

Distributed localization of networked cameras

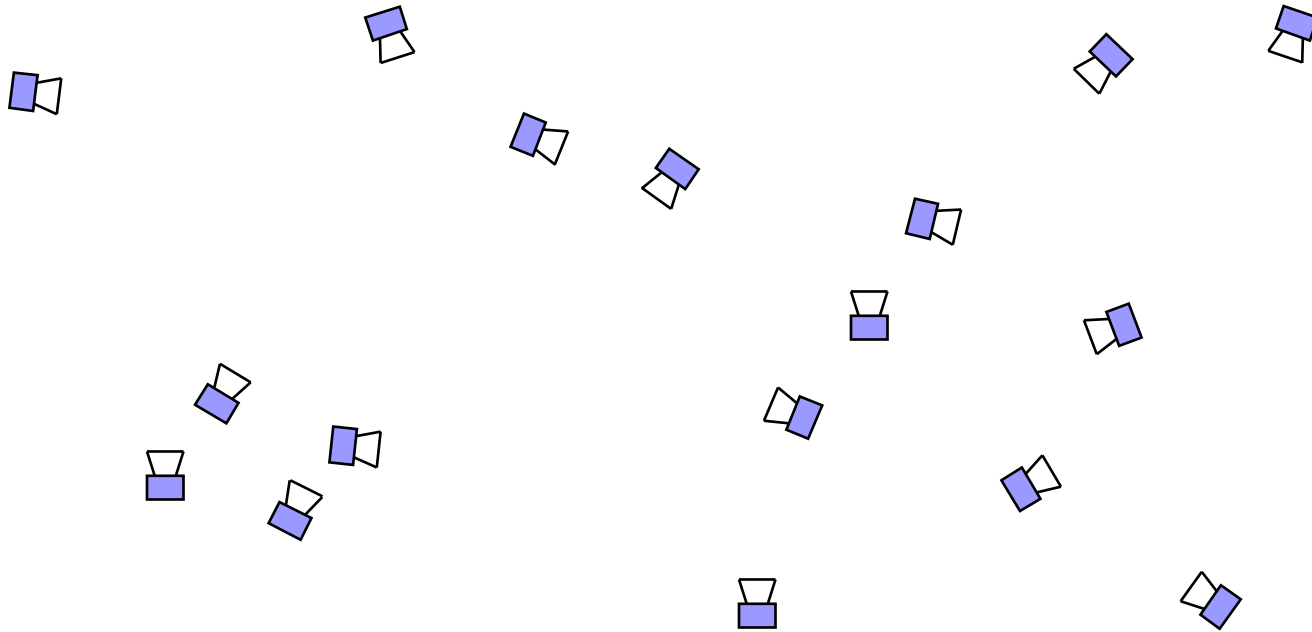
Stanislav Funiak **Carlos Guestrin**
Carnegie Mellon University

Mark Paskin
Stanford University

Rahul Sukthankar
Intel Research

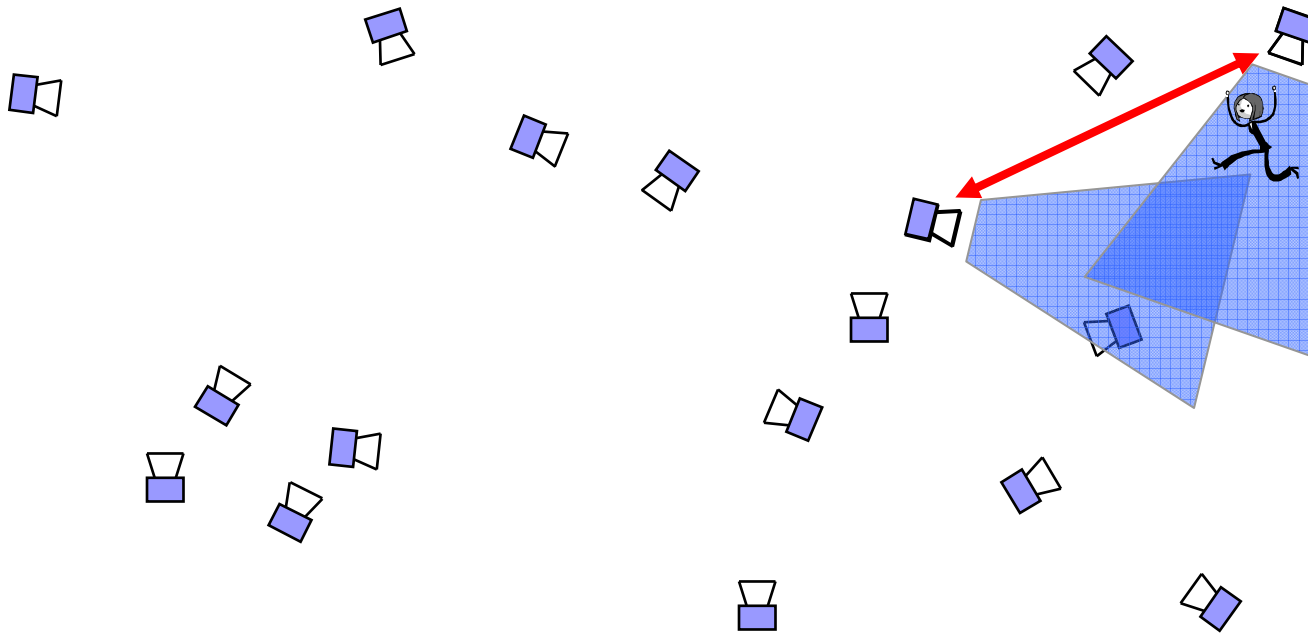
IPSN 2006 presentation, April 19, 2006

Distributed Localization of Cameras



Place wireless cameras around an environment
Need to know locations
Costly to measure locations

Distributed Localization of Cameras



If camera 1 sees person, then camera 2 sees person, learn about relative positions of cameras

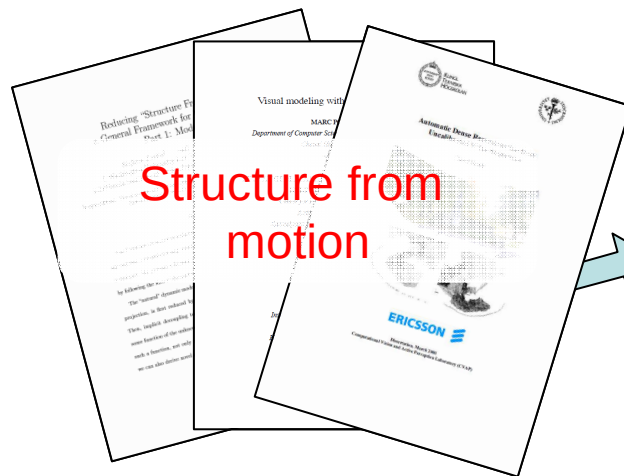
As person moves around, estimate positions of all cameras

Prior Work



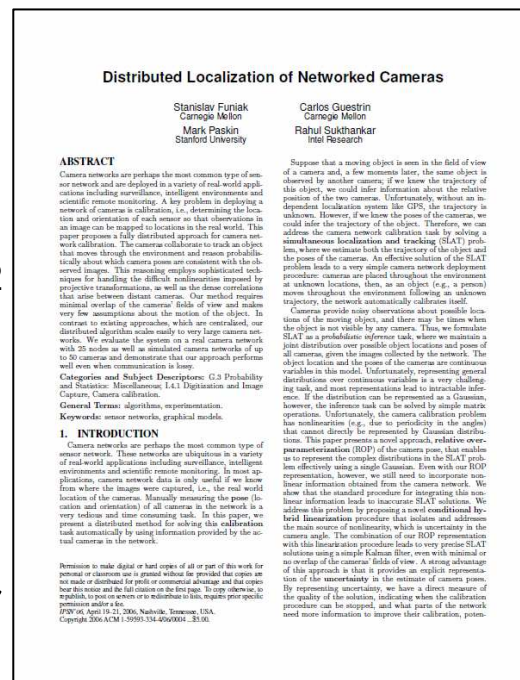
Localization from pairwise distances

Ihler *et al.*, IPSN 2004
Whitehouse, Culler, ACM WSNA 02



Structure from motion

Pollefeys, IJCV 2004
Soatto, Perona, IEEE PAMI 1998



Distributed Localization of Networked Cameras

Stanislav Furlak
Carlos Guestrin
Mark Paskin
Rahul Sukthankar
Intel Research

ABSTRACT

Camera networks are perhaps the most common type of sensor network and are deployed in a variety of real-world applications including surveillance, intelligent environments and scientific remote monitoring. A key problem in deploying a network of cameras is calibration, i.e., determining the location and orientation of each camera so that observations in an image can be mapped to locations in the real world. This paper proposes a fully distributed approach for camera network calibration. The camera collaborates to track an object that moves through the environment and reasons probabilistically about which camera poses are consistent with the observed images. This reasoning employs sophisticated techniques for handling the difficult nonlinearities imposed by projective transformation, as well as the dense correlations that arise between distant cameras. Our method requires minimal overlap of the cameras' fields of view and makes very few assumptions about the motion of the object. In contrast to existing approaches, which are centralized, our distributed algorithm scales easily to very large camera networks. We evaluate the system on a real camera network with 28 sensors as well as simulated camera networks of up to 50 cameras and demonstrate that our approach performs well even when communication is noisy.

Categories and Subject Descriptors: G.3 Probability and Statistics; Miscellaneous; I.4.1 Digitization and Image Capture; Camera calibration.

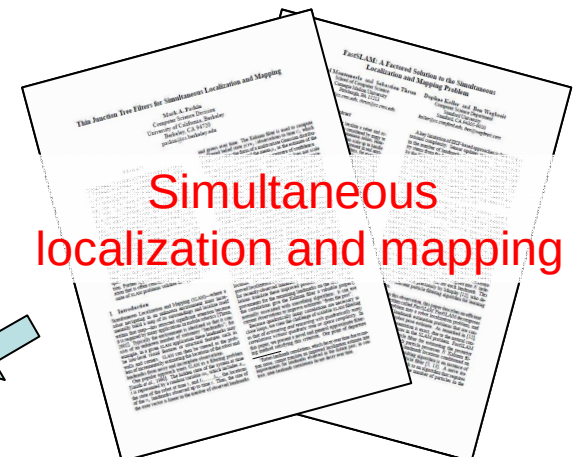
General Terms: algorithms, experimentation.

Keywords: sensor networks, graphical models.

1. INTRODUCTION

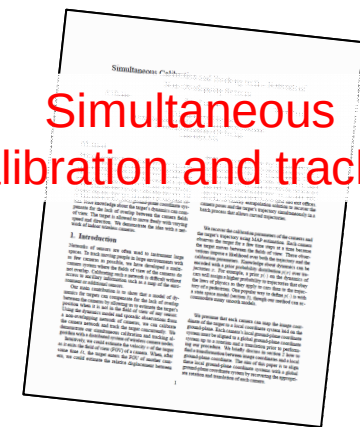
Camera networks are perhaps the most common type of sensor network. These networks are ubiquitous in a variety of real-world applications including surveillance, intelligent environments and scientific remote monitoring. In most applications, camera network data is only useful if we know from where the images were captured, i.e., the real world location of the cameras. Manually measuring the pose (location and orientation) of all cameras in the network is a very tedious and time consuming task. In this paper, we present a distributed method for solving this calibration task automatically by using information provided by the actual cameras in the network.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted by the copyright owner, provided that the copies are not made or distributed for profit or commercial advantage and that copies are not made or distributed for promotional or advertising purposes. For all other copying, permission should be sought from the copyright owner. Copyright 2006 ACM 978-1-59593-324-4/06/0000...\$5.00.



