Structural Graph Matching With Polynomial Bounds On Memory and on Worst-Case Effort

Fred DePiero, Ph.D.
Electrical Engineering Department
CalPoly State University
San Luis Obispo, California - USA

Goal: Subgraph Matching for Use in Real-Time Measurement Systems

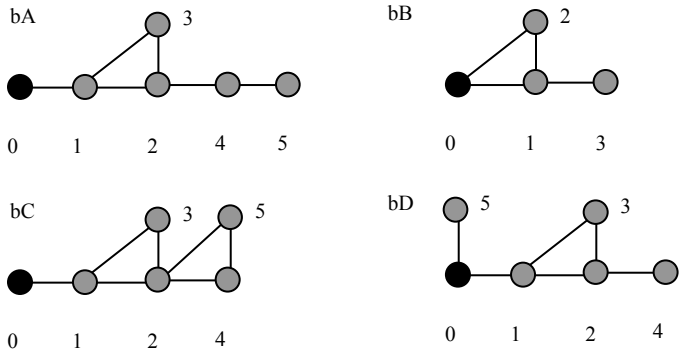
- Graph representations for measurements:
  Image registration, speech recognition, object recognition…
- *Approximate* maximum common subgraph.
- *Deterministic bounds* on processing effort and on memory requirements.
- Tolerant to low SNR, low dynamic range, required for node and edge coloring.
- Results Presented:
  - In Paper: Matching based on graph structure emphasized.
  - In Talk: Graphs with and without color attributes.
  - In Software: Multidimensional coloring supported.

Dynamic Horizon Used To Compare Graph Structure Locally

- 'Horizon' is the number of nodes included in a local comparison of graph structure.
- 'Dynamic' nature permits the horizon to vary in extent depending on similarity of structure.
- Examples of horizon distance:
  - Scene labeling via discrete relaxation, unary & binary constraints, h = 1,2
  - Superclique neighborhood comparison [Wilson & Hancock 6/97], h = constant
  - Eigenspace projection clustering [Caelli & Kosinov 4/04], h = constant
- We desire having a larger horizon to help find better node-to-node correspondences. Expand until affected by structural differences.

Local Neighborhood Created Using Basis Graphs

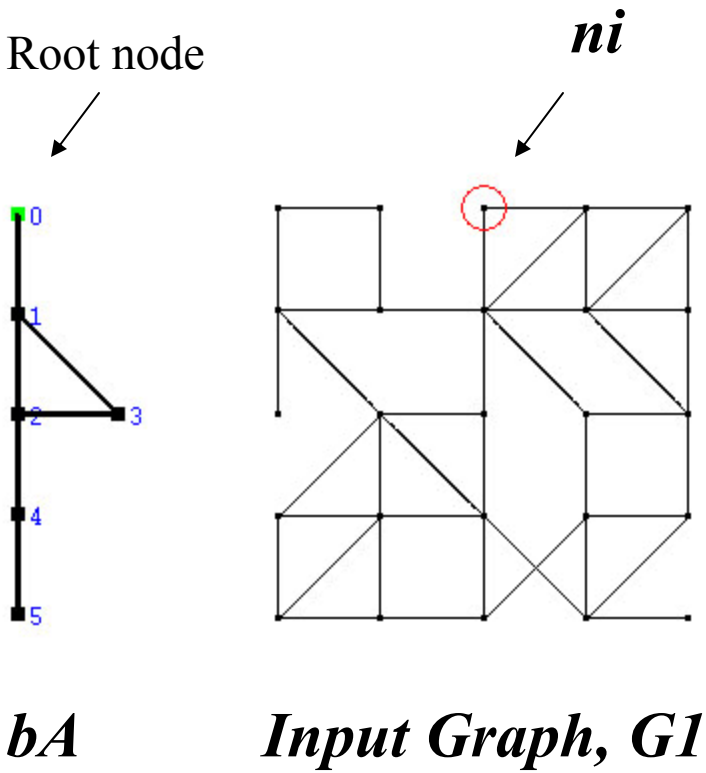
- Basis Graphs bA-bD:
- Rooted at node 0.
- Ordered nodes.
- Non symmetric.



