Distributed Vision Processing in Smart Camera Networks

CVPR-07

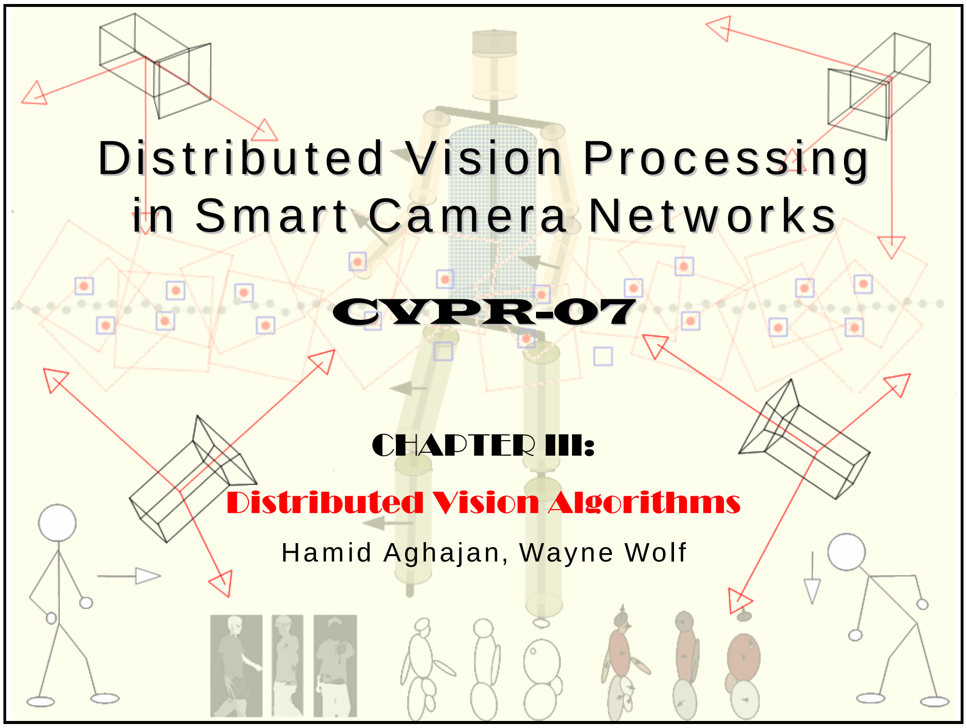
Hamid Aghajan, Stanford University, USA
François Berry, Univ. Blaise Pascal, France
Horst Bischof, TU Graz, Austria
Richard Kleihorst, NXP Research, Netherlands
Bernhard Rinner, Klagenfurt University, Austria
Wayne Wolf, Princeton University, USA

March 18, 2007
Minneapolis, USA

Course Website – http://wsnl.stanford.edu/cvpr07/index.php

Outline

I. Introduction
II. Smart Camera Architectures
  1. Wireless Smart Camera
  2. Smart Camera for Active Vision
III. Distributed Vision Algorithms
  1. Fusion Mechanisms
  2. Vision Network Algorithms
IV. Requirements and Case Studies
V. Outlook



Vision Network Algorithms

Contact: Wayne Wolf

wolf <AT> Princeton.Edu

Topics

- Target modeling for distributed systems
- Distributed state management and protocols
- Calibration:
  – Spatial
  – Temporal

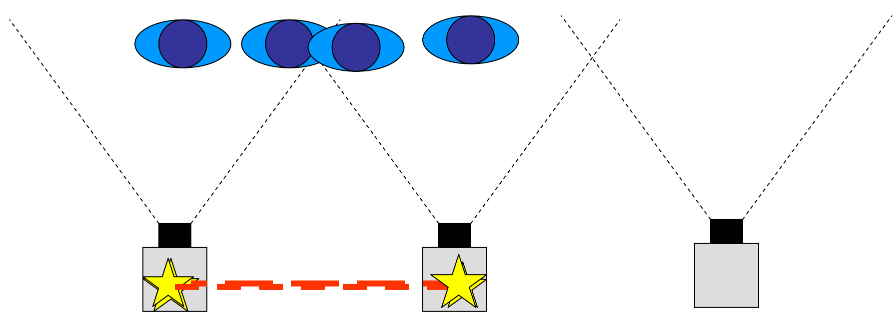
Distributed Vision Algorithms

- Starting points:
  – Single-camera algorithms
  – Multi-camera server-based algorithms
- New challenges:
  – No central repository
  – Communication delays

