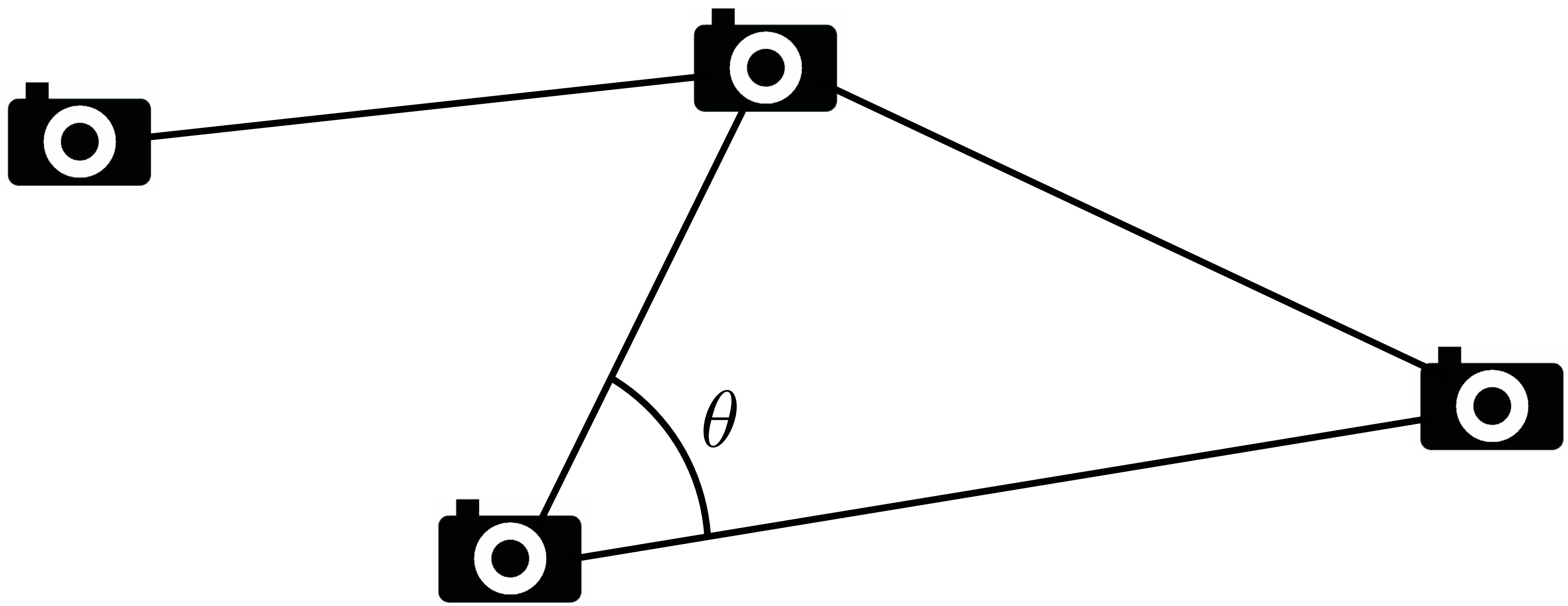
Identifying maximal rigid components in bearing-based localization

Ryan Kennedy
Oleg Naroditsky
Kostas Dandiilidis
Camillo J. Taylor
University of Pennsylvania

Network localization

- Low-cost sensors are becoming more available
- Used in many monitoring applications
  - Environmental (floods, pollution, climate change, etc.) (Hart and Martinez, 2006)
  - Surveillance (He et al., 2006)
- Useless without location of sensors
  - GPS can be expensive/unreliable