- Basis Graphs used to create local neighborhood (subgraph of input) having invariant order.

Neighborhood Constructed With Invariant Node Order

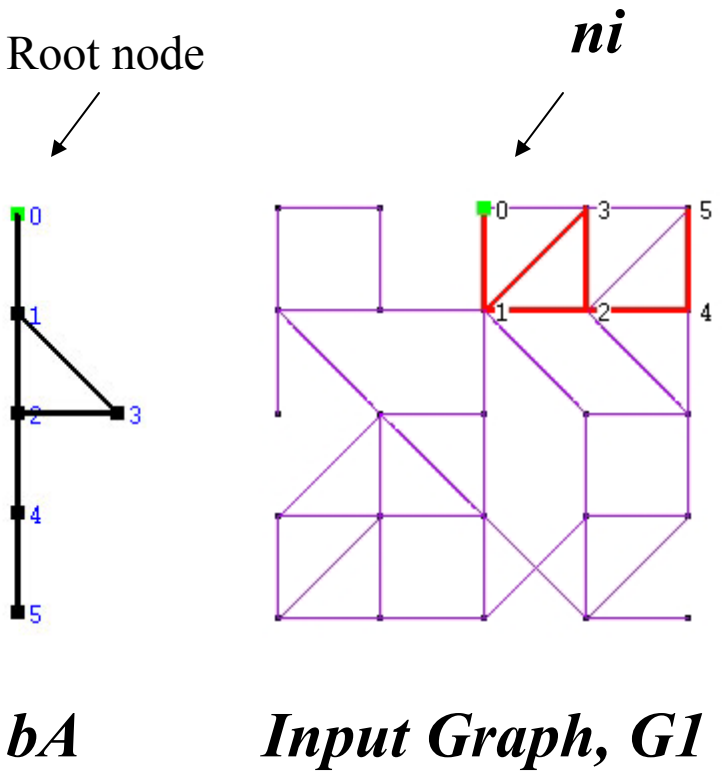
1. Depth-first placement of Basis Graph (***bA***), rooted at node ***ni*** of Input Graph (***G1***).



bA ***Input Graph, G1***

Neighborhood Constructed With Invariant Node Order

1. Depth-first placement of Basis Graph (***bA***), rooted at node ***ni*** of Input Graph (***G1***).

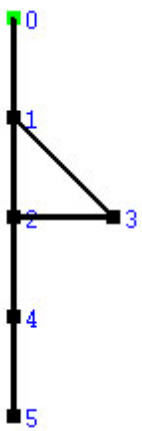


bA ***Input Graph, G1***

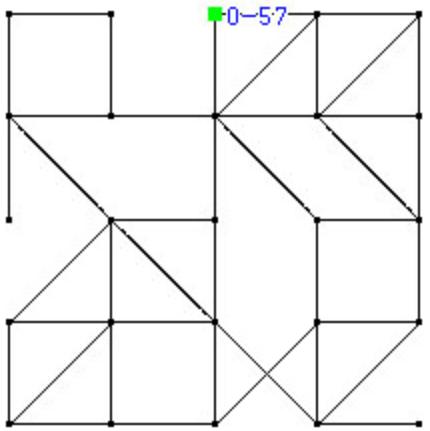
Neighborhood Constructed With Invariant Node Order

