Integrated Fusion, Performance Prediction, and Sensor Management for Automatic Target Exploitation

Graphical Models for Resource-Constrained Hypothesis Testing and Multi-Modal Data Fusion

MURI Annual Review Meeting

John Fisher, Al Hero, Alan Willsky

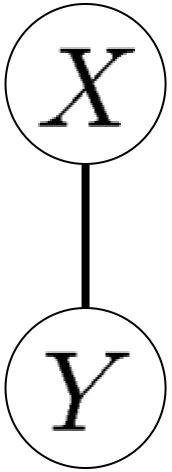
November 3, 2008

Topics

- Design of Discriminative Sensor Models Under Resource Constraints
  - Discriminative tree models
- Distributed PCA
- Multi-modal Data Fusion
  - LIDAR/EO registration

Common Thread

- Set of problems formulated as inference in graphical models.
- Exploit partial knowledge/assumptions on the structure of the measurements or latent variables

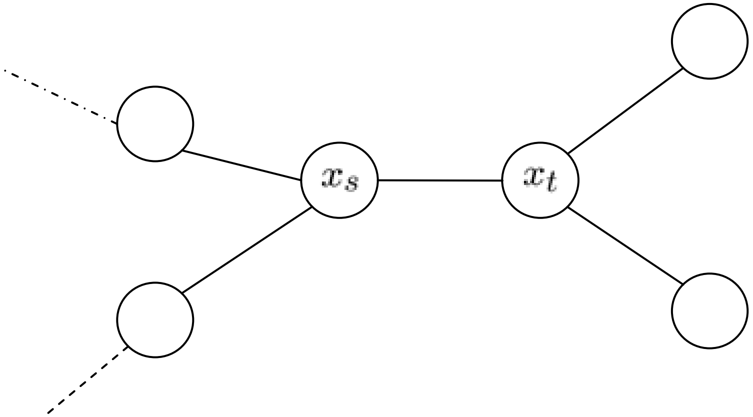


Max-weight Discriminative Forests

- Discriminative Generalization of Chow-Liu Method for Learning Generative Trees
  - Kruskal's Algorithm
  - Use of J-Divergence
    - Interpretation & Performance bounds
- Algorithm, Proof Sketch
- Empirical Results
  - Small Sparse & Non-Sparse Models

Probabilistic Tree Models

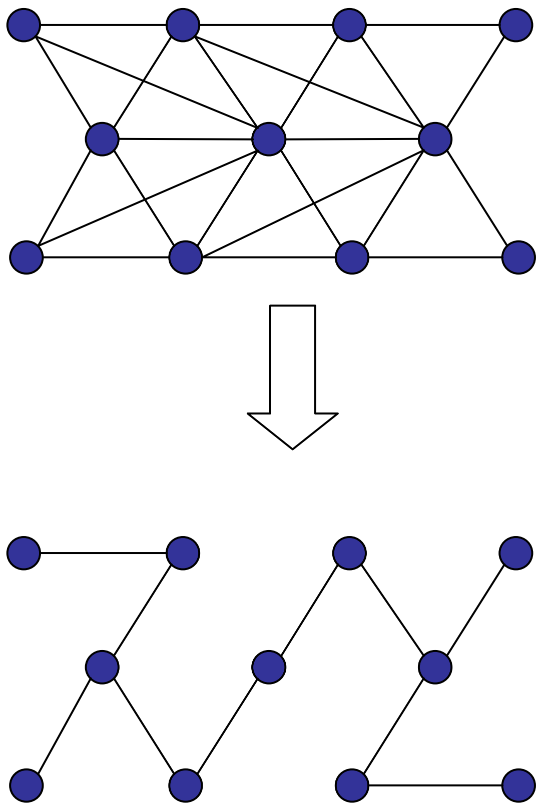
- Tree models are fully described by
  - Marginal properties on the vertex set
  - Pair-wise relationships on the edge set
- Exact inference in trees is tractable and lends itself to distributed computation
- Entropy of a tree model has a similarly decomposition



$$p(X) = \prod_{s \in \mathcal{V}} p(x_s) \prod_{s,t \in \mathcal{E}} \frac{p(x_s, x_t)}{p(x_s)\, p(x_t)}$$

$$H(p) = \sum_{s \in \mathcal{V}} H(p_s) - \sum_{s,t \in \mathcal{E}} I(x_s; x_t)$$

Generative Tree Approximations



- Let p be an N-variate distribution.
- Let $\mathcal{G}$ denote the set of all N-variate probabilistic models whose graphical structure corresponds to a tree.
- For a given $q_{\mathcal{G}} \in \mathcal{G}$ and associated $p_{\mathcal{G}}$ having the same tree structure as $q_{\mathcal{G}}$, but edge-wise marginalizations match p.

$$\begin{aligned} D\left(p \| q_{\mathcal{G}}\right) &= D\left(p \| p_{\mathcal{G}}\right) + D\left(p_{\mathcal{G}} \| q_{\mathcal{G}}\right) \\ &= H\left(p_{\mathcal{G}}\right) - H\left(p\right) + D\left(p_{\mathcal{G}} \| q_{\mathcal{G}}\right) \end{aligned}$$

- There are N^{N-2} unique spanning trees over N nodes.
- The closest tree model in a K-L divergence sense to any distribution can be found by finding the max-weight spanning-tree (MWST), complexity $O\left((N-1)\log N\right)$, over pair-wise mutual information terms, (Chow-Liu 68).

MWST Algorithms

- Many algorithms for solving the max-weight spanning (Prim, Kruskal, reverse-delete, *Chazelle*, etc.)

$$O\left(|E| \log |V|\right)$$

- Kruskal's is of particular interest
  - Greedy
  - Yields a sequence of optimal k-edge forests

Algorithm 1 The 'k-edge' MWST algorithm

Require: $1 \le k \le n-1$, w_{st};

$\mathcal{T}^{(k)} = \{\}$;

$w_{st} = \text{Sort}(w_{st})$;

for $i = 1 : k$ **do**

    if (s,t) does not form a cycle in edges in tree **then**

        $\mathcal{T}^{(k)} \leftarrow \mathcal{T}^{(k)} \cup (s,t)$;

    end if

end for

J-Divergence

- This is a well studied information measure (Jeffreys '46)

$$J(p,q) = \int (p-q)\log\left(\frac{p}{q}\right) = D(p||q) + D(q||p)$$

- Difference between the expected value of the log-likelihood ratio under each hypotheses for binary hypothesis
- Upper and lower bound on probability of error (Hoeffding & Wolfowitz '58, Kailath '67)

$$\frac{1}{2}\min(P_0, P_1)e^{-J} \leq \Pr(\text{err}) \leq \sqrt{P_0 P_1}\left(\frac{J}{4}\right)^{-1/4}$$

J-Divergence

- We'll consider an alternative measure

$$\hat{J}(p, q, p_A, q_B) = \int (p - q) \log \left(\frac{p_A}{q_B} \right)$$

where $p_A, q_B \in \mathcal{G}$

- If we restrict $\mathcal{G}$ to trees then it turns there is an efficient MWST algorithm for jointly learning tree models for p and q which optimizes $\hat{J}(p, q, p_A, q_B)$