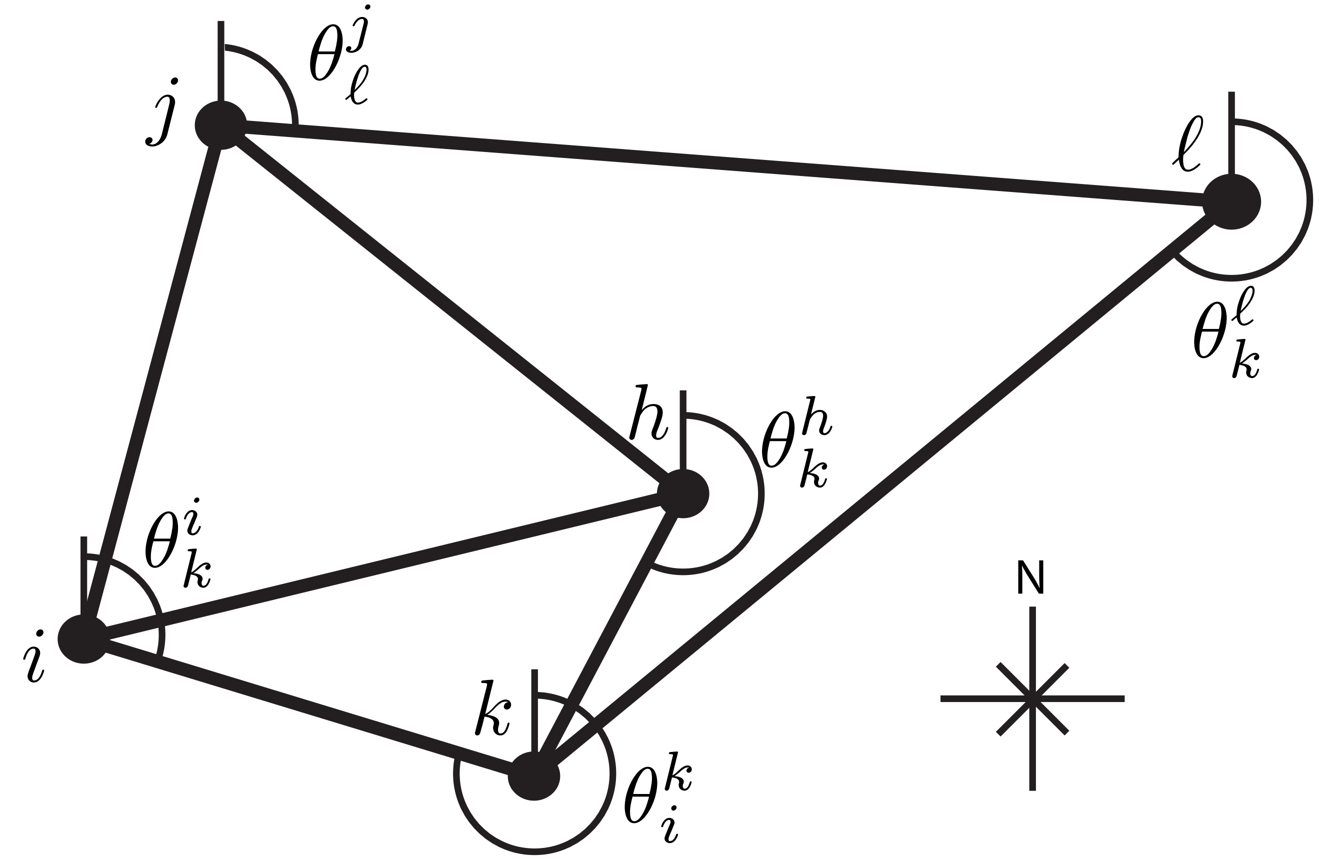
Camera network localization



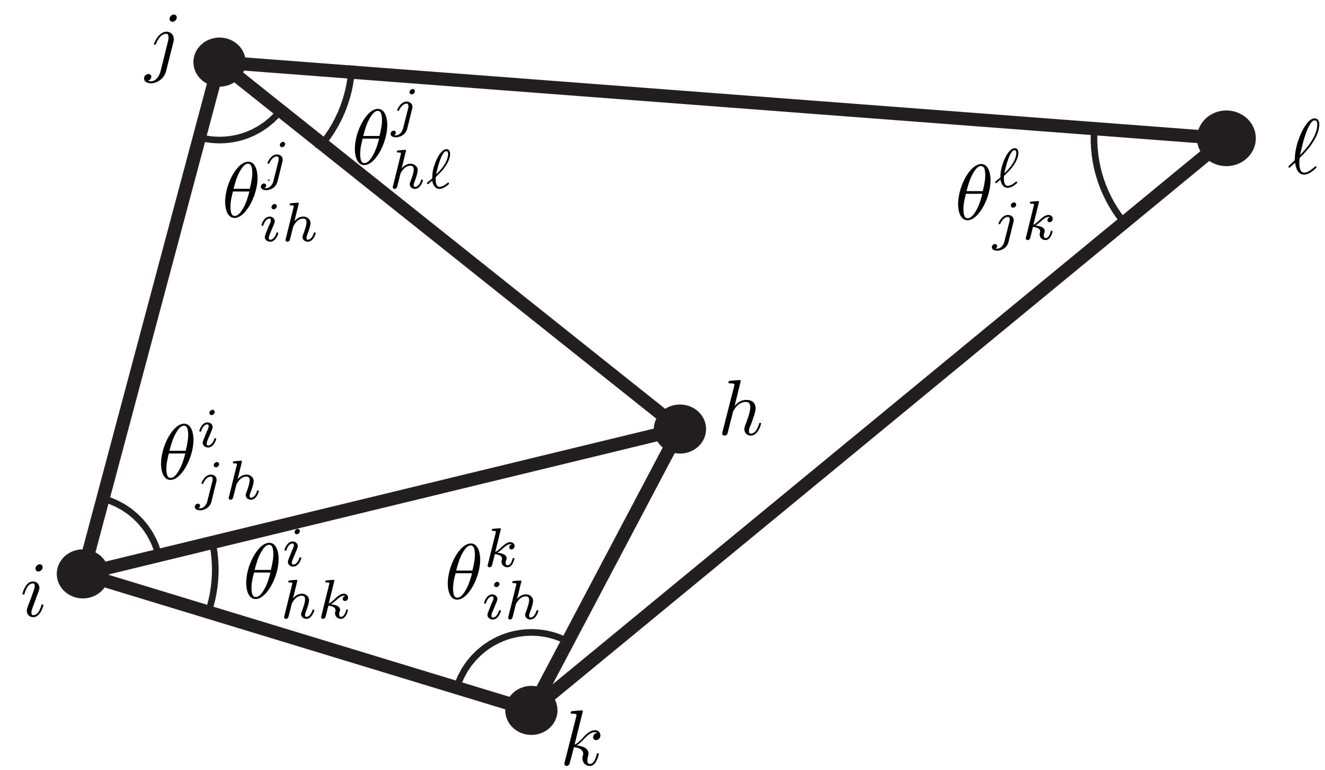
- Each camera can see a subset of others
- Camera can only measure **angles**, not distance
- What can we say about the **global** layout of cameras based on **local** measurements?