1. Depth-first placement …
2. Histogram reveals common positions for ***bA*** nodes.

Occurrences of bA.0



bA

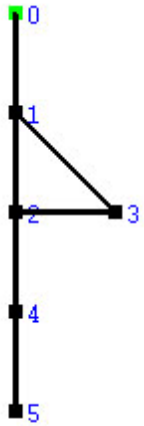


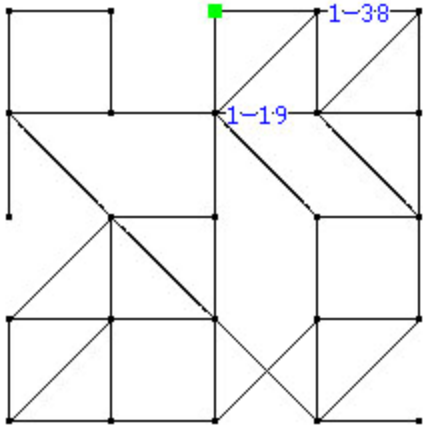
Input Graph, G1

Neighborhood Constructed With Invariant Node Order

1. Depth-first placement …
2. Histogram reveals common positions for ***bA*** nodes.

Occurrences of bA.1



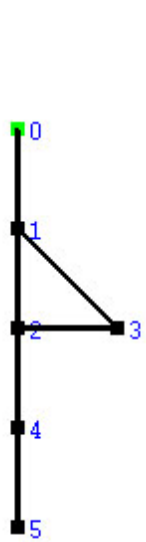


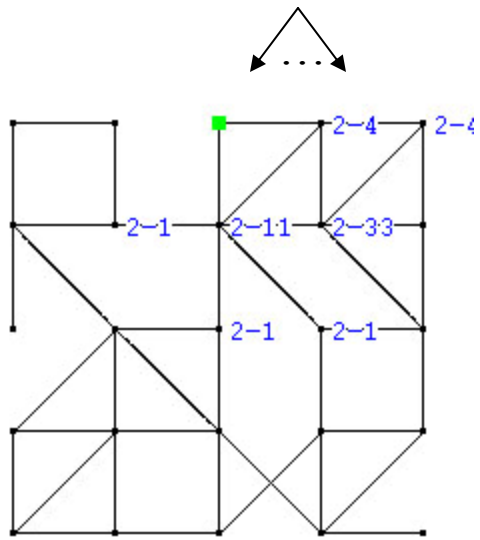
bA ***Input Graph, G1***

Neighborhood Constructed With Invariant Node Order

1. Depth-first placement …
2. Histogram reveals common positions for ***bA*** nodes.

Occurrences of bA.2



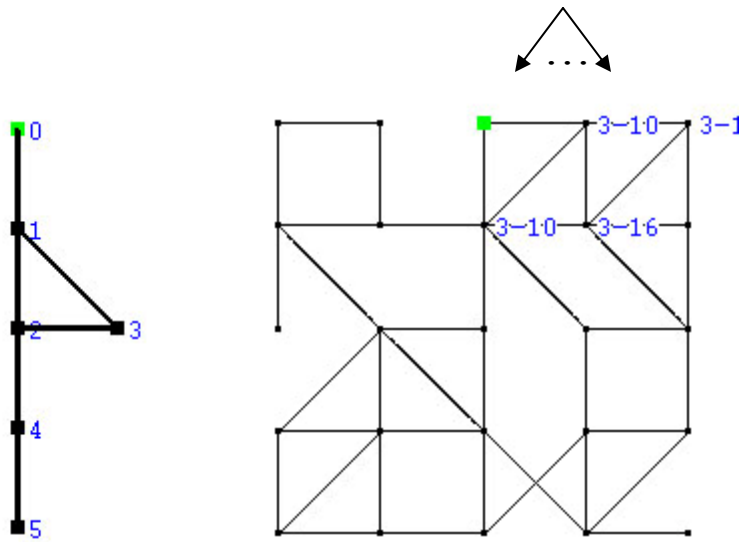


bA ***Input Graph, G1***

Neighborhood Constructed With Invariant Node Order

1. Depth-first placement …
2. Histogram reveals common positions for ***bA*** nodes.