Multi-valued edge weights

- If p_A and q_B are constrained to be trees:

$$\hat{J}(p, q, p_A, q_B) = \sum_{s \in \mathcal{V}} J(p_s, q_s) + \sum_{(s,t) \in \mathcal{E}_p \cup \mathcal{E}_q} w_{st}$$

where

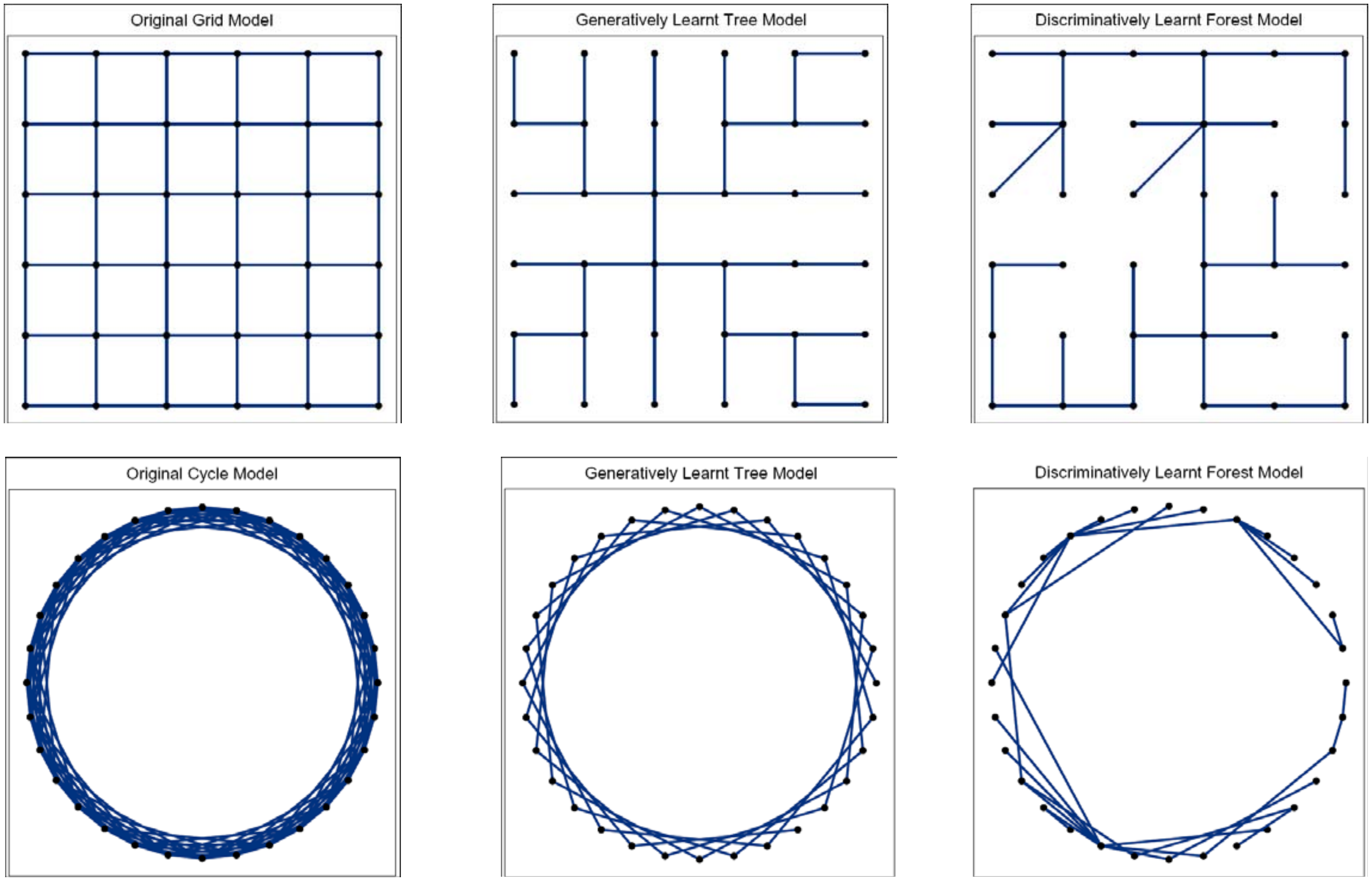
$$w_{st} = \begin{cases} I_p(x_s; x_t) - I_q(x_s; x_t) \\ \quad + D\left(q_{s,t} || p_{s,t}\right) - D\left(q_s q_t || p_s p_t\right) & (s,t) \in \mathcal{E}_p \setminus \mathcal{E}_{pq} \\ I_q(x_s; x_t) - I_p(x_s; x_t) \\ \quad + D\left(p_{s,t} || q_{s,t}\right) - D\left(p_s p_t || q_s q_t\right) & (s,t) \in \mathcal{E}_q \setminus \mathcal{E}_{pq} \\ J(p_{st}, q_{st}) - J\left(p_s p_t, q_s q_t\right) & (s,t) \in \mathcal{E}_{pq} \end{cases}$$

- In constructing such models, adding additional edge-related costs (e.g. comms costs) is straightforward.

