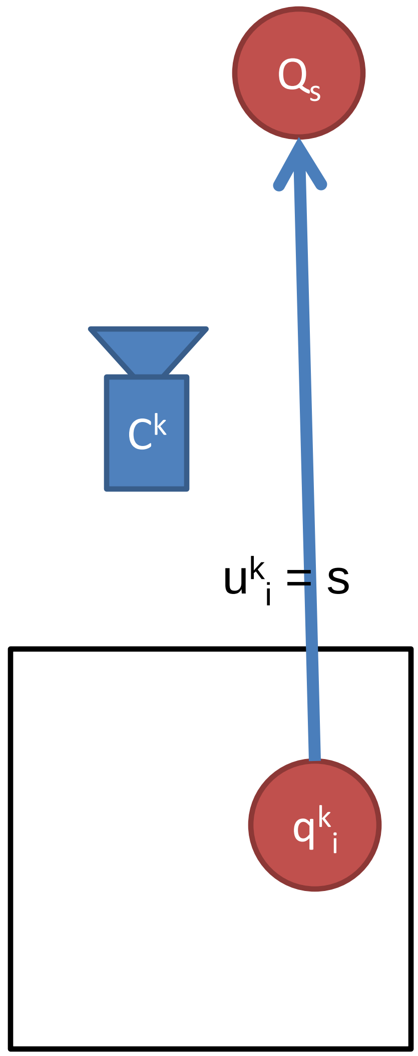
World frame coordinates

Unknown

u: Point correspondences

$u^k_i = s$ if q^k_i corresponds to Q_s

Known



Model Parameters

o: camera-space obstacle detections

o^k_j : jth obstacle detection in camera k

$o^k_j = \{x, y, w, h, \theta, \phi, c\}^k_j$

x, y: 2D location

w, h: bounding box size

θ, ϕ: **3D** pose

c: class (car, person, keyboard, etc.)

Known

Model Parameters

O: 3D objects

O_t = (X, Y, Z, Θ, Φ, c)$_t$

Similar to o except no bounding box, Z coord

Unknown

Likelihood Function

$$\begin{aligned}\{\mathbf{Q},\mathbf{O},\mathbf{C}\} &= \arg\max_{Q,O,C} \Pr(\mathbf{q},\mathbf{u},\mathbf{o}|\mathbf{Q},\mathbf{O},\mathbf{C}) \\ &= \arg\max_{Q,O,C} \Pr(\mathbf{q},\mathbf{u}|\mathbf{Q},\mathbf{C})\Pr(\mathbf{o}|\mathbf{O},\mathbf{C})\end{aligned}$$

Assumption: Points independent from objects

Why?

- Splits likelihood, makes inference easier
- Would require complicated model of object 3D appearance otherwise

Camera parameters appear in both terms

Point Term

$$\Pr(\mathbf{q}, \mathbf{u} | \mathbf{Q}, \mathbf{C})$$

- Compute by measuring agreement between predicted, actual measurements
- Compute predictions by projecting 3D-> cam
- Assume predicted, actual locations vary by Gaussian noise

$$\Pr(q_i^k | Q_s, C^k) \propto \exp(-(q_i^k - q_{u_i^k}^k)^2 / \sigma_q)$$

$$\Pr(\mathbf{q}, \mathbf{u} | \mathbf{Q}, \mathbf{C}) = \prod_i^{N_Q} \prod_k^{N_k} \exp(-(q_i^k - q_{u_i^k}^k)^2 / \sigma_q)$$

Point Term (Alternative)