Occurrences of bA.3

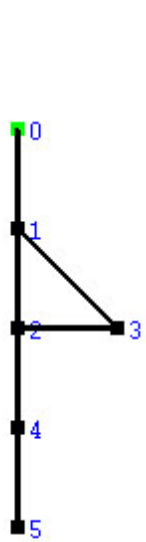


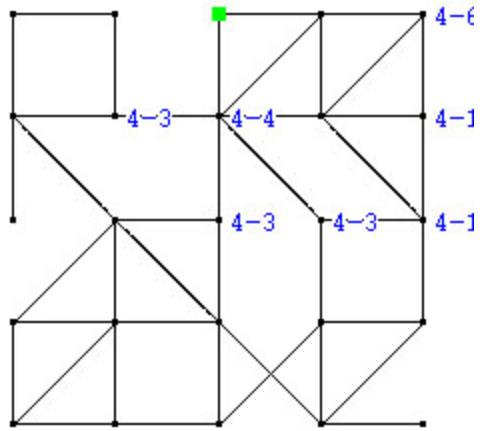
bA ***Input Graph, G1***

Neighborhood Constructed With Invariant Node Order

1. Depth-first placement …
2. Histogram reveals common positions for ***bA*** nodes.

Occurrences of bA.4



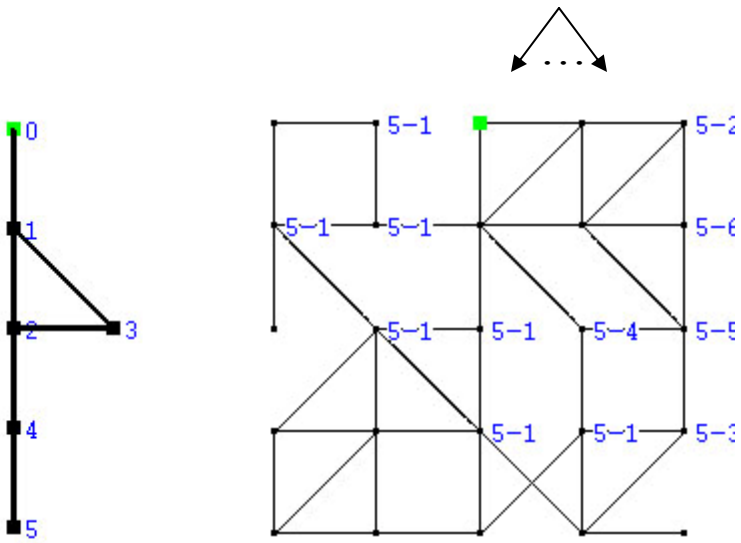


bA ***Input Graph, G1***

Neighborhood Constructed With Invariant Node Order

1. Depth-first placement …
2. Histogram reveals common positions for ***bA*** nodes.

Occurrences of bA.5

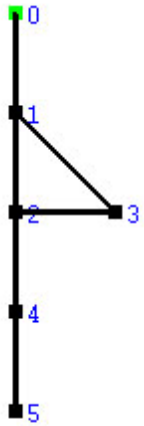


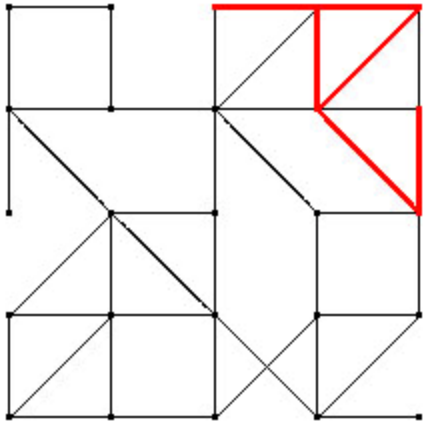
bA ***Input Graph, G1***

Neighborhood Constructed With Invariant Node Order

1. Depth-first placement …
2. Histogram reveals common positions …
3. Overlay instances of ***bA*** onto ***G1***, depth-first. Select nodes with highest histogram count. Halt at ties.