Simultaneous localization and mapping

Montemerlo *et al.*, AAAI 2002
Paskin, IJCAI 2003



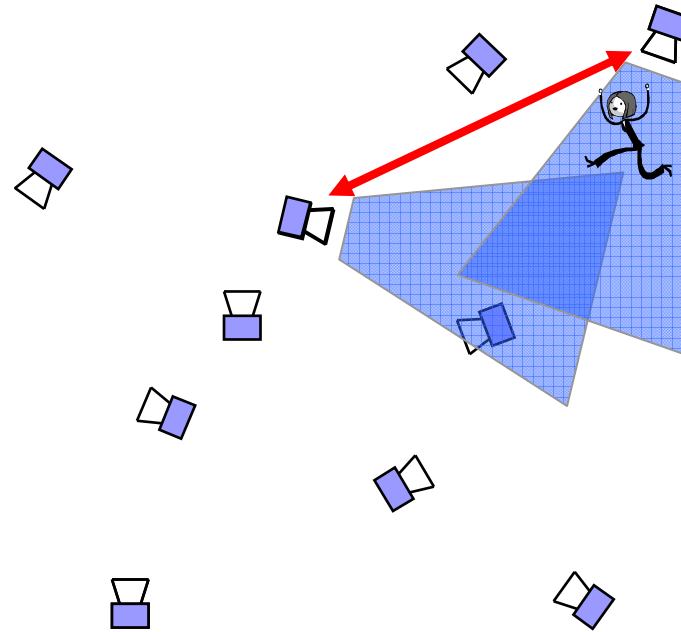
Simultaneous calibration and tracking

Rahimi *et al.*, CVPR 2004

Distributed Localization of Cameras

Want a solution:

- **online**
- **distributed**
- **represents uncertainty about estimated locations**
e.g. for active control



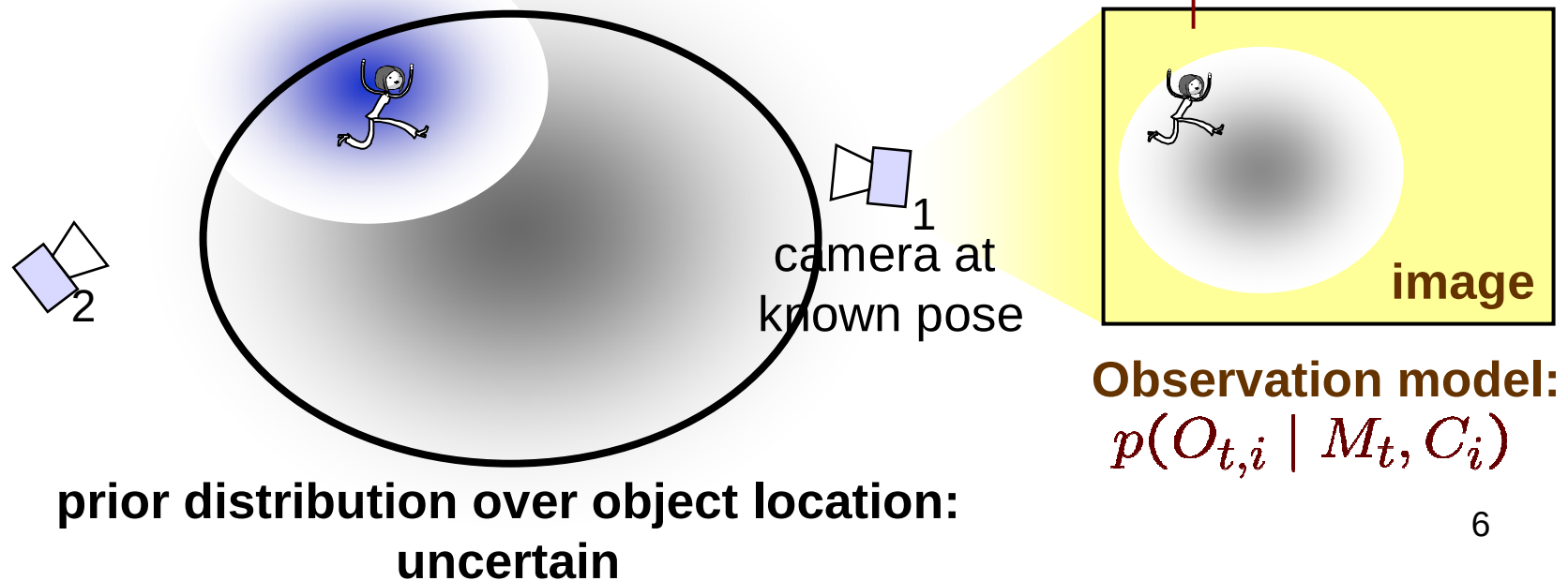
If camera 1 sees person, then camera 2 sees person, learn about relative positions of cameras

As person moves around, estimate positions of all cameras

Tracking with Kalman Filter: Estimation

$$\underbrace{p(M_t \mid \mathbf{C}, \mathbf{o}_{1:t-1}, \mathbf{o}_t)}_{\text{posterior distribution}} \propto \underbrace{p(M_t \mid \mathbf{C}, \mathbf{o}_{1:t-1})}_{\substack{\text{prior distribution} \\ \begin{array}{l} \text{object location} \\ \text{camera poses} \\ \text{previous observations} \end{array}}} \times \underbrace{p(O_{t,i} = o_{t,i} \mid M_t, C_i)}_{\text{observation likelihood}}$$

**posterior distribution:
even more uncertain**



Tracking with Kalman Filter: Prediction

$$\underbrace{\int_{M_t} p(M_t | \mathbf{C}, \mathbf{o}_{1:t-1}, \mathbf{o}_t)}_{\text{posterior distribution}} \times \underbrace{p(M_{t+1} | M_t)}_{\text{motion model}} = \underbrace{p(M_{t+1} | \mathbf{C}, \mathbf{o}_{1:t})}_{\text{predicted distribution (prior at } t+1 \text{)}}$$

M_t



M_{t+1}

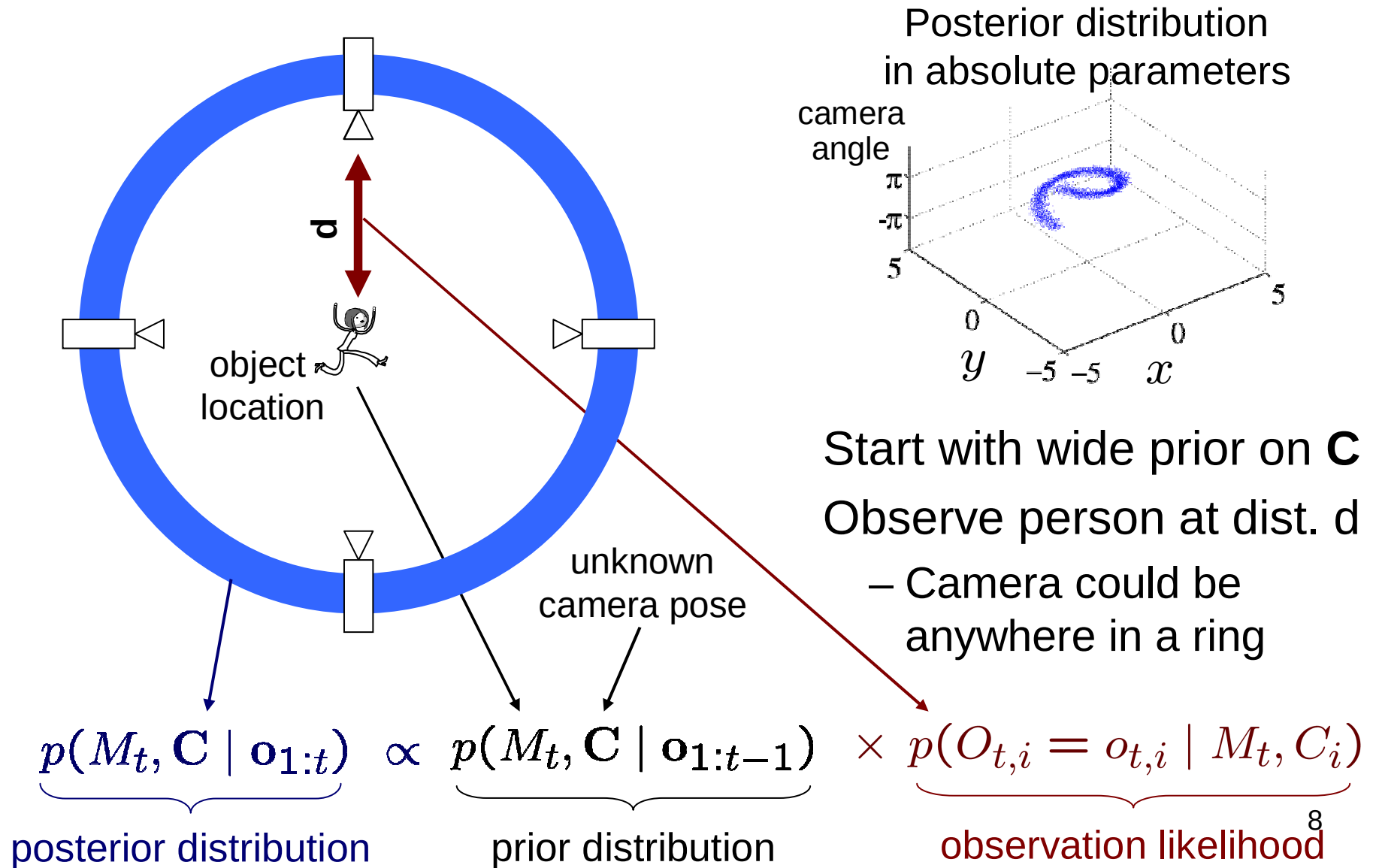
Motion model:



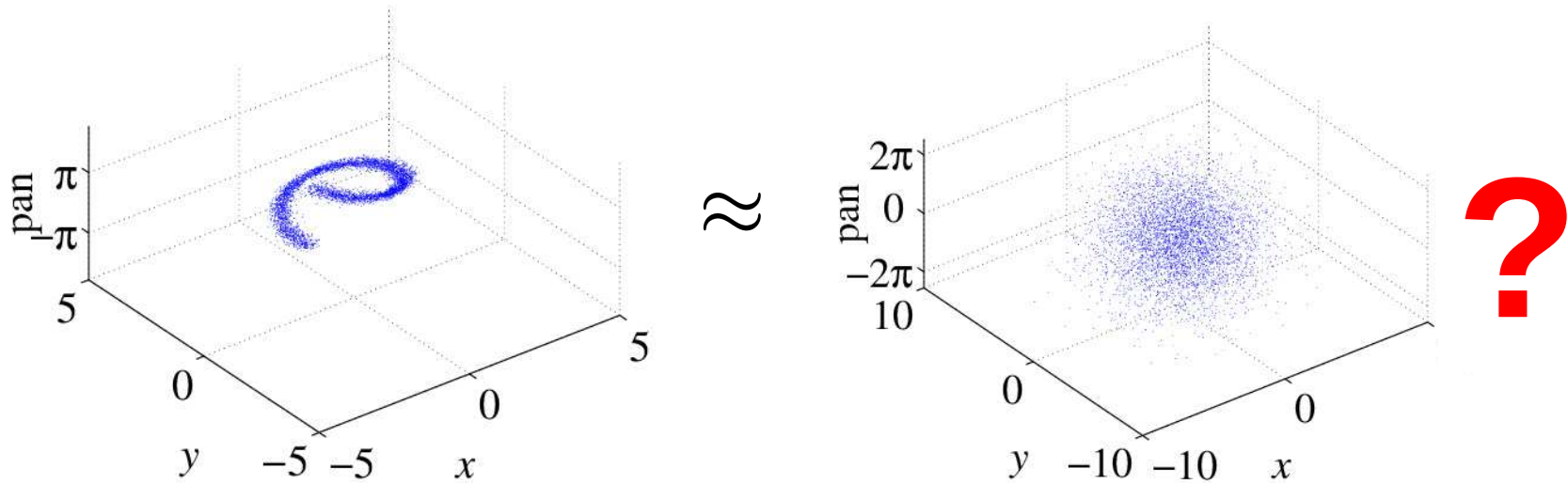
t

$t+1$

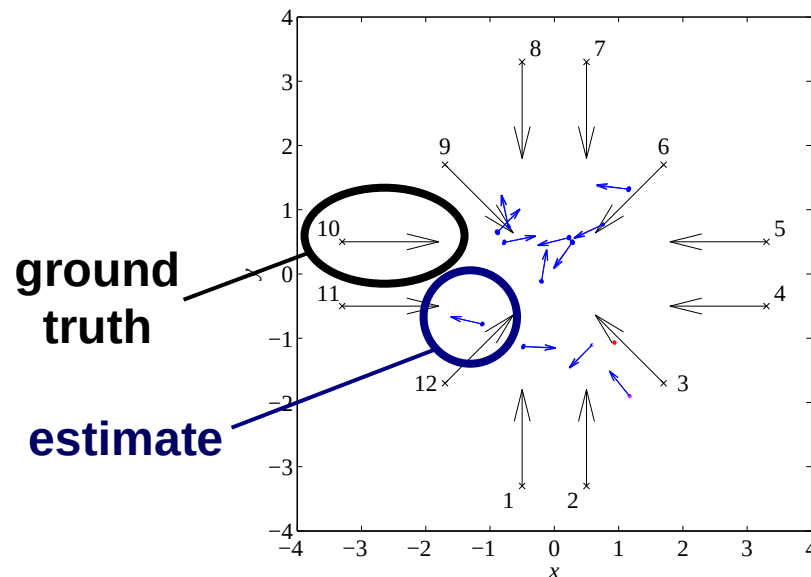
Camera Localization: Estimation



Kalman Filter uses a linear representation...



Problem structure lost with Gaussian approximation



Far from ground truth
Overconfident

Relative Over-Parameterization (ROP)

Intuition: a ring structure can be represented with polar coordinates

Not enough: Camera does not view person head on

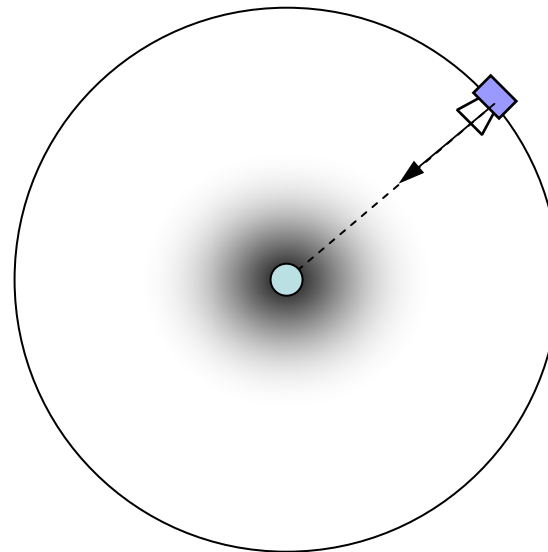
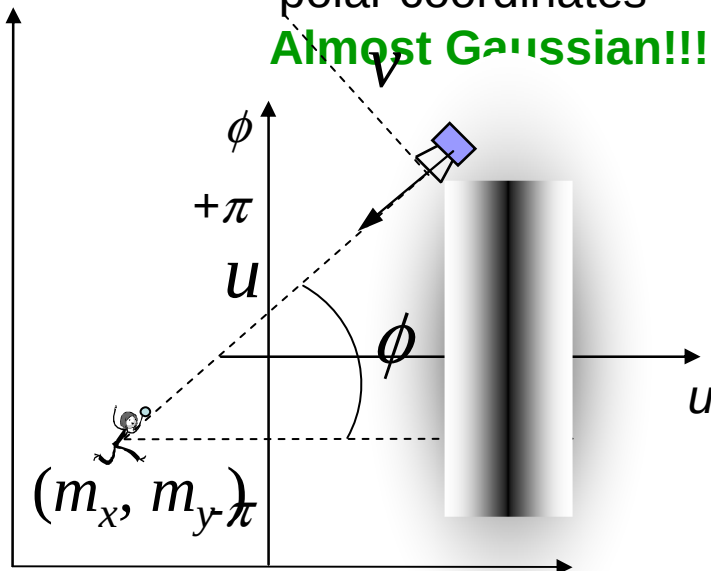
Relative over-parameterization – position relative to location of person

1. Distance u , angle ϕ
2. Lateral displacement v
3. The center – the *unknown* location of object

ROP

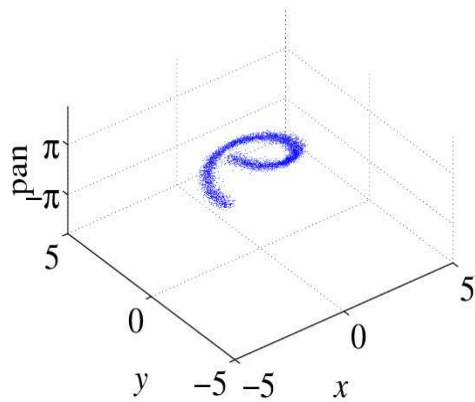
$\langle u, \phi, \phi, m_x, m_y \rangle$ but in

polar coordinates –
Almost Gaussian!!!

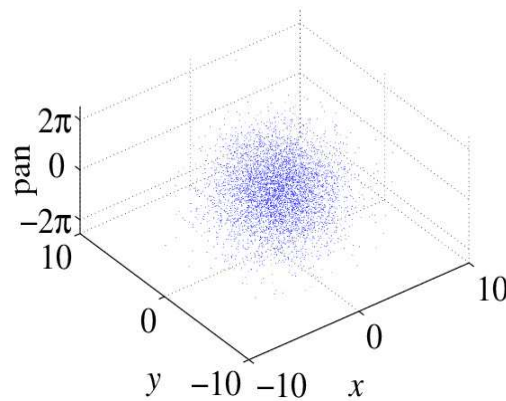


Comparing parameterizations

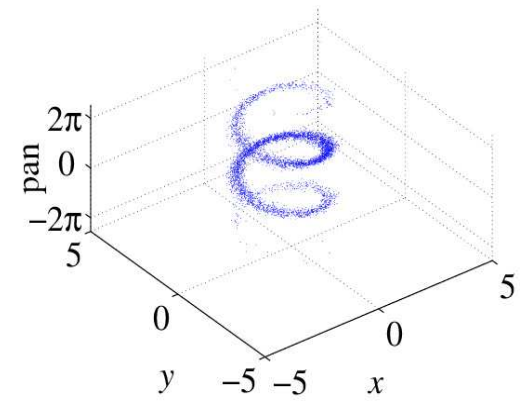
true posterior:



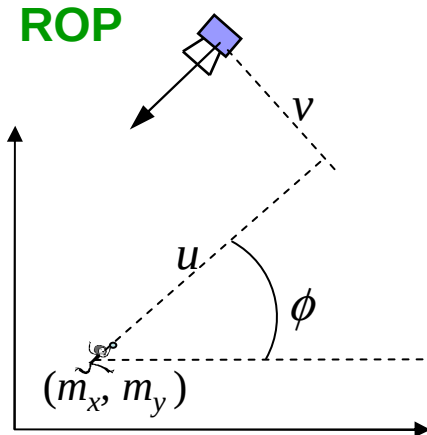
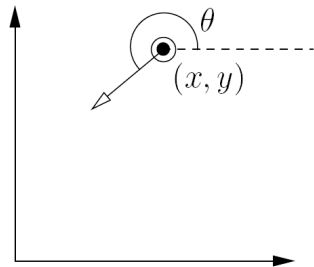
best Gaussian
approx. in x, y, θ :



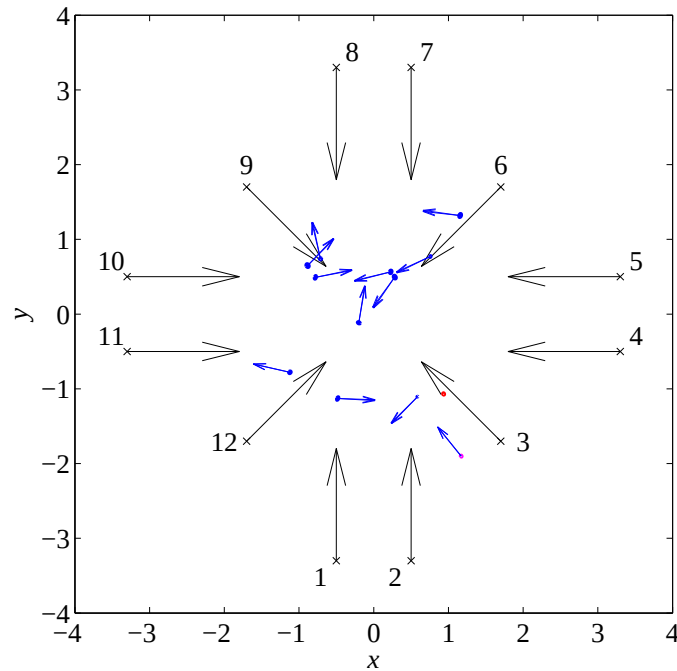
best Gaussian
approx. in ROP:



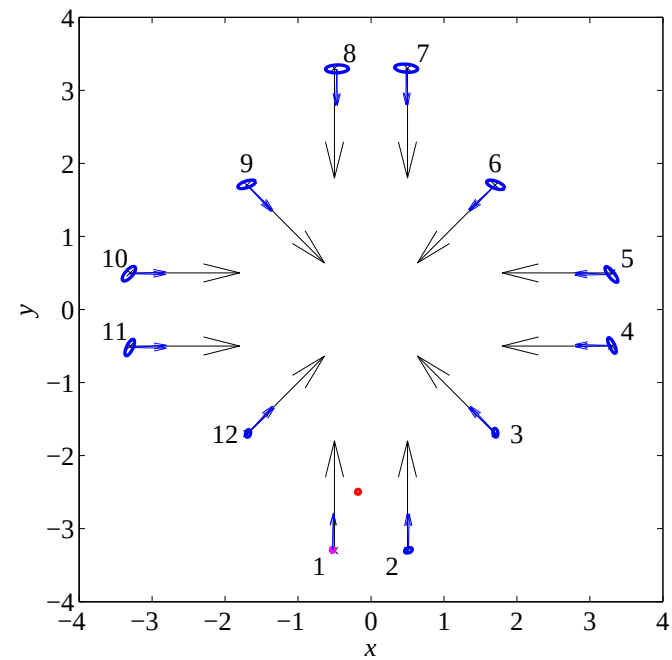
Standard
parameterization



Test run on Tower Scenario

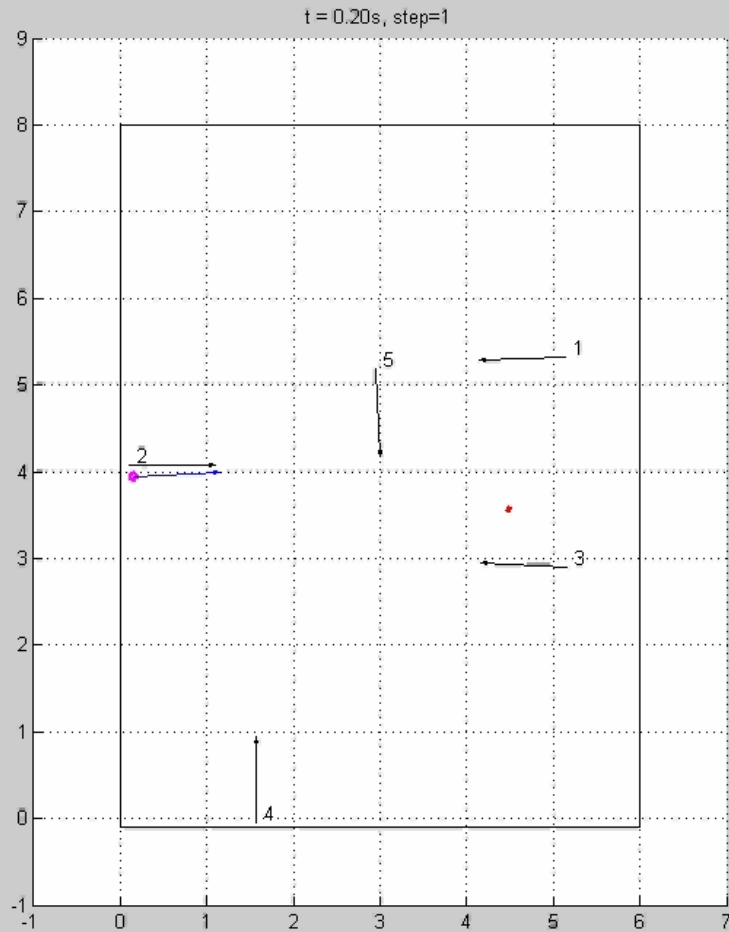


standard parameterization



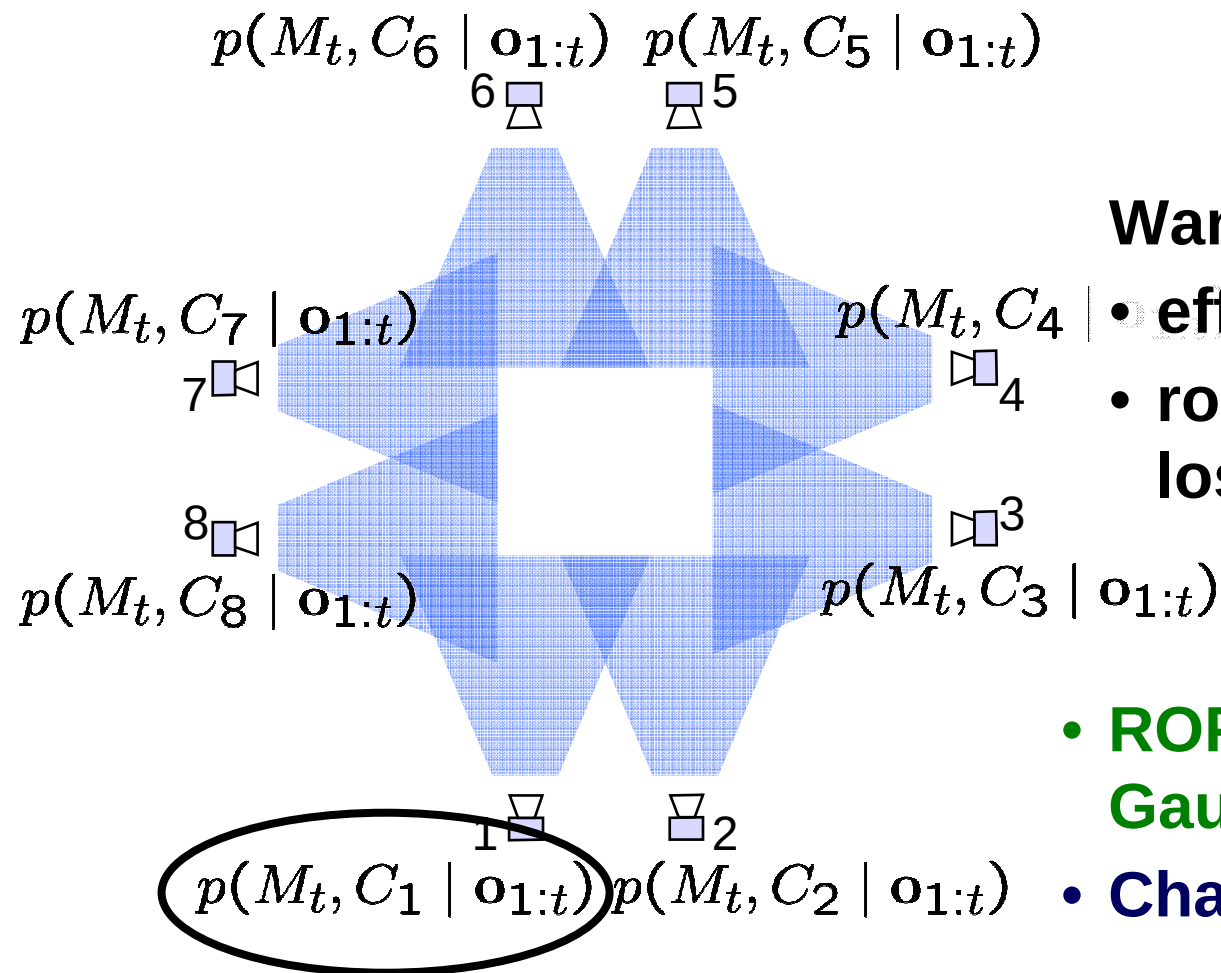
ROP with further improvements
(see paper)

Donuts and Bananas on real data – Network of 5 cameras



Distributed Localization of Cameras

Goal: each camera estimates the location of itself and the object

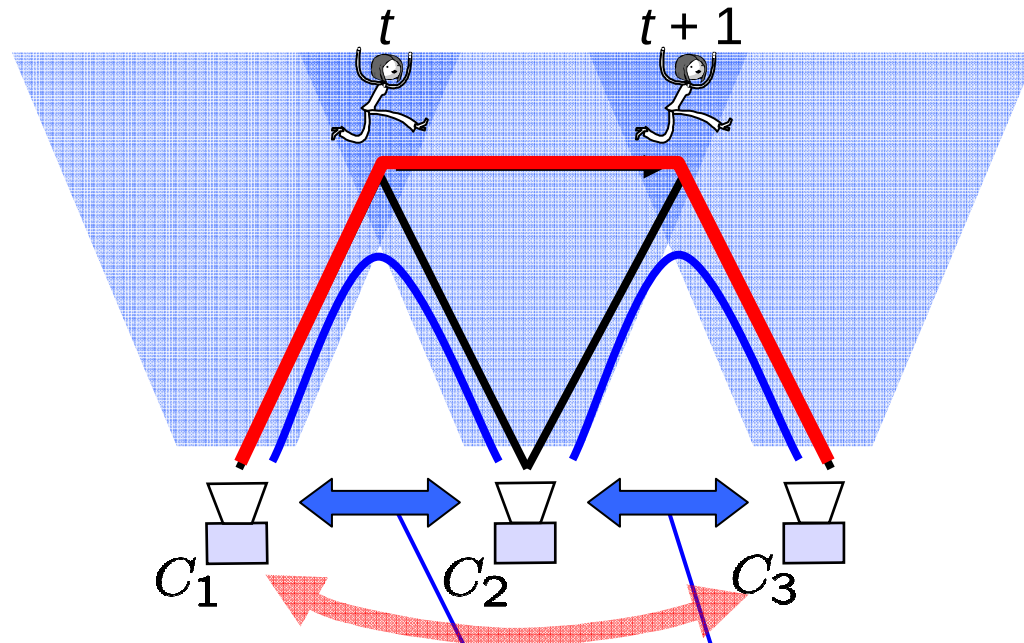


Want algorithm:

- **efficient**
- **robust to message loss, node loss**

- **ROP lets us use a single Gaussian**
- **Challenges?**

Motion model introduces dependencies



Motion model introduces dependencies among distant cameras

➡ communication and computation inefficiency

Estimation at t :

$$p(o_{t,1} | M_t, C_1) \times p(o_{t,2} | M_t, C_2)$$

Prediction:

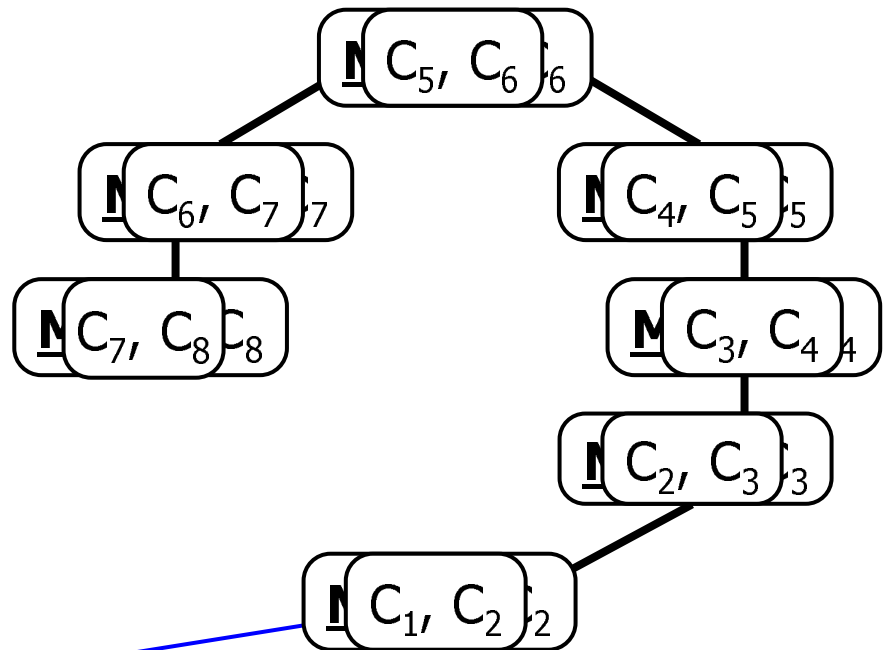
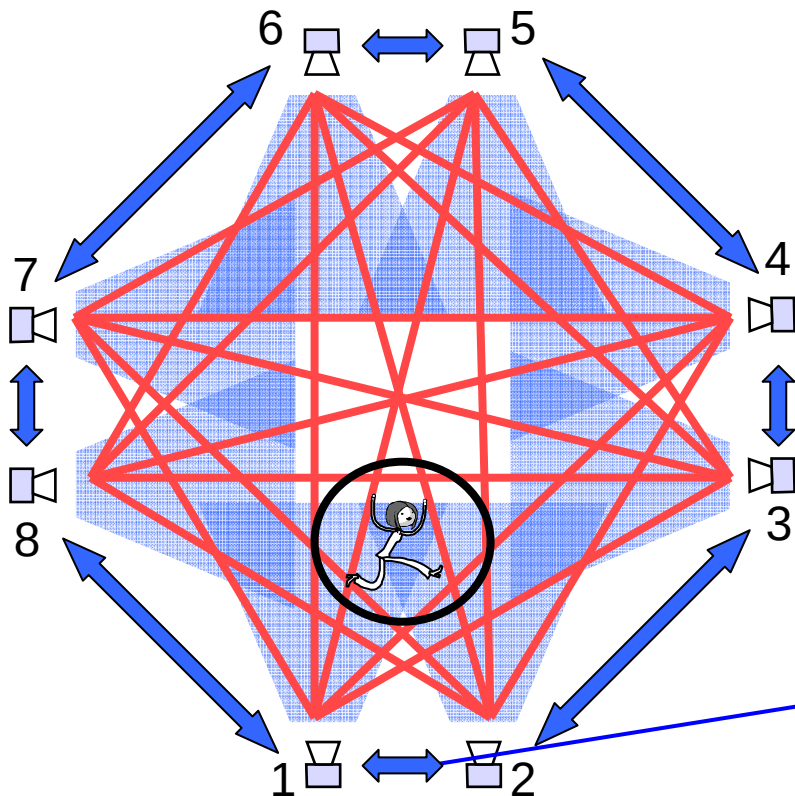
$$\times p(M_{t+1} | M_t)$$

Estimation at $t+1$:

$$p(o_{t+1,2} | M_{t+1}, C_2) \times p(o_{t+1,3} | M_{t+1}, C_3)$$

Assumed density filtering

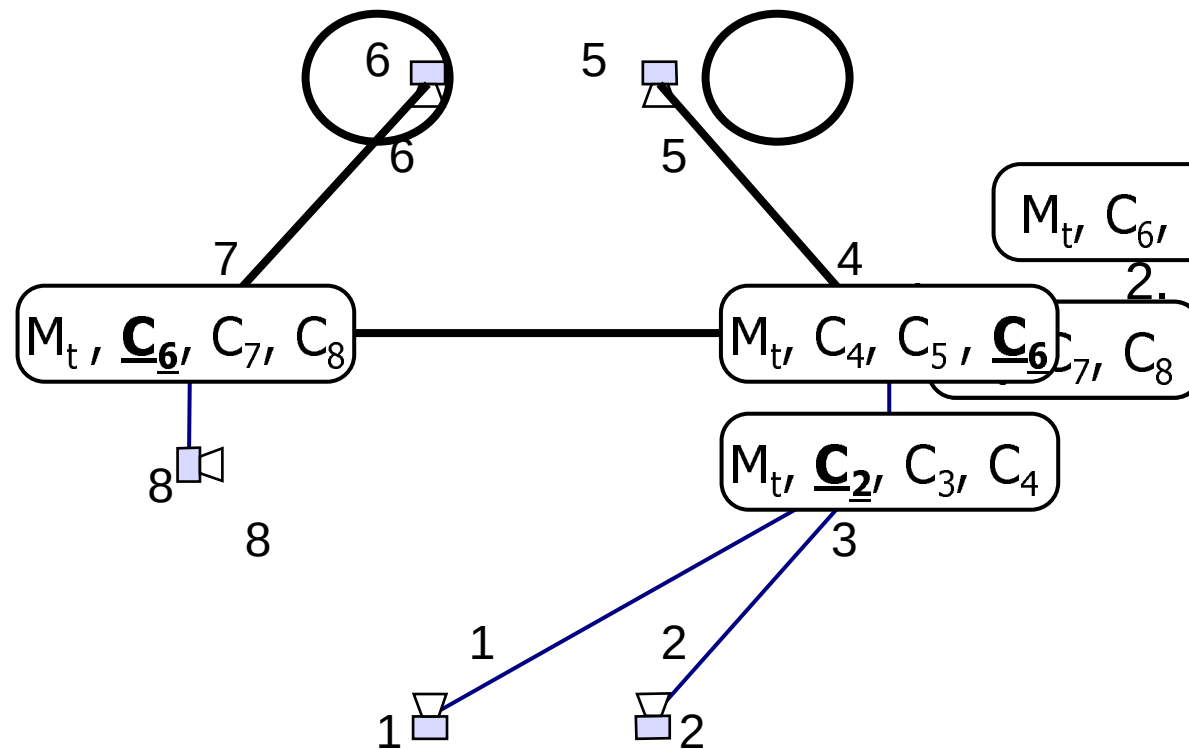
Intuition: only capture strong dependencies among cameras
based on [Boyen Koller 1998]



Each **clique** contains:

- Camera and its neighbor
- Object location

Distributed Filtering: Initialization



1. Assign each clique to one or more nodes

• M_t, C_5, C_6 clique to > 1 node for robustness

M_t, C_6, C_7

M_t, C_4, C_5

2. The nodes build a

network junction

[Paskin et al. 2005]

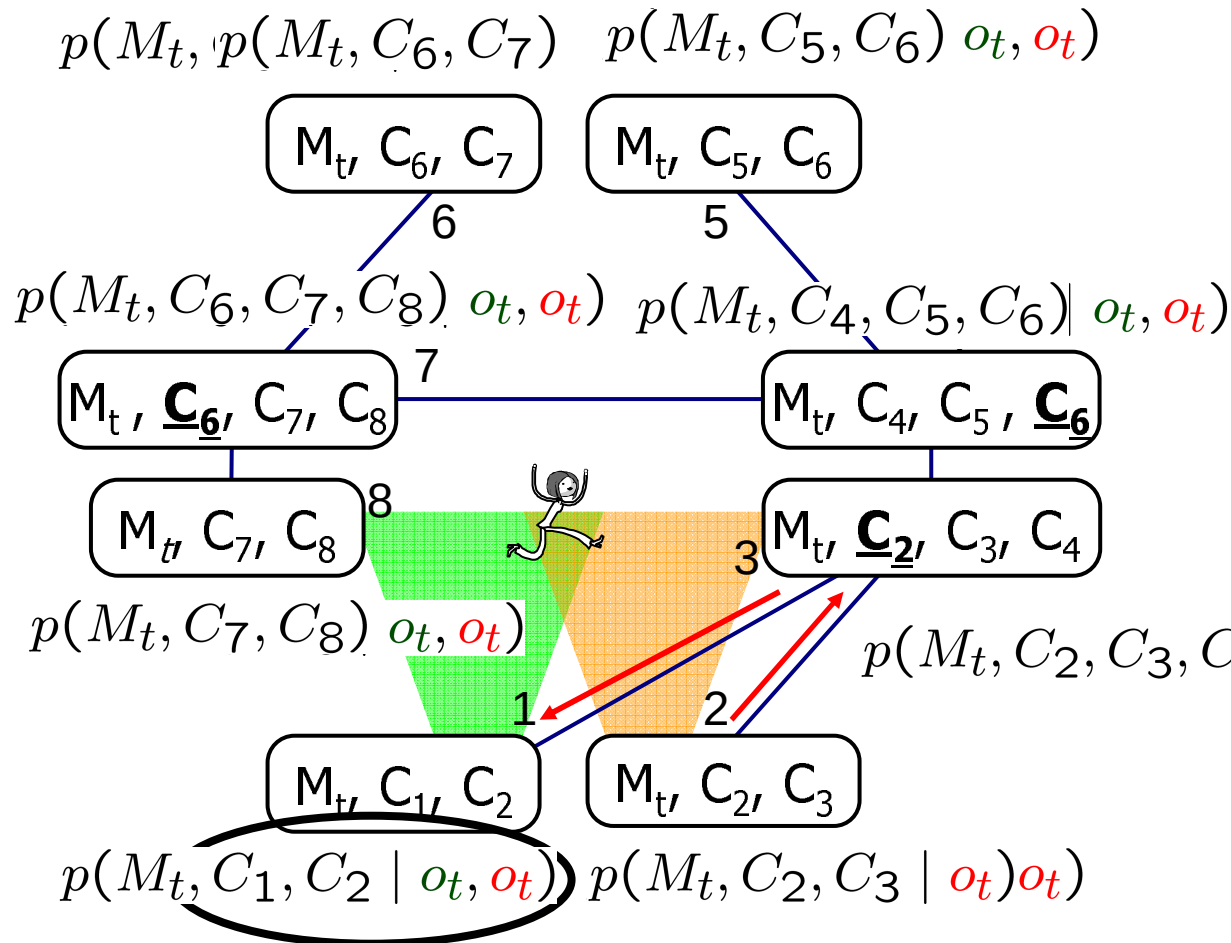
• build a red M_t, C_2, C_3

• ensure the flow of

M_t, C_1, C_2 on

Distributed Filtering: Estimation

Instance of Robust Distributed Inference [Paskin Guestrin, UAI 2004]



