Consensus Algorithms for Camera Sensor Networks

Roberto Tron
Vision, Dynamics and Learning Lab
Johns Hopkins University
Camera Sensor Networks

Motes
- Small, battery powered
- Embedded camera
- Wireless interface
- Limited processing power

Limited resources
We cannot use existing algorithms
Challenges to Existing Technologies

**Traditional computer vision**
- Algorithms are centralized: all images are sent to the same node for processing

*Camera sensor networks have limited resources*
- Limited processing power
- Limited memory
- Slow wireless channel
- Nodes can only communicate with neighbors

**Computer vision algorithms require resources not available in a camera sensor network**
Traditional distributed algorithms for sensor networks

• *Consensus algorithms* estimate low-dimensional, Euclidean quantities from simple measurements across the network.

In computer vision applications:

• Measurements (images) are high-dimensional.
• Measurements are corrupted by noise, outliers.
• Estimates are non-Euclidean (e.g., rotations) and different from the measurements.

Advantages:

• Each node needs to communicate only with its neighbors.
• Low memory requirements.
• Mild conditions on the network topology.

Consensus algorithms cannot be directly used for computer vision applications.
Our Research

Extend consensus algorithms to obtain distributed solutions for computer vision problems
Averaging Consensus Algorithms

Assume each node has an initial scalar measurement $u_i$ and we want to compute the average $\bar{u}$ across the network.

Consensus:
1. Represent the network as

$$G = (V, E)$$

Nodes

Links

$$\bar{u} = \frac{1}{N} \sum_{i \in V} u_i$$
Averaging Consensus Algorithms

2. Define a cost function as sum of squared differences between neighboring nodes

\[ \varphi = \frac{1}{2} \sum_{(i,j) \in E} (x_i - x_j)^2 \]

3. Apply gradient descent and obtain a distributed algorithm

\[ x_i(t+1) = x_i(t) + \varepsilon \sum_{(i,j) \in E} (x_j(t) - x_i(t)), \quad x_i(0) = u_i \]

Minimum achieved for \( x_i = x_j, \forall (i,j) \in E \)

4. All the nodes converge to the average

\[ \lim_{t \to \infty} x_i(t) = \bar{u}, \forall i \]

Convergence relatively easy to show
Research Directions

**Task:** use a camera network to locate an object of interest

**Challenges**
- Localizing the object
- Localizing the cameras
- Dealing with packet losses
- Discovering the vision graph
Distributed Network Localization

**Goal:** find relative positions and orientation using images only

**Approach**
1. Estimate relative pose for cameras with overlapping fields of views
2. Integrate pairwise measurements across the network

**Challenges**
- Measurements are not consistent: going around a cycle does not give the identity transformation
- Rotations are not Euclidean quantities
- Translations can be estimated only up to scale
- Global scale ambiguity
Proposed solution

Minimize sum of square distances between estimates and measurements

\[
\begin{align*}
R_{ij} &= R_i^T R_j \\
T_{ij} &= R_i^T (T_j - T_i)
\end{align*}
\]

\[
\min_{R_i, T_i} \sum_{i \sim j} \frac{1}{2} \left( \sum_{k} \eta_{ij,k} (R_i R_j R_k - \tilde{R}_{ij,k}) \right)^2 + \frac{1}{2} \left( \sum_{k} \eta_{ij,k} (R_i (T_j - T_i) - \tilde{T}_{ij,k}) \right)^2 - \lambda_{ij} \sum_{k} \eta_{ij,k} \tilde{t}_{ij,k} \geq 1
\]

- Use the Riemannian distance in the space of rotations \(SO(3)\)
- Reparametrize the solution to enforce global consistency
- Add variables to account for unknown scales
- Introduce constraints to fix the global scale ambiguity

Use Riemannian gradient descent for the minimization

\(^2\text{Tron, Vidal - “Distributed Image-Based 3-D Localization of Camera Sensor Networks”}

Appeared at CDC09
Distributed Network Localization

**Experiment:** 8 cameras, 30 scene points, 4-regular graph

Rotation errors (degrees)