First instance of bA



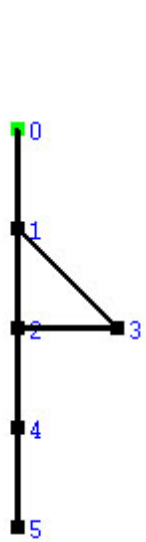


bA ***Input Graph, G1***

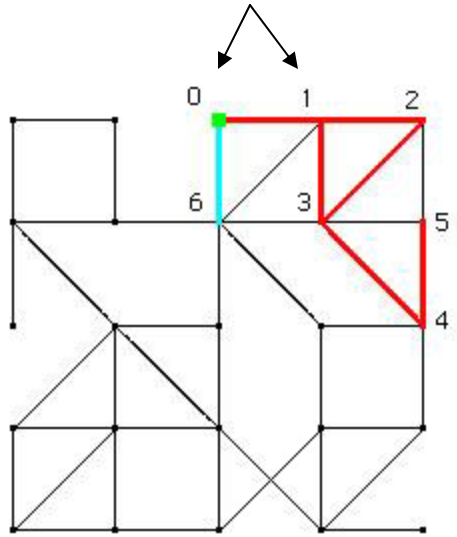
Neighborhood Constructed With Invariant Node Order

1. Depth-first placement …
2. Histogram reveals common positions …
3. Overlay instances of ***bA***…
4. Neighborhood is the induced subgraph of ***G1*** for nodes touched by ***bA*** instances. Neighborhood order given by ordering of nodes in ***bA*** instances.

Ordered Neighborhood



bA



Input Graph, G1

Common Subgraph Extracted Via Local & Global Processing

1. Establish local (invariant) neighborhoods for each input graph.
2. Compare structure of local neighborhoods in each graph.
   - Define initial mapping probability using complement of edit distance.
   - Functional approximation to PDF, “discriminative”.
3. Use continuous relaxation to update mapping probabilities.
   - Fixed number of iterations. Introduces global constraints to mapping.
4. Find final mapping in best-first manner.
   - Using updated mapping probabilities.

Additional Features Improve Size Of Common Subgraph

1. Similarity of degree also used to adjust mapping probabilities.

   Slight improvement.

2. Use multiple basis graphs, select result having larger common subgraph.

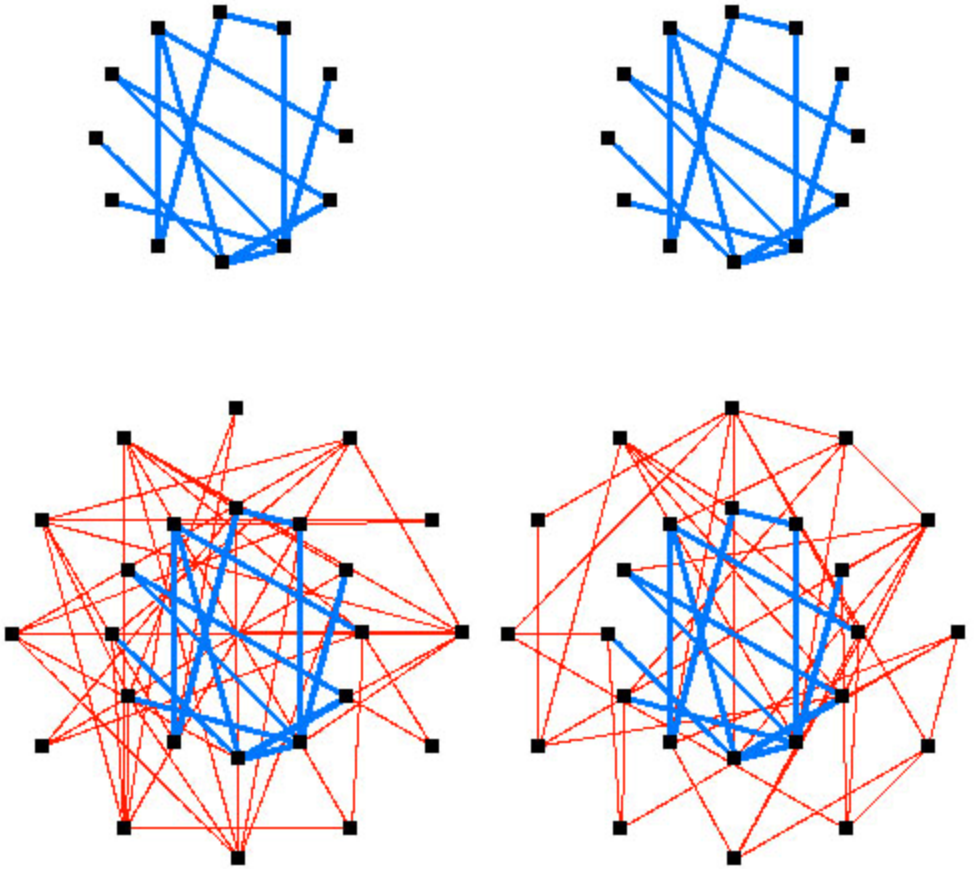
   Good improvement.

