

Visualization of Large Graphs

Romain Bourqui
Maître de Conférence

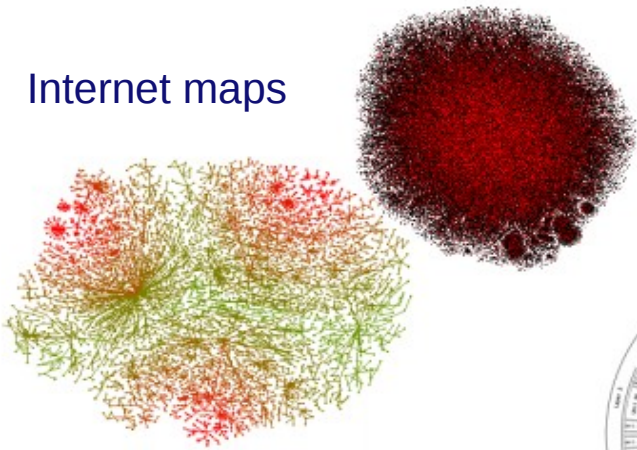


- I. Introduction**
- II. Multiscale force directed algorithms**
- III. Compound Graph Visualization**
- IV. Fitting the Level Of Detail with the user interest**
- V. Conclusion and Future Work**

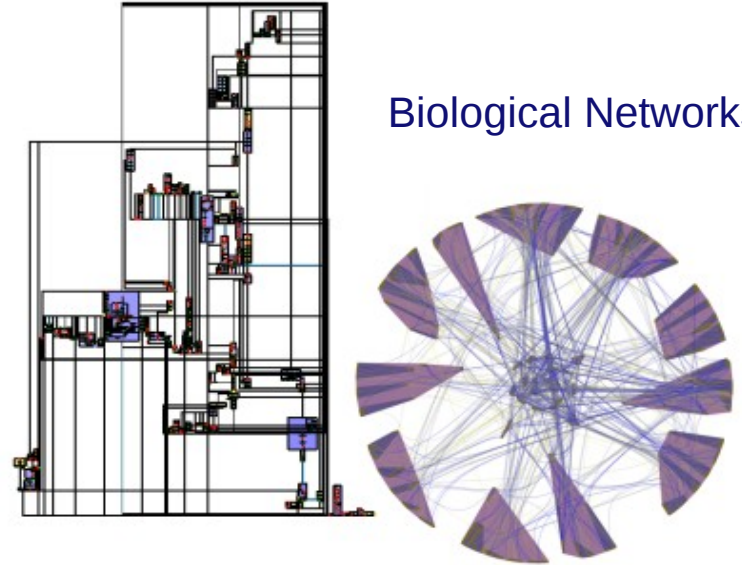
- I. Introduction**
- II. Multiscale force directed algorithms**
- III. Compound Graph Visualization**
- IV. Fitting the Level Of Detail with the user interest**
- V. Conclusion and Future Work**

Introduction

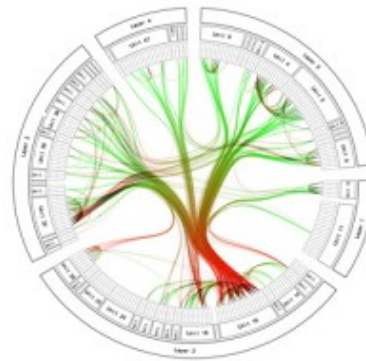
Internet maps



Biological Networks



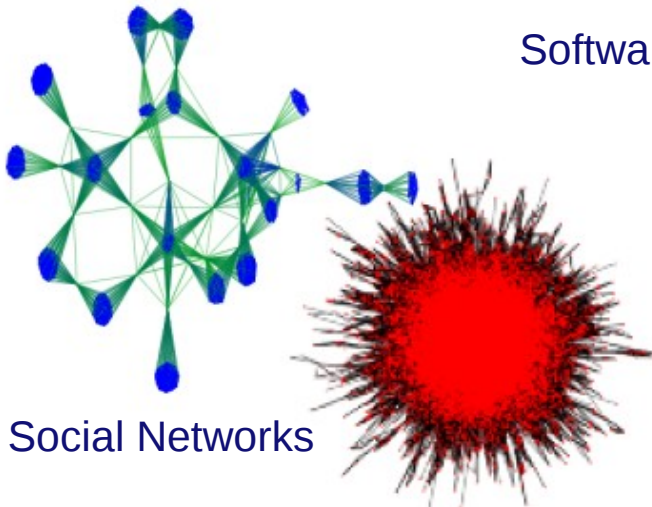
Software architectures



...

Graphs are useful

Social Networks



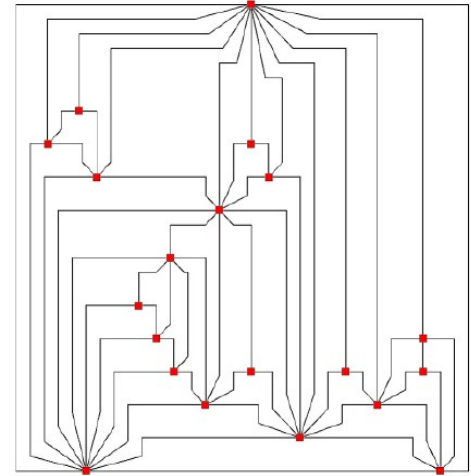
Introduction: Graph Drawing

**Dedicated algorithms for
particular graphs**

Introduction: Graph Drawing

Dedicated algorithms for particular graphs

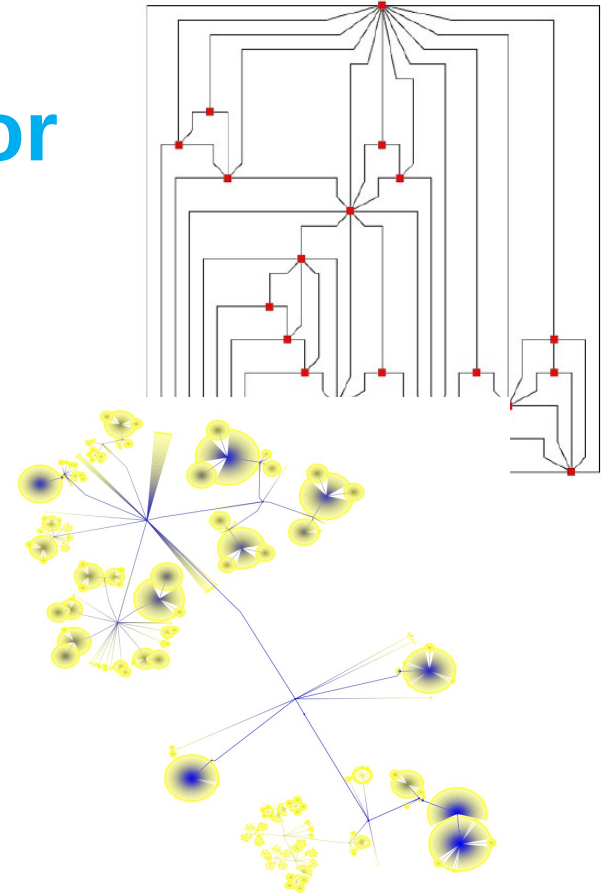
- Planar graphs



Introduction: Graph Drawing

Dedicated algorithms for particular graphs

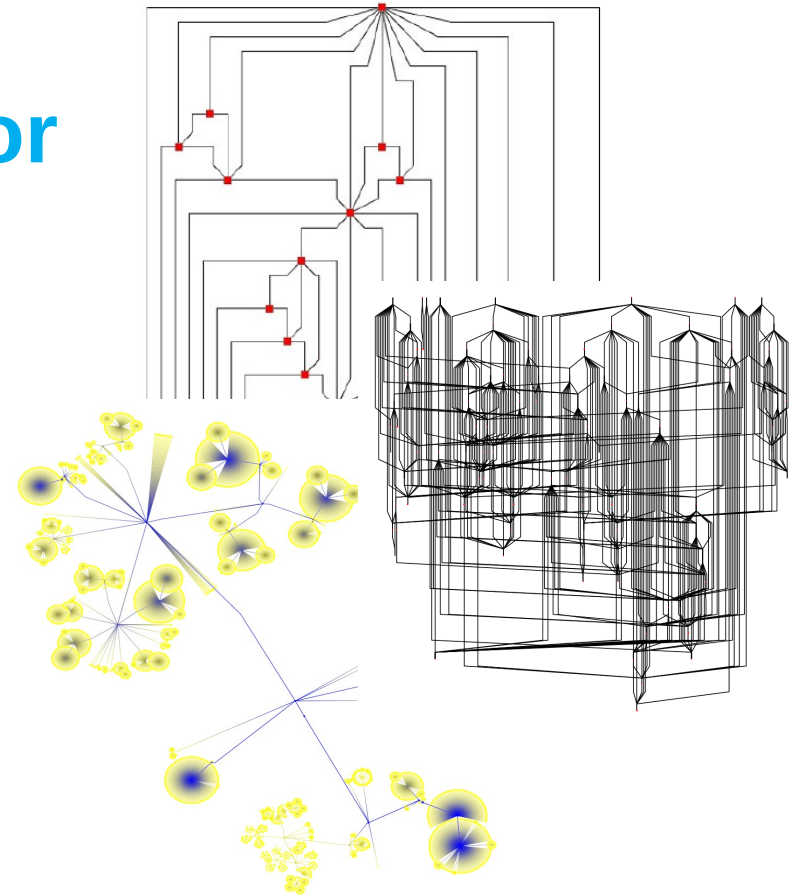
- Planar graphs
- Trees



Introduction: Graph Drawing

Dedicated algorithms for particular graphs

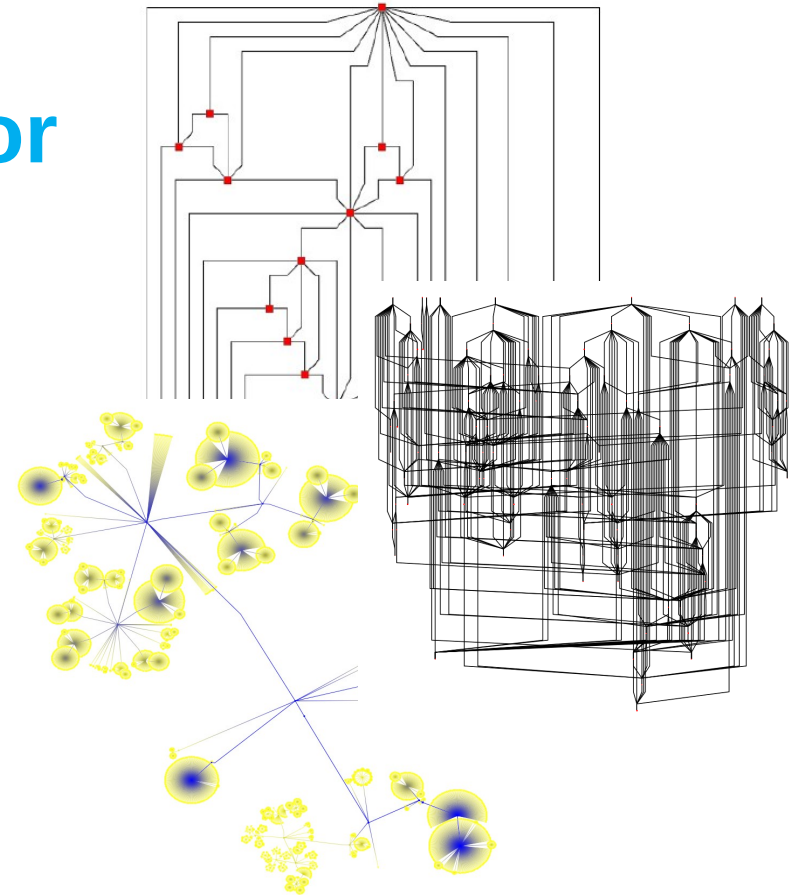
- Planar graphs
- Trees
- Directed Acyclic Graphs



Introduction: Graph Drawing

Dedicated algorithms for particular graphs

- Planar graphs
- Trees
- Directed Acyclic Graphs
- ...