1. Each node starts with **prior over its clique**

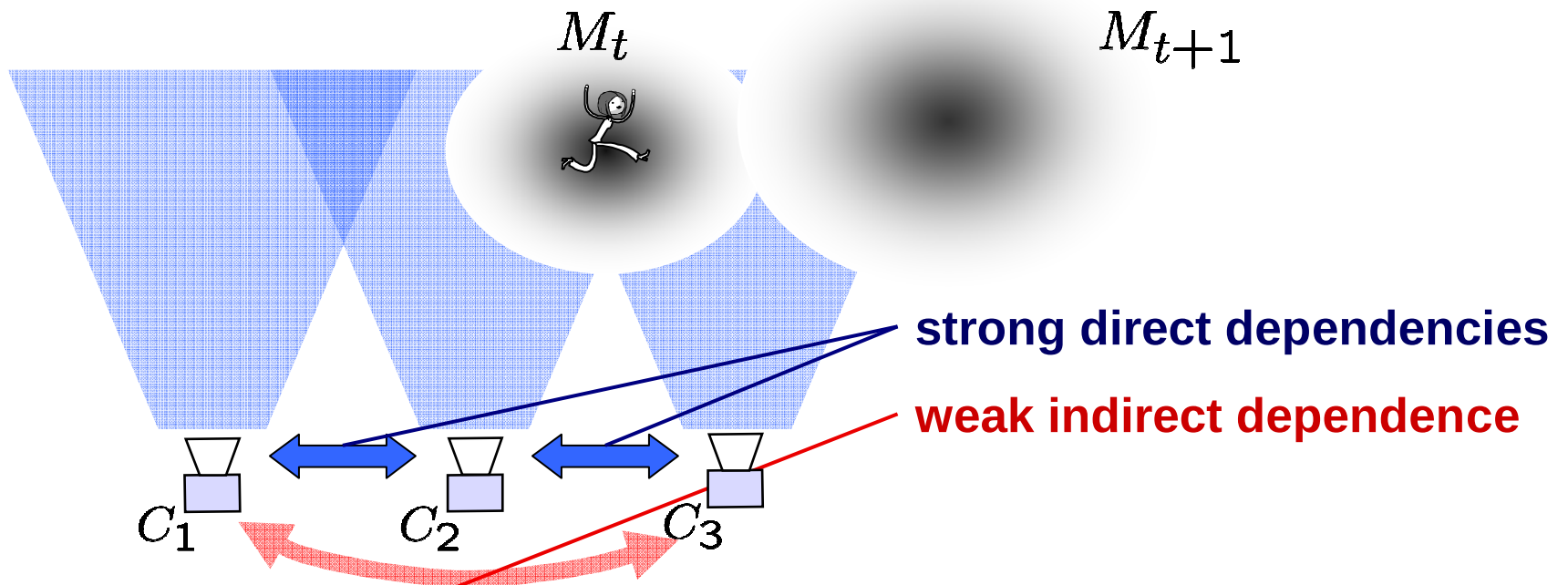
2. Nodes make **observations**

3. Nodes **communicate** relevant likelihoods & priors neighbors

4. At convergence: **condition on all measurements** made in the network

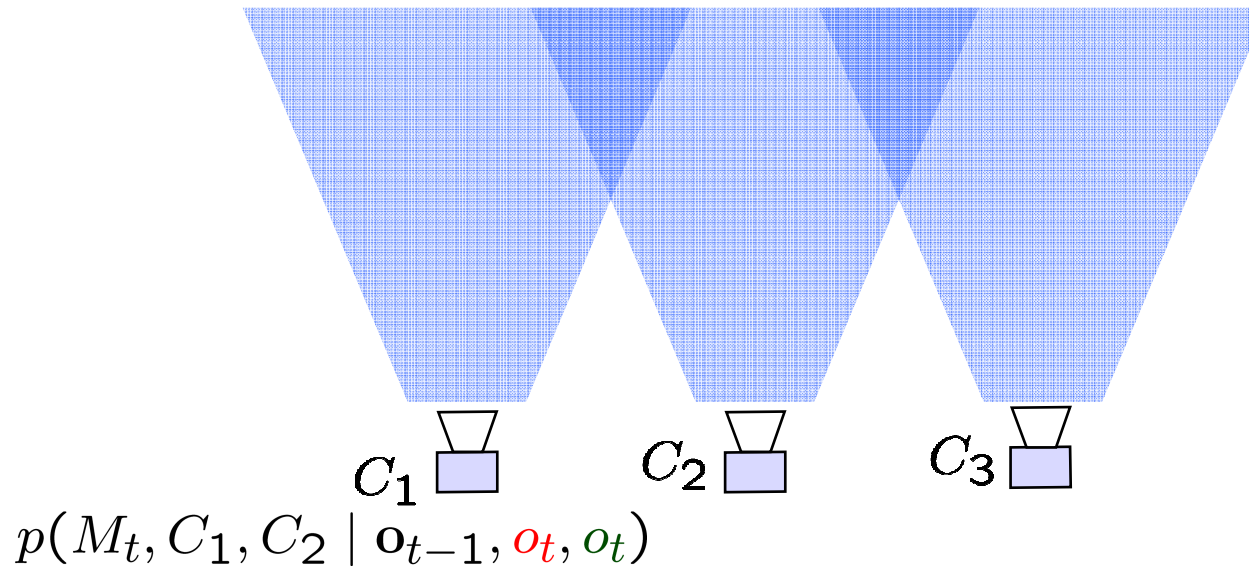
Prediction Revisited

$$\overbrace{p(M_{t+1}, \mathbf{C} \mid \mathbf{o}_{1:t})}^{\text{prediction}} = \int_{M_t} \overbrace{p(M_t, \mathbf{C} \mid \mathbf{o}_{1:t-1}, \mathbf{o}_t)}^{\text{posterior distribution}} \times \overbrace{p(M_{t+1} \mid M_t)}^{\text{motion model}}$$



**How to implement the prediction step distributedly?
prune weak dependencies?**

Distributed Filtering: Prediction



$$\tilde{p}(M_{t+1}, C_1, C_2 \mid \mathbf{o}_{1:t}) = \int p(m_t, C_1, C_2 \mid \mathbf{o}_{1:t}) p(M_{t+1} \mid m_t) dm_t$$

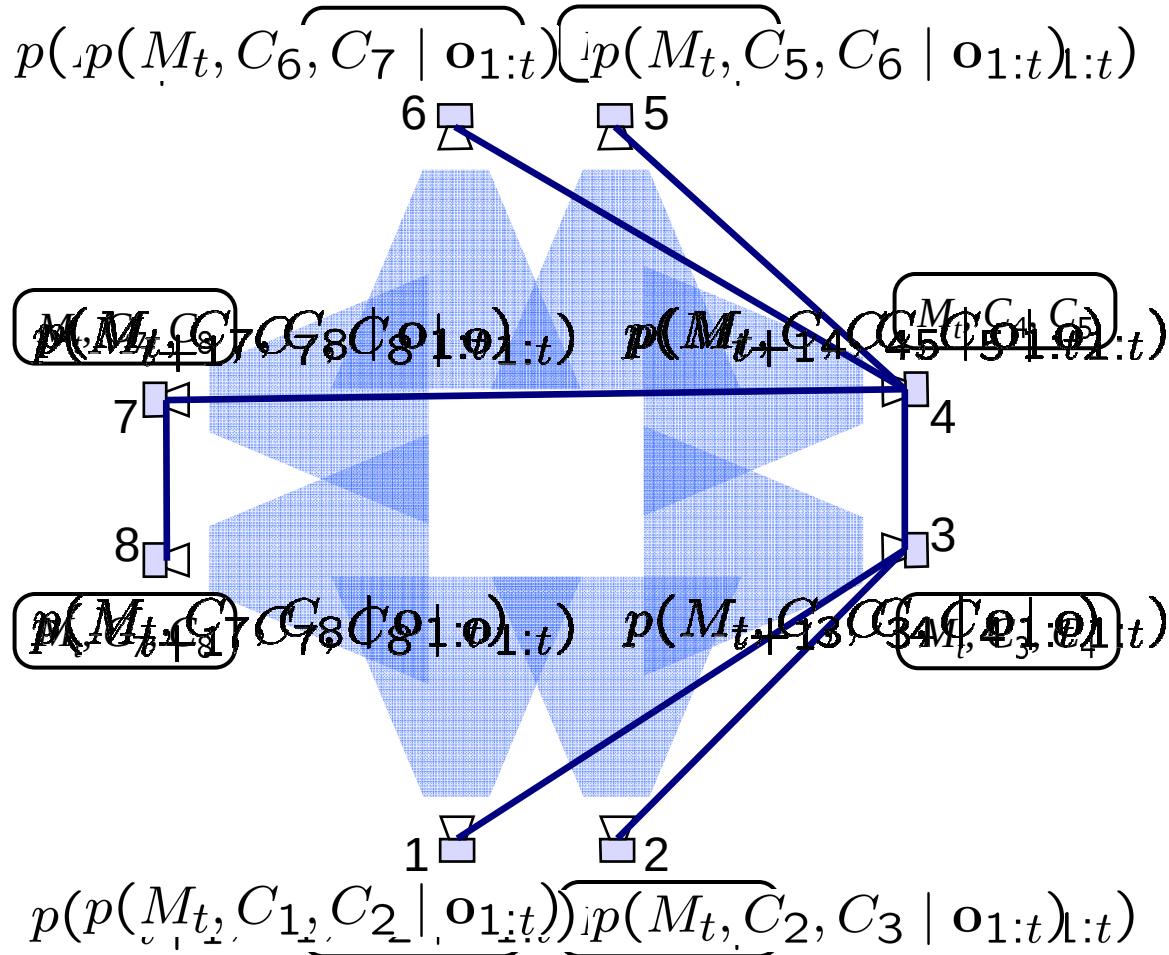
Want the best approximation (minimizing KL divergence):

- captures short-range dependencies
- drops long-range dependencies

Sufficient to compute the marginals over cliques

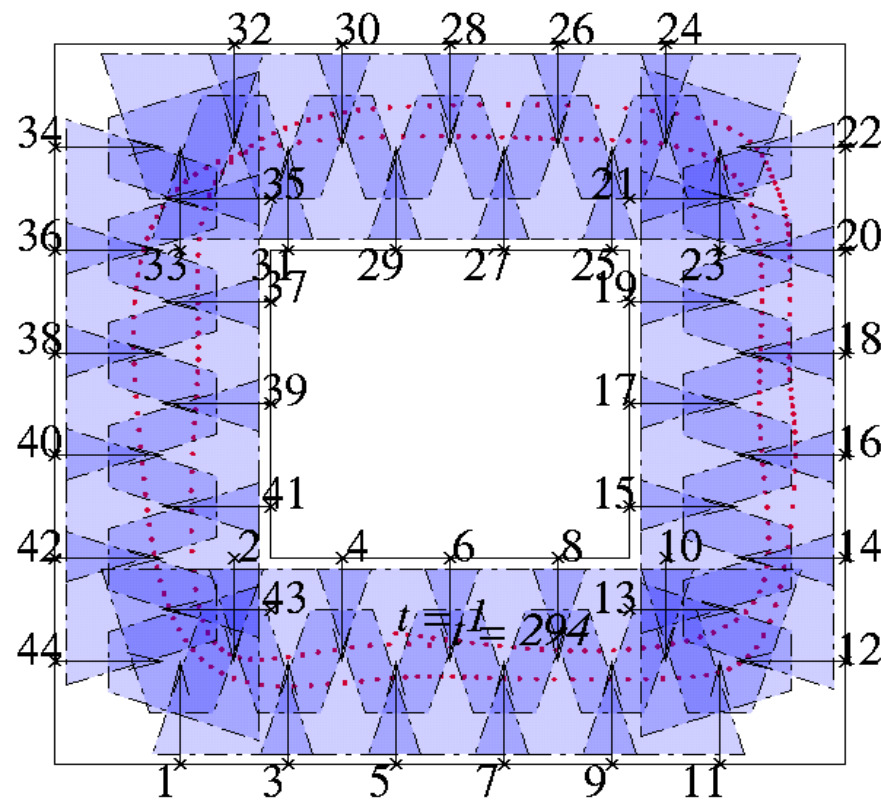
[Boyen, Koller 1998]