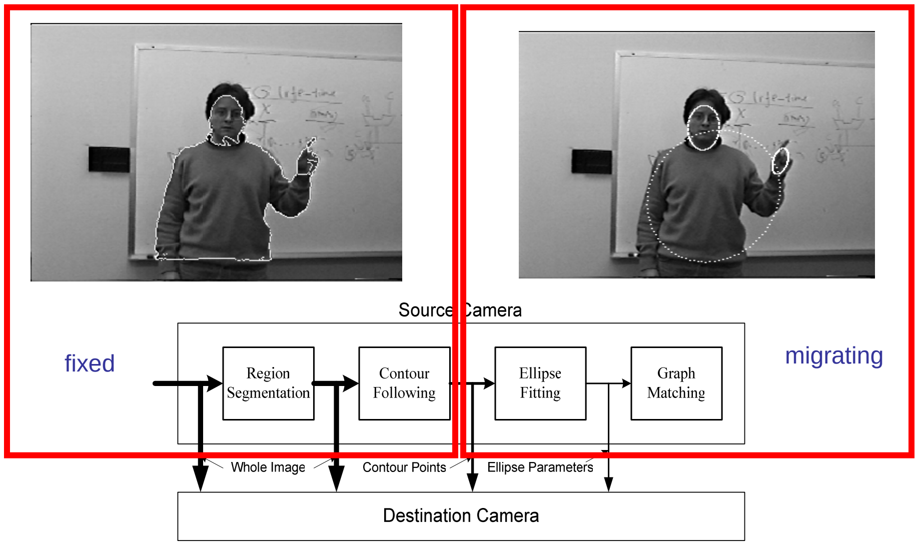
Example: Distributed Gesture Recognition

- *Lin et al*: recognize gestures using graph model
  - Started with single-camera algorithm
  - Split block diagram and inserted protocol to transform into distributed algorithm

Token Passing

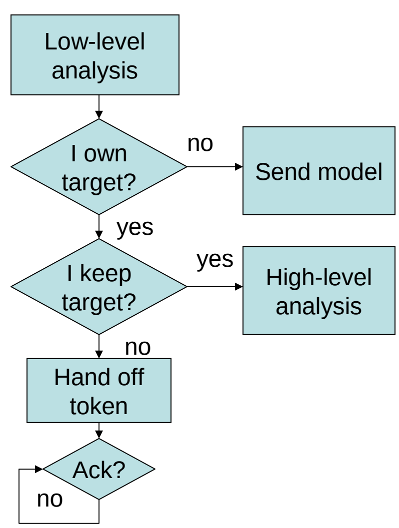


Algorithm Partitioning



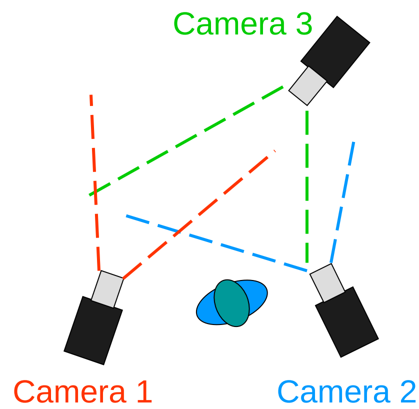
Token-Passing Protocol

- Token represents ownership of high-level analysis
- Nodes periodically determine when token should move
  - Use target centroid as heuristic
- Handshake transfers token



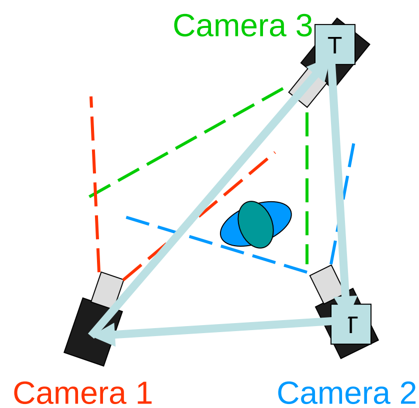
Peer-to-Peer Tracking

- *Velipasalar et al*: Nodes must agree on target identity
  - In the presence of multiple views
- Need distributed algorithm for agreement