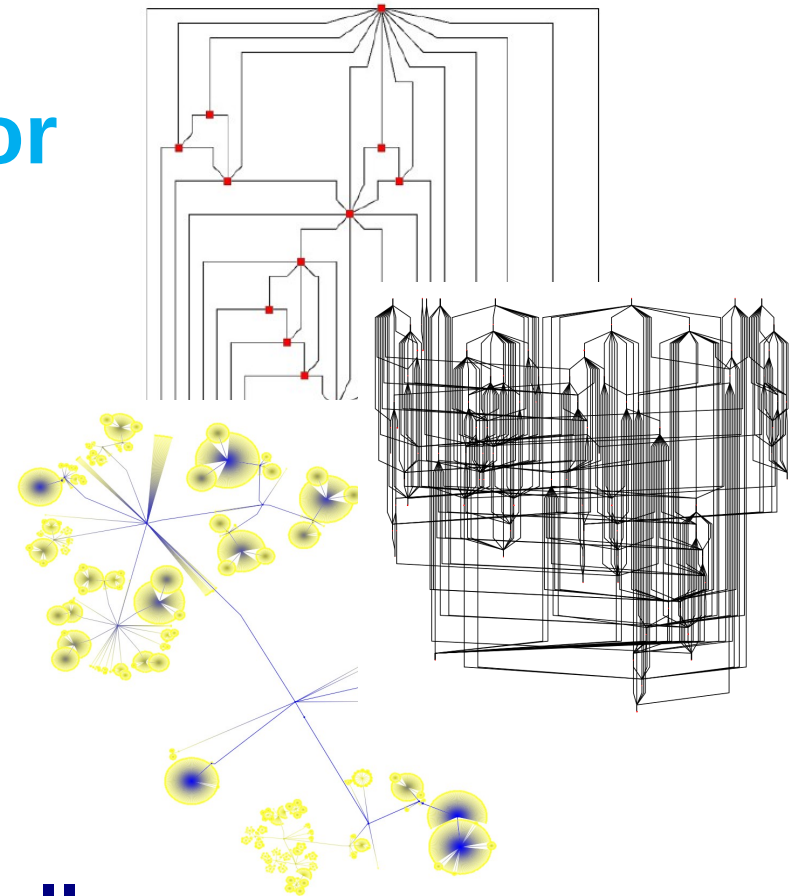
Introduction: Graph Drawing

Dedicated algorithms for particular graphs

- Planar graphs
- Trees
- Directed Acyclic Graphs
- ...

Problem:

real-world graphs usually
do not belong to these classes

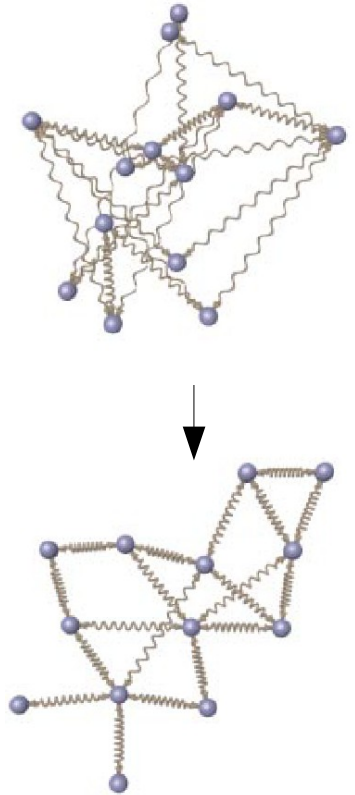


Introduction: General Graph Drawing

Force directed algorithms:

- Nodes are physical objects
- Edges are springs linking the objects

⇒ Many algorithms (Eades 84, Kamada and Kawai 89, Fruchterman and Reingold 91, Frick et al. 94)



Introduction: General Graph Drawing

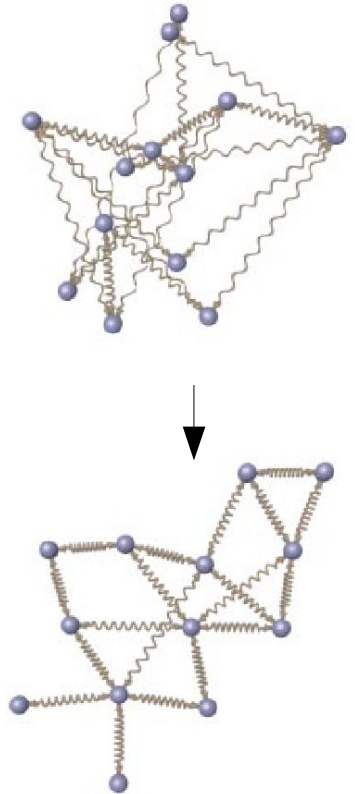
Force directed algorithms:

- Nodes are physical objects
- Edges are springs linking the objects

⇒ Many algorithms (Eades 84, Kamada and Kawai 89, Fruchterman and Reingold 91, Frick et al. 94)

Problem:

Time complexity is too high for large graphs



Introduction: Large Graph Drawing

Solution:

- Trade off between computation time and aesthetic criteria
- Multi-scale techniques

⇒ Many algorithms (*Gajer and Kobourov 00, Hachul and Junger 04, Adai et al. 04, Lauther 06*)

- I. Introduction**
- II. Multiscale force directed algorithms
- III. Compound Graph Visualization
- IV. Fitting the Level Of Detail with the user interest
- V. Conclusion and Future Work

Outline

- I. Introduction
- II. Multiscale force directed algorithms**
- III. Compound Graph Visualization
- IV. Fitting the Level Of Detail with the user interest
- V. Conclusion and Future Work

Multiscale force directed:

- Visually pleasant and structurally significant results
- Good computation time
- ...

Among them:

- **GRIP** [*Gajer and Kobourov 00*]
- **FM3** [*Hachul and Junger 04*]
- **LGL** [*Adai et al. 04*]

Among them:

- **GRIP** [*Gajer and Kobourov 00*]
- **FM3** [*Hachul and Junger 04*]
- **LGL** [*Adai et al. 04*]

Main Idea:

- **Election of pivots (several levels)**
- **Placement of each level using force directed algorithms**

Two main steps:

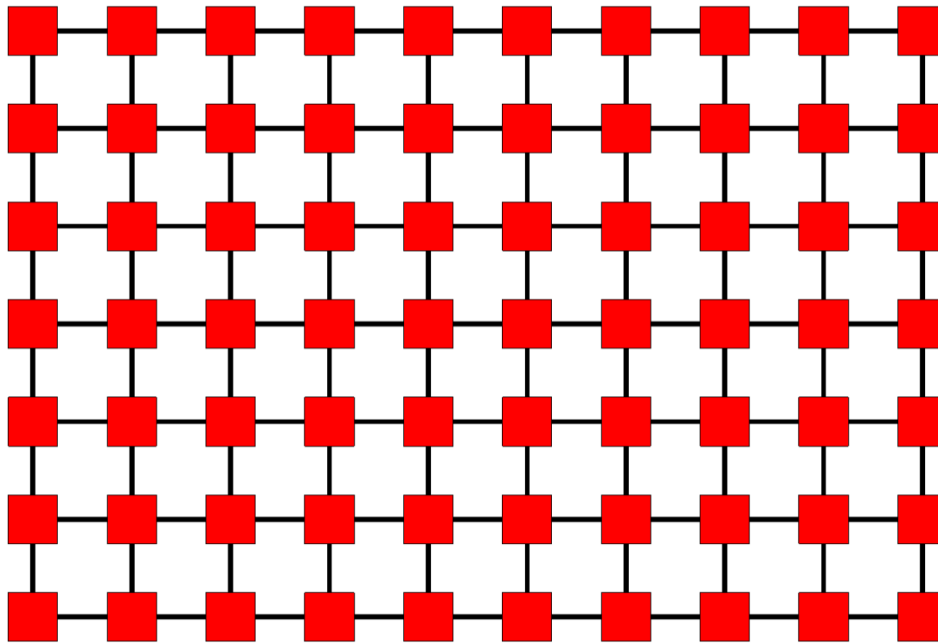
- **Vertex filtration: *Maximal Independant Set Filtration (MISF)***
- **Vertex placement using force directed algorithms (Fruchterman-Reingold and Kamada-Kawai)**

Vertex Filtration (MISF)

- Let $G=(V, E)$ be a simple and connected graph

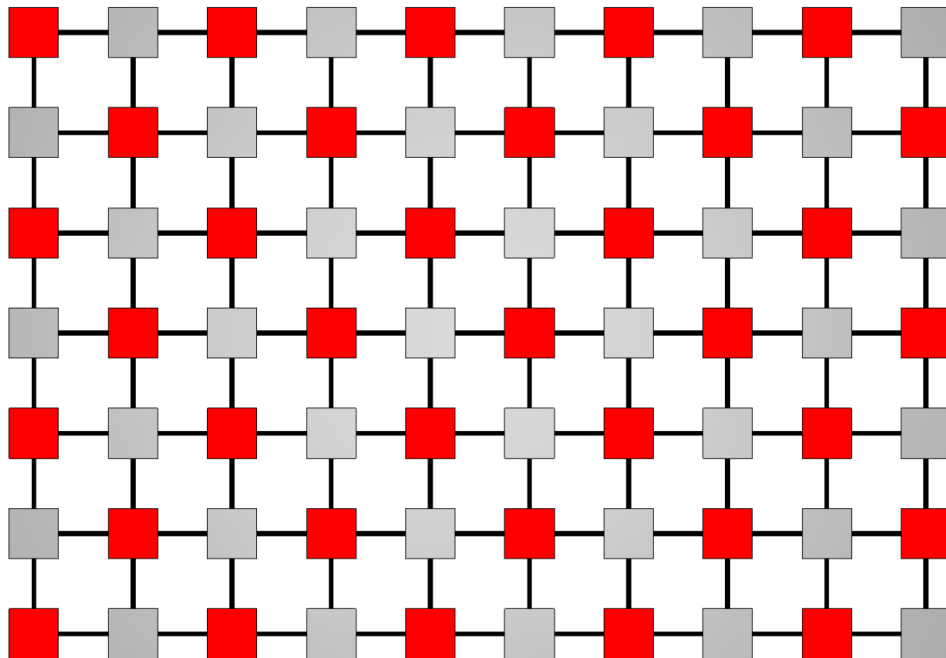
Maximal Independent Set Filtration (MISF) \mathcal{V} of the set of vertices V of G , $\mathcal{V} : V_0 \supset V_1 \supset V_2 \supset \dots \supset V_k \supset \emptyset$ where $V_0 = V$ and $\forall 0 < i \leq k$, V_i is a maximal subset of V_{i-1} such that for all pairs u, v of vertices of V_i the graph distance between u and v is greater or equal to 2^i .