<table>
<thead>
<tr>
<th>Noise</th>
<th>0 pixels</th>
<th>1 pixels</th>
<th>3 pixels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>0.00 (0.00)</td>
<td>2.77 (0.58)</td>
<td>4.80 (1.74)</td>
</tr>
<tr>
<td>Final</td>
<td>0.00 (0.00)</td>
<td>0.13 (0.00)</td>
<td>0.39 (0.03)</td>
</tr>
</tbody>
</table>

Translation direction errors (degrees)

<table>
<thead>
<tr>
<th>Noise</th>
<th>0 pixels</th>
<th>1 pixels</th>
<th>3 pixels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>0.00 (0.00)</td>
<td>0.11 (0.00)</td>
<td>0.33 (0.05)</td>
</tr>
<tr>
<td>Final</td>
<td>0.00 (0.00)</td>
<td>0.09 (0.00)</td>
<td>0.29 (0.03)</td>
</tr>
</tbody>
</table>

Geometric variance of scale ratios

<table>
<thead>
<tr>
<th>Noise</th>
<th>0 pixels</th>
<th>1 pixels</th>
<th>3 pixels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final</td>
<td>1.000</td>
<td>1.002</td>
<td>1.005</td>
</tr>
</tbody>
</table>
Distributed Object Localization

Each camera can detect the pose of an object (e.g. a face)

**Goal:** Averaging consensus on the pose of the object

For the translation part: Use standard consensus

For the rotation part: We need to derive a new algorithm
Object localization

Consensus in the manifold

Define the cost function and minimize using gradient

\[ \varphi = \frac{1}{2} \sum_{(i,j) \in E} (x_i - x_j)^2 \]

Same as consensus, but use Riemannian distance

\[ \varphi = \frac{1}{2} \sum_{(i,j) \in E} d^2_{SO(3)}(S_i, S_j) \]

Result

- Converges to a consensus configuration \( S_i = S^* \)
- In general \( S^* \) is not equivalent to a centralized solution

We proposed a distributed algorithm that obtains the same results as the centralized solution

\(^1\text{Tron, Vidal, Terzis - “Distributed pose averaging in camera networks via consensus on SE(3)” ICDSC08}\)
Current / Future work directions

Discovery of the vision graph
• How to find matching images?
• What information to share?
• How to route information?

Robust consensus
• Solutions robust to packet losses and outliers

Distributed machine learning
• Use consensus for dimensionality reduction and classification
• E.g. object classification
Conclusion

Camera Sensor Networks
• Represent a new, interesting venue of research
• Require re-thinking algorithms from both computer vision and sensor networks communities
• Use consensus to solve problems beyond simple averaging

Acknowledgments
• This work was funded by the NSF grant “Distributed Sensing via Robust Consensus on Manifolds” (CNS-0834470)
• We would like to thank our collaborators Ehsan Elhamifar, Yin Chen, Prof. René Vidal, Prof. Andreas Terzis and Prof. I-Jeng Wang
Centralized average of rotations

Definition of mean in the space of rotations

**Karcher mean**

\[
\overline{S} = \text{argmin}_{S \in SO(3)} \sum_{i \in W} d_{SO(3)}^2(S, S_i)
\]

Centralized algorithm

\[\overline{S} = S_0\]

Do

\[w = \frac{1}{N} \sum_{i \in V} \log \bar{S}(S_i)\]

If \[\|w\| < \delta\]

Break

End

No closed form solution, only a Riemannian gradient descent iterative algorithm

1) Map rotations to tangent space

2) Average tangent vectors

3) Update estimated mean
Object localization

Result

• Converges to a consensus configuration, $S_i = S^* \ \forall i$
• In general, $S^*$ is not equivalent to the centralized solution $\bar{S}$

Why

In the Euclidean case, the average is maintained between iterations

$$\text{mean}\{x_i(t)\} = \text{mean}\{x_i(t + 1)\}$$

This is not true in the Riemannian case!