Test Trials Contained High Noise

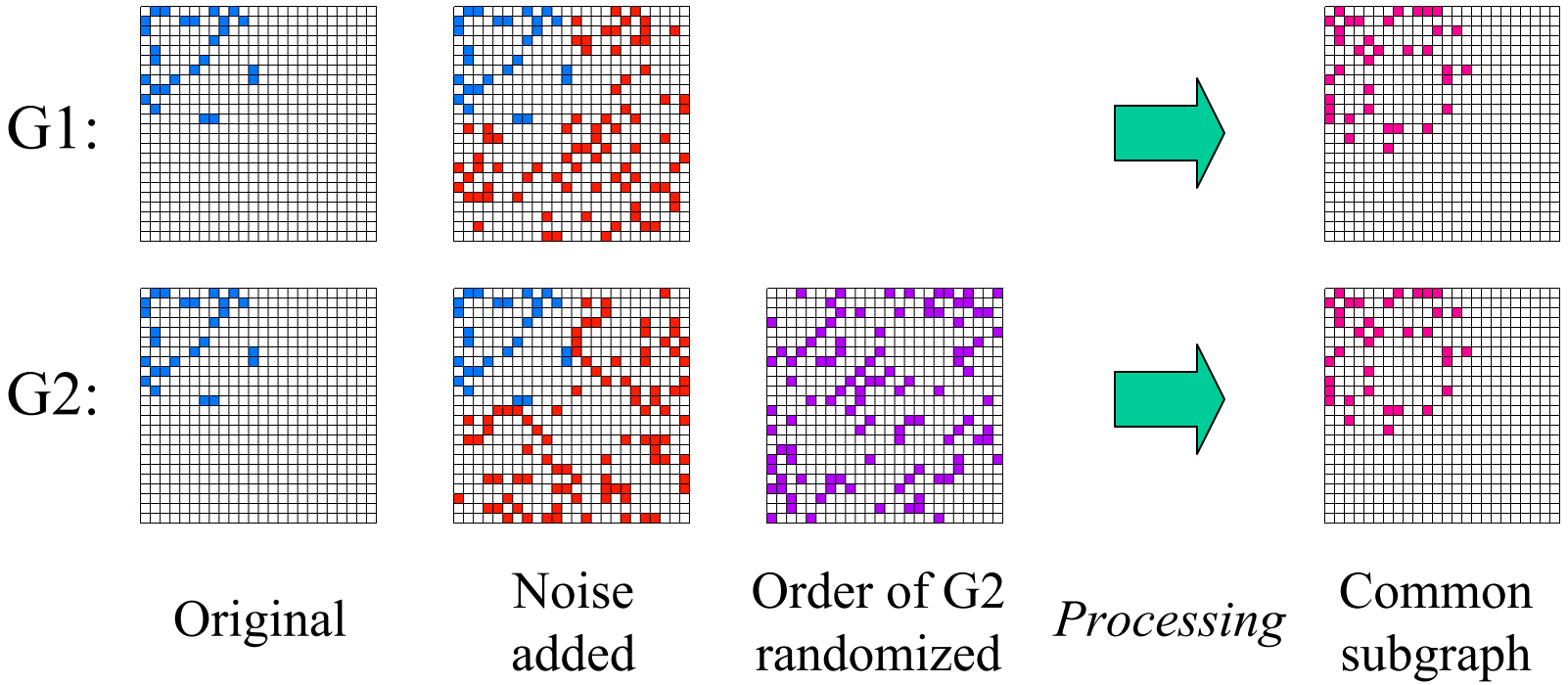
- Input graphs randomly generated.
- Types: Model A, Strongly regular Near-planar.
  Varying sizes, edge densities.
- Varying amounts of noise added to each input.
  Outer ring of nodes, red edges.