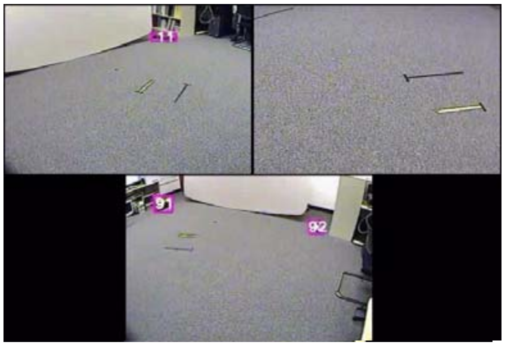
Distributed Tracking Agreement Protocol

- Each node runs a tracker for each target in its field of view
- Nodes form ring for synchronization communication
- Nodes share tracker information every N frames



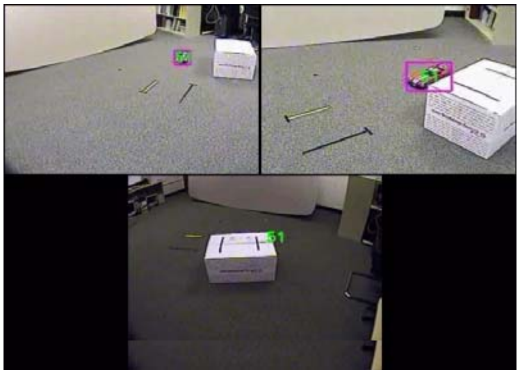
Label Consistency

- Cameras must agree on label for target in shared field of view
- Negotiate agreement and label at next communication round after target crosses field of view line

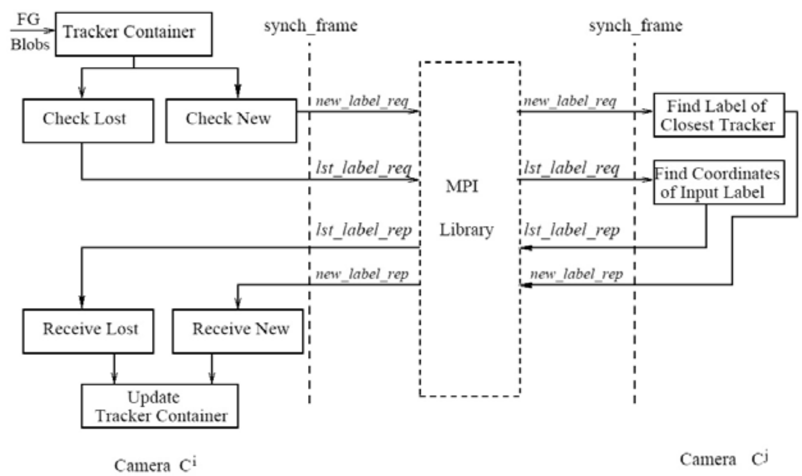


Target Position Sharing

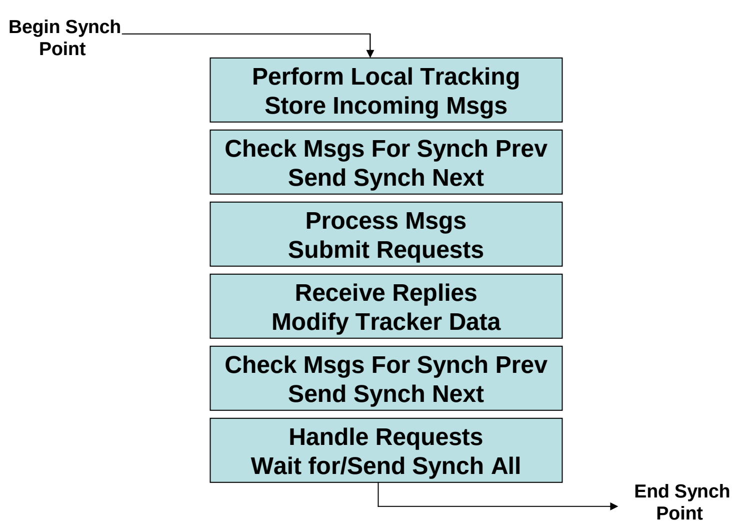
- When target is occluded, node can get position from other nodes
- Provides fault tolerance