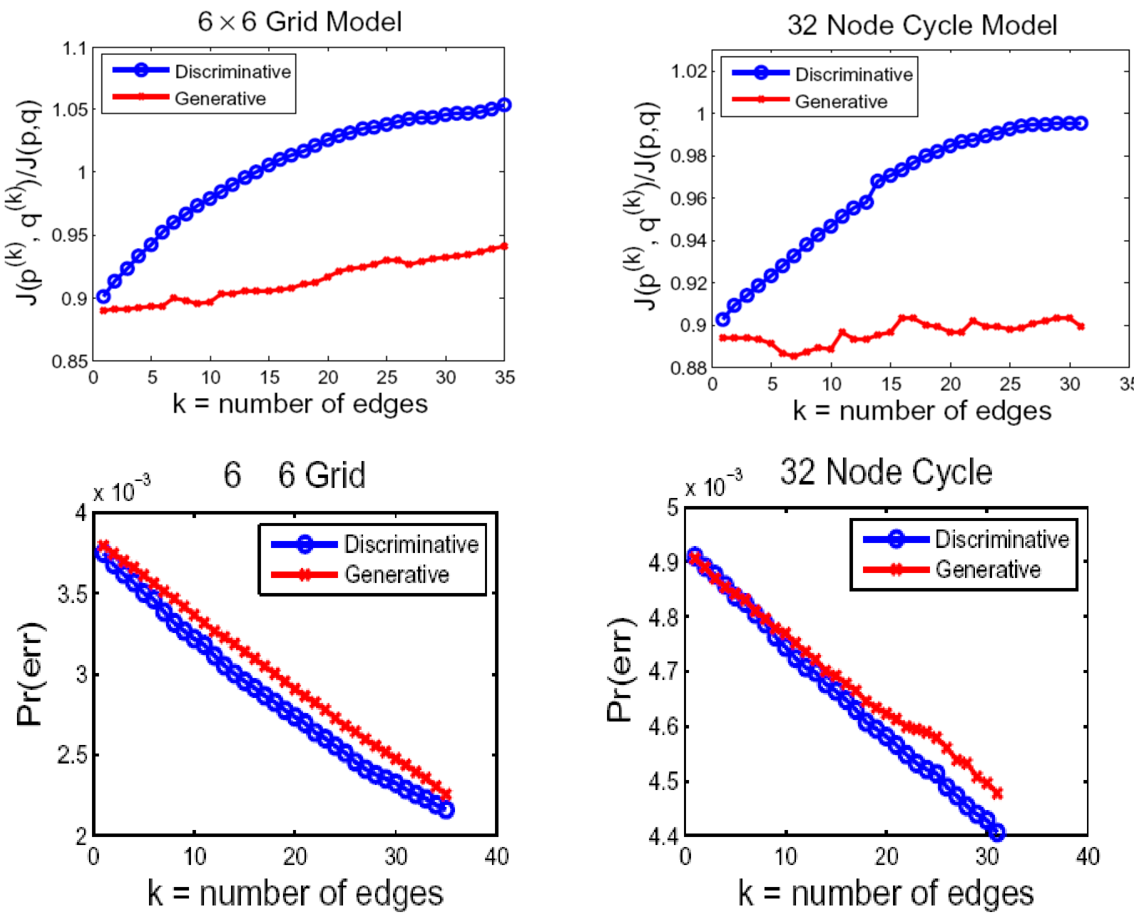
On the use Kruskal's Algorithm

- The sequence of forests are optimal.
- At each iteration, the number of edges in the respective models may be different
- Early termination possible (i.e. p_A and/or q_B may not be *spanning* trees upon termination)
- The proof optimality is similar to the generative case, the primary difference is that some edges may be precluded in one tree, but not the other.
  - The $w_{s,t}$ become single-valued
  - The max is always reduced

Empirical Results

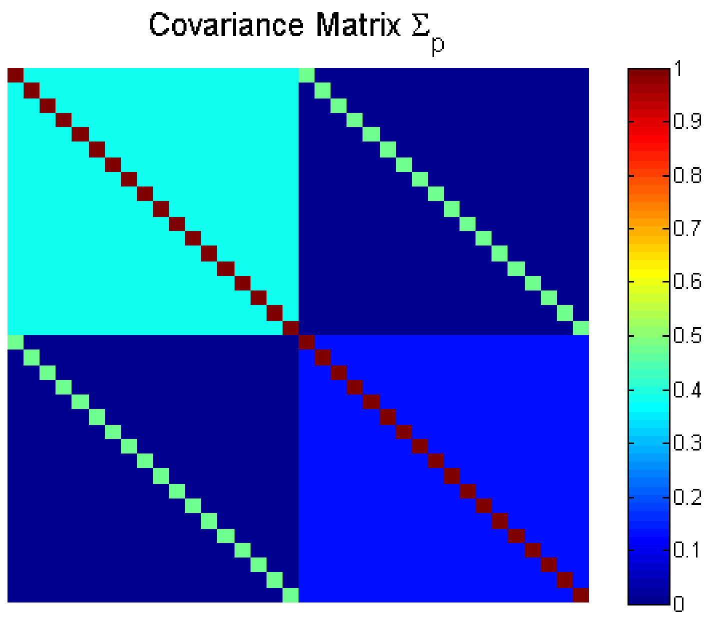


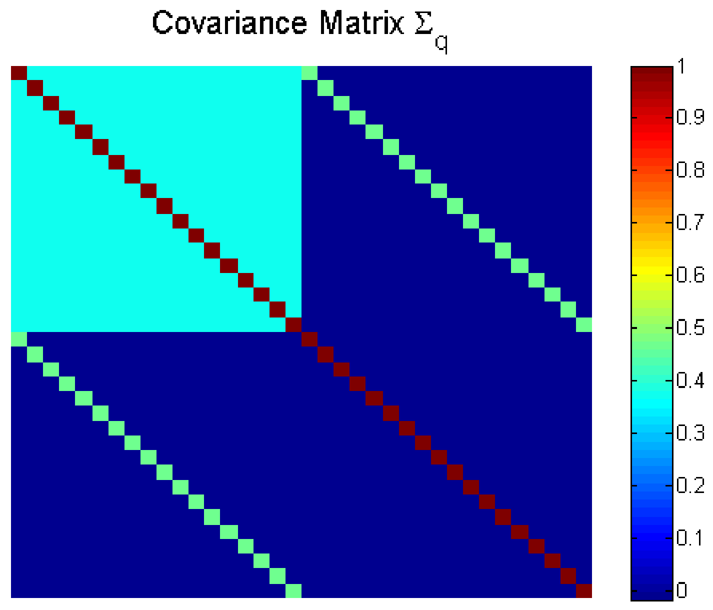
Empirical Results



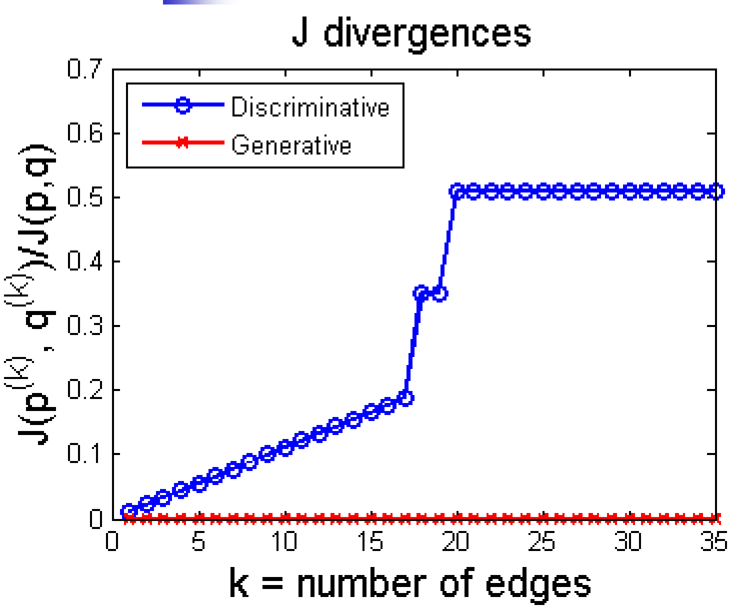
A somewhat contrived example

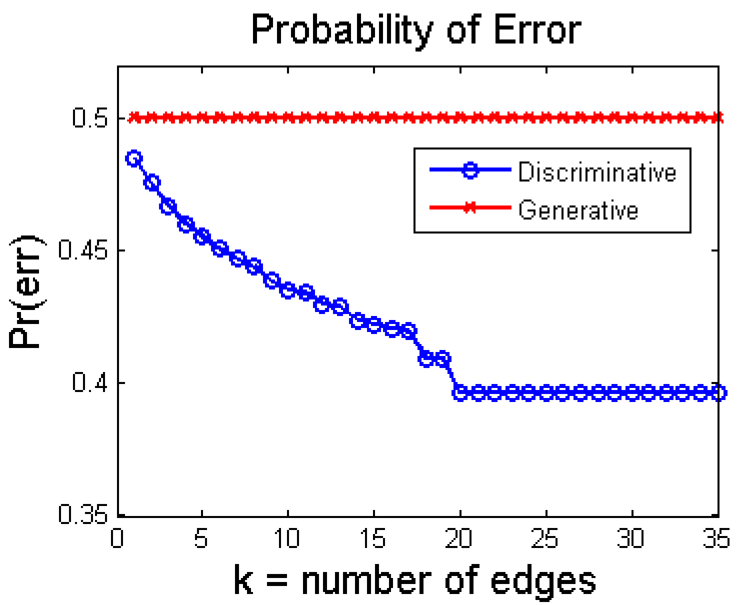
$$\Sigma_p = \begin{bmatrix} \Sigma_c & 0 \\ 0 & P \end{bmatrix} + \Sigma_n \qquad \Sigma_q = \begin{bmatrix} \Sigma_c & 0 \\ 0 & Q \end{bmatrix} + \Sigma_n \qquad \Sigma_n = \begin{bmatrix} I_{\frac{N}{2}} & \rho_n I_{\frac{N}{2}} \\ \rho_n I_{\frac{N}{2}} & I_{\frac{N}{2}} \end{bmatrix}$$





