Multi-camera processing, analysis and applications

Andrea Cavallaro  
Centre for Intelligent Sensing  
Queen Mary University of London  
www.eecs.qmul.ac.uk/~andrea  
@smartcameras

Juan Carlos SanMiguel  
Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
www.vpu.eps.uam.es/~jcs  
@jc_sanmiguel

Outline

• Multi-camera processing: the past, the present and the future
• Information fusion
• Applications and practical challenges

• Simulation tools and WiSE-MNet
• Single camera object tracking
• Multi-camera communication: ideal vs realistic network conditions
• Hands-on examples

• Multi-camera object tracking
• Consensus algorithms and Distributed Particle Filters
• Task-optimised multi-camera coalitions
• Hands-on examples

• Resources: videos and additional code
• Concluding Q&A session and research outlook
Traditional approach

- Traditional multi-camera systems
  - stream raw images to a powerful centralised processing facility
  - limited scalability

Distributed smart cameras

- Key enabling technology for sensing the world
  - from centralised to distributed
  - processing at the nodes
  - exchange of metadata
  - distributed inference
  - scalability
Information fusion

• Centralized
  – nodes in the network send information to a single fusion center, which makes decisions

• Decentralized
  – nodes are clustered and communicate with a cluster-level fusion center, which makes decisions
  – fusion centers communicate among themselves

• Distributed
  – nodes communicate among themselves, normally with neighbors
Fields of view

(partially) overlapping FOVs  non-overlapping FOVs

FOV: field of view

$r_c$: communication range

●: viewing node
Smart-camera networks

• Smart cameras
  – in-built processing and communication capabilities
  – process large volumes of data
  – share high-data-rate messages
  – (generally) operate with limited resources

• How to effectively design and test new algorithms?

• Applications
  – search and rescue operations
  – wide-area surveillance
  – vehicular ad-hoc networks
  – smart cities & home automation

What are we looking for in a simulator?

• Configurability
  – scripting languages, configuration interfaces

• Abstraction levels
  – design complexity depends on researcher skills

• Support for standard sensing & communication protocols

• Extensibility
  – easy integration with third party libraries

• Open source
Simulators for camera networks

<table>
<thead>
<tr>
<th>code</th>
<th>type</th>
<th>resources</th>
<th>processing</th>
<th>comms</th>
<th>extendable</th>
<th>focus</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVVV</td>
<td>C++</td>
<td>3D</td>
<td>-</td>
<td>-</td>
<td>No</td>
<td>virtual worlds</td>
</tr>
<tr>
<td>(CVPR2007)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLCNR*</td>
<td>C++</td>
<td>3D</td>
<td>-</td>
<td>vision routines</td>
<td>Yes</td>
<td>virtual worlds</td>
</tr>
<tr>
<td>(JETCAS2013)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAMSIM</td>
<td>Java</td>
<td>2D</td>
<td>-</td>
<td>protocols</td>
<td>Yes</td>
<td>coordination</td>
</tr>
<tr>
<td>(SISO2013)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WSVN**</td>
<td>C++</td>
<td>2D</td>
<td>battery, clock, memory</td>
<td>wireless, MAC</td>
<td>~</td>
<td>video monitoring</td>
</tr>
<tr>
<td>(WMNCS2010)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M3WSN**</td>
<td>C++</td>
<td>2D</td>
<td>battery, clock, memory</td>
<td>wireless, MAC</td>
<td>code not released</td>
<td>multimedia TX</td>
</tr>
<tr>
<td>(SimuTools13)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WISE-Mnet**</td>
<td>C++</td>
<td>2D</td>
<td>battery, clock, memory</td>
<td>cameras &amp; trackers, wireless, dummy</td>
<td>Yes</td>
<td>camera networks</td>
</tr>
<tr>
<td>(SSPD2011)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Paid license required   **Requires the libraries: Omnet++ and Castalia