Two situations

1. Global coordinate frame (e.g., each node has a compass)



2. No global coordinate frame



Two questions

1. Based on only **local** angle measurements can we determine the possible ways that network is organized?

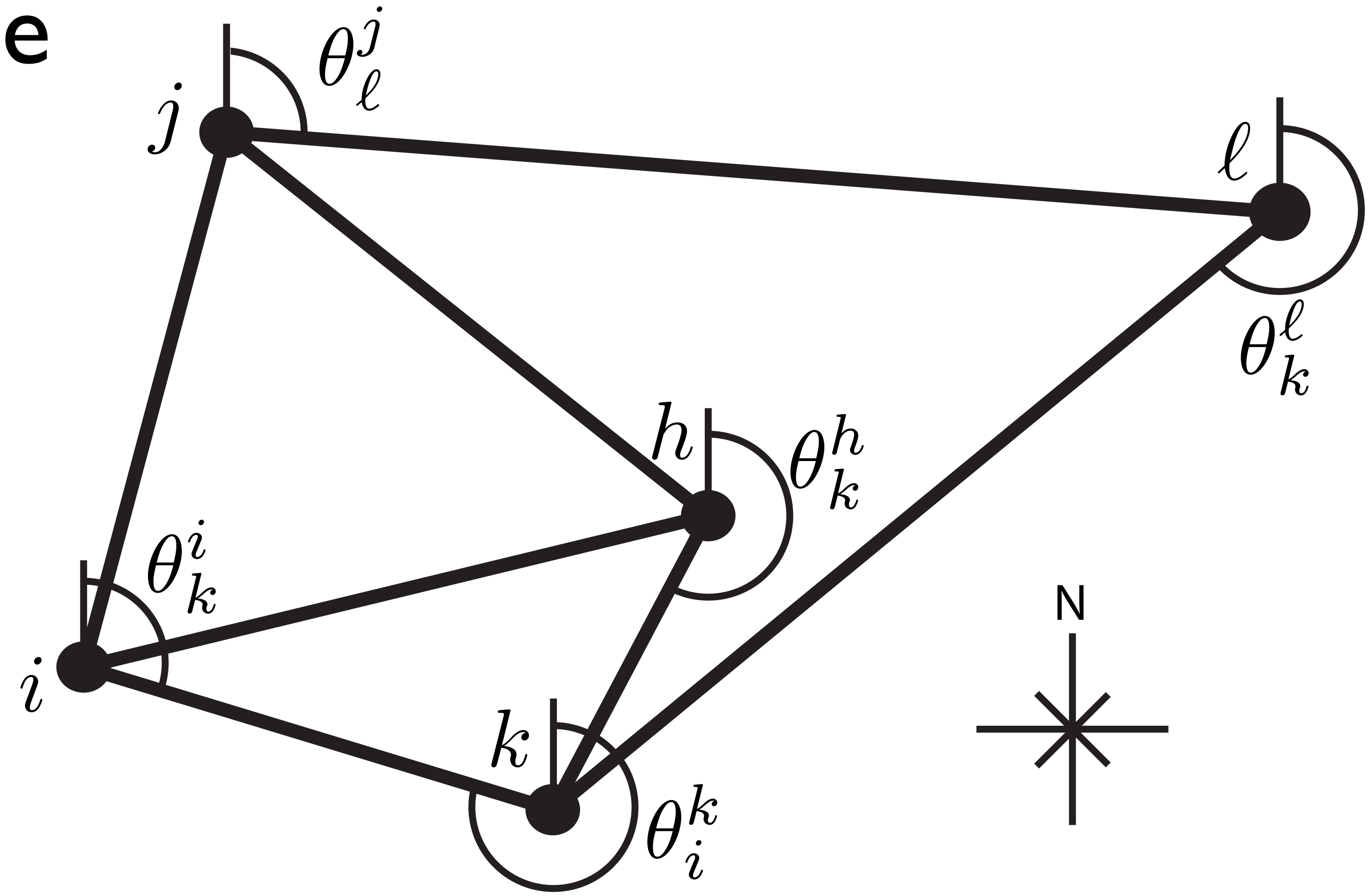
2. Solutions may not be unique; can we find subnetworks which have a unique (up to scale and translation) solution?

   - We can find a unique solution for these subnetworks

   - Tells us where we need to add more constraints

Global coordinate frame

- All nodes have access to a **global reference frame**
- Angles are measured with respect to this reference



Global coordinate frame

- Each angle measurement can be written as a **linear constraint**

$$\mathbf{a}_i^T \mathbf{x} = 0$$

- All such constraints can be written as

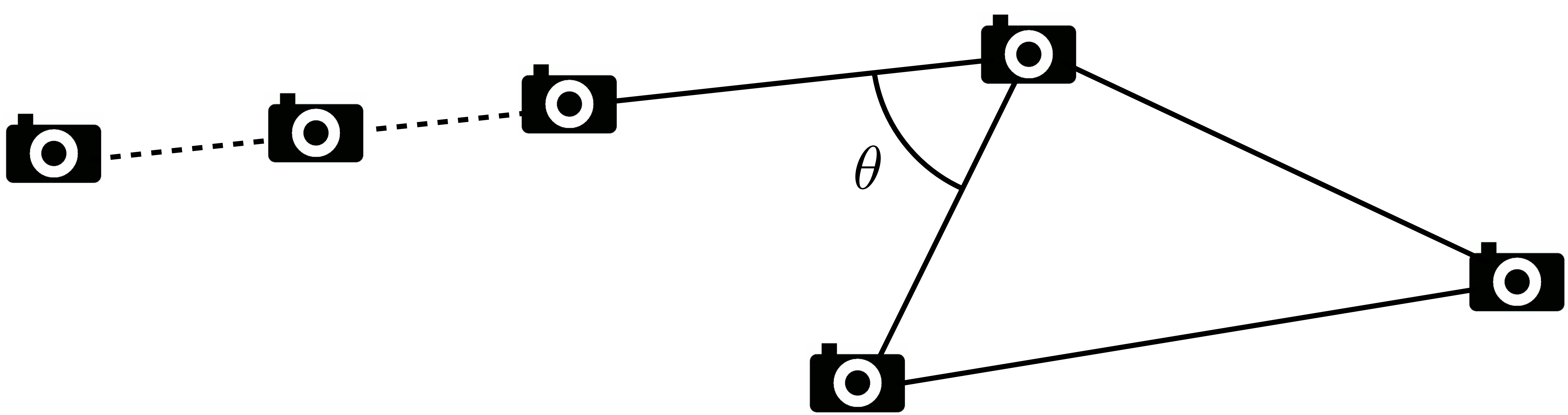
$$\mathbf{A}\mathbf{x} = \mathbf{0}$$

- **Solution**: any vector in the **null space** of $\mathbf{A}$

- (Brand et. al, 2004) showed that with noise, the optimal solution is given by the eigenvectors corresponding to the smallest eigenvalues of $\mathbf{A}$