- Node order of input graphs randomized…

Model A, N=12, with 12 added noise nodes



Adjacency Matrix Used to Verify Result

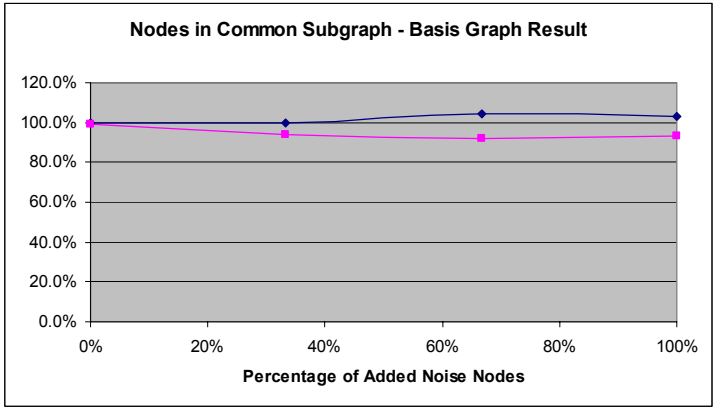


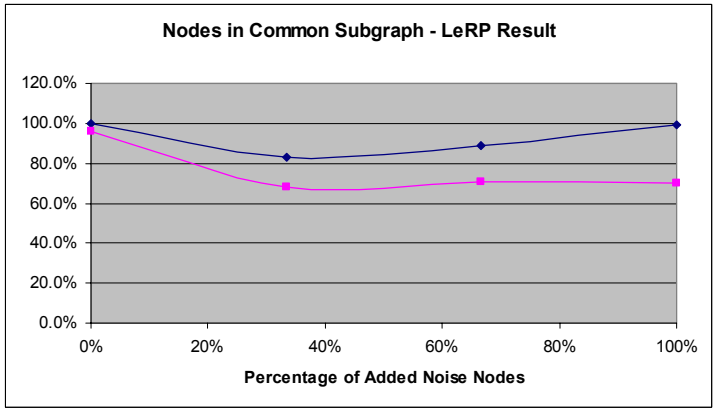
Adjacency matrix of common subgraph checked for validity.

Results Show Improvement Compared to Previous Method

- Graph Inputs:
  - Model A (0.15, 0.2, 0.3)
  - Strongly Regular (d=2,3,4)
  - Noise added to each input (0-100%)
  - 12 nodes, nominal.
- Size of common subgraph shown.
  - Min & max value of mean plotted, for each above type.
  - 1200 trials total.
- Comparisons made versus previous approach: 'LeRP', Compared Length-R Paths.

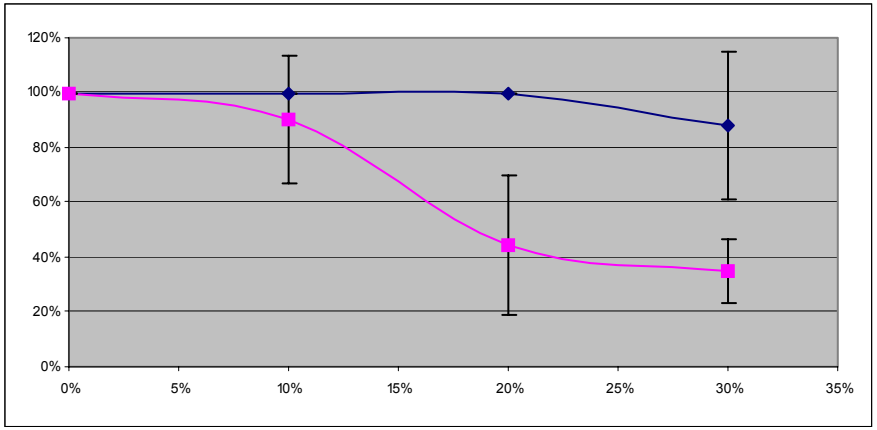
Basis graphs encounter high density of noise nodes & edges in these tests – as shown on previous slides!