- Take q^k_i and q^l_j as matching points from cameras C^k and C^l
- Determine epipolar line of q^k_i w/r/t C^l sd
- Take $d^{l,k}_{j,i}$ as the distance from q^l_j to this line

$$\Pr(q^k_i, q^l_j | Q_s, C_l, C_k) \propto \exp(-d^{l,k}_{j,i}/\sigma_u)$$

- Consider appearance similarity: $\exp(-\frac{\alpha(a^k_i, a^l_j)}{\sigma_\alpha})$

$$\begin{aligned}\Pr(\mathbf{q}, \mathbf{u} | \mathbf{Q}, \mathbf{C}) &\propto \prod_{k \neq l}^{N_k} \prod_{i \neq j}^{N_s} \Pr(q^k_i, q^l_j | Q_s, C_l, C_k) \\ &\propto \prod_{k \neq l}^{N_k} \prod_{i \neq j}^{N_s} \exp(-\frac{d^{l,k}_{j,i}}{\sigma_u}) \exp(-\frac{\alpha(a^k_i, a^l_j)}{\sigma_\alpha})\end{aligned}$$

Object Term

$$\Pr(\mathbf{o}|\mathbf{O},\mathbf{C})$$

- Also uses agreement
- Projection more difficult
- Recall: 3D object parameterized by XYZ coords, orientation, class
- 2D also has bounding box params

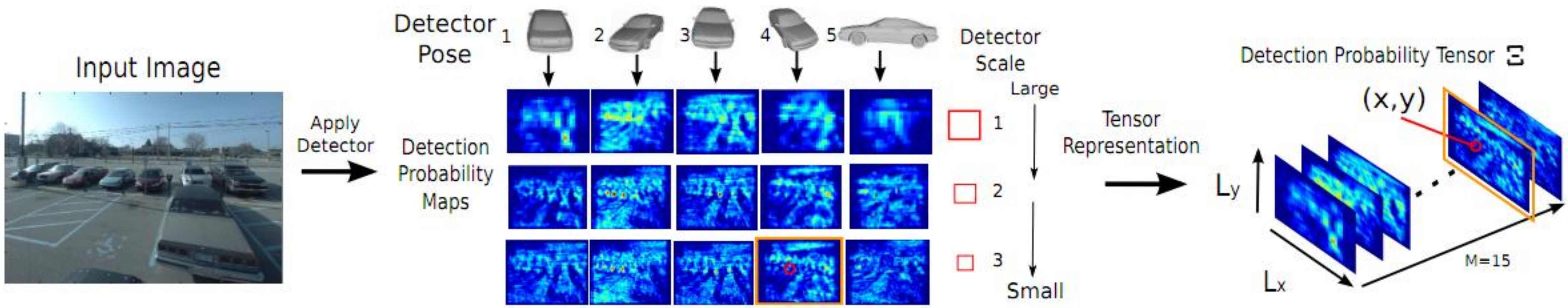
Projecting 3D->2D object

- Location, pose easy using camera params
- For BB width, height:

$$w_t^k = f_k \cdot W(\Theta_t^k, \Phi_t^k, c_t)/Z_t^k$$
$$h_t^k = f_k \cdot H(\Theta_t^k, \Phi_t^k, c_t)/Z_t^k$$

- f_k : camera focal length
- W, H: mapping from object bounding cube to bounding box
- “learned by using ground truth 3D object bounding cubes and corresponding observations using ML regressor”

Object Probability



- Scale proportional to bounding box size
- Highly quantized pose, scale
- Stack maps as tensor, index based on pose, scale
- Tensor denoted as Ξ (Chi)
- Tensor index denoted as $\pi(w_t^k, h_t^k, \phi_t^k, \theta_t^k, c_t^k)$

Object Term

$$\Pr(o|O_t, C^k) = \Xi^k(x_t^k, y_t^k, \pi(w_t^k, h_t^k, \phi_t^k, \theta_t^k, c_t^k))$$

$$\Pr(\mathbf{o}|\mathbf{O}, \mathbf{C}) \propto \prod_t^{N_t} \Pr(\mathbf{o}|O_t, \mathbf{C}) \propto \prod_t^{N_t} (1 - \prod_k^{N_k} (1 - \Pr(o|O_t, C^k)))$$

- Probability of object observation proportional to the probability of not not seeing it in each image (yes a double negative)

Why do it this way?