Simulators: references


• **Wireless Simulation Environment for Multimedia Networks**
  
  – based on Omnet++/Castalia
  – enables the simulation of distributed computer-vision algorithms at a high level of abstraction *(real communication protocols)*
  – models the communication layers, the sensing and distributed applications of camera networks *(resource constraints)*
  – networks capturing complex vectorial data *(e.g. video)*
  – includes implementations of state-of-the-art algorithms for smart-camera networks
  – flexible and extensible environment

Networked Computer Vision: the importance of a holistic simulator
J. SanMiguel and A. Cavallaro,
IEEE Computer, Vol. 50, No. 7, July 2017

• **Omnet++**

  • Generic discrete-event simulation engine
  • Generic modules interactions can be defined
    – behaviour is coded in C++
    – module interconnections defined by Network Description *(NED)* language
    – configuration ini files to set parameters
  • Highly flexible and extensible with external libraries
  • Network elements
    – nodes, protocols, channels
    – provided (externally) as simulation models *(INET, MiXiM, Castalia)*
Castalia

- Defines the wireless environment and the node architecture for
  - wireless sensor networks (WSNs)
  - body area networks (BANs)
  - networks of low-power embedded devices

WiSE-MNet: discrete event simulation

- Nodes operate independently
  - Omnet++ automatically starts nodes and physical processes
  - No linear script (Matlab) or main (C/C++ projects)

- Communication: message exchange between nodes
- Processing: received messages in discrete units

**Tic Toc example**

More info at
https://omnetpp.org/doc/omnetpp/tictoc-tutorial/
WiSE-MNet architecture

- WiSE-MNet extends Castalia for Wireless Camera Networks

WiSEXXX files → Castalia extensions

WiSE-MNet architecture: connectivity

- NED files describe internal/external connections
WiSE-MNet camera node

• Layer-wise simulation for smart camera operations
  – sensing
  – processing
  – communication

• Abstracts the key elements shared by these layers
  – framerate and framesize for sensing
  – memory and operating frequency for processing
  – bandwidth and transmission modes for communication

• Models power consumption for embedded hardware
• Implements algorithms to provide the functionality of each layer

WiSE-MNet camera node: application layer

• Implements decision-making
  – image/video processing
  – receive/send data
  – ...

Source file WiseAppTest.cc

Header file WiseAppTest.h
WiSE-MNet camera node: application layer

- Hierarchical layer development for complex applications

Comms (neighbours, interface)

FOVs (neighbours)

Template for target tracking (finite-state-machine)

Logic of the application (processing & data exchange)
WiSE-MNet camera node: sensing

- Physical processes: phenomena to measure
  - target detections: 2D (linear/random motion, variable speed, …)
  - video streams: 3D data (resolution, framerate, …)
- Sensor model **WiseCameraManager**
  (extends WiseBaseSensorManager module)
  - automatic sensing of existing physical processes
  - provides video feeds or target detections

![Diagram of WiSE-MNet camera node: sensing](image)

**WiSE-MNet camera node: sensing**

```plaintext
// Called when a new sample is ready
if (isSampleAvailable()) {
    // Check if the sample type is from the sensor manager
    if (sample.type == SensorManager.TYPE_SAMPLE) {
        // Create a new message
        newSample = createMessage(sample);
        // Send the new sample
        sendMessage(newSample);
    }
}
```
WiSE-MNet camera node: communication

• Communication channel
  – direct communication (avoids protocols, testing purposes)
  – dummy (ideal conditions)
  – wireless (packet loss, interferences,...)

• Via packets
  – defined in *.msg files
  – depend on the application
  – contains the data exchanged
    • variables
    • commands
    • images
    • ...

![WiseCameraICFMsg.msg](image)

WiSE-MNet camera node: communication

• Communication functionality hierarchically provided
  – WiseBaseApplication:
    direct message exchange with specific nodes
    connection with specific nodes (wireless/dummy)
    connection with neighbors within radio range (wireless/dummy)
  – WiseCameraApplication:
    connection with neighbors with co-visibility (wireless/dummy)

• Response to received packets or messages

```c
void WiseApplication::fromNetworkLayer(WiseApplicationPacket * rcvPacket) {
  const char *str, double rssi, double lqi;

  //... Method fromNetworkLayer()
```