Vertex Filtration (MISF)



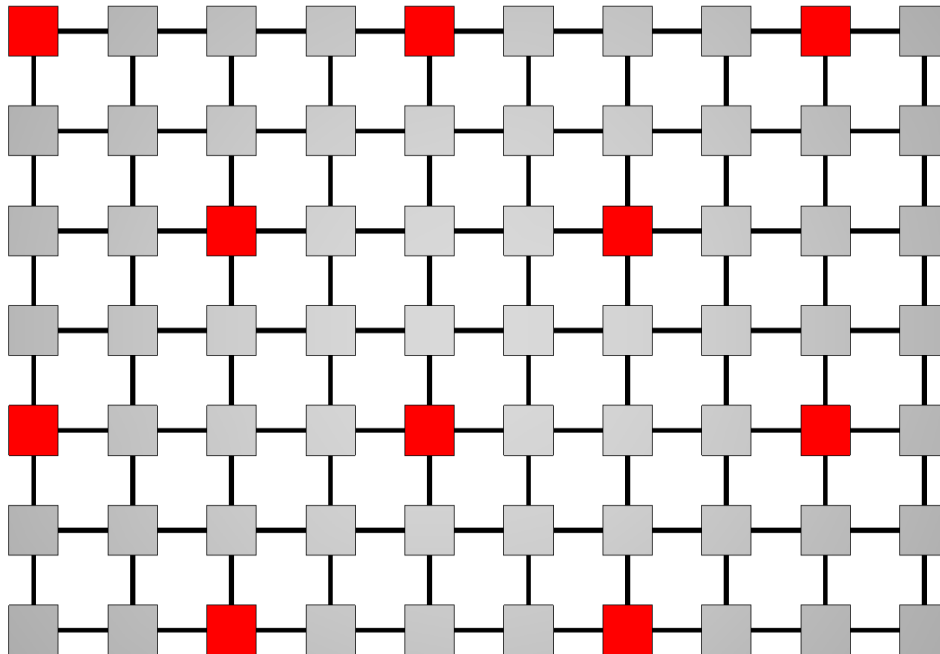
- $|V_0| = |V| = 70$

Vertex Filtration (MISF)



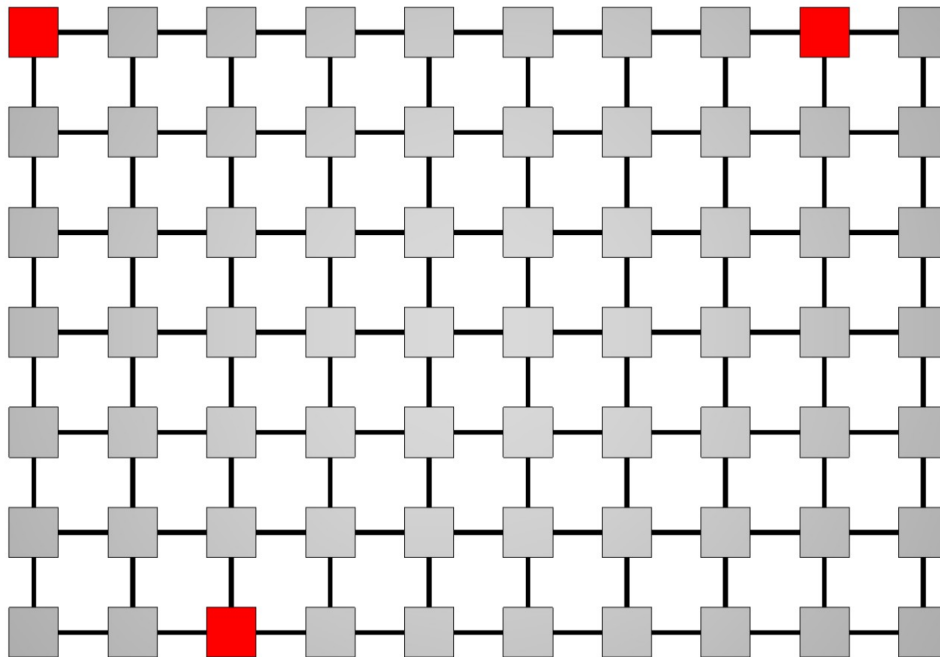
- $|V_0| = |V| = 70$
- $|V_1| = 35$

Vertex Filtration (MISF)



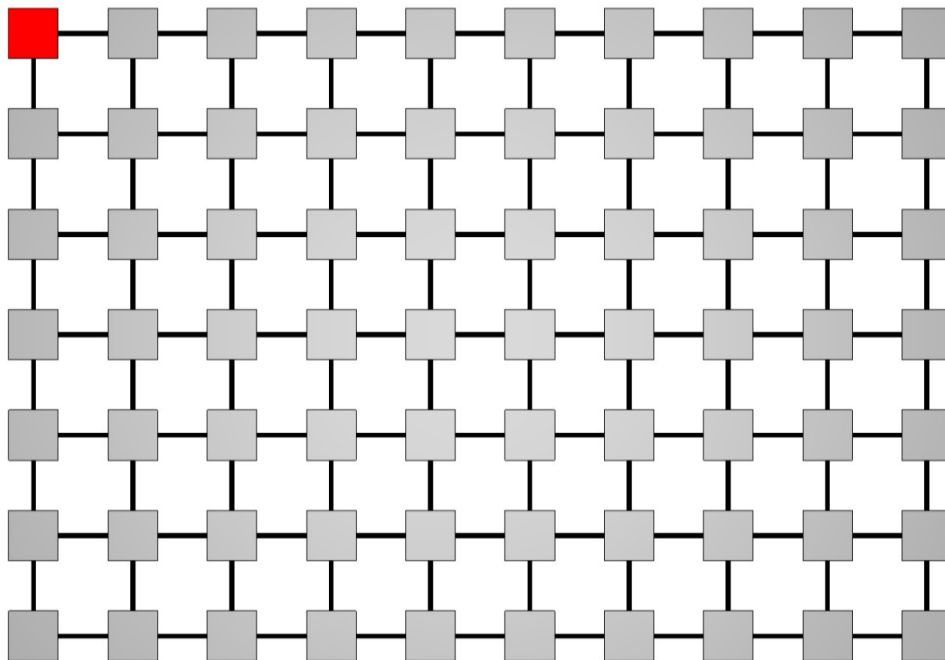
- $|V_0| = |V| = 70$
- $|V_1| = 35$
- $|V_2| = 10$

Vertex Filtration (MISF)



- $|V_0| = |V| = 70$
- $|V_1| = 35$
- $|V_2| = 10$
- $|V_3| = 3$

Vertex Filtration (MISF)



- $|V_0| = |V| = 70$
- $|V_1| = 35$
- $|V_2| = 10$
- $|V_3| = 3$
- $|V_4| = |V_k| = 1$

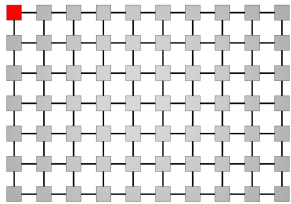
Vertex Filtration (MISF)

For each i from 1 to k :

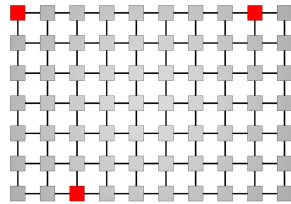
1. Let $V^* = V_{i-1}$
2. Randomly choose a vertex u of V^* .
3. Remove u and all vertices at distance at most 2^{i-1} from V^* and add u in V_i .
4. Go to step 2 until there is no vertex in V^* .

Vertex Placement

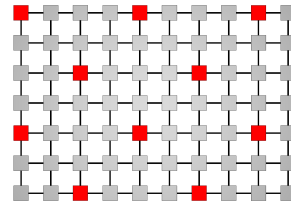
Placement of $V_k, V_{k-1} \setminus V_k, \dots, V_1 \setminus V_2, V_0 \setminus V_1$



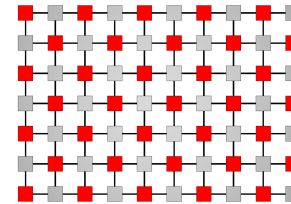
V_4



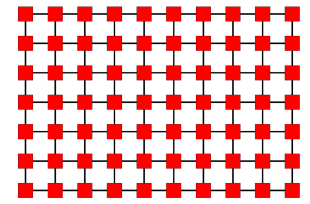
V_3



V_2



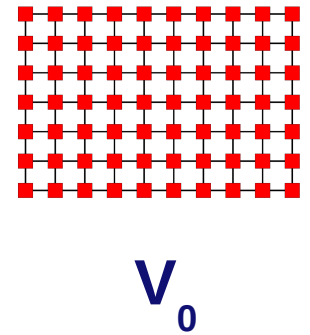
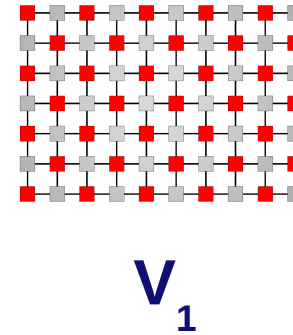
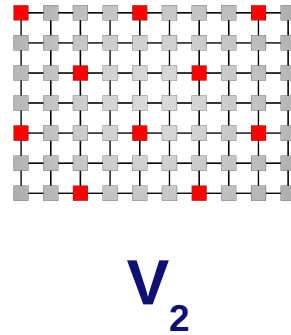
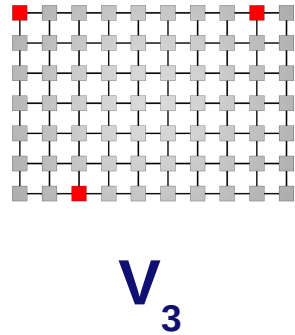
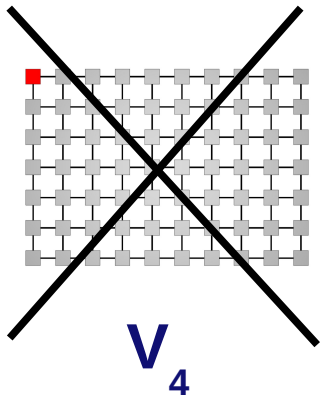
V_1