---

$^1$Tron, Vidal, Terzis - “Distributed pose averaging in camera networks via consensus on SE(3)”
Appeared at ICDSC08
Another approach: Consensus in the tangent space

Map the centralized algorithm to a distributed equivalent

Centralized algorithm

\[ \overline{S} = S_0 \]

Do

\[
\begin{align*}
    w &= \frac{1}{N} \sum_{i \in V} \log \overline{S}(S_i) \\
    \text{If } \|w\| < \delta & \quad \text{Break} \\
    \overline{S} &= \exp_{\overline{S}}(w)
\end{align*}
\]

End

Compute this average of tangent vectors with consensus

Converges to the correct average!

Disadvantages

- Needs appropriate initialization
- Each iteration requires a full consensus

---

\(^1\)Tron, Vidal, Terzis - “Distributed pose averaging in camera networks via consensus on SE(3)”
Appeared at ICDSC08
Object localization

Consensus in the manifold
• Simple initialization
• Does not converge to the correct average

Consensus in the tangent space
• Hard initialization
• Converges to the correct average

Final solution
• Use one to initialize the other
Problem: for large camera networks, manual camera calibration is impractical

Proposed approach
• Calibrate only a few cameras
• Propagate calibration to other cameras (solution of a linear equation system)
• Obtain the 3D structure with consensus

Limits
• Number of estimated parameters depends on topology
• No globally optimal solution (yet!)

---

Elhamifar, Vidal – “Distributed Calibration of Camera Sensor Networks”
Appeared at ICDSC09
Robust consensus (ongoing work)$^4$

**Problem**

- Real-life networks are subject to packet losses
- Consensus does not work in the case of asymmetric losses

\[ \text{mean}\{x_i(t)\} \neq \text{mean}\{x_i(t + 1)\} \]

**Proposed solution**

- Add new state variables and new corrective iterations
- Converge to the correct average even with packet losses

$^4$In collaboration with Yin Chen and Andreas Terzis
How to find the vision graph?
Current algorithms use flooding
1. Extract features from images
2. Transfer features from all to all

Problem
• Does not scale well

Is there a better way?
Object Recognition (future work)

Learning
• In-network learning of models for recognition (think of eigenfaces)
• Distributed PCA and Distributed GPCA

Recognition
• Integrate information from multiple views to better recognize objects

Use consensus to solve these problems
Results

**Experiment:** 20 localized cameras, with random placement

Histograms of rotation errors

**Initial**

**Consensus in the manifold**

**Final algorithm**
Camera Networks

**Motivation:** Monitoring and surveillance of a large environment

**Classic approach:** multiple cameras wired to a central processing unit

**Problems**
- Wiring makes it hard to deploy and reduces flexibility
- Central processing does not scale well with the number of nodes
Motes are
• small,
• battery powered devices with
• an embedded camera
• a wireless interface
• limited processing power
Motes can form Camera Sensor Networks that are easily deployable and fault tolerant.

Can we use camera sensor networks for computer vision?
Classical Average Consensus

Assume each node has an initial scalar measurement $u_i$ and we want to compute the average $\bar{u}$ across the network.

Consensus Approach:
1. Represent the network as a graph $G = (V, E)$

$$\bar{u} = \frac{1}{N} \sum_{i \in V} u_i$$
Classical Average Consensus

2. Define a cost function as sum of squared differences between neighboring nodes

\[ \varphi = \frac{1}{2} \sum_{(i,j) \in E} (x_i - x_j)^2 \]

3. Apply gradient descent and obtain a distributed algorithm

\[ x_i(t + 1) = x_i(t) + \varepsilon \sum_{(i,j) \in E} (x_j(t) - x_i(t)), \quad x_i(0) = u_i \]

Minimum achieved for \( x_i - x_j, \forall (i, j) \in E \)

Property: \( \text{mean}\{x_i(t)\} = \text{mean}\{x_i(t + 1)\} \)
3. If the graph is connected, the nodes converge to the minimum of the function, where all the nodes reach consensus on the global average

\[
x_1(t_\infty) = x_2(t_\infty) = \ldots = x_1(t_\infty) = \bar{u} = \frac{1}{N} \sum_{i \in W} u_i
\]