- Occlusion -> probability of not seeing = 1
- Doesn't affect likelihood term

Estimation

- Have a model, now how do we maximize it?
- Answer: Markov Chain Monte Carlo
- Estimate new params from current ones
- Accept depending on ratio of new/old prob

Two questions remain:

- What are the initial parameters?
- How do we update?

Initialization

Camera location/pose – two approaches:

Point-based:

- Use five-points solver to compute camera parameters from five corresponding points
- Scale ambiguous, so randomly pick several

Object-based:

- Form possible object correspondences between frames, initialize cameras using these

Initialization

Object & point locations:

- Use estimated camera parameters (prev slide)
- Project points, objects from 2D->3D
- Merge objects which get mapped to similar locations
- Determine 2D-3D correspondences (u)

Update

- Order: C, O, Q (updated versions: C’, O’, Q’)
- Pick C’ with Gaussian probability around C
- Pick O’ to maximize Pr(o|O’,C’) (within local area of O)
- Pick Q’ to maximize Pr(q,u|Q’,C’)
  - Unless alternative term was used

Algorithm

Algorithm 1 MCMC sampling from r^{th} initialization. See [1] for details

1: Start with rth proposed initialization $\mathbf{C}_r, \mathbf{O}_r, \mathbf{Q}_r$. Set counter $v = 0$.
2: Propose new camera parameter $\mathbf{C}'$ with Gaussian probability whose mean is the previous sample and the co-variance matrix is uncorrelated.
3: Propose new $\mathbf{O}'$ within the neighborhood of previous object's estimation to maximize $\Pr(\mathbf{o}|\mathbf{O}', \mathbf{C}')$.
4: Propose new $\mathbf{Q}'$ with $\mathbf{C}'$ to minimize the point projection error.
5: Compute the acceptance ratio $\alpha = \frac{\Pr(\mathbf{q},\mathbf{u},\mathbf{o}|\mathbf{C}',\mathbf{O}',\mathbf{Q}')}{\Pr(\mathbf{q},\mathbf{u},\mathbf{o}|\mathbf{C},\mathbf{O},\mathbf{Q})}$
6: If $\alpha \geqslant \varrho$ where ϱ is a uniform random variable $\varrho \sim U(0,1)$, then accept $(\mathbf{C}, \mathbf{O}, \mathbf{Q}) = (\mathbf{C}', \mathbf{O}', \mathbf{Q}')$. Record $(\mathbf{C}, \mathbf{O}, \mathbf{Q})$ as a sample in $\{\mathbf{C}, \mathbf{O}, \mathbf{Q}\}_r$.
7: $v = v + 1$. Goto 2 if v is smaller than the predefined max sample number; otherwise return $\{\mathbf{C}, \mathbf{O}, \mathbf{Q}\}_r$ and end.

Obtaining Results

- Intuition: MCMC visit probability proportional to probability function (what we're trying to maximize)
- Cluster MCMC points using MeanShift
- Cluster with most corresponding samples wins
- Read out Q, O, C as average from cluster

Results

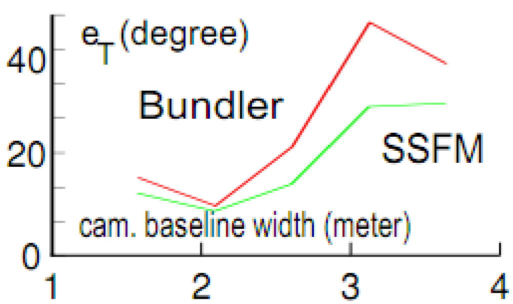
- http://www.eecs.umich.edu/vision/projects/ssfm/index.html