A somewhat contrived example





Comments

- We have presented a discriminative method for learning trees using an approximation to J-divergence
  - Standard MWST algorithms enabled by defining a multi-valued weight function.
  - Resulting structures differ from generative models.
  - Early termination is possible resulting in a forest or non-spanning tree.
- Empirical Results
  - Marginal improvements for graphs which are already sparse.
  - Constructed example demonstrates improvement over generatively learned trees can be significant.
- Open Questions/Directions
  - Is there a Hoeffding-Wolfowitz type bound using the approximate measure?
  - What is the behavior on sparse graphs as the size of the graph grows?
  - Evaluate performance with empirical distributions.
  - Incorporation of data fusion costs in distributed systems.

Progress in Front-end Processing

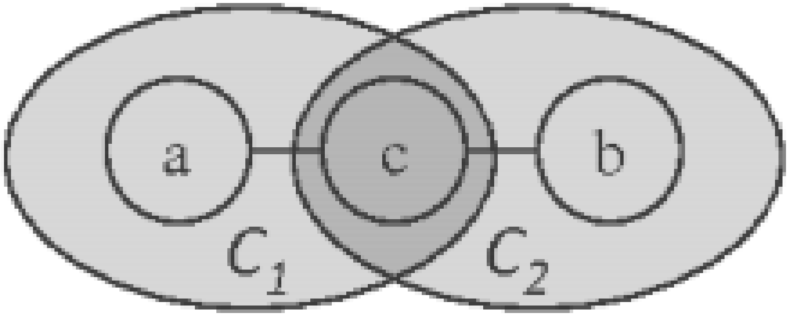
- Graphical models for distributed decomposable PCA

Distributed PCA

- Principle components analysis (PCA) is a model-free dimensionality reduction technique used for high level data fusion (variable importance, regression, variable selection)
- Deficiencies:
  - PCA does not naturally incorporate priors on
    - Dependency structure (graphical model)
    - Matrix patterning (decomposability)
- Scalability problem: complexity is O(N^3)
  - Unreliable/unimplementable for high dimensional data
  - Ill-suited for distributed implementation, e.g., in sensor networks

Networked PCA

- Network model: measure sensor outputs X_a, X_b, X_c
  - Two cliques {a,c} and {b,c}
  - Separator {c}
- Decomposable model: covariance matrix R unknown but conditional independence structure is known.
- PCA of covariance matrix R finds linear combinations $y=U^TX$ that have maximum or minimum variance



DPCA formulation

- Precision matrix $K=R^{-1}$
- For decomposable model K has structure

$$\begin{bmatrix} K_{a,a} & K_{a,c} & 0 \\ K_{c,a} & K_{c,c} & K_{c,b} \\ 0 & K_{b,c} & K_{b,b} \end{bmatrix}$$

- General Representation

$$\mathbf{K} = \begin{bmatrix} \tilde{\mathbf{K}}^{C_1,C_1} & & 0 \\ & & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & \tilde{\mathbf{K}}^{C_2,C_2} & \\ 0 & & \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & \tilde{\mathbf{K}}^{S,S} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

1-dimensional DPCA

- PCA for minimum eigenvector/eigenvalue solves

$$\lambda = \mathrm{eig}_{\min}\left(\mathbf{K}\right)$$

- Key observation: $\lambda = \sup \quad t \quad \text{s.t.} \quad t < \mathrm{eig}_{\min}\left(\mathbf{K}\right)$
- This constraint is equivalent to

$$\begin{aligned} t &< \mathrm{eig}_{\min}\left(\mathbf{K}_{b,b}\right) \\ t &< \mathrm{eig}_{\min}\left(\mathbf{K}_{C_1,C_1} - \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}\left(t\right) \end{bmatrix}\right) \end{aligned}$$

where $\mathbf{M}\left(t\right) = \mathbf{K}_{c,b}\left(\mathbf{K}_{b,b} - t\mathbf{I}\right)^{-1}\mathbf{K}_{b,c}$

Extension to k-dimensional DPCA

- k-dimensional PCA solves sequence of eigenvalue problems

$$\lambda_k = \begin{cases} \min_{\mathbf{u}} & \mathbf{u}^T \mathbf{K} \mathbf{u} \\ \text{s.t.} & \mathbf{u}^T \mathbf{u} = 1 \\ & \mathbf{u}^T \mathbf{u}_i = 0 \quad , i = 1, \cdots, k-1 \end{cases}$$

- Dual optimization

$$\lambda_k \geq \max_{t, \{v_i\}_{i=1}^{k-1}} \quad \text{s.t.} \quad t \leq \text{eig}_{\min} \left(\mathbf{K} + \sum_{i=1}^{k-1} v_i \mathbf{u}_i \mathbf{u}_i^T \right)$$

k-dimensional DPCA (ctd)

- Dual maximization splits into local minimization with message passing

$$t < \mathrm{eig}_{\min}\left(\mathbf{K}_{b,b} + [\mathbf{U}]_{b,:}[\mathbf{U}]_{b,:}^T\right)$$

$$t < \mathrm{eig}_{\min}\left(\mathbf{K}_{C_1,C_1} - \mathbf{E}\mathbf{M}_{\mathbf{U}}(t)\,\mathbf{E}^T\right)$$

- Message passing

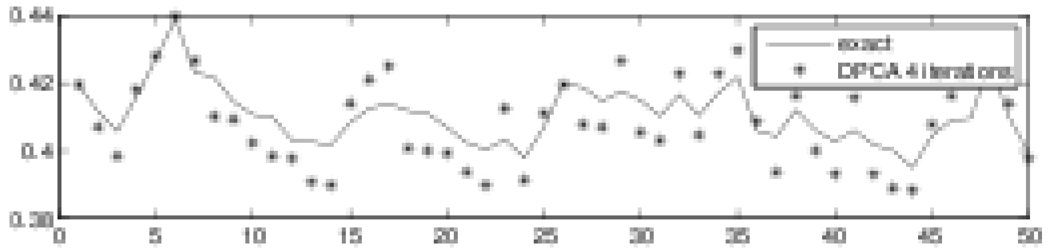
$$\mathbf{M}_{\mathbf{U}}(t) = \begin{bmatrix} \mathbf{K}_{c,b} \\ [\mathbf{U}]_{b,:}^T \end{bmatrix} \times \left(\mathbf{K}_{b,b} + [\mathbf{U}]_{b,:}[\mathbf{U}]_{b,:}^T - t\mathbf{I}\right)^{-1} \begin{bmatrix} \mathbf{K}_{b,c} & [\mathbf{U}]_{b,:} \end{bmatrix}.$$