![Implementation](image)
WiSE-MNet camera node: execution

- Timers to define the behavior of the node
  - sensing/processing rate
  - sequential execution
- Response to received data
  - automatically triggered
- **Startup()** method
  - App layer initialization
  - Set timers
  - Set variables

```cpp
#include "WiseAppTest.h"
using namespace std;

int main() {
  WiseAppTest::logger();
}
```

WiSE-MNet camera node: execution

Timer callback (repetitive tasks)
WiSE-MNet: examples (1/3)

- **Lifetime prediction**
  [SanMiguel and Cavallaro, TCSVT16]

Hardware modelled:
- processor ARM Cortex-A9 (0.25-1.5GHz)
- Sensor B3 (10-24MHz)
  [Likamwa et al., MOBYSIS13]
- Radio transceiver C2420 (max. 250kbps)

WiSE-MNet: examples (2/3)

- **Accuracy-consumption trade-off**
  [SanMiguel and Cavallaro, TCSVT2016]
WiSE-MNet: examples (3/3)

- View & role variability
  [SanMiguel and Cavallaro, TCSVT2016]

- Task: single-camera tracking

- How does performance (tracking accuracy) change with the reduction of resources?

Processing with a resource-limited camera

- **Task**: single-camera tracking

- How does performance (tracking accuracy) change with the reduction of resources?

Networked Computer Vision: the importance of a holistic simulator
J. SanMiguel and A. Cavallaro
IEEE Computer, Vol. 50, No. 7, July 2017
DEMONSTRATION 1

Target tracking with a resource-limited camera

Target re-identification
Target re-identification

- **Task**: transmit target features to other cameras
- What is the influence of a limited bandwidth and a growing number of cameras?

**DEMONSTRATION 2**

Target re-identification with realistic network conditions
Some additional definitions

- Viewing nodes
  - cameras observing the same target

- Combining the estimates of a node and its neighbours
  - local posterior
  - global posterior

- Multiple time scales
  - measurement collection (observations)
  - task allocation (coordination)
  - processing (iterations) between measurement collections

Multi-heartbeat operation

- Measurement collection
  - observation

- Task processing
  - execution the task
  - e.g. object localisation based on current observations and past information
Data sharing

Key elements

- Cameras
- Tasks
- Quality of Service
- Resources

Distributed and decentralized multi-camera tracking
M. Taj, A. Cavallaro

Image credit: PETS 2009 dataset http://www.cvg.reading.ac.uk/PETS2009/a.html
Key elements

- **Cameras** $C_i \in \{C_1, \ldots, C_N\}$
- Tasks
- Quality of Service
- Resources
Task processing: example

Continuous estimation of the state of an object (target) given a set of measurements (observations) obtained from spatially distributed cameras.

\[
Z_k = (z_k^1, z_k^2, \ldots, z_k^N)
\]
Measurements

\[
x_k = f(Z_k, Z_{1:k-1}, x_{0:k-1})
\]
State estimation

Fusion schemes

- Centralised
- Flooding
- Consensus
- Token passing
- Dynamic clustering
Centralised fusion

• Viewing nodes
  – send their local posteriors to a Fusion Centre (FC) that computes the global posterior

• Fusion Centre
  – sends the global posterior back to all nodes for them to predict the posterior for the next time step

• Disadvantages
  – suitable for small-scale networks only
  – high communication overheads near the FC
  – vulnerability to FC failures
  – limited robustness to topology changes

Flooding (or dissemination)

• Viewing nodes
  – broadcast their local posterior to all (or subsets) of nodes
    • single iteration with full network connectedness
    • multi-hop or multiple iterations with partial network connectedness
      – each node sends its own and the previously received information to neighbours
      – each node performs fusion and updates its local posterior

• Suitable
  – to share low amounts of data
  – with nodes with high connectivity

• Disadvantages
  – high communication and processing cost for large networks
  – high memory requirements
Consensus

• Each node exchanges information with neighbours
  – fusion performed by
    • average
    • gossip
    • maximum or minimum
  – speed of convergence depends on the number of nodes
  – non-viewing ones send zeros as information