Summary of our approach

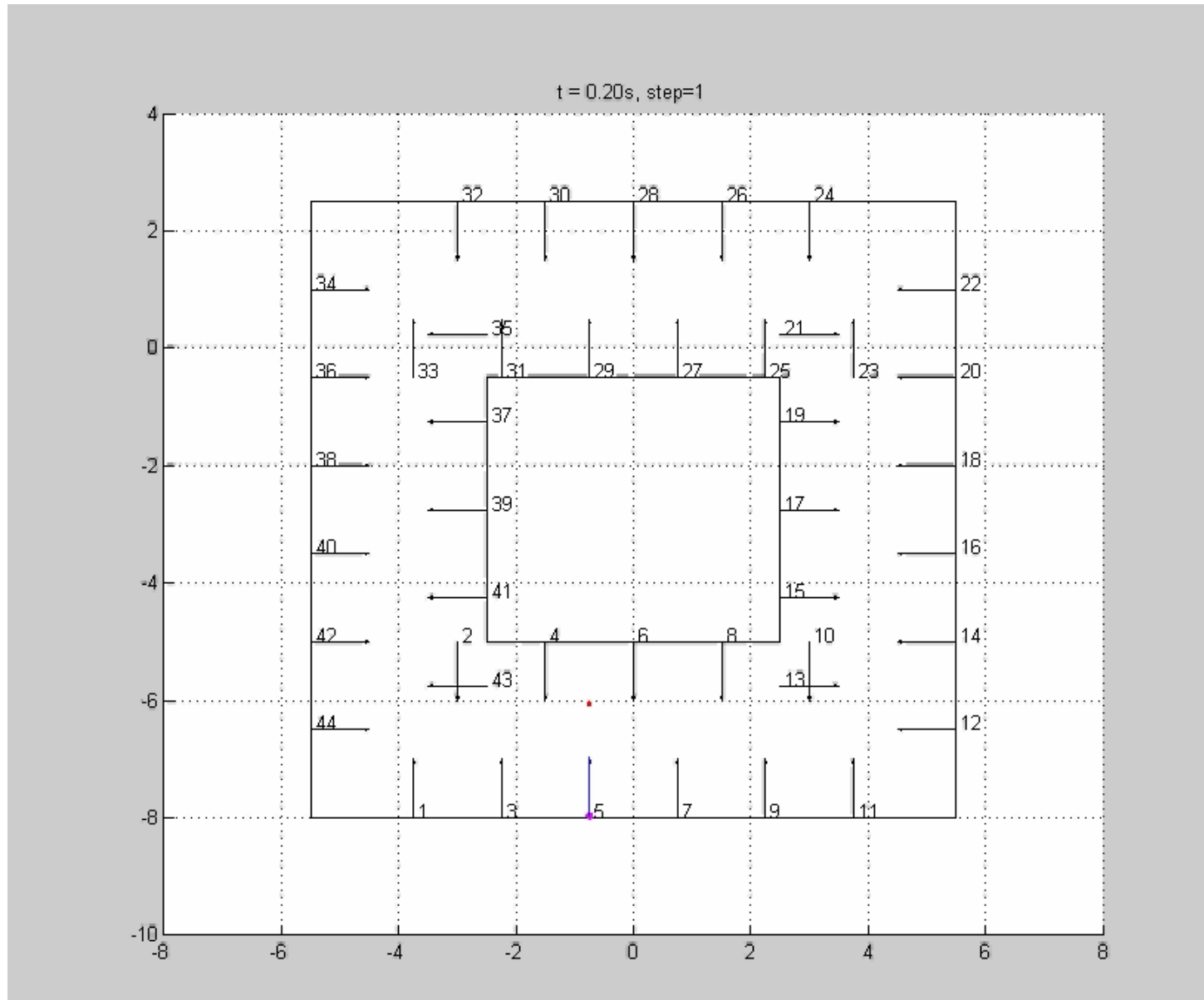


1. Each node maintains a clique marginal
2. Nodes build communication structure, **network junction tree** [Paskin et al. 2005]
3. Estimation: nodes condition on observations [Paskin & Guestrin UAI 04]
4. Prediction: best approximation computed locally

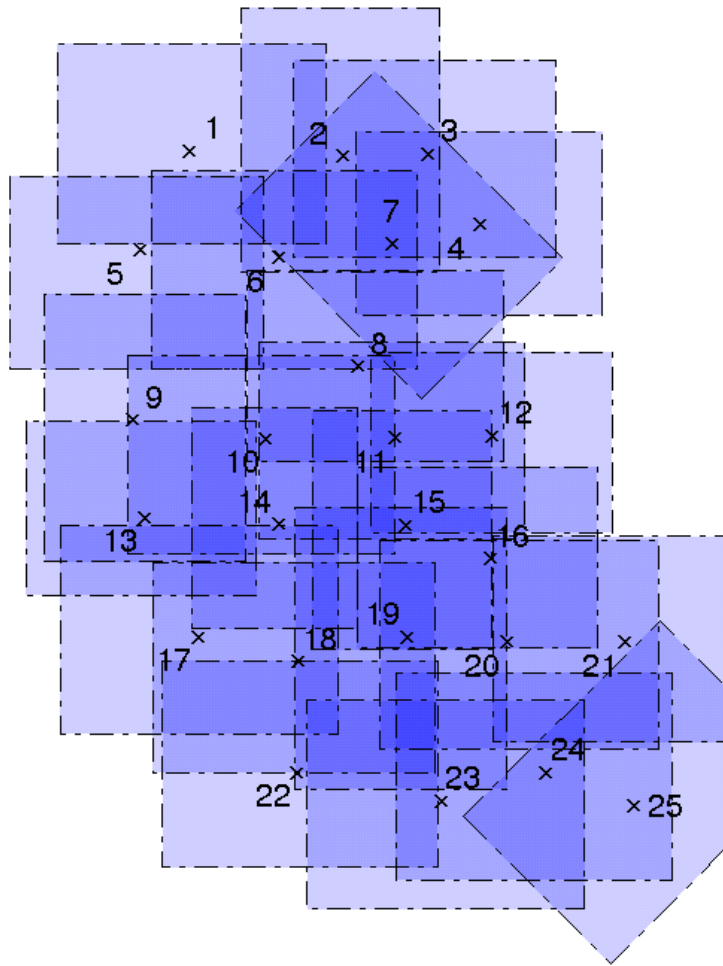
Results: 44 simulated side-facing cameras



Results: 44 simulated side-facing cameras

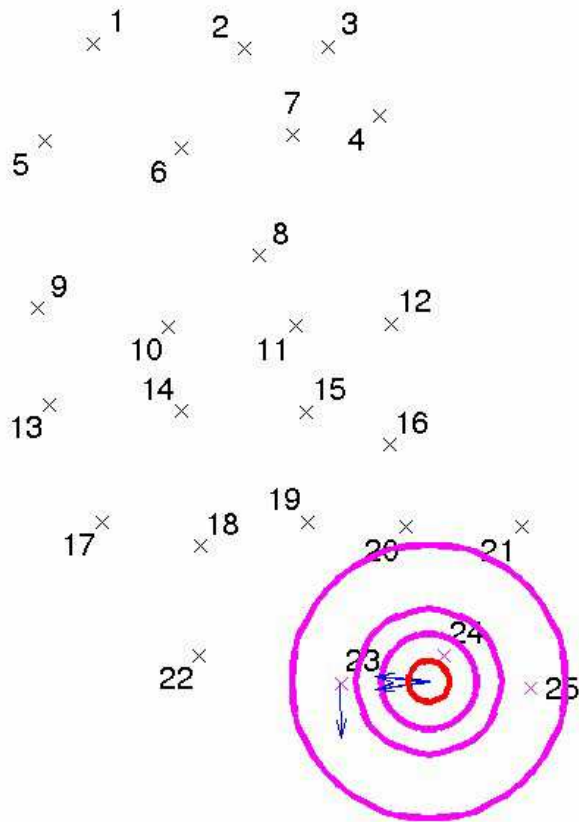


Network of 25 cameras at Intel Research Pittsburgh

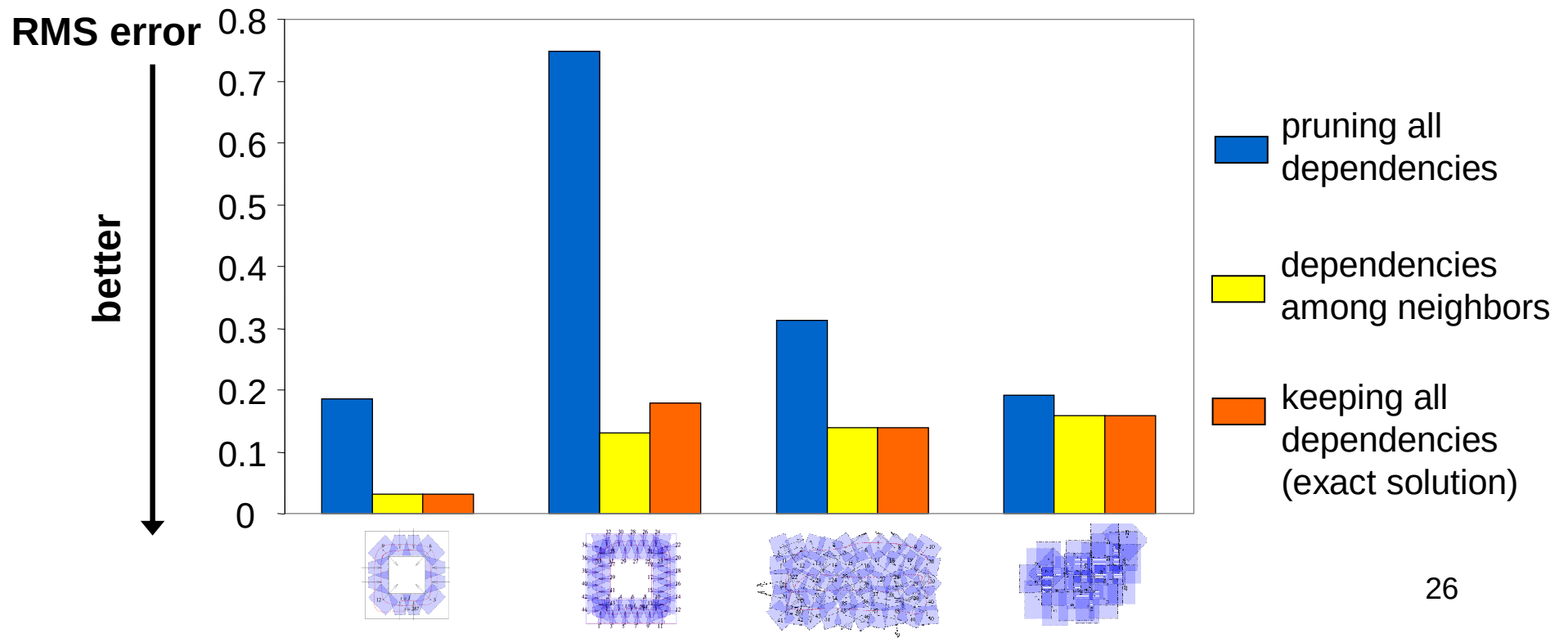
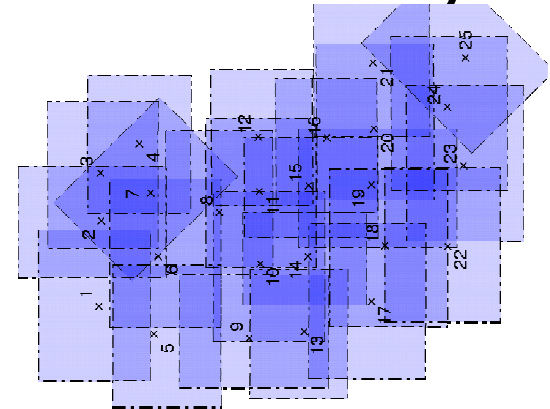
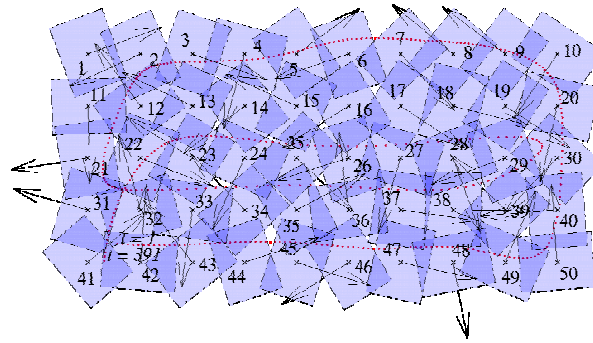
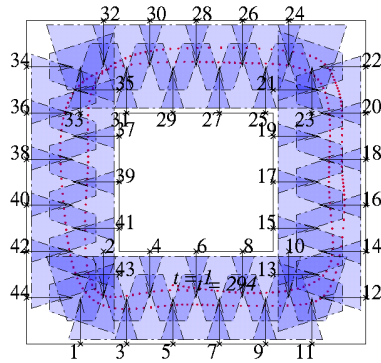