Tracking illustration of DPCA

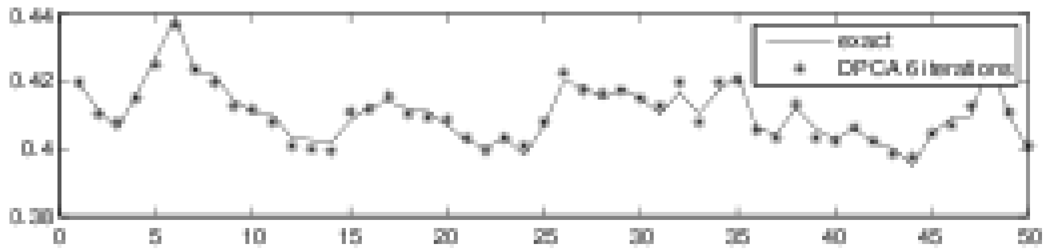
- Scenario: Network with 305 nodes representing three fully connected networks with only 5 coupling nodes
  - C1 = {1, · · · , 100, 301, · · · , 305},
  - C2 = {101, · · · , 200, 301, · · · , 305}, and
  - C3 = {201, · · · , 300, 301, · · · , 305}.
- Local MLEs computed over
  - sliding time windows of length n = 500
  - 400 samples overlap.
- Centralized PCA computation: EVD O(305)^3 flops
- DPCA computation: EVD O(105)^3 flops + message passing of a 5x5 matrix M

DPCA min-eigenvalue tracker

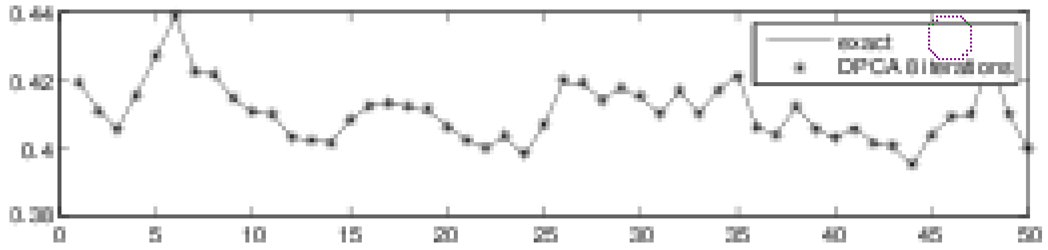
Iteration 1



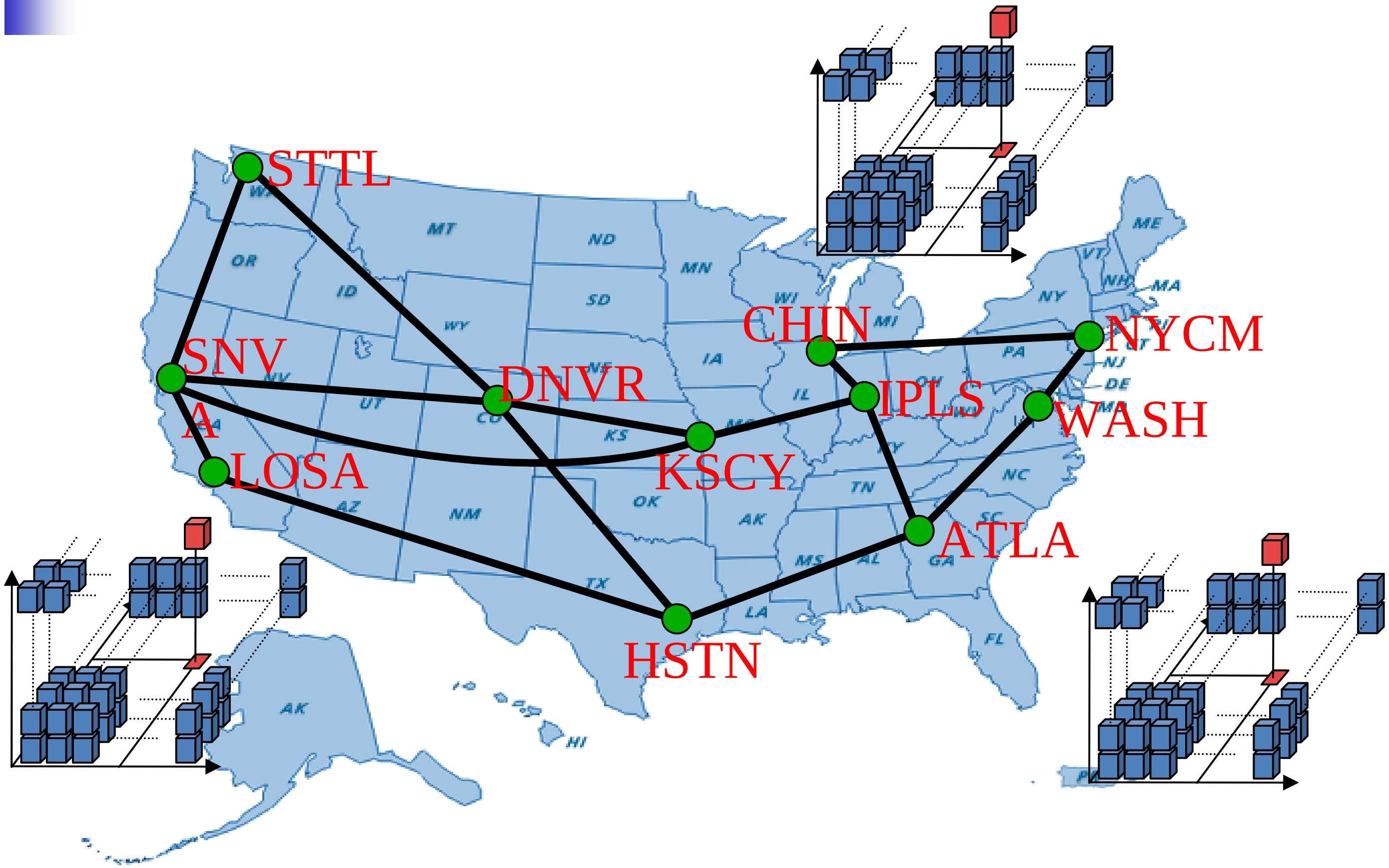
Iteration 2



Iteration 3



DPCA network anomaly detection



Multiple measurement sites (Abilene)