Results Show Improvement Compared to Recent Publication

- Graph Inputs:
  - Model A (0.2)
  - Noise added to one input (0-30%)
  - 50 nodes, nominal.
- Mean size of common subgraph, with one standard deviation error bars.
- Upper Curve: Basis Graphs.
- Lower Curve: LeRP.



- Caelli-Kosinov, IEEE-PAMI 4/'04.
  ~30% match for 30% added noise.

BG Can Yield Good Approximation To Maximal Subgraph

	Basis Graphs	LeRP
Edit Distance	0.6 +/- 1.5	4.9 +/- 2.1
% Size Difference	2 +/- 7 %	25 +/- 13 %

- Maximum common subgraph found exhaustively.
- Input graphs had 10 nodes nominal, 100% noise nodes additional.
- Six different graph types. 120 trials reported.

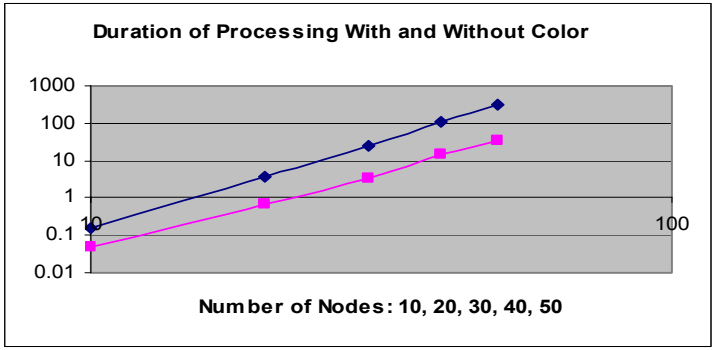
Deterministic
Effort and Memory Required

- Effort for histogram processing bounded by O(N^V) for N-node data graph and V-node basis graph, for worst-case.
- Effort for the mean case O(N D^(V-1)), for graphs with mean degree D.
- Basis graph suite, and mean V, are constant in reported trials.
- Maximum value of V=6 in reported tests.

- O(V N^2) memory required.

Lower curve had integer-valued node & edge coloring. Dynamic range: 4.

Duration of LeRP: under 0.1 seconds.



Factors Limit Increased Size of Basis Graphs

- *Do larger graphs necessarily require larger basis graphs?*

- Increase in V depends on structure of noise & input graph.
  - Should keep V small to reduce corruption of local description by noise.
  - Neighborhoods can be potentially large for modest V.
- Tests demonstrate best performance for low noise, implying choice of basis is less critical in these cases.

- Hence cases with both high and low noise may not always require large basis graphs for acceptable performance.

- Purpose of basis graph is to provide *local* comparison!
- Alternative: Increase number of relaxation iterations.

Revised Version of Paper Available

- Improvements to
  - Algorithm
  - Testing
  - Implementation
- Available via: www.ee.calpoly.edu/~fdepiero, See references.
- Encouraging use of approach…

Freeware Available!

- **No charge for non-profit, non-commercial, non-military, non-defense applications.**
- Executables only, currently.
- Find common subgraph for specific input data-graphs, defined in ASCII files. Mapping reported.
- Perform Monte Carlo simulations, specifying number of nodes, dynamic range of integer colors. Reports mean size of common graph.

Summary & Future Studies

- Basis Graphs provide local description of graph structure.
- Common subgraphs with same size as nominal input graphs can be recovered, with 100% additional noise nodes added to each input.
- Freeware available.

- Investigating:
  - A-priori PDFs formally relating edit distance of neighborhoods to mapping probability.
  - Optimal set of Basis Graphs. May vary depending on structure of noise & graph style?
  - Process to find A-priori mapping PDFs and optimal basis graphs, given a style of input graphs.
  - Faster processing, forming approximate histogram.

- Applications:
  - Graphs as measurements, real-time comparison.
  - Image registration & speech recognition.
  - Clustering analyses of graph-based measurements, to find probabilities of nodes & edges for typical measurements. Incorporate probabilities into matching process.