V_0

Vertex Placement

Placement of $V_k, V_{k-1} \setminus V_k, \dots, V_1 \setminus V_2, V_0 \setminus V_1$



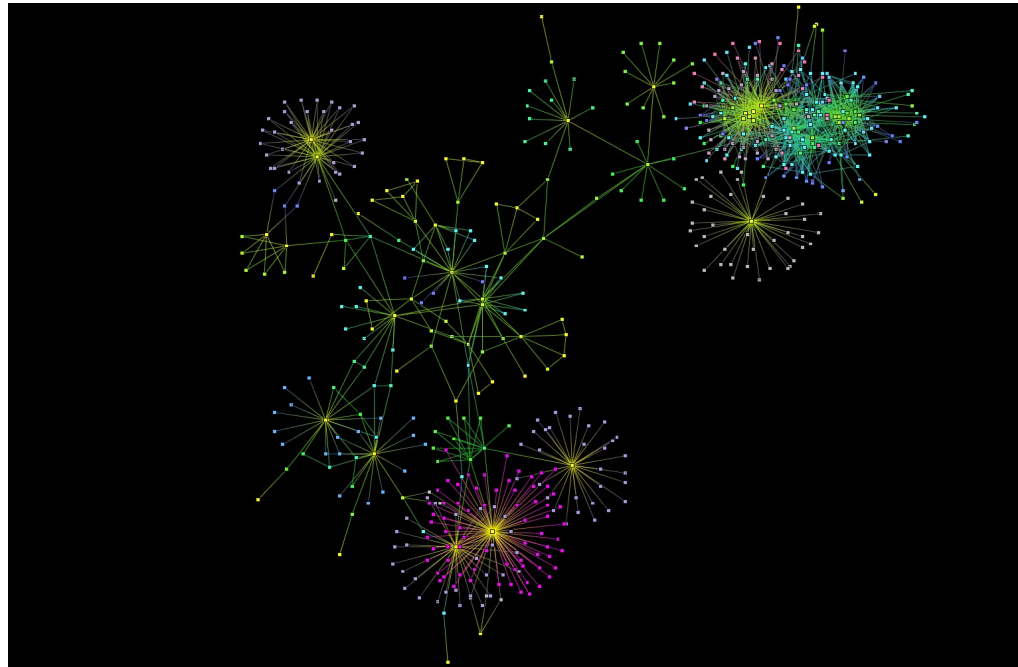
Vertex Placement

- Placement of V_k : vertices are laid out such that:

$$\forall u, v \in V_k : d_R(u, v) = d_G(u, v)$$

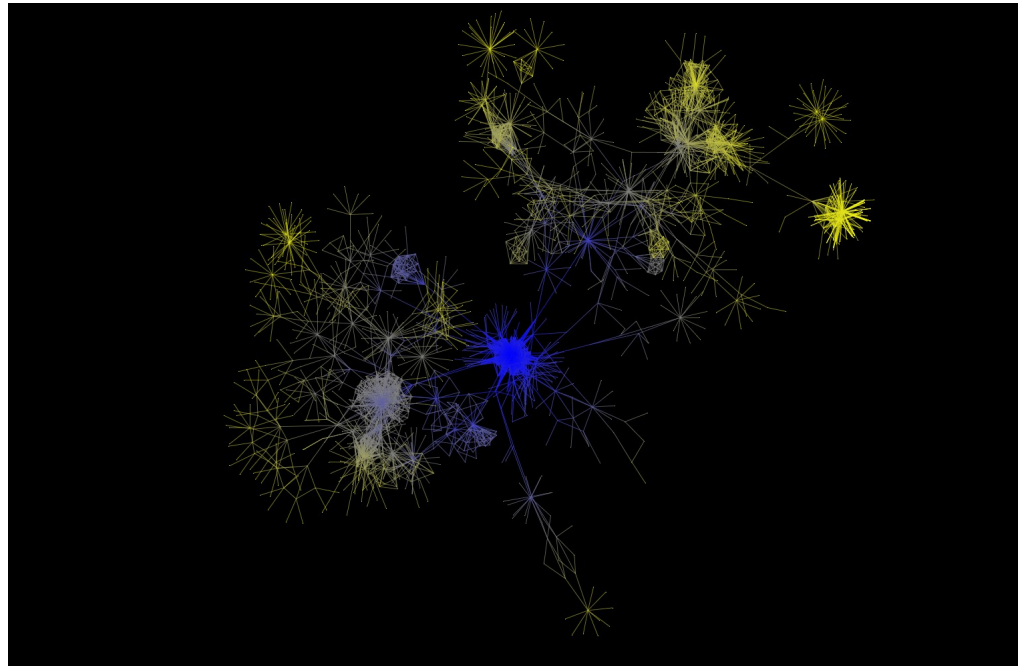
- Placement of $V_i \setminus V_{i-1}$ $0 < i < k$: Kamada-Kawai algorithm
- Placement of $V_0 \setminus V_1$: Fruchterman-Reingold algorithm

Examples/Results



GRIP: 0,13s GEM: 30s

Examples/Results

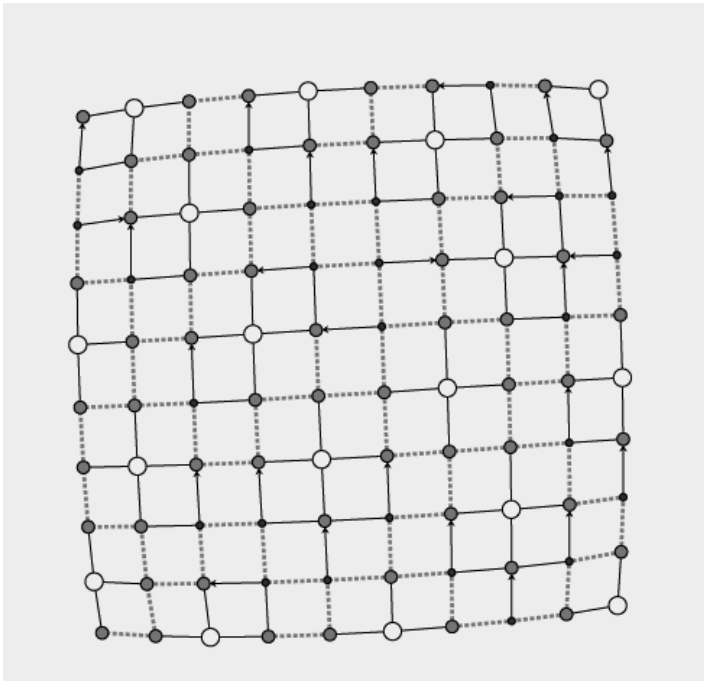


GRIP: 0,26s GEM: 15min

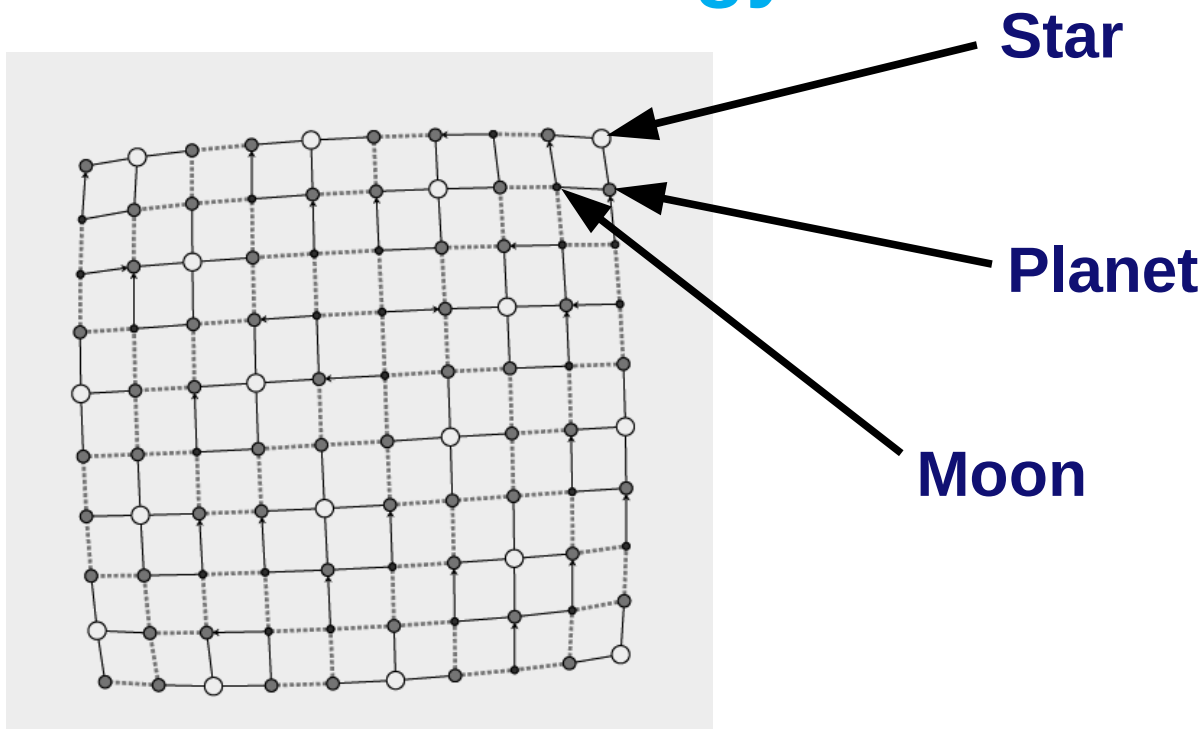
Main Steps:

- **Election of pivots: analogy the solar systems**
- **Placement using force directed approach**

Multilevel strategy:

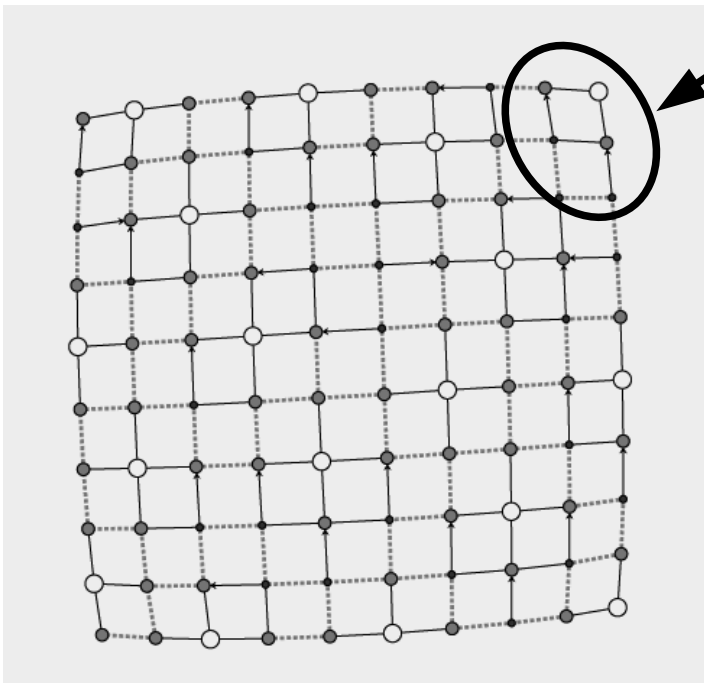


Multilevel strategy:



Multilevel strategy:

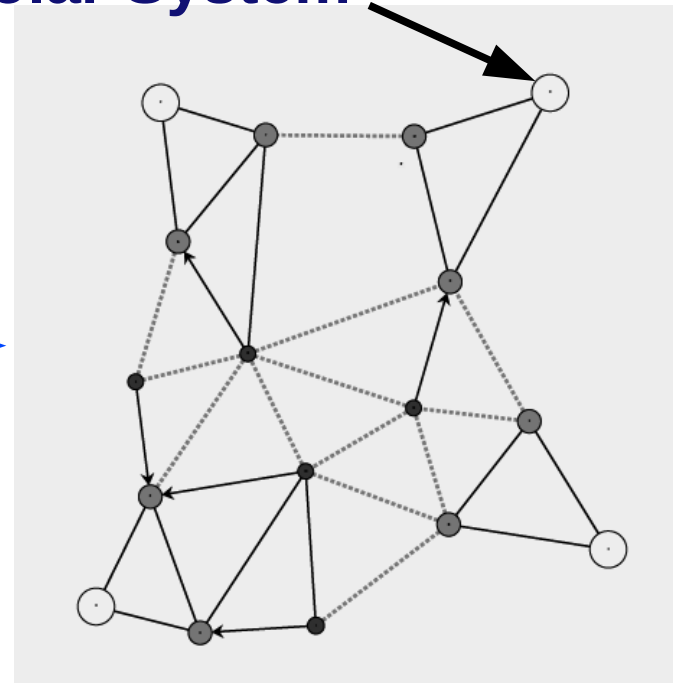
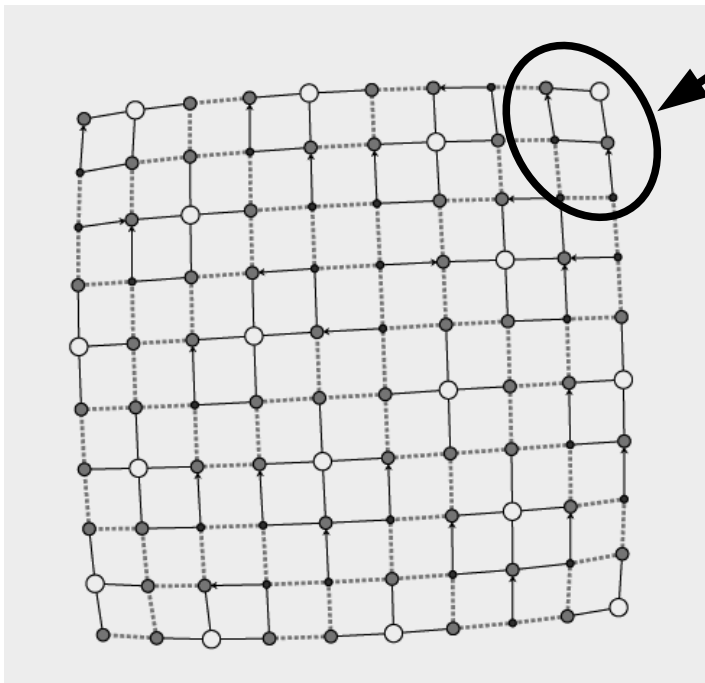
Solar System



Multiscale Force Directed: FM3 [HJ 04]

Multilevel strategy:

Solar System



Main Steps:

- Election of pivots: analogy the solar systems
- Placement using force directed approach

=> Time Complexity: $O(|V| \log|V| + |E|)$

Main Steps:

- **Election of pivots: spanning tree**
- **Placement using force directed approach**

Main Steps:

- Election of pivots: spanning tree
- Placement using force directed approach

=> Can take edge weights into account

Multiscale force directed:

- Visually pleasant and structurally significant results
- Good computation time
- ...

Remaining issues:

- Clutter (node-node or node-edge overlaps, edge-edge crossings)
- Rendering speed

Outline

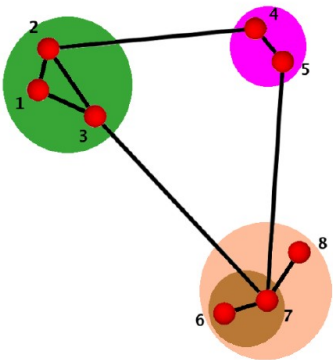
- I. Introduction
- II. Multiscale force directed algorithms**
- III. Compound Graph Visualization
- IV. Fitting the Level Of Detail with the user interest
- V. Conclusion and Future Work

Outline

- I. Introduction
- II. Multiscale force directed algorithms
- III. Compound Graph Visualization**
- IV. Fitting the Level Of Detail with the user interest
- V. Conclusion and Future Work

Introduction: Large Graph Visualization

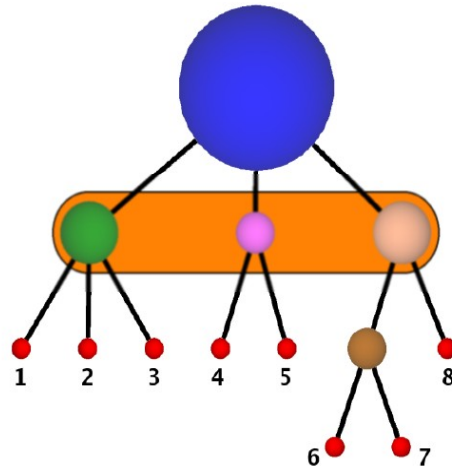
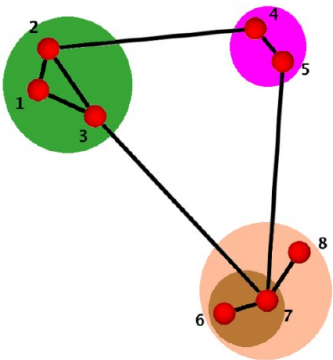
Building an *abstraction*



Introduction: Large Graph Visualization

Building an *abstraction*

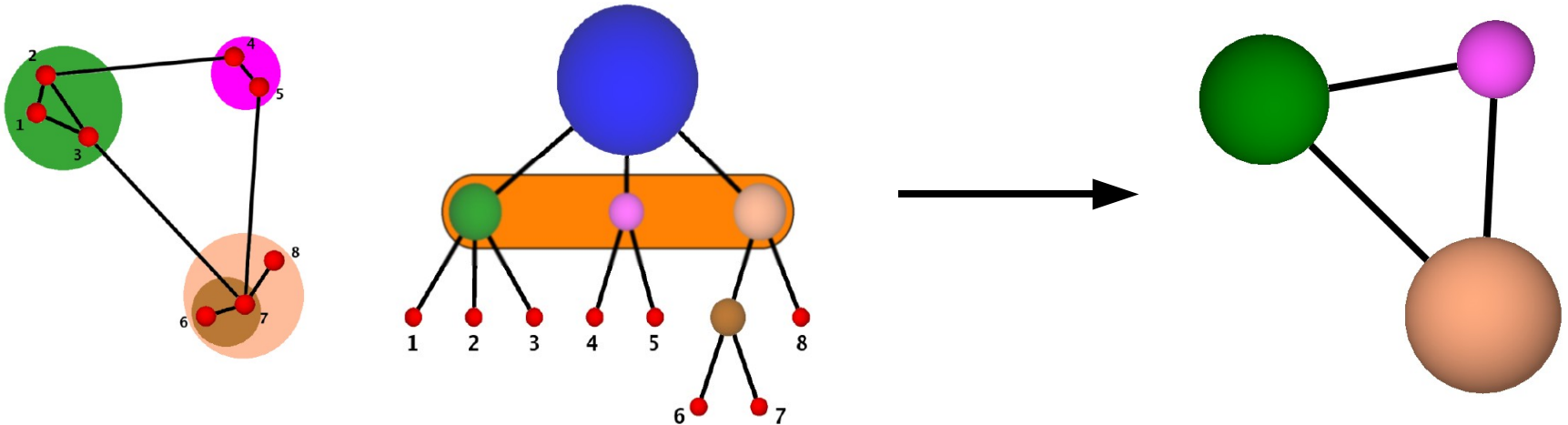
- Clustering algorithm (or input partition)



Introduction: Large Graph Visualization

Building an *abstraction*

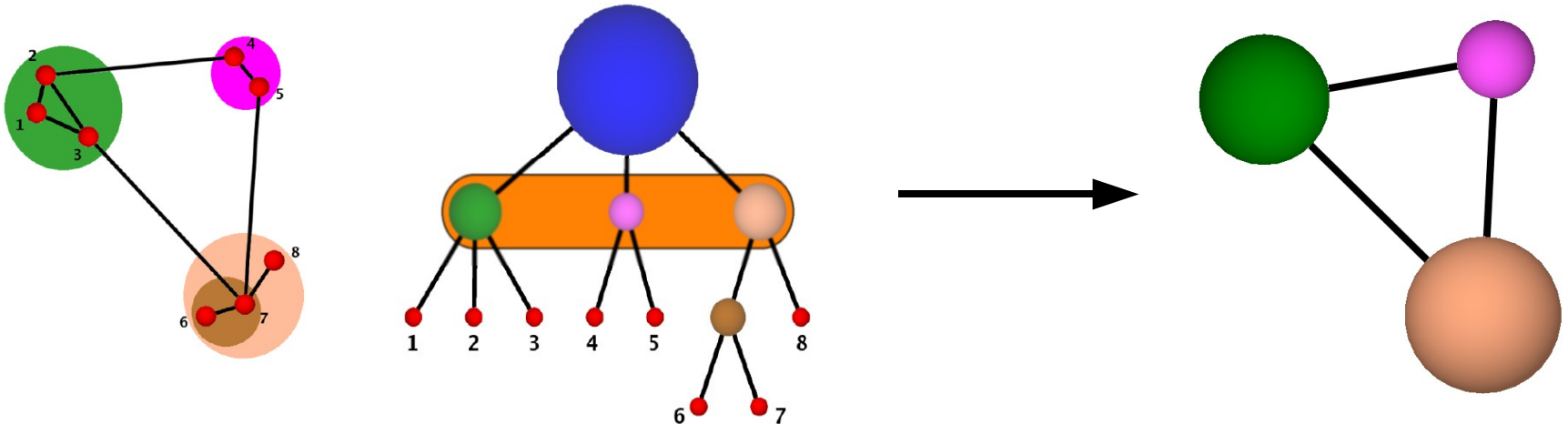
- Clustering algorithm (or input partition)
- Compound graph



Introduction: Large Graph Visualization

Building an *abstraction*

- Clustering algorithm (or input partition)
- Compound graph



⇒ Needs a dedicated interaction system

Compound Graph Visualization

Definitions

Let $G=(V,E)$ be a graph and $T=(V_T,E_T)$ a hierarchy tree

A set C is a **cut** in the hierarchy tree if C is a set of vertices such that no vertex of C is a successor/ancestor of another vertex of C .