Network of 25 cameras at Intel Research Pittsburgh

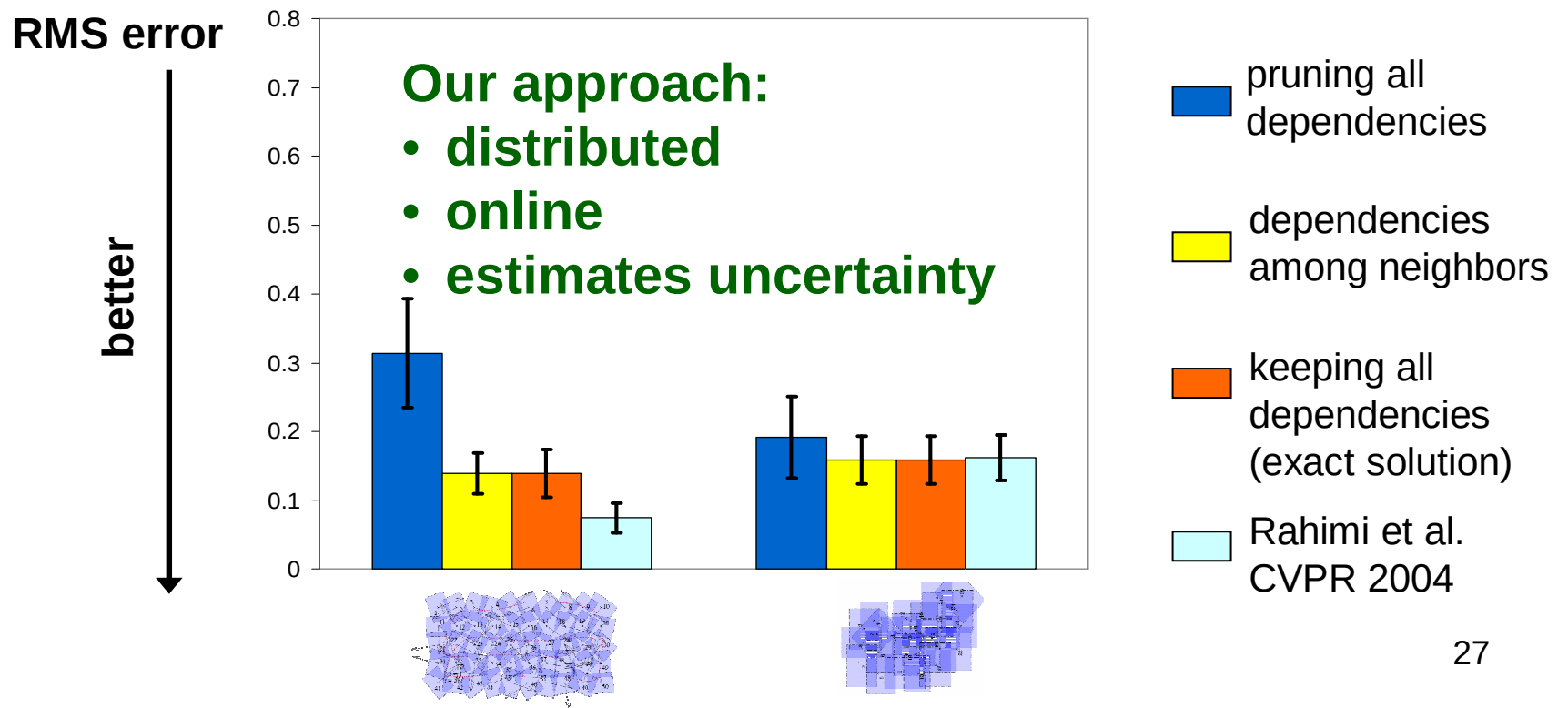
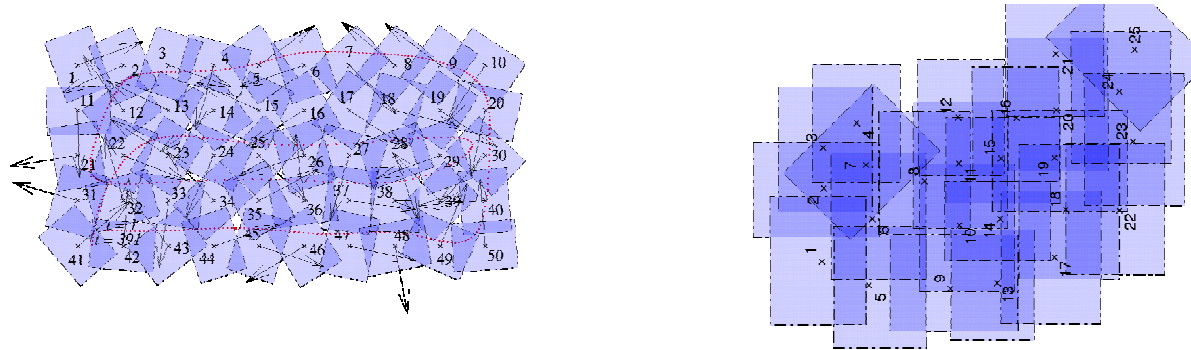
$t = 0.27s$, step=1



Results: Model Complexity vs. Accuracy

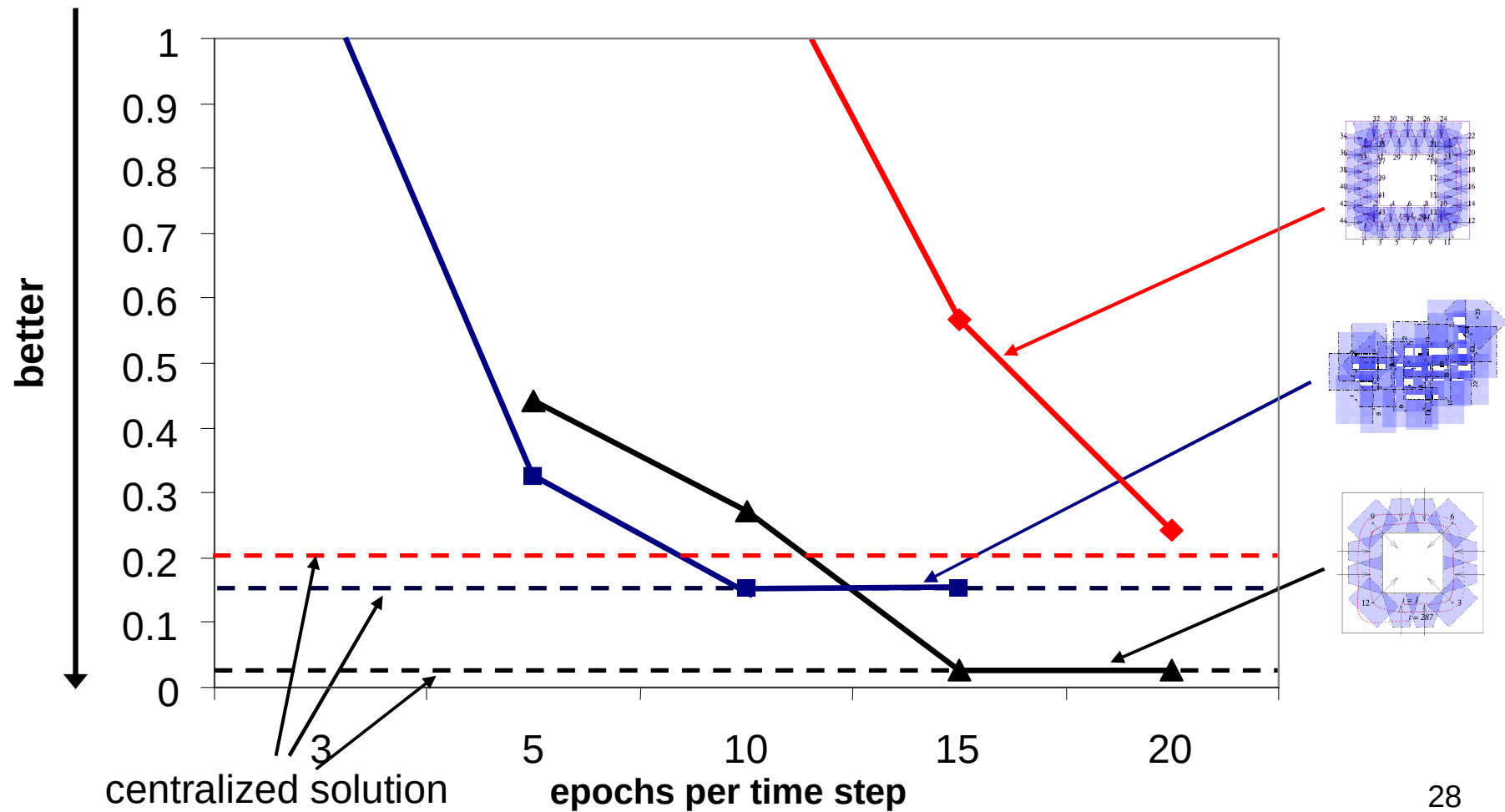


Comparison with Rahimi et al., CVPR 2004



Results: Communication vs. Accuracy

RMS error



Conclusion

- **Accurate camera localization with only a single Gaussian!!!**
 - ROP – parameterization accurately representing **ring-like distributions**
 - Effective technique for incorporating **nonlinear observations**
- Distributed **online** algorithm for camera localization that **represents uncertainty**
- Algorithm for distributed filtering for **general dynamic models**
- Evaluated on network of **25 real cameras**