Solution may not be unique

- Multiple connected components
- Co-linear constraints:

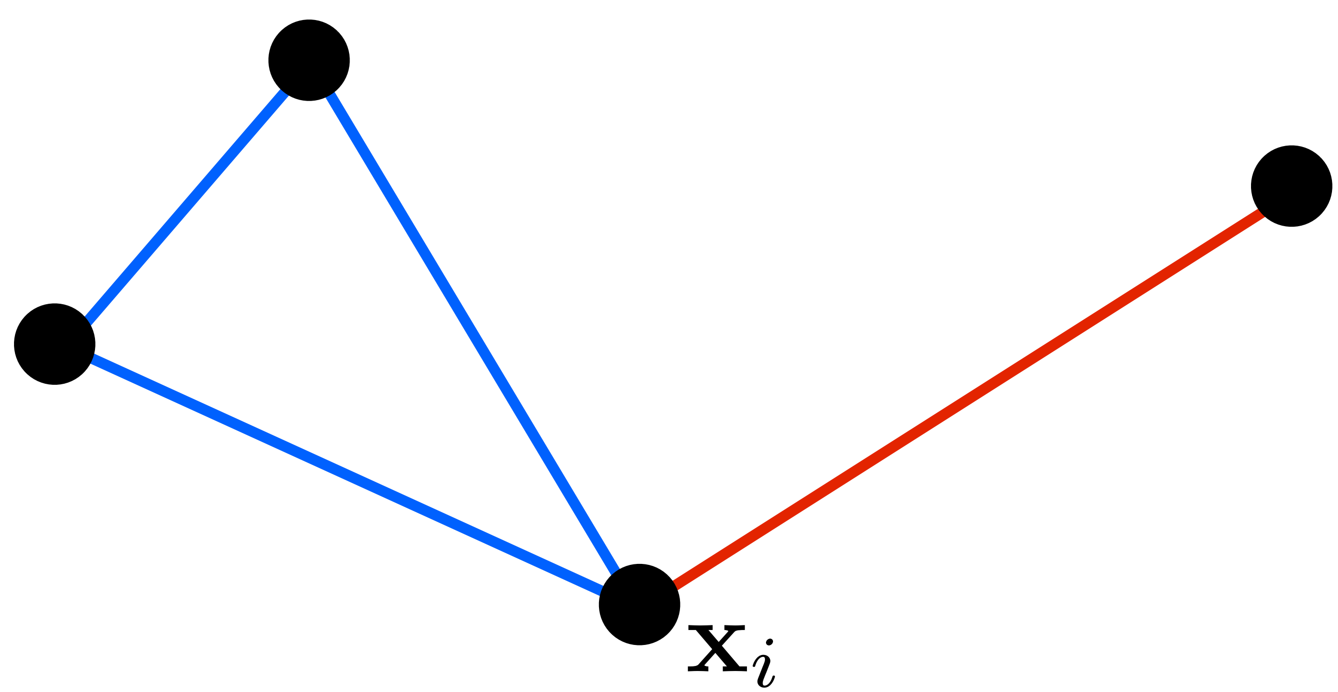


Solution may not be unique

- Can we determine which subproblems are **rigidly-constrained**?
  - Such subnetworks have a unique solution (up to scale and translation)
  - Tells us where to add more constraints

Finding rigid components

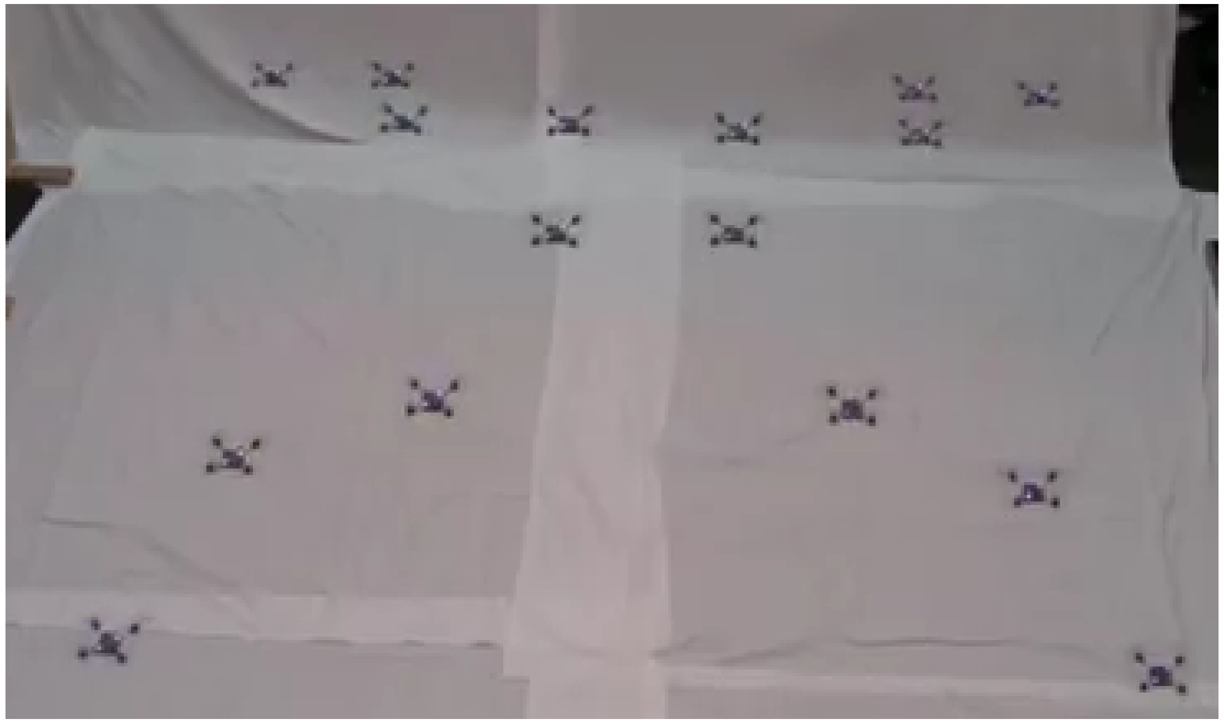
1. Find null space of the constraint matrix
   - Represents all possible solutions
2. Which nodes must be scaled together to still be a solution?
   - We show this can be done by manipulating the null space matrix