A cut C is **maximal** iff there exists no $u \in V_T \setminus C$, such that $C \cup \{u\}$ is a cut.

A set C is a **sub-cut** iff C is a cut of a sub-hierarchy of T .

Compound Graph Visualization

We consider two approaches

Compound Graph Visualization

We consider two approaches

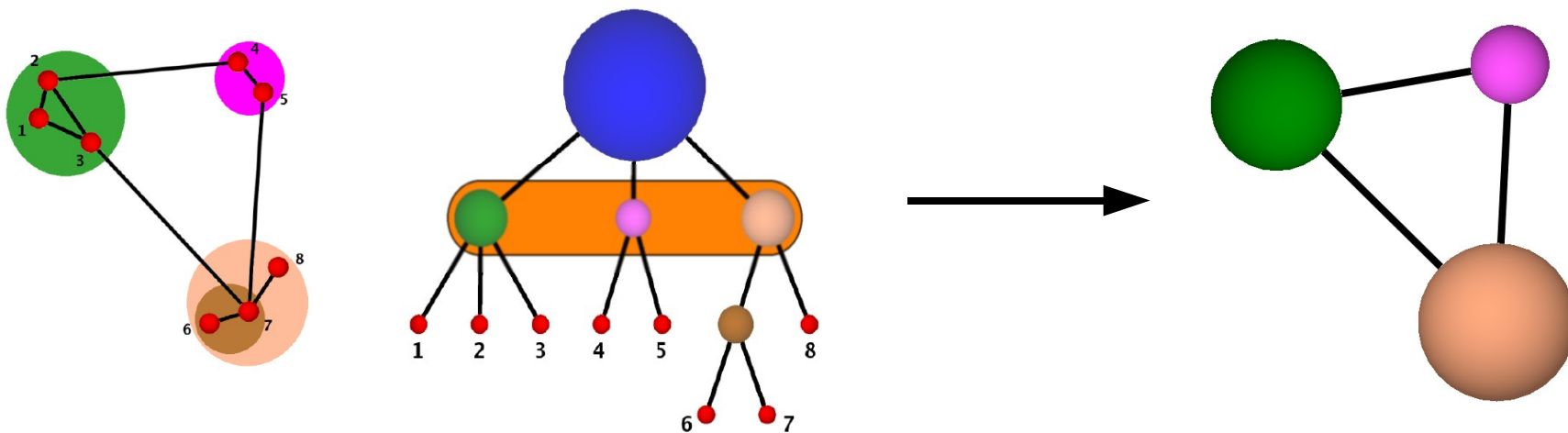
- **Classical compound visualization**

We consider two approaches

- **Classical compound visualization**
- **Multilvel compound visualization**

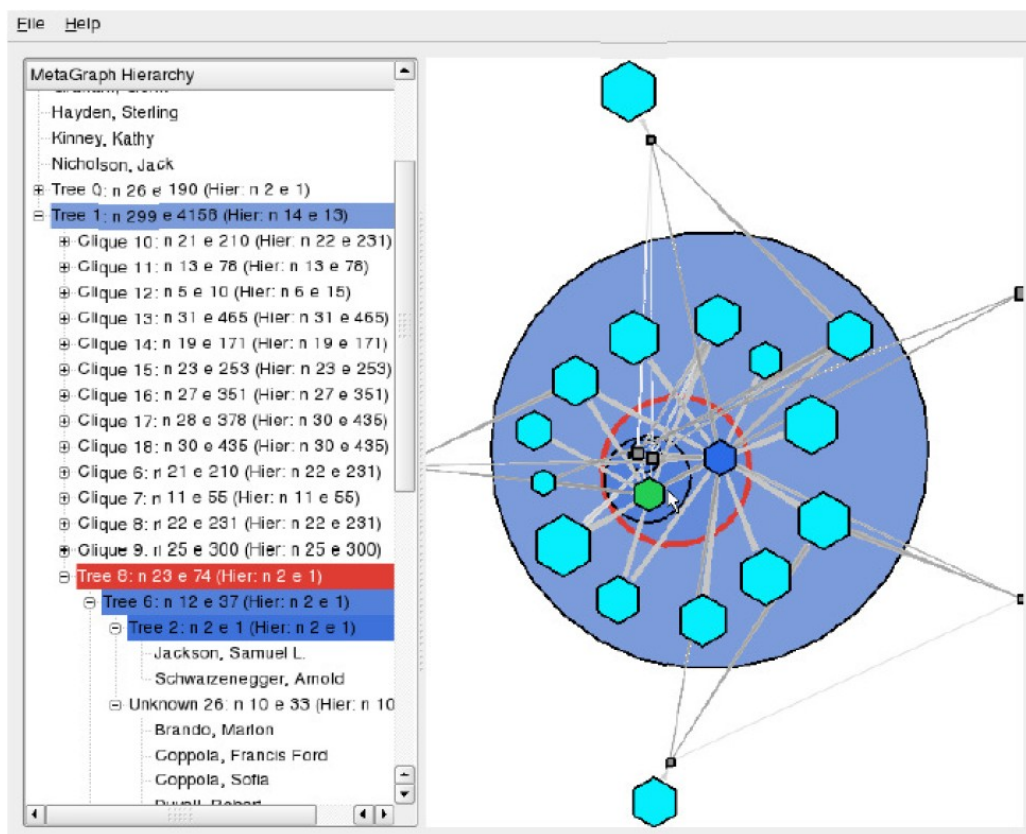
Classical Compound Graph Visualization

Visualize **one maximal cut** of the hierarchy tree



Classical Compound Graph Visualization

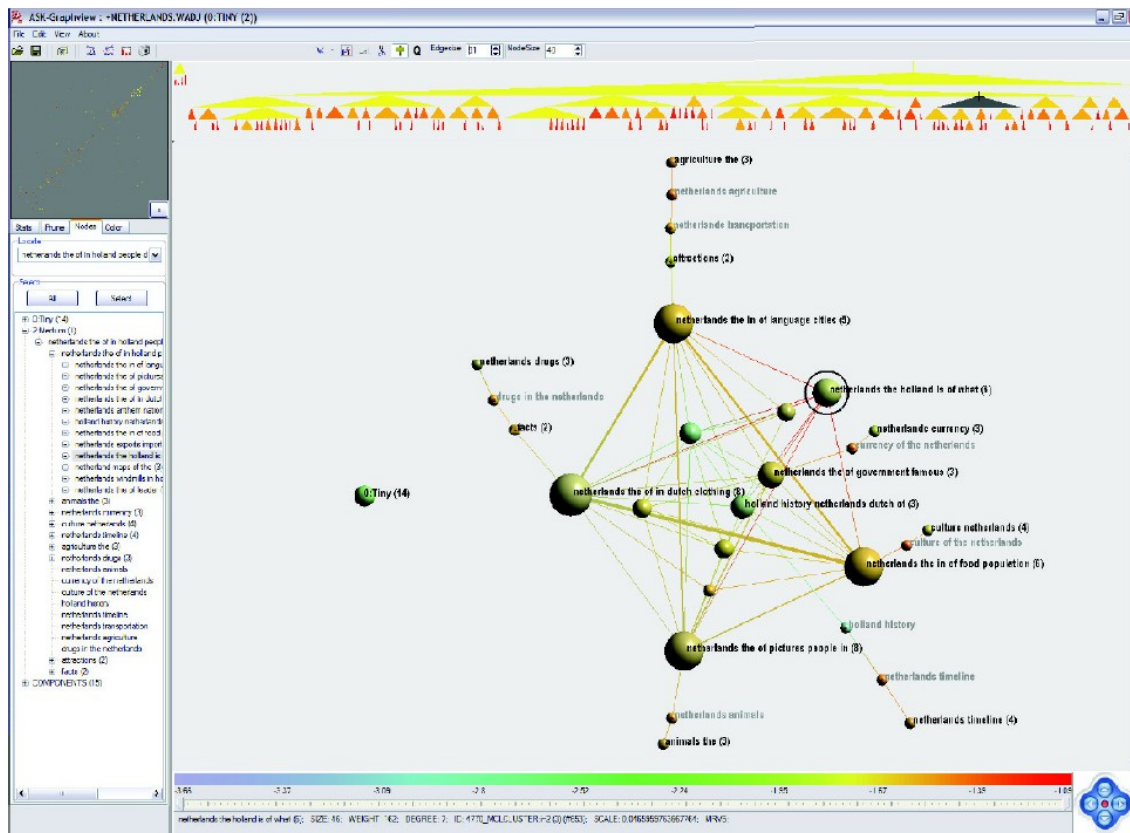
Examples: Grouse [Archambault et al. 07]



- Node shape shows cluster topology
- Cluster drawn on demand using the appropriate algorithm

Classical Compound Graph Visualization

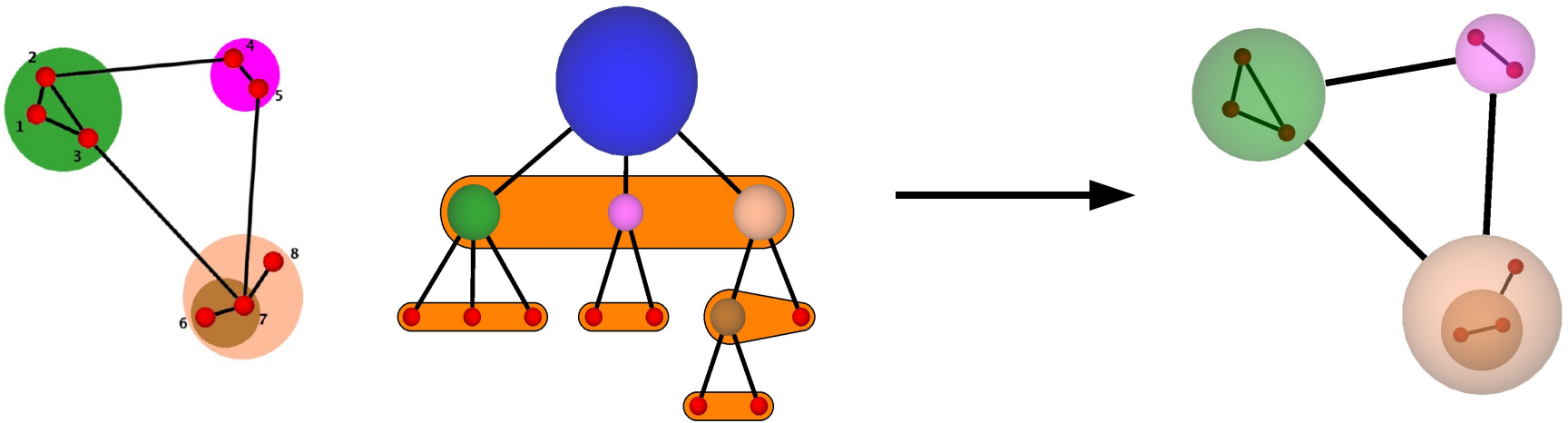
Examples: Ask-GraphView [Abello et al. 06]