Results vs. Bundler

Dataset	$\bar{e}_T$ Bundler/SSFM	$\bar{e}_R$ Bundler/SSFM
Ford Campus Car	$26.5/\mathbf{19.9}^\circ$	$\mathbf{0.47}^\circ/0.78^\circ$
Street Pedestrian	$27.1^\circ/\mathbf{17.6}^\circ$	$21.1^\circ/\mathbf{3.1}^\circ$
Office Desktop	$8.5^\circ/\mathbf{4.7}^\circ$	$9.6^\circ/\mathbf{4.2}^\circ$

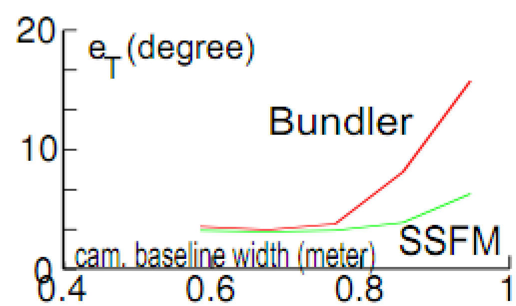
Table 1: Evaluation of camera pose estimation for two camera case. $\bar{e_T}$ represents the mean of the camera translation estimation error, and $\bar{e}_R$ the mean of the camera rotation estimation error.

Cars



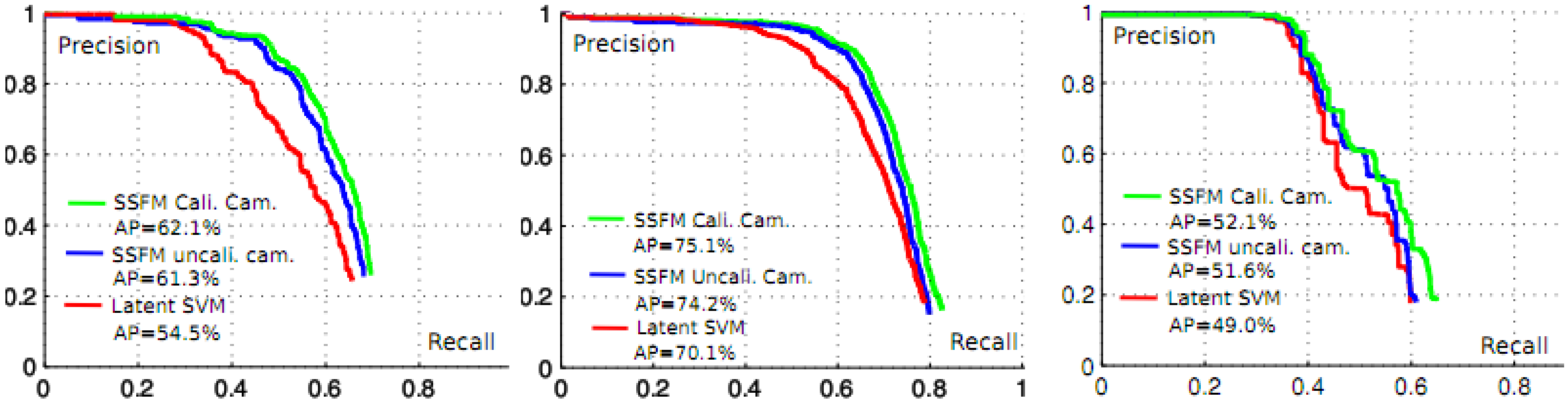
(a) T est. error v.s. the baseline

Office



(c) T est. error v.s. the baseline.

Object Detection Results



(a) Car (b) Pedestrian (c) Office

Figure 4: Detection PR results by SSFM with calibrated cameras (green), SSFM with uncalibrated cameras (blue) and LSVM [9] (red). Fig. 4c shows average results for mouse, keyboard and monitor categories. SSFM is applied on image pairs randomly selected from the testing set (unless otherwise stated). Calibration is obtained from ground truth.

Runtime

- 20 minute runtime for 2 images
- Results not presented for more than 4
- Bad scaling?
- Code released, but 0.1 alpha vers…
- Ran Bundler on 4 images, took < 3 minutes

Questions?