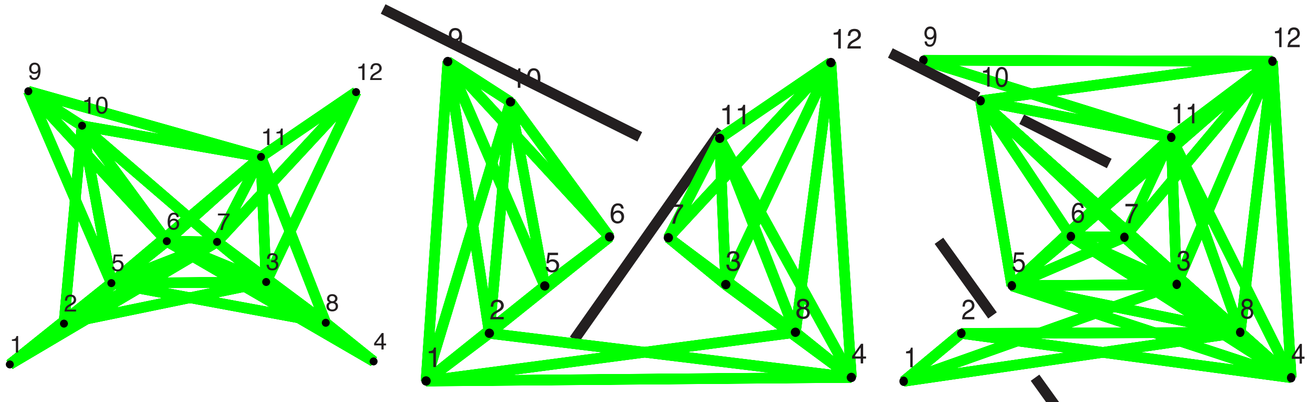


Rigidly-constrained subproblems

- **Scenario I**: Random points in the plane

- **Scenario II**: Quadrotor formations





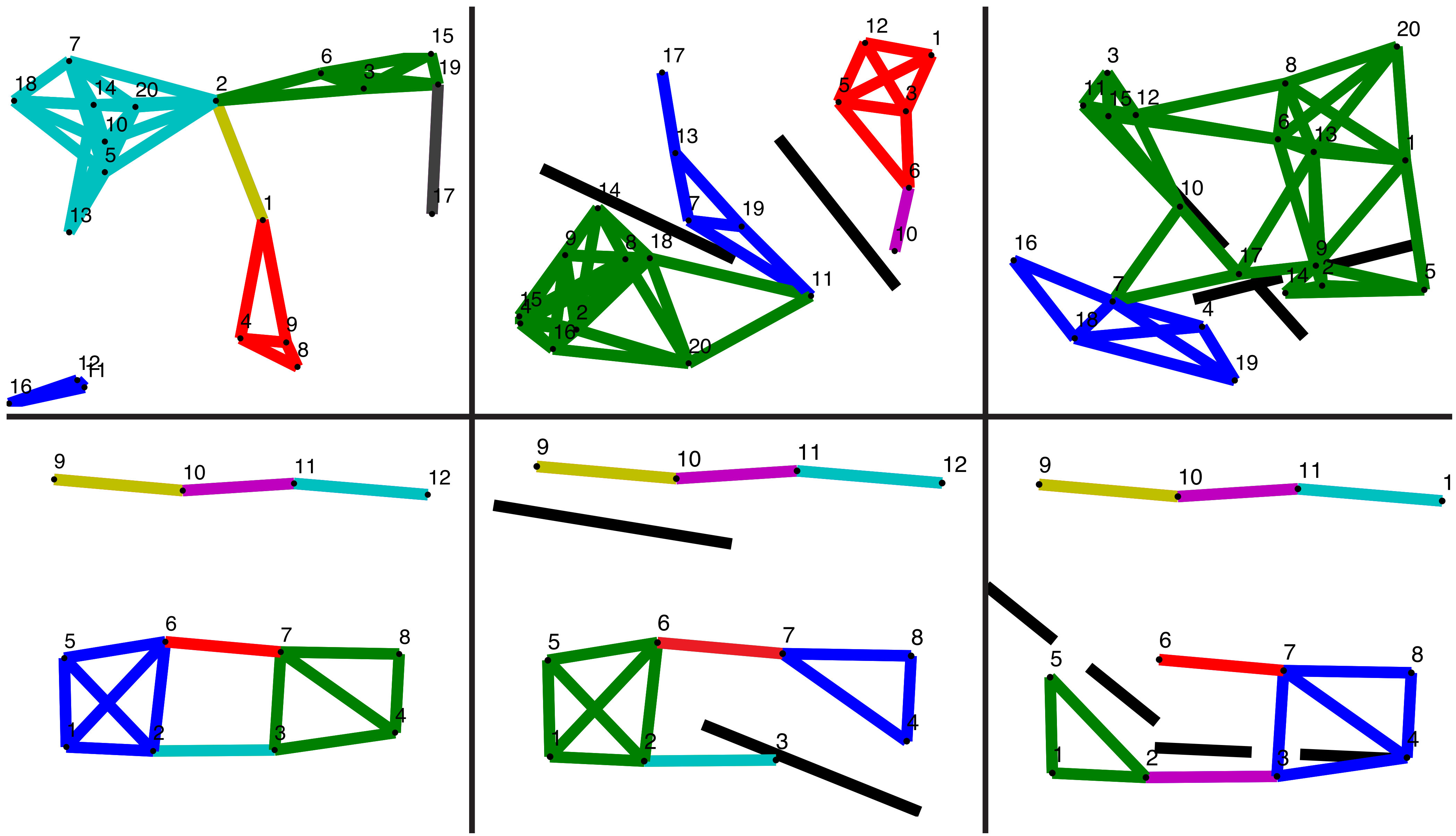
(a) (b) (c)