Communication Diagram



Communication Protocol



Target Modeling

- We use a fast blob tracker:
  - Bounding box
  - Bhattacharya coefficient:

$$\rho(\mathbf{y}) \equiv \rho[p(\mathbf{y}), q] = \int \sqrt{p_{\mathbf{z}}(\mathbf{y}) q_{\mathbf{z}}} d\mathbf{z},$$

  - q**z** is density function of the feature **z** representing the color of the target model
  - p**z(y)** is the feature distribution of the FG blob centered at **y**

- All the usual problems:
  - Multiple views and models
  - Target merging and splitting

- We don't build target merge/split into the protocol---handled camera-by-camera

Video: Indoor Setup 1

The synch_rate is 1 frame

Video: Indoor Setup 2

The synch_rate is 30 frames

Video: Indoor Setup 3

The synch_rate is 60 frames

Spatial Calibration

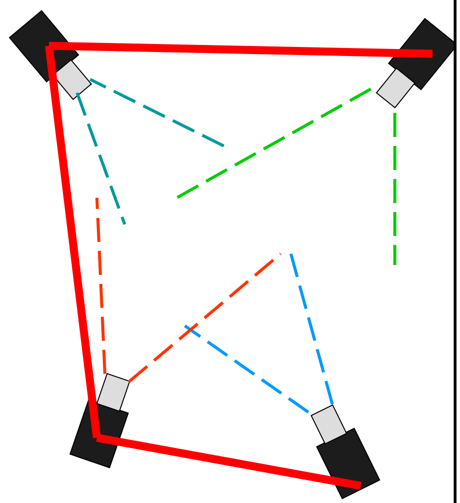
- Need to find camera position/orientation, possibly camera internal parameters
- All the usual multi-camera problems
- Algorithm must work without centralized server

Calibration using Belief Propagation

- *Devarajan and Radke*: Leverage dependencies introduced by camera geometries
- Belief propagation is used in sensor networks
  - Camera calibration introduces similarity transforms that make the problem more difficult

Vision Graph

- Nodes are connected if they share enough visible points
  - Stable accurate estimate of epipolar geometry can be obtained
- N(i): neighbors of node i
- Constructing initial estimate:
  - Estimate reconstruction from nucleus of i, N(i)
  - Use RANSAC to reject outliers
  - Adjust



Basic Belief Propagation

- Each node updates its belief using messages from neighbors
- Update equations:

Message from i to j at time t

Compatibility potential between i and j

Belief potential at i

$$m_{ij}^{t} \propto \int_{Y_i} \Psi(Y_i, Y_j)\phi(Y_i) \prod_{k \in N(i)/j} m^{t-1}{}_{ki}(Y_i)\,dY_i$$

$$b^{t}{}_{i}(Y_i) \propto \phi(Y_i) \prod_{j \in N(i)} m^{t}{}_{ji}(Y_i)$$

Camera Calibration

- Joint density of camera position based on camera observations can be factored:

$$p(Y_1, Y_2, \ldots, Y_M \mid Z_1, Z_2, \ldots, X_M) = \prod_{i \in V} p(Z_i \mid Y_i) \prod_{(i,j) \in E} p(Y_i, Y_j)$$