• Advantages
  – availability of global estimates (e.g. posterior average) at all nodes
  – does not require
    • routing protocols
    • knowledge about nodes (e.g. observation models or FOVs)
    • knowledge about the network (communication graph)
  – robustness to node failures, topology changes, link failures

Distributed tracking: strategies

Distributed Kalman Consensus Filter (KCF)
- computes local estimates via Kalman Filters (KF)

Extended Information Filter (EIF) or Particle Filter (PF)
- with non-linear measurement or motion models
Example: Distributed Particle Filters (DPFs)

• Basics:
  – each node executes a local Particle Filter (PF)
  – measurements are synchronized, calibration is known
  – some information is exchanged

• Likelihood sharing
  – exchange information to have a common model of the likelihood
  – random number generators are synchronized

• Posterior sharing
  – the network has a common knowledge of the posterior pdf
  – consensus-based approach
  – aggregation-based approach
    • spatial sequence of aggregation steps
    • Partial Posterior (PP) is exchanged among the nodes

Distributed tracking: strategies
Token passing

• Sequential estimator
  – aggregation chain (AC) of viewing nodes
  – each node
    • receives a partial estimate from the previous one
    • updates this estimate using its local information
    • sends the result to the next node
  – next node selection
    • most informative node (based on prior knowledge)
    • requires global knowledge of the network
  – the process terminates when all nodes are visited once
    • the last node broadcasts the result (global posterior)
    • nodes in its communication range update their local posteriors
  – the last node initiates the AC for the next time step
    (often becoming the initial AC node)

Token passing

• Limitation
  – latency

• Suitable when
  – overlapping cameras are connected or
  – routing tables are provided
Problem: growing cost of particle dissemination through the network
Solution: Gaussian Mixture Model of the Partial Posterior (GMM-PP)

Independence from the # of particles

Distributed tracking with realistic networks

• Problems
  – most approaches are theoretical and designed for WSNs
  – need adaptation for limited Field-Of-View sensors (cameras)
    • detection miss | target hand-over | target loss
  – need mechanisms for the definition of the aggregation chain
    • first node → starts iteration
    • intermediate nodes → aggregate local measurement to the PP
    • last node → performs estimation

• Goal
  – To simulate realistic wireless camera networks

Distributed target tracking under realistic network conditions
C. Nastasi, A. Cavallaro

WISE-MNet: an experimental environment for Wireless Multimedia Sensor Networks
C. Nastasi, A. Cavallaro
First node

1. Knows previous posterior and local measurement

2. Prediction and Update:
   - re-sampling
   - draw from state-transition
   - weight update from likelihood

3. GMM-PP creation

4. Next-hop selection

5. Sends GMM-PP

\[
\{ x_{k-1}^{(i)}, u_{k-1}^{(i)} \}_{i=1}^P \xrightarrow{GMM-PP} \int_{GMM-PP}^1
\]

Intermediate node \( h \)

1. Receives PP from node \( h-1 \)

2. Importance sampling:
   - use the incoming PP as importance function \( g() \)
   - draw from importance function
   - weight update: CONDENSATION

3. GMM-PP creation

4. Next-hop selection

5. Sends GMM-PP

\[
\{ x_{k}^{(i)}, u_{k}^{(i)} \}_{i=1}^P \xrightarrow{GMM-PP} \int_{GMM-PP}^{1:h}
\]
1. Receives PP from node \( N-1 \)
2. Importance sampling as for intermediate nodes
3. Last PP is also the global PP
4. **Target state estimation**
5. **Next tracking step starts here!**

After importance sampling:

\[
\hat{x}_k = \sum_{i=1}^{P} u_k^{(i)} \cdot x_k^{(i)}
\]

**Distributed tracking: strategies**

**Consensus**

**Aggregation**
Camera network

Camera network

$\tau_c$ : communication range

$\bullet$ : viewing node

Consensus-based fusion

$\kappa$ : communicative neighbourhood of camera $i$

object detector

observation $z_i$

previous posterior $\hat{p}(x_{i-1} | Z_{i-1})$

Filter

local posterior $p_i = p(x_i | Z_i)$

consensus update $\hat{p}(x_i | Z_i)$

global posterior $\tilde{p}(x_i | Z_i)$

$\rho_j$ : communication with network node(s)
N-consensus

Neighbour consensus for distributed visual tracking
S. Katragadda, A. Cavallaro
Proc. of IEEE Intelligent Sensors, Sensor Networks and Information Processing, Singapore, April 2015