Rigidly-constrained subproblems

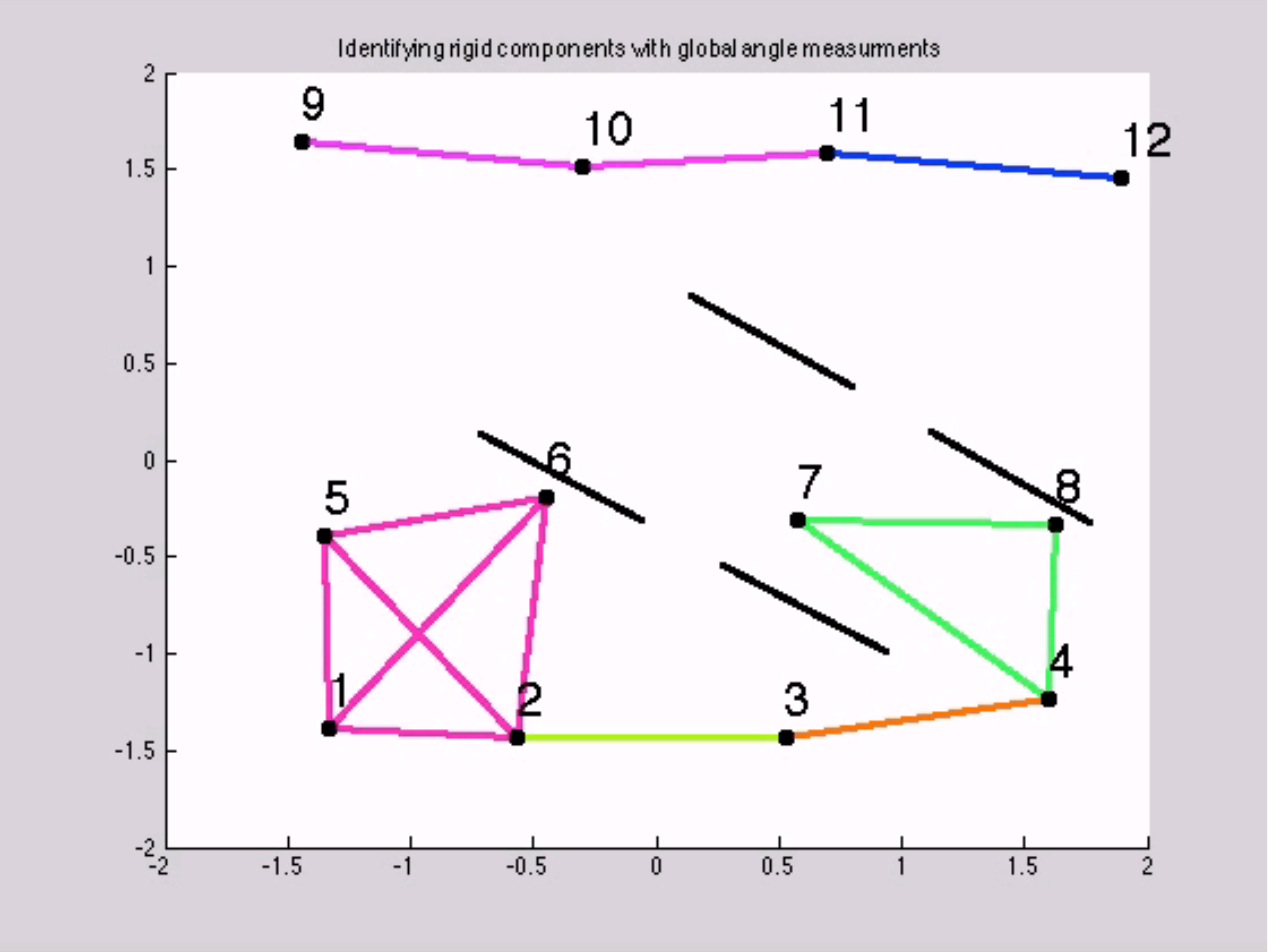
Fixed visibility radius | Walls | Pillars