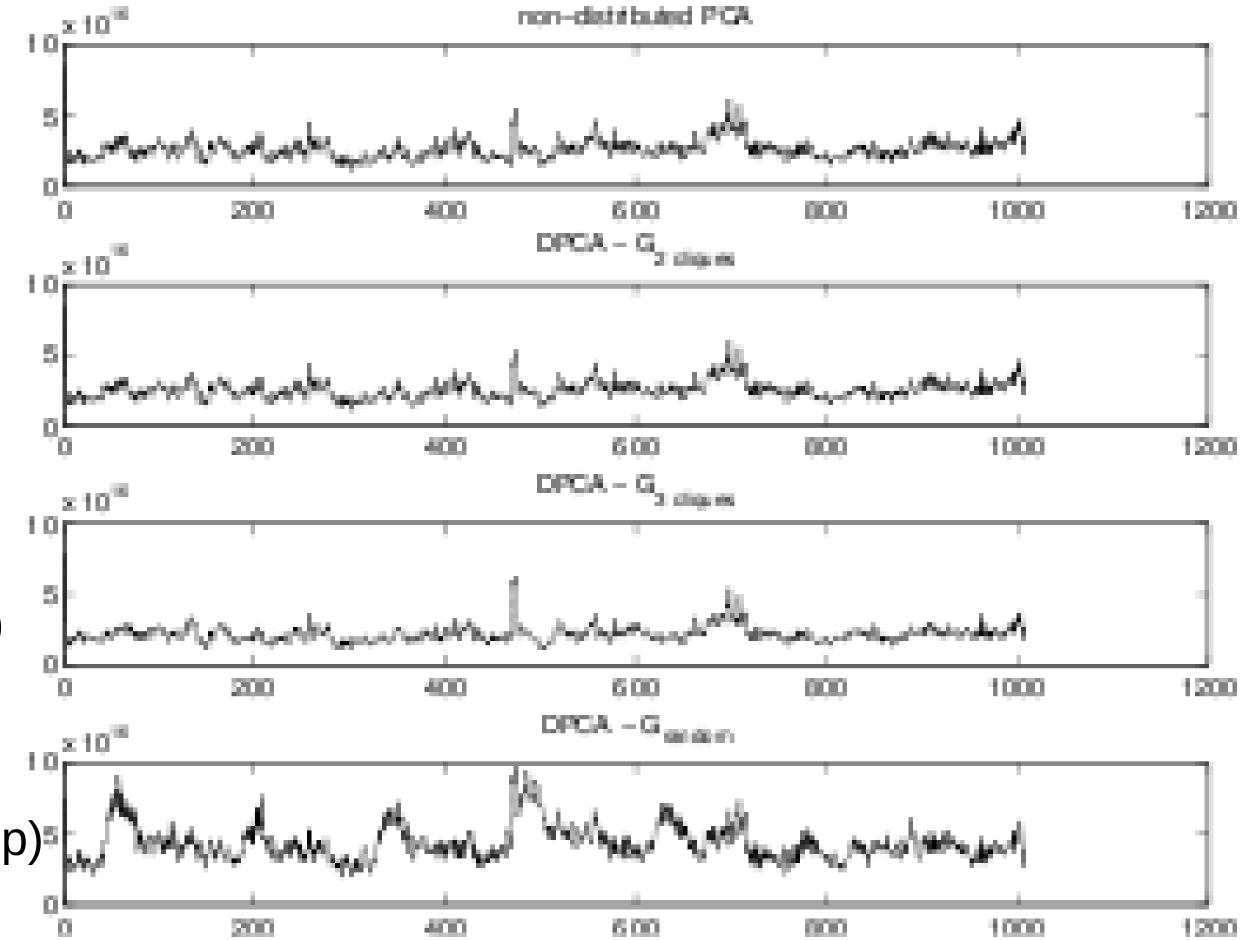
DPCA anomaly detection

PCA (centralized)

DPCA (E-W decomp)

DPCA (E-W-S decomp)

DPCA (Random decomp)



Discussion

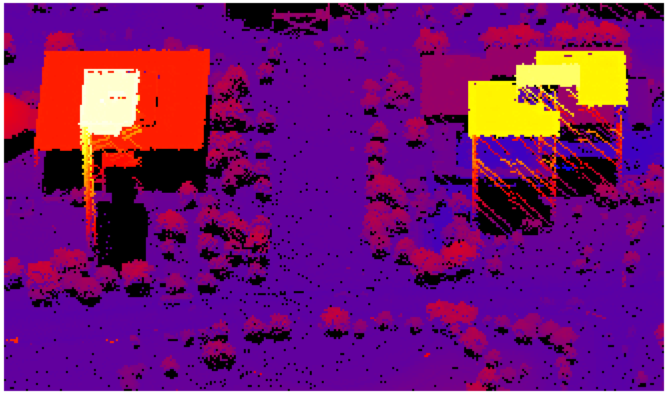
- **Take home message**: Combination of model-free dimensionality reduction and model-based graphical model can significantly reduce computational complexity of PCA-based high-level fusion
- Complexity scales polynomially in clique size not in overall size of problem. Example: 100,000 variables with 500 cliques each of size 200
  - Centralized PCA: complexity is of order 1015
  - DPCA: complexity is of order 106
- If one can impose similar decomposability constraints on graph Laplacian matrix, be extended to non-linear dimensionality reduction: ISOMAP, Laplacian eigenmaps, dwMDS.

LIDAR/Optical Registration

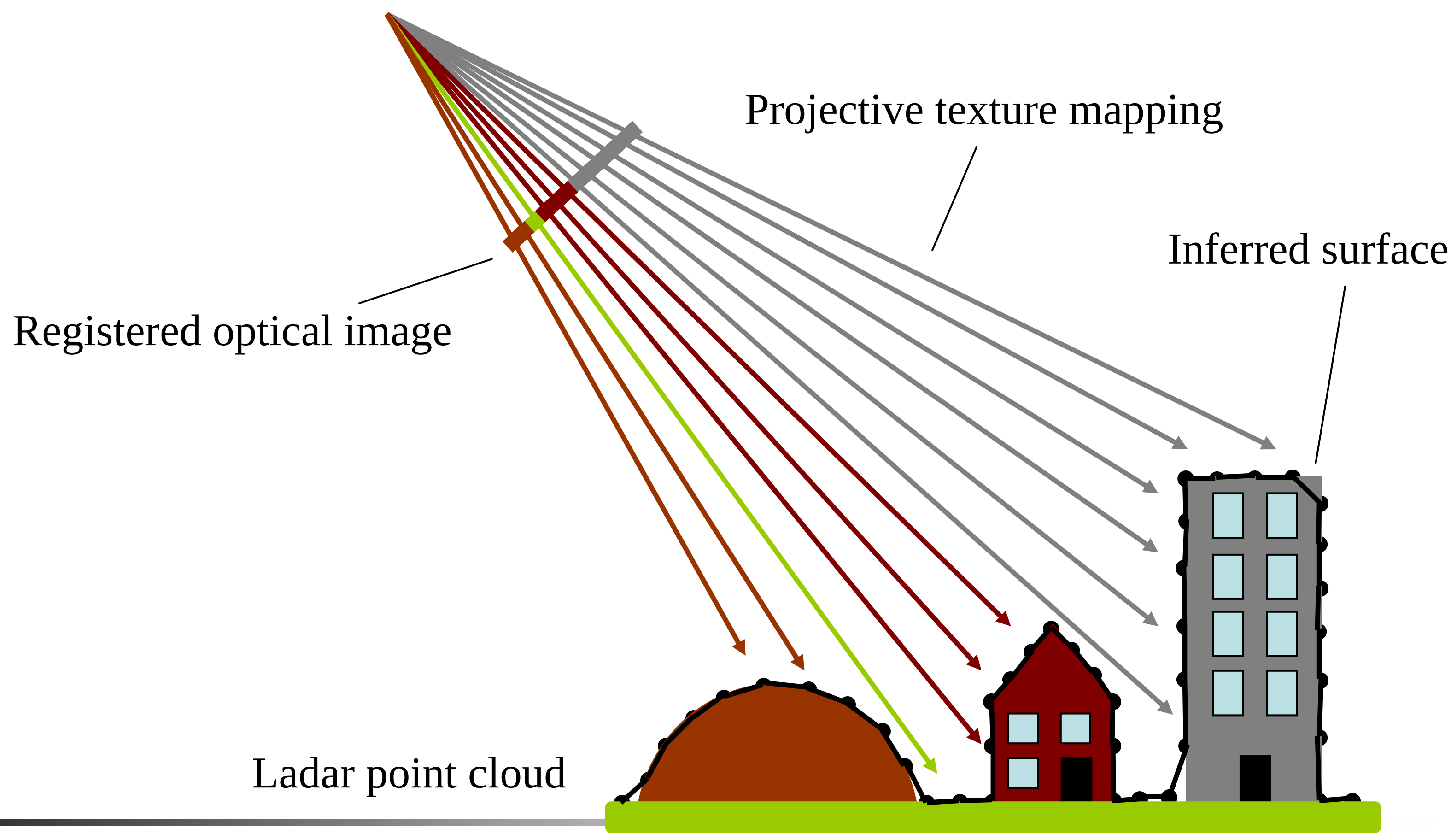
- Laser Radar
- 3D Model Generation
  - Delaunay mesh formation
  - Model Demo
- Camera Model
- Statistical Registration Methods
- Results
  - Probing Experiments
  - Registration Demo