N-consensus: idea

$N_i$ : communicative neighbourhood of camera $i$

$\hat{z}$ : observation

$P(x_{i-1} | Z_{i-1})$ : previous posterior

$P(x_i | Z_i)$ : local posterior

$P(x_{i-1} | Z_{i-1})$ : extended information filter

$P(x_i | Z_i)$ : N-consensus update

$P(x_{i-1} | Z_{i-1})$ : global posterior

$r_c$ : communication range

$r_v$ : viewing range
N-consensus: idea

E.g. when $D = 2$
N-consensus: idea

E.g. when $D = 2$
N-consensus: idea

E.g. when $D = 2$

Distributed tracking: strategies
Multi-camera target tracking

• **Task**: cameras collaborate to track a target

• What is the cost of processing, communication and decision making?

• What is the influence of limited bandwidth and the number of cameras on the iterations of the algorithm?

Networked Computer Vision: the importance of a holistic simulator
J. SanMiguel and A. Cavallaro
IEEE Computer, Vol. 50, No. 7, July 2017
Multi-camera multi-target tracking

- **Task**: cameras collaborate to track multiple targets
- What is the influence of the iterations of the algorithm?
- What is the influence of the number of targets on the accuracy of the tracker?

*Networked Computer Vision: the importance of a holistic simulator*

J. SanMiguel and A. Cavallaro

IEEE Computer, Vol. 50, No. 7, July 2017
Multi-heartbeat scheduling

- **Task allocation**
  - tasks are chosen, assigned and coordinated (e.g. decision on which cameras operate)

- **Task processing**
  - execution of the task for achieving a certain goal (e.g. running the video processing algorithm)

Key elements

- **Cameras** $C_i$
- **Tasks** $T_j \in \{T_1, ..., T_M\}$
- Quality of Service
- Resources $R_i$

Final and intermediate goals of the camera network
Dynamic clustering

- Dynamic clusters
  - distributed (collaborative) process
    - iterative message exchange (negotiation)
      - to decide which nodes have target observations
      - to propagate the cluster over time
    - cluster membership adaptation depending on
      - target location (viewing nodes form clusters)
      - network topology
  - cluster head selection
    - fusion: decentralised process
      - cluster head fuses its own + received cluster information to generate the global estimate

- Inter-cluster data fusion
  - when multiple (single-hop) clusters are formed

- Multi tasking: different clusters for different tasks

Key elements

- Cameras $C_i$
- Tasks $T_j \in \{T_1, ..., T_M\}$
- Quality of Service
- Resources $R_i$

[Dynamic tasks: computational complexity depends on the visual data]

[Multi-target Target tracking]

[Kamal et al, CVPR 2013]
Key elements

- Cameras $C_i$
- Tasks $T_j$
- Quality of Service (QoS)
- Resources $R_i$

Examples
- track a target with the maximum accuracy
- minimize total energy consumption
- capture view of targets with high resolution

Performance criteria
- needed to quantify the success of tasks
Task allocation

Inputs
- sensing framerate
- frame resolution
- spatial location
- FOV
- range
- processing mode
- multiple cores
- operating frequency
- memory speed
- transmit/receive power
- communication range
- network protocol
- multi-hops
- resource usage

Objectives
- to maximize
  - energy efficiency
  - QoS
  - network lifetime
  - task performance
  - coverage
- to minimize
  - resource usage
  - latency
  - packet error rate
  - bandwidth usage

Constraints
- network connectivity
- co-visibility
- QoS
- coverage
- energy budget
- resource availability

Outputs
- optimal camera location
- camera scheduling
- camera activation
- extended lifetime
- improved task performance
- increased robustness
- optimal QoS