- Two cameras have some variables Yi, Yj that must agree
  - Enforce using binary selector matrix C

$$P(Y_i, Y_j) \propto \delta(C_{ij} Y_i - C_{ji} Y_j)$$

Update Algorithm

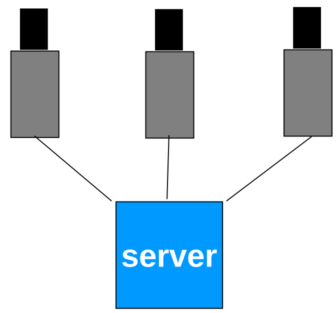
- Belief function: $\phi(Y_i) \propto p(Z_i \mid Y_i)$
- Compatibility potential: $\psi_{ij}(Y_i, Y_j) \propto \delta(C_{ij} Y_i - C_{ji} Y_j)$
- Belief update computation:
  - Can be calculated iteratively in pairwise computations

$$\mu_i^i \leftarrow [\Sigma_i^{-1} + \sum_{j \in N(i)} (\Sigma_i^{\ j})^{-1}]^{-1} (\Sigma_i^{-1} \mu_i + \sum_{j \in N(i)} (\Sigma_i^{\ j})^{-1} \mu_i^{\ j})$$

$$\Sigma^i_{\ i} \leftarrow [\Sigma_i^{-1} + \sum_{j \in N(i)} (\Sigma_i^{\ j})^{-1}]^{-1}$$

Temporal Calibration

- Server-based systems are easy to synchronize:
  - Gen-lock
  - Cabling
- Peer-to-peer systems cannot be implicitly synchronized
  - Analog synchronization is too expensive

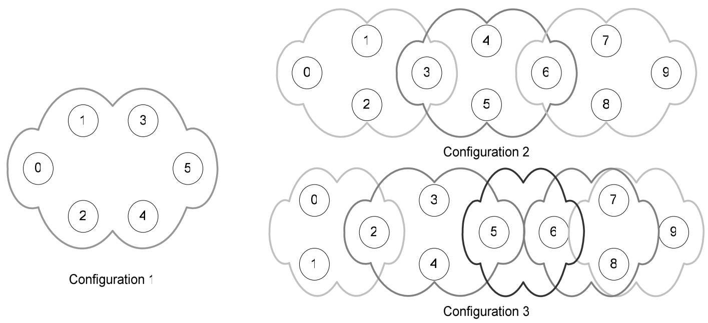


Timing Synchronization Algorithms

- *Lamport*:
  - Read clock value of nodes periodically
  - Use average clock difference to adjust clock
- *Lundelius*:
  - Nodes broadcast timestamp at expected time
  - Nodes gather timestamp messages
  - Use medium value to adjust clock
- *Halpern*:
  - Nodes broadcast timestamp at expected time
  - Nodes update its clock when a faster timestamp is received

System Configuration

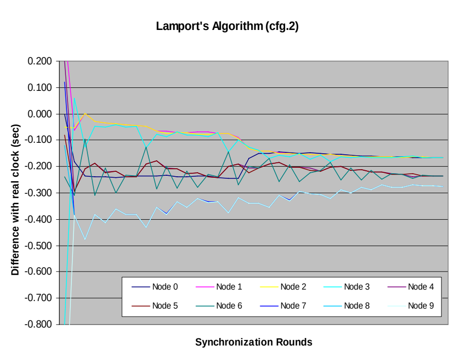
- *Lin et al*: Network configuration:

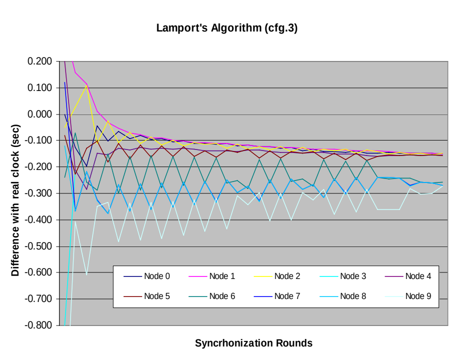


  - Configuration 1: 6 cameras in a single network
  - Configuration 2 & 3: 10 cameras distributed in several networks

Simulation Results

- Lamport's algorithm:
  - The 10 nodes can only be synchronized within 300 ms
  - Sub-networks can converge if initial difference is small
  - Clock value of routers might oscillate between average values of neighboring sub-networks