Laser Radar

- 3D point cloud
- Not gridded
- Typical lateral resolution of 0.5 – 1.0 meters
- Linear mode sensors:
  - Slower collects
  - Probability of detection (pdet) values
- Geiger mode sensors:
  - Faster collects
  - Geometry only

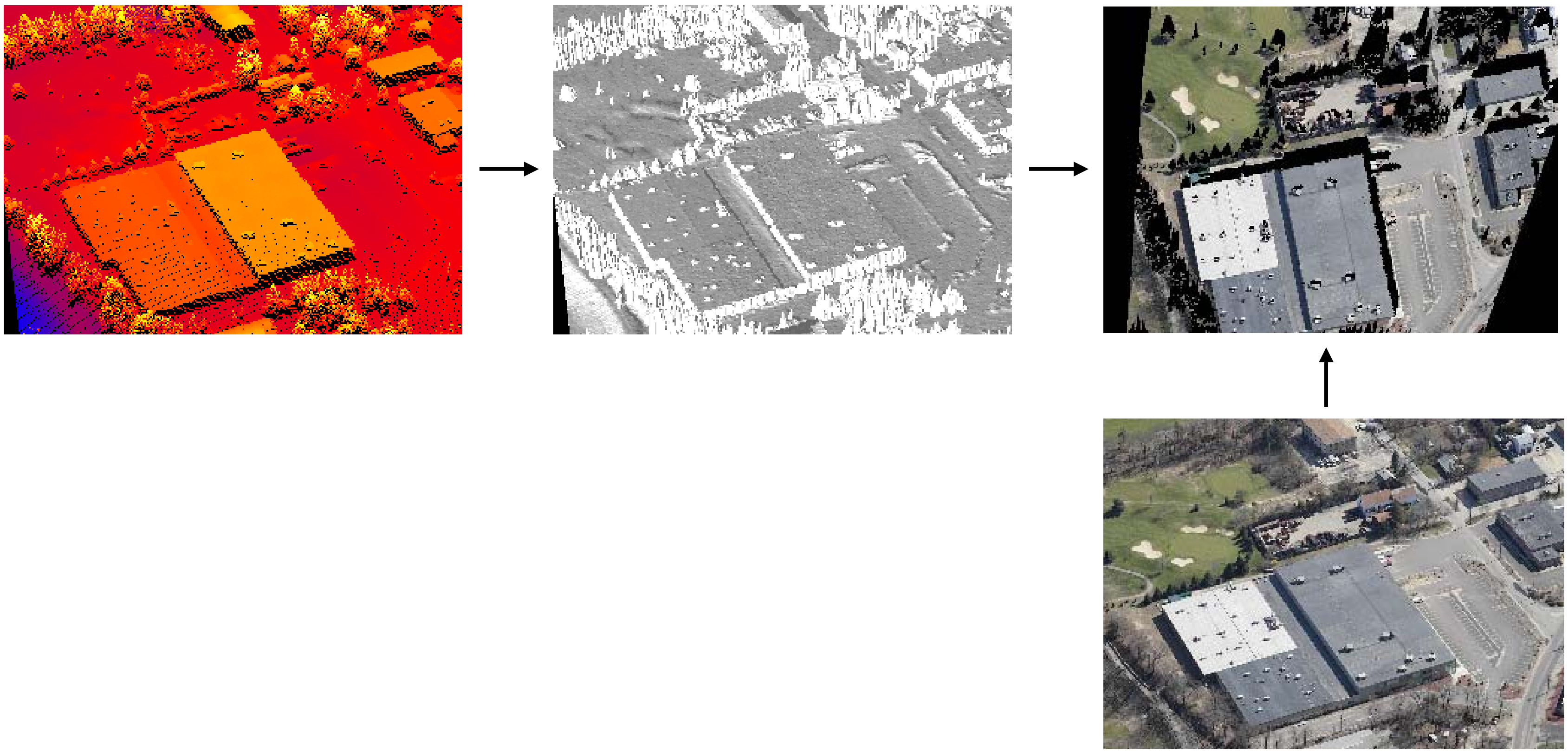


3D Model Generation



3D Model Generation

Point cloud | Mesh | 3D Model

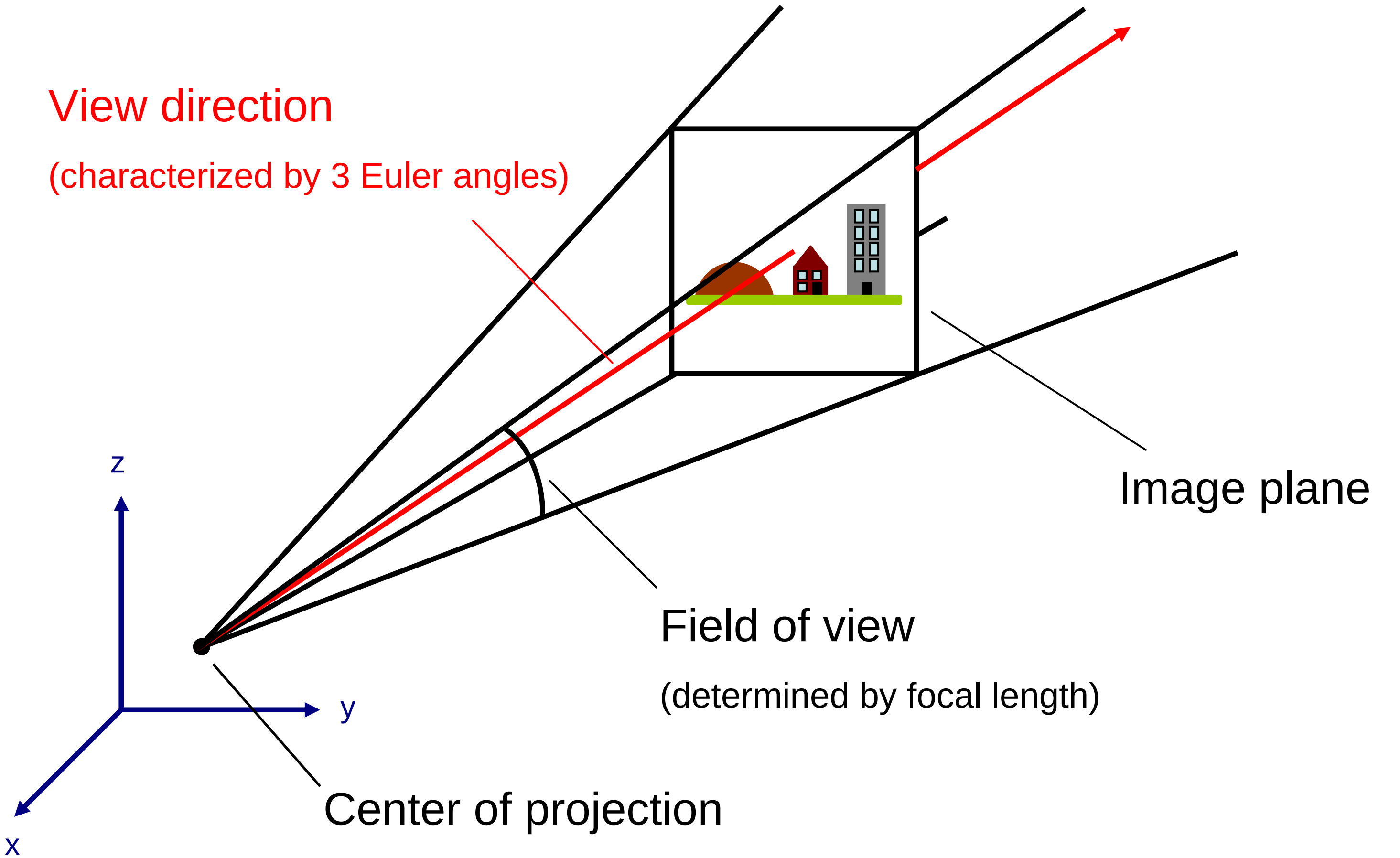


Optical image

Automatic Registration Challenges

- GPS/INS has low precision
- Matching colors with geometry (inherently low mutual information)
- Occlusion reasoning with point cloud projection
- 3D rendering
- Resolution differences
- Previous work
  - Line correspondences (Frueh, Sammon, Zakhor 2004)
  - Vanishing points (Ding, Lyngbaek, Zakhor 2008)

Projective Camera



$$\begin{bmatrix} f \\ C_x \\ C_y \\ C_z \\ \alpha \\ \beta \\ \gamma \end{bmatrix}$$

Initial Registration

- Manual calibration is prohibitive
- User selected correspondence points
- Minimization of sum of algebraic error

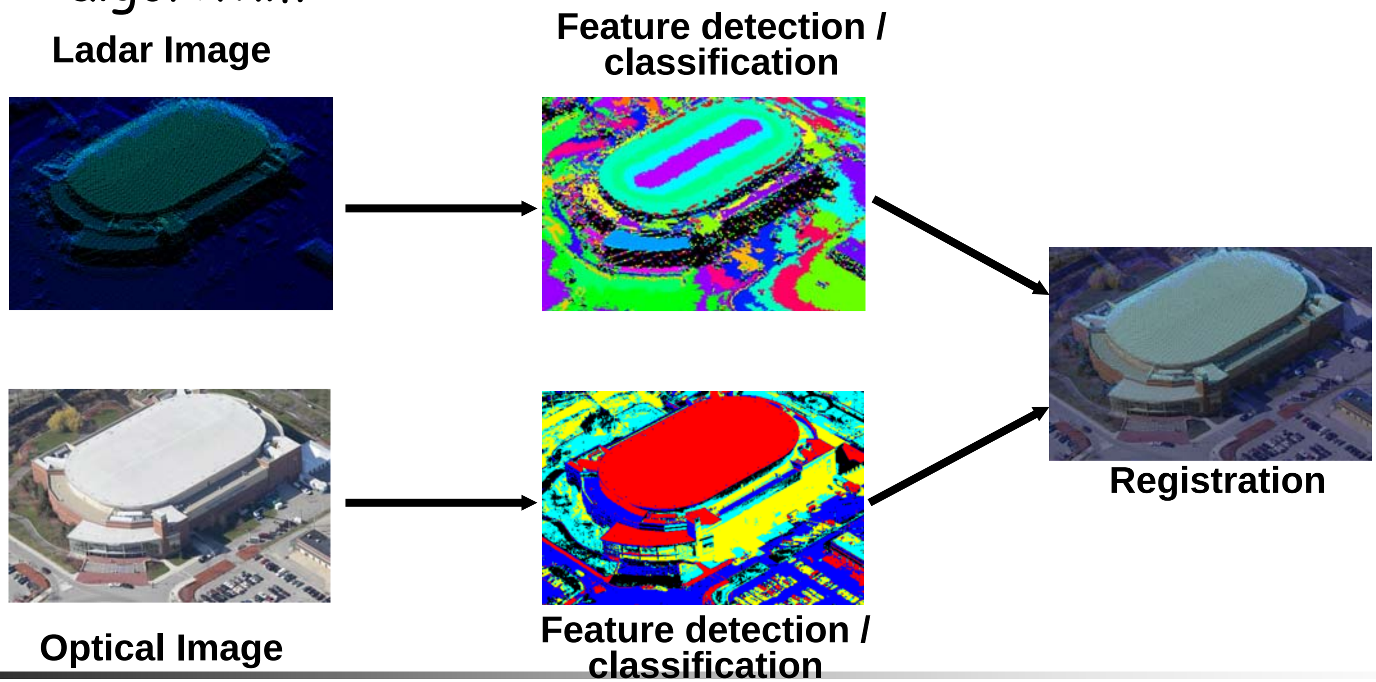
$$\min_T \sum_i d_{\text{alg}}(\mathbf{X}_i', T\mathbf{X}_i)^2$$