Mapping cameras to tasks

High flexibility

Multi-camera
single-target

Low allocation
complexity

Single-camera
single-target

High allocation
complexity

Multi-camera
multi-target

Low flexibility

one to one

many to one

many to many

one to many

Single-camera
multi-target
Task allocation: basics

• Scope
  – local
  – global

• Frequency
  – periodic
  – event-triggered

• Diversity
  – homogeneous capabilities
  – heterogeneous capabilities

In-node task

To maximize performance and minimize energy consumption by modifying sensing, processing and communication

• Examples:
  – Optimal sensing (framerate, resolution) to minimize energy consumption
    [LiKamWa et al, MOBYSIS 2013]
  – Feature selection to minimize the storage and computational cost
    [Tahir and A. Cavallaro, IEEE TCSVT 2014]
  – Power-rate-distortion model for wireless video communication under energy constraints
    [Z. He et al, IEEE TCSVT 2005]
  – Run-time workflow estimation for video-based applications
    [SanMiguel et al, MVAP 2013]
In-node task

- Single camera example: optimized sensing

Given the frame rate (R) and resolution (N), find the optimal configuration (active, idle and standby times) of an image sensor to achieve the lowest energy per frame.

Clock scaling

Energy Characterization and Optimization of Image Sensing Toward Continuous Mobile Vision
R. Likamwa, B. Priyantha, M. Philipose, Matthai, L. Zhong, P. Bahl
International conference on Mobile systems, applications, and services (MOBISYS), pages 69–81, 2013

Network-wide task

To maximize performance and minimize energy consumption by modifying sensing, processing and communication over multiple cameras jointly (cooperation)

- Strategies
  - centralized
  - distributed
    - distributed constraint optimization problems (DCOPs)
    - auctions
    - game theory
    - ...
  - decentralized
    - coalitions
Centralized solution

- Multi-objective optimization problem
  (1) inputs
  (2) required output
  (3) objectives
  (4) constraints

Task allocation: centralized example

to find the optimal number of cameras, their configuration and assign tasks

**Tasks:** observation points $T = \{t_1, \ldots, t_m\}$

**QoS:** pixels on target $pot_{t_i}$, framerate $fps_{t_i}$, algorithms $a_{t_i}$

**Algorithms:** composed of procedures $P_{t_i}$ with required resources $r_{t_i} = (c, m, e)$
- computation
- memory
- energy

**Camera FOV:** $\omega, \delta, \theta$

**Camera resources:**
- $r_i = (c, m, e)$
- maximum computation,
- maximum memory
- available energy

Resource-Aware Coverage and Task Assignment in Visual Sensor Networks
B. Dieber, C. Micheloni and B. Rinner
Task allocation: centralized

Objectives (optimization criteria)

<table>
<thead>
<tr>
<th>Min global energy usage</th>
<th>$\min \left( \sum_{n=1}^{N} e_{n} \cdot f_{pn} \right) $</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max overall/local lifetime</td>
<td>$\max \left( L_{i} \right) $</td>
</tr>
<tr>
<td>Min processing</td>
<td>$\min \left( \sum_{n=1}^{N} c_{n} \cdot f_{pn} \right) $</td>
</tr>
<tr>
<td>Max quality</td>
<td>$\max \left( \sum_{i=1}^{N} q_{i} \cdot P_{n} \right) $</td>
</tr>
</tbody>
</table>

Constraints

1) QoS: pixels on target $p_{ot_{i}}$, & framerate $f_{ps_{i}}$
2) Task: desired activity $a_{x}$ must be done
3) Every observation point covered
4) Available resources must not exceed allocated

Resource-Aware Coverage and Task Assignment in Visual Sensor Networks
B. Dieber, C. Micheloni and B. Rinner

Task allocation: centralized

- Some results...

Deployment of the cameras (black circles) on the monitored environment.