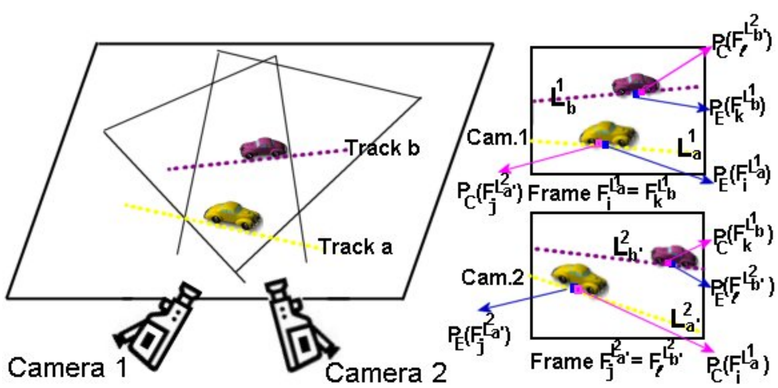


Simulation Results

- Synchronization requirements
  - Distributed gesture recognition system
    - Current: 15.23 frames/sec @ Pentium III 1.0 GHz
    - Goal: 30 frames/sec
  - All three algorithms can achieve 15 frames/sec requirement in configuration 1
  - Only Halpern's algorithm fulfills the requirements in configuration 2 and 3
  - Lamport's and Lundelius' algorithms need $O(n^2)$ messages in each round
  - Halpern's algorithm need amortized O(n) messages in each round
  - Halpern's algorithm advantages:
    - Simplest in computation
    - Most precise synchronization
    - Fewer message exchanges
    - Timestamps can be hidden within data messages

Image-based Temporal Calibration

- *Velipasalar*: Correlate images sequences to find synchronizing offset



Temporal Synchronization Algorithm

- Each camera calculates position of its target(s) in other camera's field-of-view
- Each camera finds a possible match in the frame sequence of other camera and identifies initial frame offset between cameras
- Perform a confidence check for each track pair
- Refine the match

Confidence Check and Refinement

- Not all target identifications will be equally good
  - Confidence check weights match by confidence value
- After confidence check, frame offset is refined using local search

Results without Final Refinement Step

Ground truth	100	200	300	400	500	800	1000
output	99	199	301	401	501	795	993
Accuracy (%)	99	99.5	99.67	99.75	99.8	99.37	99.3

References

- Wayne Wolf, Burak Ozer, and Tiehan Lv, “Smart cameras as embedded systems,” *IEEE Computer*, 35(9) September 2002, pp. 48-53.
- Chang Hong Lin, Tiehan Lv, Wayne Wolf, and I. Burak Ozer, “A peer-to-peer architecture for distrib-uted real-time gesture recognition,” in *Proceedings, International Conference on Multimedia and Exhibi-tion*, IEEE, 2004, vol. 1, pp. 27-30.
- Senem Velipasalar, Wayne Wolf, "Frame-Level Temporal Calibration of Video Sequences from Unsynchronized Cameras by Using Projective Invariants", *IEEE International Conference on Advanced Video and Signal Based Surveillance*, Como, Italy, September 15-16, 2005.
- Senem Velipasalar, Jason. Schlessman, Cheng-Yao Chen, Wayne Wolf, and Jaswinder Pal Singh, “SCCS: a scalable clustered camera system for multiple object tracking communicating via Message Pass-ing Interface,” in *Proceedings, International Conference on Multimedia and Expo,* IEEE, 2006.
- Zhaolin Cheng, Dhanya Devarajan, and Richard J. Radke, Determining Vision Graphs for Distributed Camera Networks Using Feature Digests. *EURASIP Journal on Advances in Signal Processing: Special Issue on Visual Sensor Networks*, Volume 2007, Article ID 57034, 11 pages.
- Dhanya Devarajan and Richard J. Radke, Calibrating Distributed Camera Networks Using Belief Propagation. *EURASIP Journal on Advances in Signal Processing: Special Issue on Visual Sensor Networks*, Volume 2007, Article ID 60696, 10 pages.
- C. H. Lin and W. Wolf, “A case study in clock synchronization for distributed camera systems,” *Proceedings of the SPIE*, vol. 5683, January 2005.

Outline

I. Introduction
II. Smart Camera Architectures
   1. Wireless Smart Camera
   2. Smart Camera for Active Vision
III. Distributed Vision Algorithms
   1. Fusion Mechanisms
   2. Vision Network Algorithms
IV. Requirements and Case Studies
V. Outlook