$$d_{\text{alg}}(\mathbf{X}', T\mathbf{X}) = d_{\text{alg}}(\mathbf{X}', \hat{\mathbf{X}}') = (w\hat{u} - u\hat{w})^2 + (v\hat{w} - w\hat{v})^2$$

- *T* is parameterized by *f*, C_x, C_y, C_z, α, β, γ
- Levenberg-Marquardt iterative optimization (derivative free)

Statistical Registration Methods

- Commonly used for registration of multi-modal medical imagery
- Information theoretic similarity measure with optimization algorithm



Example: Joint Entropy

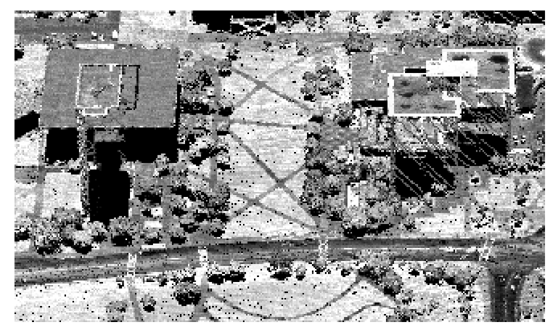
Optical image

$$T_{JE} = \arg\min_{T} - \sum_{i} p([u, v_T])_i \log p([u, v_T])_i$$

T: camera projection matrix
u: optical image features/labels
v: ladar image features/labels
p(·): probability mass function

Luminance

→ JE Registration ←



Pdet Values

Automatic Registration Demo

- Minimum joint entropy with optical luminance and ladar elevation
- Downhill simplex optimization (derivative free)
- Ladar rendered using OpenGL
- Entire ladar data set in graphics memory
- Approximate initial registration

Unsupervised LIDAR/Optical Registration