Resource-Aware Coverage and Task Assignment in Visual Sensor Networks
B. Dieber, C. Micheloni and B. Rinner
Task allocation: distributed

- Market-based approaches: auctions

**Auction:** protocol that allows agents to indicate their interests in one or more resources and that uses these indications to determine both an allocation of resources and a set of payments by the agents

[Shoham & Leyton-Brown 2009]

Task allocation: distributed

- Market-based approaches: auctions
  - object value dynamically assessed by the market
  - flexible: any object type
  - dynamic: interactive
  - automated
    - simple rules
    - low complexity of negotiations
  - revenue-maximising and efficient allocations are achievable

**Bidding value = utilities - costs**
Task allocation: distributed

**Multi-task bidding value for target** $T_i$
- camera energy $E_j$
- resource usage $R_j$
- framerate provided $F_{ji}$
- resolution provided $D_{ji}$

$$U_j(T_i) = \omega_E E_j + \omega_R R_j + \omega_F F_{ji} + \omega_D D_{ji}$$

Adaptive energy-oriented multitask allocation in Smart camera networks
C. Kyrkou, C. Laoudias, T. Theocharides, C. Panayiotou, M. Polycarpou

Task allocation: in-network (decentralized)

**Coalition:** pact among individuals or groups, during which they cooperate in joint action, each in their own self-interest, joining forces together for a common cause

http://en.wikipedia.org

Coalitions require data exchange (centralized or decentralized) through auctions, cooperative game theory,…
Coalitions

- Coalition-based processing stages for each camera $C_i$

Cost-aware coalitions for collaborative tracking in resource-constrained camera networks
J. SanMiguel and A. Cavallaro

Task allocation: in-network (decentralized)

- Manager selection: determine the coalition coordinator & exchange data with previous managers
- Utility estimation: evaluate quality of results & willingness to participate
- Cost estimation: estimate the costs to process & communicate
- Camera allocation: manager negotiates with cameras to be part of the coalition (marginal utility)

Coalitions that adapt to dynamic tasks

Cost-aware coalitions for collaborative tracking in resource-constrained camera networks
J. SanMiguel and A. Cavallaro
Task allocation: in-network (decentralized)

- Marginal analysis: select optimal cameras for coalitions
  - Contribution of $c_i$ using marginal utility theory [Zopounidis2010]
    $\gamma(c_i, \tilde{L}^k) = MU(c_i, \tilde{L}^k) - \sum_{s=1}^{N_{cost}} \lambda_s MC_s(c_i, \tilde{L}^k)$
    $c_i$ - $i^{th}$ camera
    $\tilde{L}^k$ - sub-set of viewing cameras

- Sequential selection of best camera $c_i^*$ to join coalition considering available resources (battery and computation)
  $c_i^* = \arg\max \gamma(c_i, \tilde{L}^k)$

Cost-aware coalitions for collaborative tracking in resource-constrained camera networks
J. SanMiguel and A. Cavallaro

Task allocation: in-network (decentralized)

- Marginal analysis: example

Global vs local decision
Task allocation: in-network (decentralized)

- Cost-aware coalitions
  - High energy savings
  - Small error increase

Camera coalitions for multi-target tracking

- **Task**: cameras form coalitions to track multiple targets

- What is the influence of the iterations of the algorithm?

- What are the benefits of forming coalitions of cameras?
DEMOSTRATION 5

Camera coalition formation for multi-target tracking

What did we learn today?

• Collaboration protocols for cameras (information fusion)
• The key elements for autonomous decision making
• The basic principles of task allocation and processing

• How to avoid common errors and misconceptions about information processing in camera networks

• How to implement and test the concepts learnt today using a specific simulator: WiSE-MNnet
• How to extend the simulator with new algorithms
Summary

• Networks of wireless (battery powered) smart cameras
  – challenges related to sensing, processing & communication
  – Wise-MNet simulator to account for all constraints

• Fusion schemes for camera networks
  – centralised, decentralised, distributed
  – flooding, consensus, token passing and dynamic clustering
  – collaboration / cooperation / allocation strategies / coordination / coalitions

• Cameras may share resources to improve performance
  – resources mostly depend on hardware capabilities
  – resource usage implies various costs: latency, energy,…
  – power consumption is one of the most important costs

Online resources with source code

Distributed visual processing

• Neighbour consensus for distributed visual tracking
  S. Katragadda, A. Cavallaro,
  IEEE Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Apr 2015