Random points

Quadrotor formations



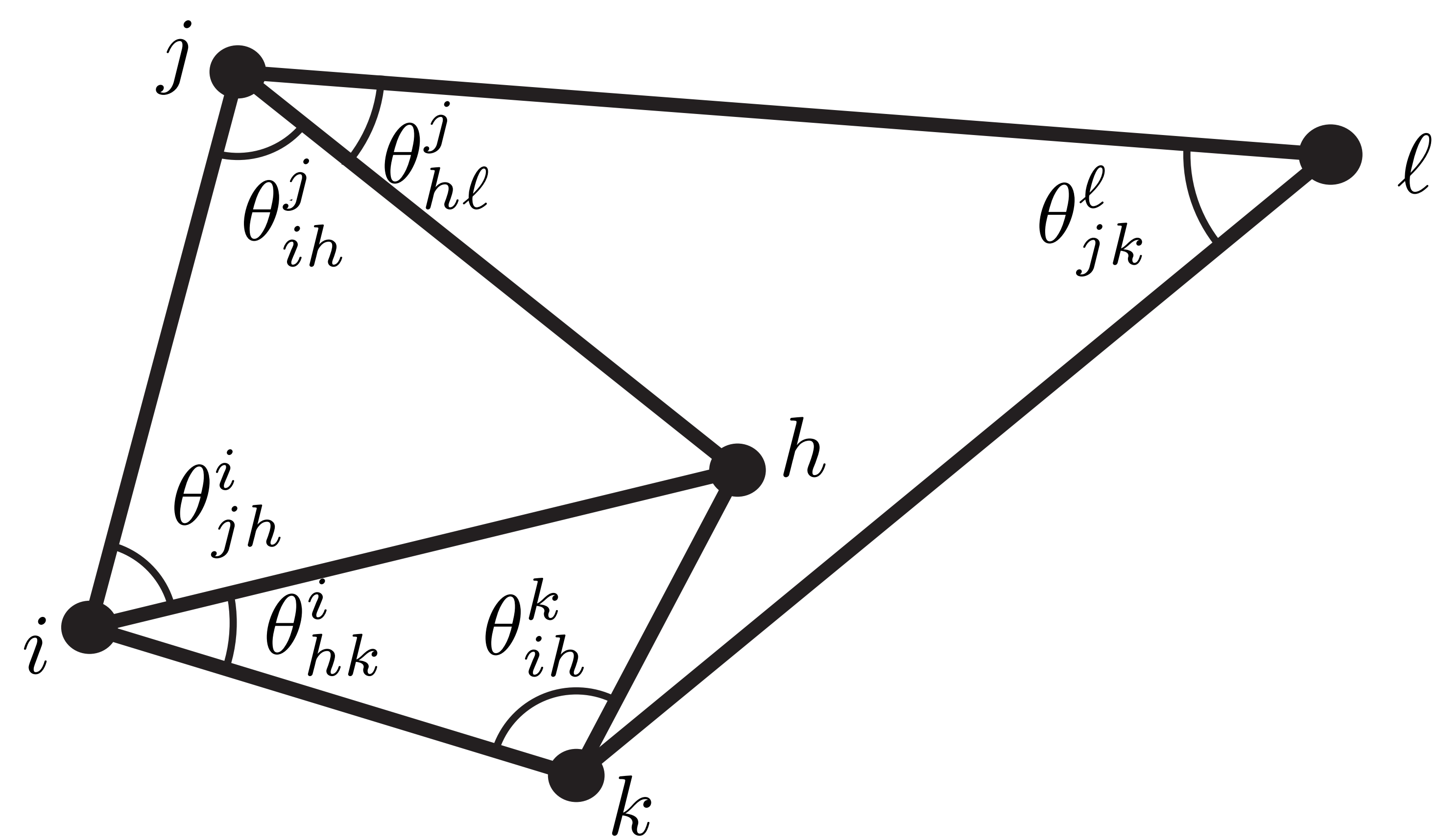
Rigidly-constrained subproblems



No global coordinate frame

- Nodes don't know which way they're facing

- Angles are measured relative to other nodes



No global coordinate frame

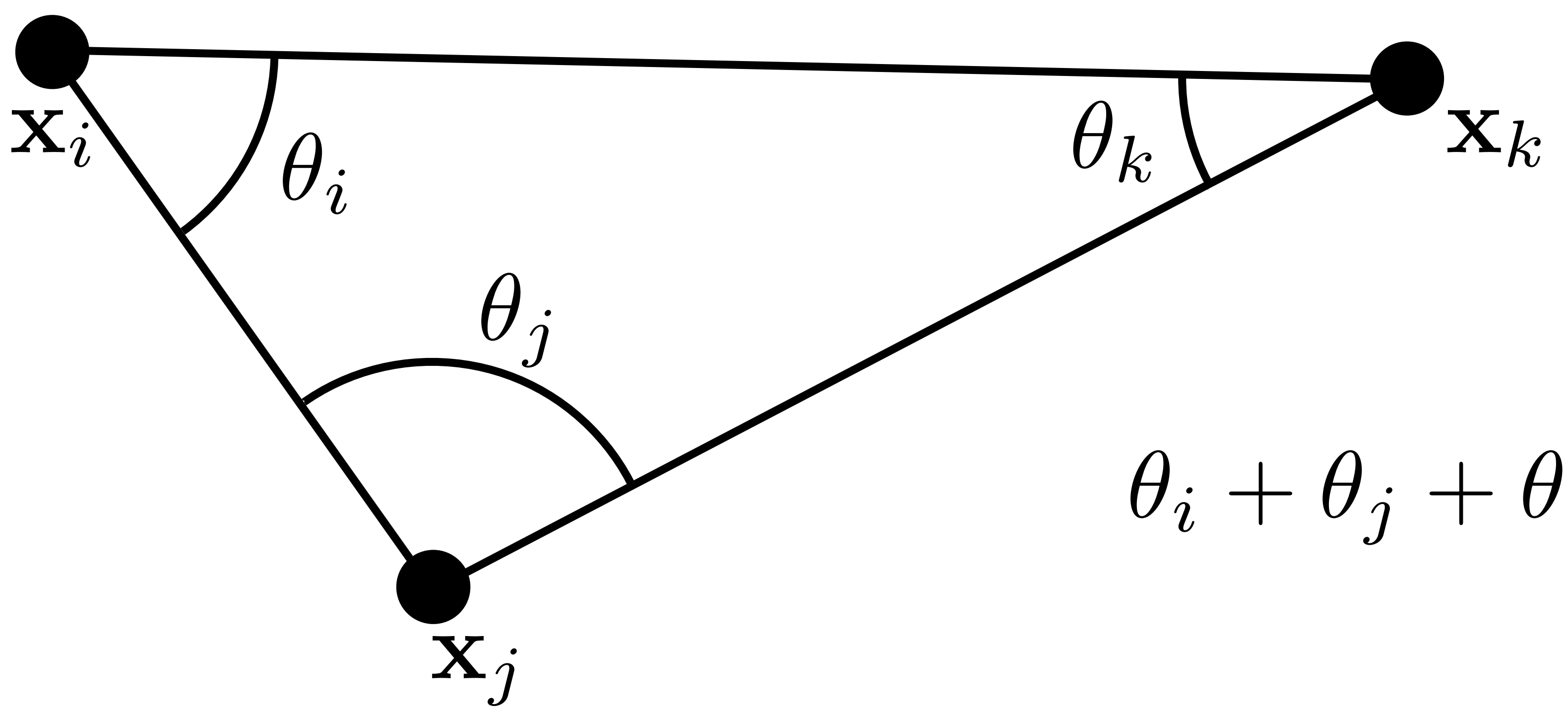
- Each angle measurement gives a **quadratic** constraint (not linear like before)