• The costs of fusion in smart camera networks
  S. Katragadda, J. SanMiguel, A. Cavallaro,
  ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC), Nov 2014

• Consensus protocols for distributed tracking in wireless camera networks
  S. Katragadda, J. SanMiguel, A. Cavallaro,
  IEEE Int. Conf. on Information Fusion (FUSION), July 2014

  http://www.eecs.qmul.ac.uk/~andrea/software.htm

Resource-aware visual processing

• Low-cost multi-camera object matching
  S.F. Tahir, A. Cavallaro
  IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP), May 2014

• Cost-effective features for re-identification in camera networks
  S.F. Tahir, A. Cavallaro
  IEEE Transactions on Circuits and Systems for Video Technology, 2014

  http://www.eecs.qmul.ac.uk/~andrea/matching
Online resources with source code

Utility estimators

• Adaptive on-line performance evaluation of video trackers
  J. SanMiguel, A. Cavallaro and J. Martínez
  http://www.vpu.eps.uam.es/publications/TrackQuality/

• Temporal validation of particle filters for video tracking
  J. SanMiguel and A. Cavallaro
  Computer Vision and Image Understanding, 131(1):42-55, February 2015
  http://www.vpu.eps.uam.es/publications/PFConsistency

Power modeling

• Energy consumption models for smart-camera networks
  J. SanMiguel and A. Cavallaro
  IEEE Trans. on Circuits and Systems for Video Technology, in press
  http://www.vpu.eps.uam.es/publications/camera_power/

Simulator: WISE-MNet

• Networked Computer Vision: the importance of a holistic simulator
  J. SanMiguel and A. Cavallaro
  IEEE Computer, Vol. 50, No. 7, July 2017
  http://www.eecs.qmul.ac.uk/~andrea/wise-mnet.html

References

References


References


References


References


Acknowledgements

- C. Nastasi
- S. Katragadda

High Availability Video Analysis for People Behaviour Understanding
http://www.vpu.eps.uam.es/HAVideo/
Supported by the Ministerio de Economía y Competividad (Spanish Government)

EU Artemis JU project COPCAMS
Cognitive & perceptive Cameras
http://www.copcams.eu

EU FP7 project CENTAUR
Crowded environments monitoring for activity understanding and recognition
http://www.centaur-project.eu

EU Erasmus Mundus Joint Doctorate ICE
Interactive & Cognitive Environments
http://www.icephd.org

Appendix
WiSE-MNet: installation

1. Install dependencies of Omnet++
2. Install Omnet++
3. Install dependencies of OpenCV
4. Install OpenCV
5. Download WiSE package*
6. Setup a project using the WiSE package*

*Identical installation for Castalia (not required as it is included in WiSE)

Note - runs in Linux (runs also in Windows but without OpenCV)

WiSE-MNet: GUI
WiSE-MNet: creating an app (1/3)

- Create new algorithms
  - build on top of existing functionality

```
Comms (neighbours, interface)

WiseBaseApplication

WiseCameraApplication

WiseCameraManager

Interface for camera sensing

FOVs (neighbours)

WiseCameraSimplePeriodicTracker

Template for target tracking
(finite-state-machine)

Logic of the application
(processing & data exchange)

WiseCameraDPF
```

WiSE-MNet: creating an app (2/3)

- Create new trackers
  - subclasses of WiseCameraSimplePeriodicTracker
  - finite-state-machines for node execution using timers

```
Startup

INIT

t = 0

WAIT FIRST SAMPLE

t = sample_life

WAIT END SAMPLE

At_end_sample()

WAIT SAMPLE

At_sample()

t = sample_tracker - sample_life

At_first_sample()

Init resources()

Defining sampling instants

At_first_end_sample()

```

```
```
```
```
```
```
WiSE-MNet: creating an app (3/3)

Functions to implement from tracking template

Functions to implement from base template

WiSE-MNet: running an app (1/4)

Configuration (ini files)
WiSE-MNet: running an app (2/4)

Configuration (ini files)

WiSE-MNet: running an app (3/4)

Debug configuration
WiSE-MNet: running an app (4/4)