$$\mathbf{x}^T \mathbf{M}_i \mathbf{x} = 0 \quad \forall \quad i \in \{1, \ldots, n\}$$

- Can't just find null space of constraint matrix
- Why not? Constraint matrices are **not positive semi-definite**
  - If they were, optimization is convex and we could find a solution easily

Triangle constraints

- **Scenario**: each node in a triangle can see the other two



$$\theta_i + \theta_j + \theta_k = \pi$$

- **Constraints**: $\mathbf{M}_i, \mathbf{M}_j, \mathbf{M}_k$
- **Claim**: it is possible to combine these into a **single** constraint which is **p.s.d.**

Triangle constraints

- **Scenario**: each node in a triangle can see the other two
- **Constraints**: $\mathbf{M}_i, \mathbf{M}_j, \mathbf{M}_k$
- **Claim**: it is possible to combine these into a **single** constraint which is **p.s.d.**

$$\begin{aligned} \mathbf{x}^T \mathbf{M}_i \mathbf{x} &= 0 \\ \mathbf{x}^T \mathbf{M}_j \mathbf{x} &= 0 \\ \mathbf{x}^T \mathbf{M}_k \mathbf{x} &= 0 \end{aligned} \quad \Longleftrightarrow \quad \mathbf{x}^T \mathbf{M} \mathbf{x} = 0, \mathbf{M} \text{ is p.s.d.}$$

Triangle constraints

- **Scenario**: each node in a triangle can see the other two
- **Constraints**: $\mathbf{M}_i, \mathbf{M}_j, \mathbf{M}_k$
- **Claim**: it is possible to combine these into a **single** constraint which is **p.s.d.**

$$\begin{aligned} \mathbf{x}^T \mathbf{M}_i \mathbf{x} &= 0 \\ \mathbf{x}^T \mathbf{M}_j \mathbf{x} &= 0 \\ \mathbf{x}^T \mathbf{M}_k \mathbf{x} &= 0 \end{aligned} \quad \Longleftrightarrow \quad \mathbf{x}^T \mathbf{M} \mathbf{x} = 0, \mathbf{M} \text{ is p.s.d.}$$

$$\mathbf{M} = \sin\theta_i \mathbf{M}_i + \sin\theta_j \mathbf{M}_k + \sin\theta_k \mathbf{M}_k$$

Triangle constraints

- Are triangle constraints common?
- **Yes!**
- In Euclidean space, if one node can see two others, they are likely to be close enough to see each other as well
- In our experiments with quadrotor formations, **60-99%** of all constraints were triangular

Experiments

Random configurations (20 nodes)			
Noise (°)	**4m**	**walls**	**pillars**
$\mathcal{U}(-1,1)$	2.265 ± 0.384 mm	69.04 ± 255.6 mm	19.45 ± 133.6 mm
$\mathcal{N}(0,0.5^2)$	1.952 ± 0.302 mm	79.59 ± 285.9 mm	7.747 ± 26.16 mm
$\mathcal{N}(0,0.3^2)$	1.172 ± 0.187 mm	8.576 ± 28.97 mm	1.901 ± 1.932 mm
$\mathcal{N}(0,0.1^2)$	0.379 ± 0.052 mm	10.60 ± 91.33 mm	0.802 ± 1.289 mm
Proportion triangular	0.98	0.84	0.80

Quadrotor configurations			
Noise (°)	**4m**	**walls**	**pillars**
$\mathcal{U}(-1,1)$	2.720 ± 0.747 mm	16.25 ± 73.45 mm	10.71 ± 53.41 mm
$\mathcal{N}(0,0.5^2)$	2.337 ± 0.701 mm	20.25 ± 80.02 mm	7.949 ± 38.75 mm
$\mathcal{N}(0,0.3^2)$	1.408 ± 0.403 mm	11.68 ± 58.78 mm	2.918 ± 5.136 mm
$\mathcal{N}(0,0.1^2)$	0.470 ± 0.139 mm	1.793 ± 4.961 mm	1.099 ± 3.271 mm
Proportion triangular	0.99	0.86	0.85

Heterogeneous networks?

- What if only **some** nodes have a shared coordinate frame?
- Triangle constraints = p.s.d. quadratic constraints
- We can write our p.s.d. quadratic constraints as **linear constraint**! Just like before
- Can potentially combine constraints from nodes **with** and **without** a global coordinate frame to optimize **heterogeneous** networks

Summary

- **Global coordinate frame**: can identify components that are rigidly-constrained
- **No global coordinate frame**: can efficiently find solutions for sets of triangular constraints
- **Heterogeneous networks**: can be solved by combining both types of linear constraints