- Huge graphs
- Cluster drawn on demand

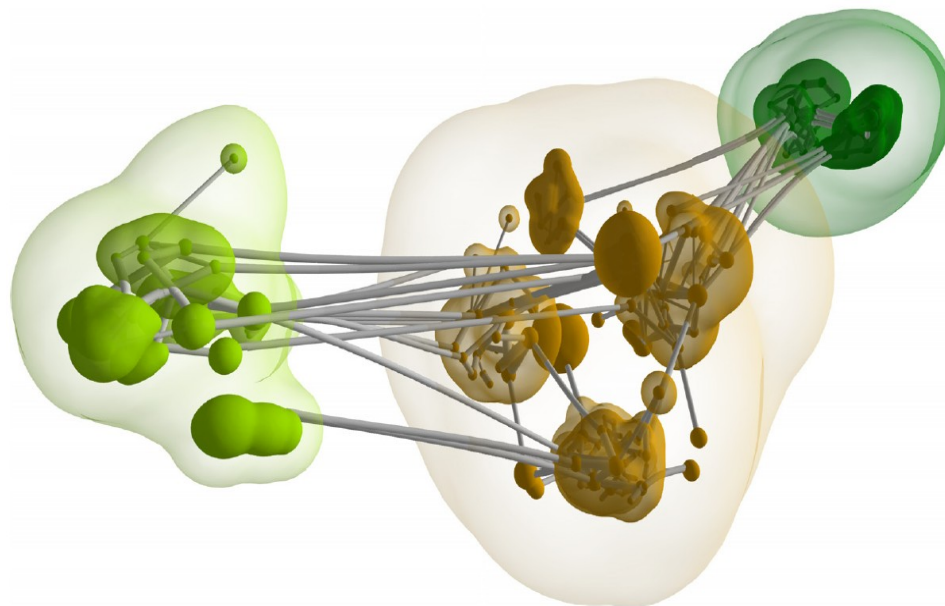
Multilevel Compound Graph Visualization

Visualize **several** *maximal sub-cuts* of the hierarchy tree



Multilevel Compound Graph Visualization

Examples: *[Balzer and Deussen 07]*



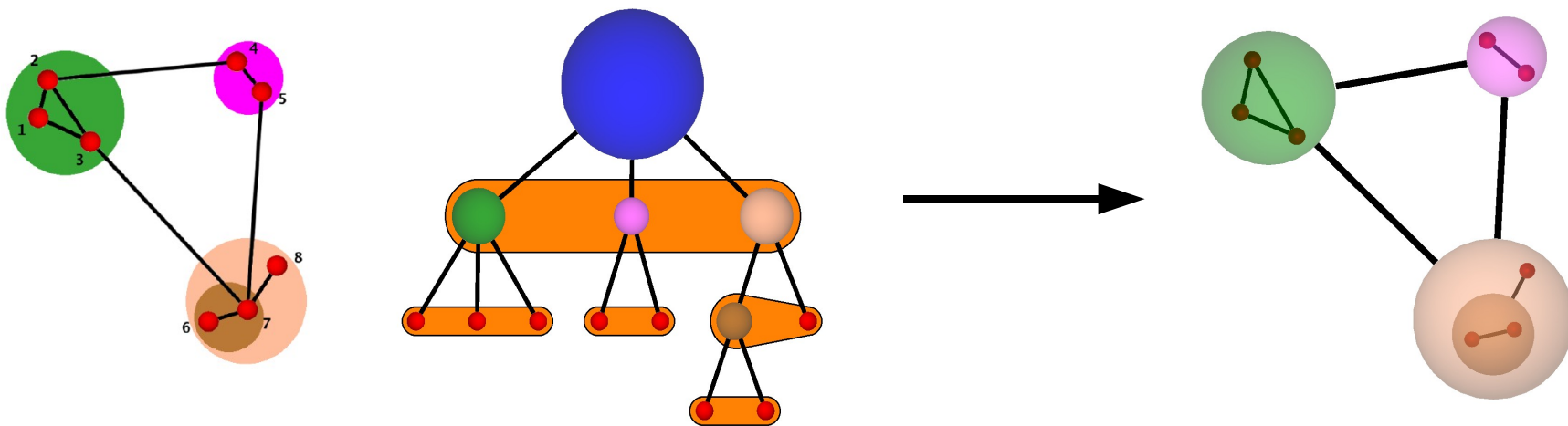
Problem

How to retrieve the relations between elements of the original network?

⇒ Enable the user to modify the *Level Of Detail (LOD)*

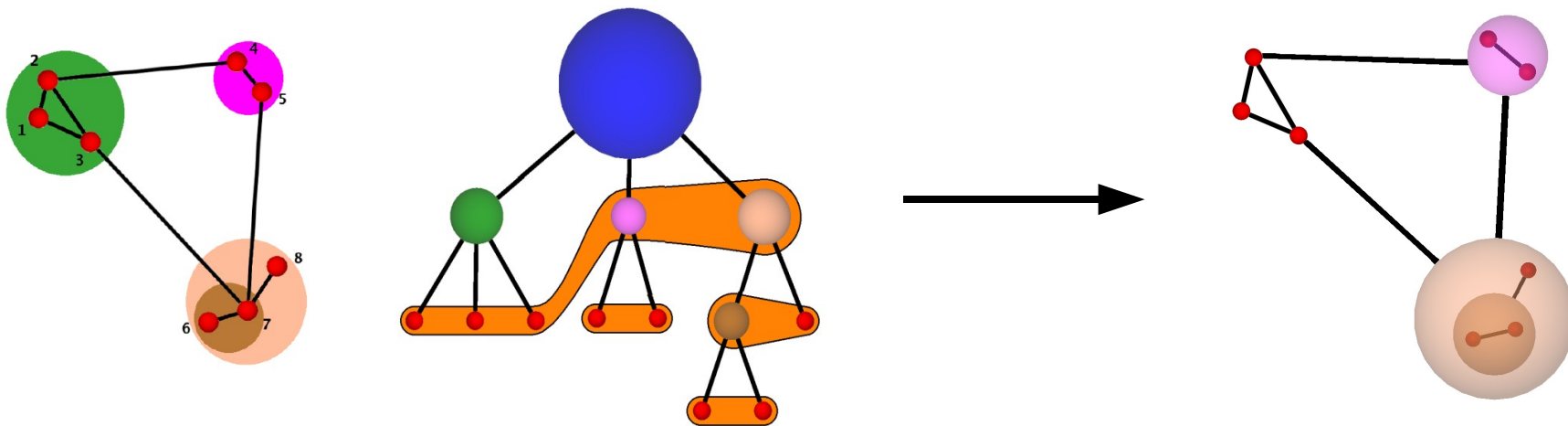
Compound Graph Visualization

Modifying the *LOD*: Expand/Collapse Operation



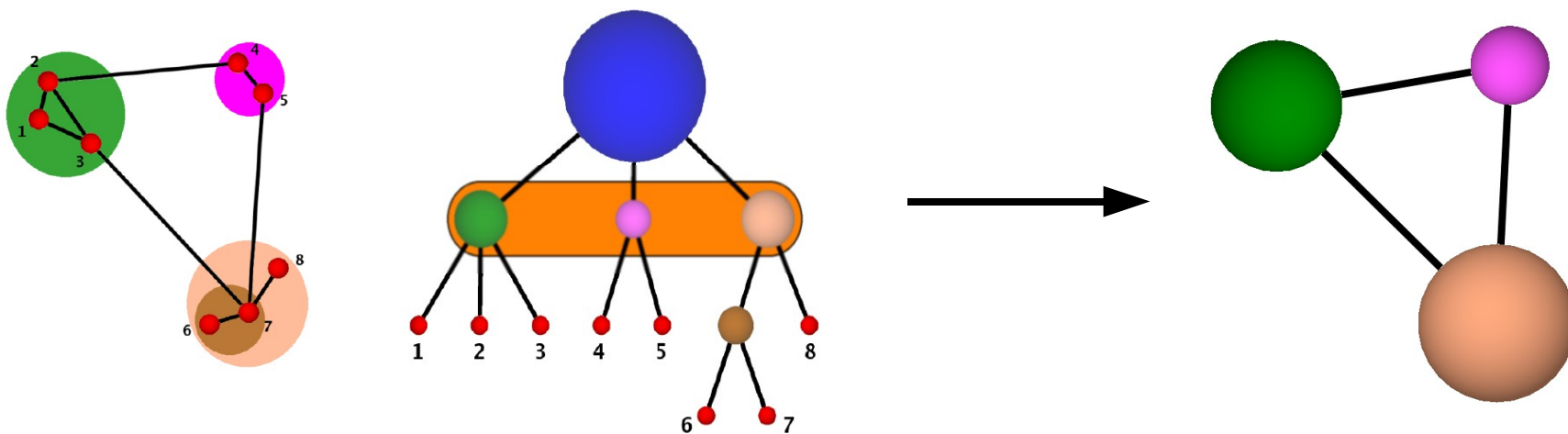
Compound Graph Visualization

Modifying the *LOD*: Expand/Collapse Operation



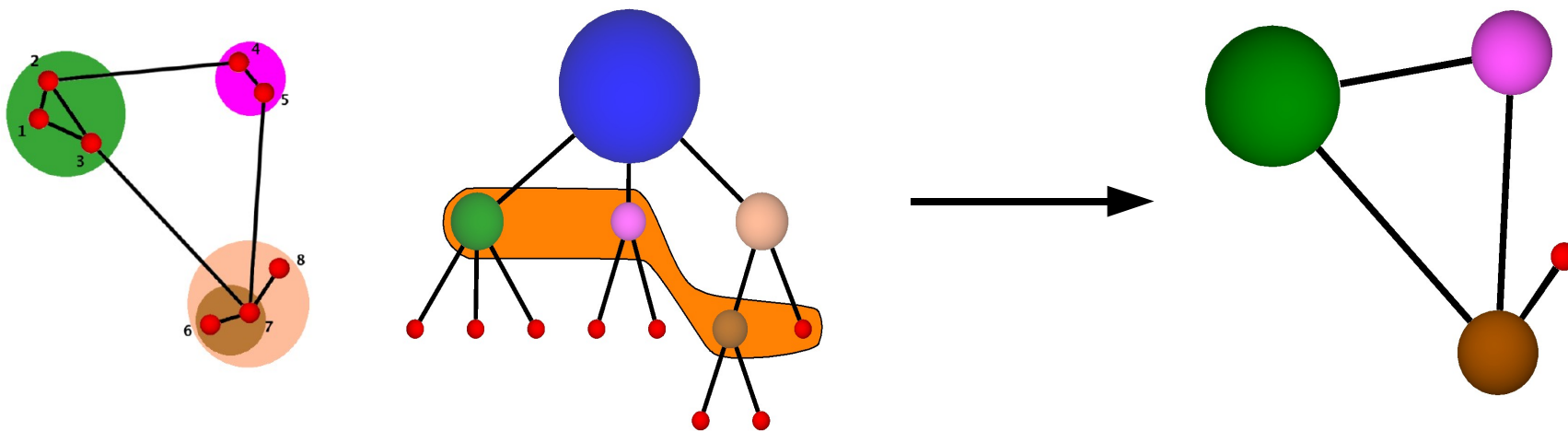
Classical Compound Graph Visualization

Modifying the *LOD*: Expand/Collapse Operation



Classical Compound Graph Visualization

Modifying the *LOD*: Expand/Collapse Operation



Combination of *Expand/Collapse* + *transparency*

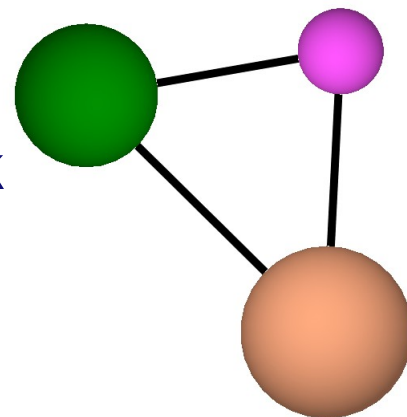
Classical Compound Graph Visualization

Advantages:

- **Allows fast rendering**
- **Gives an overview of the global structure of the graph**

Drawbacks:

- **Relationships within a sub-network**
- **Level Of Detail (LOD) too low**
- **Labeling problem**



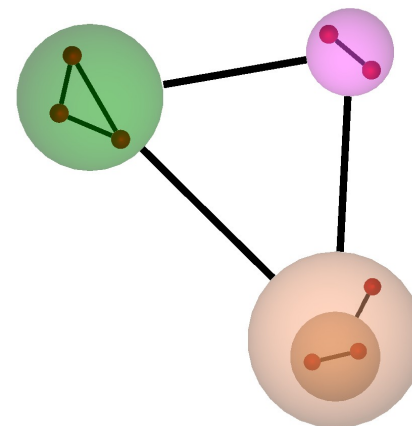
Multilevel Compound Graph Visualization

Advantages:

- Can reduce the number of displayed elements
- Gives overview of both the global and the local structures of the graph

Drawbacks:

- Level Of Detail (LOD) may be unnecessary high in some regions of the visualization



Outline

- I. Introduction
- II. Multiscale force directed algorithms
- III. Compound Graph Visualization**
- IV. Fitting the Level Of Detail with the user interest
- V. Conclusion and Future Work

Outline

- I. Introduction
- II. Multiscale force directed algorithms
- III. Compound Graph Visualization
- IV. Fitting the Level Of Detail with the user interest**
- V. Conclusion and Future Work

Three main aspects:

- **Mixing classical and multilevel compound visualization**
- **Multiple views**
- **Dedicated interaction**

Mixing classical and multilevel visualization

Summary:

- **Classical compound visualization: low *LOD***
- **Multilevel compound visualization: too high *LOD* in some parts of the visualization**

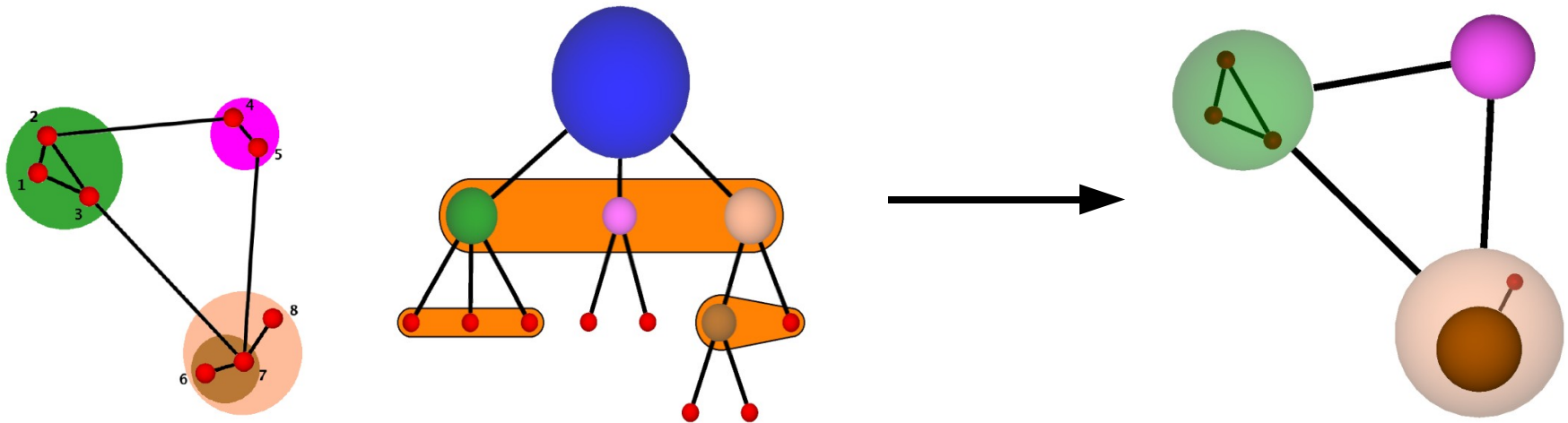
Mixing classical and multilevel visualization

Use of the classical visualization in parts of the visualization that are uninteresting for the user and the multilevel visualization for the remaining parts

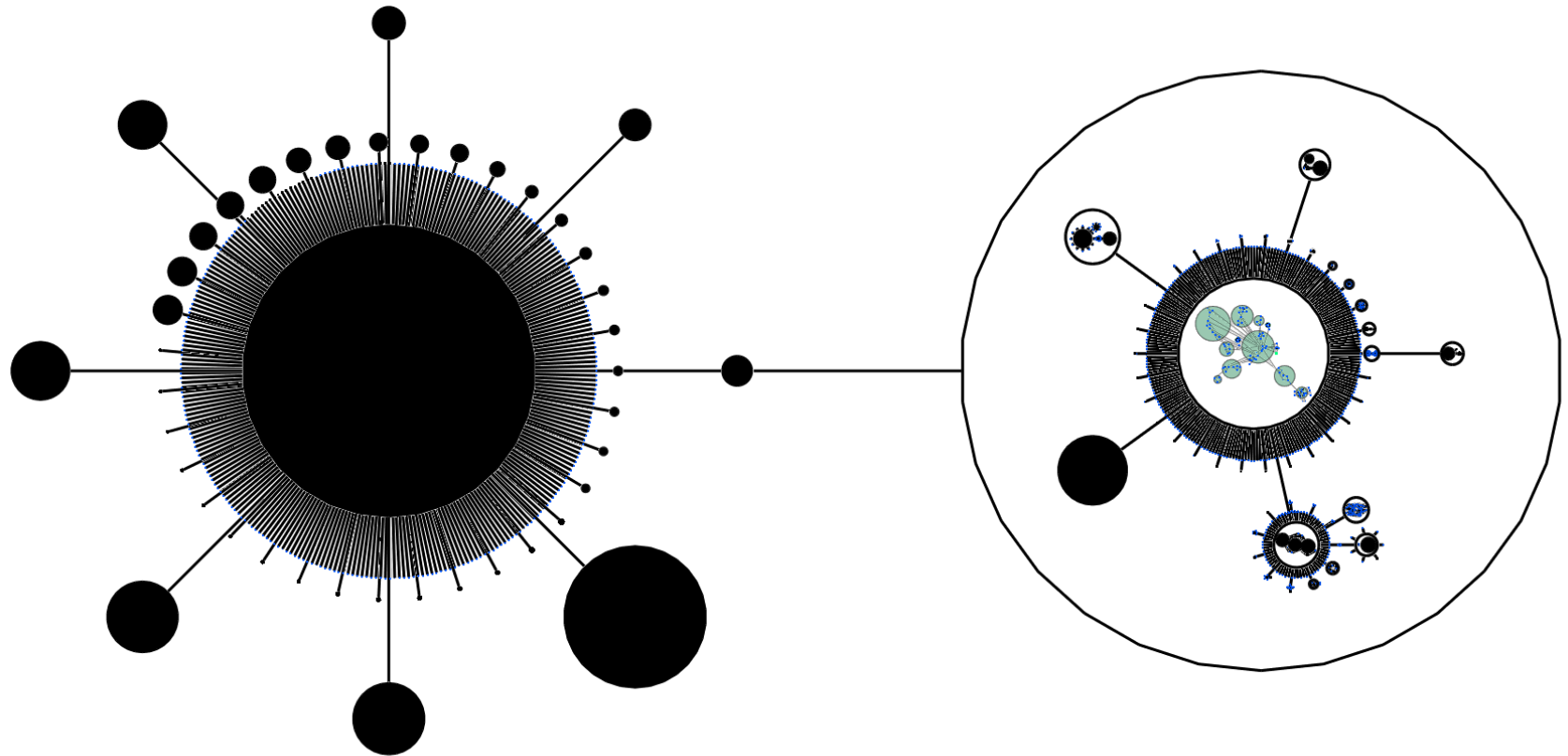
⇒ Combines advantages of both techniques

Mixing classical and multilevel visualization

Use of the classical visualization in parts of the visualization that are uninteresting for the user and the multilevel visualization for the remaining parts



Mixing classical and multilevel visualization



Example on UBC website

Problem:

**Clustering algorithms are usually
heuristics**

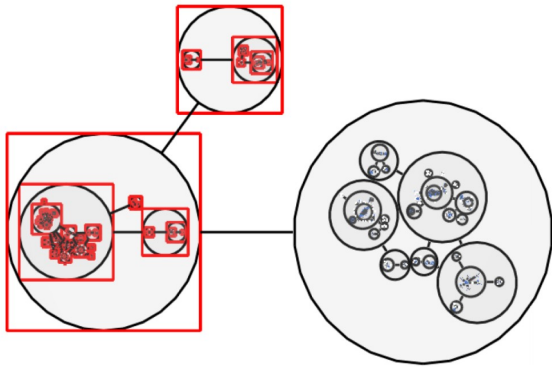
Problem:

Clustering algorithms are usually heuristics

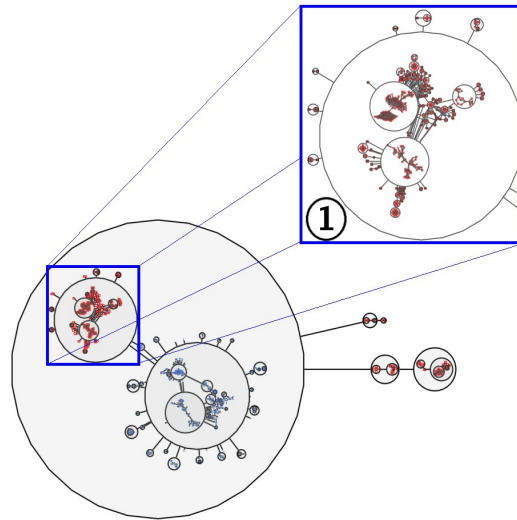
Solution:

- **Provide multiple clustering algorithms**
- **Support multiple views and *Linking & Brushing***

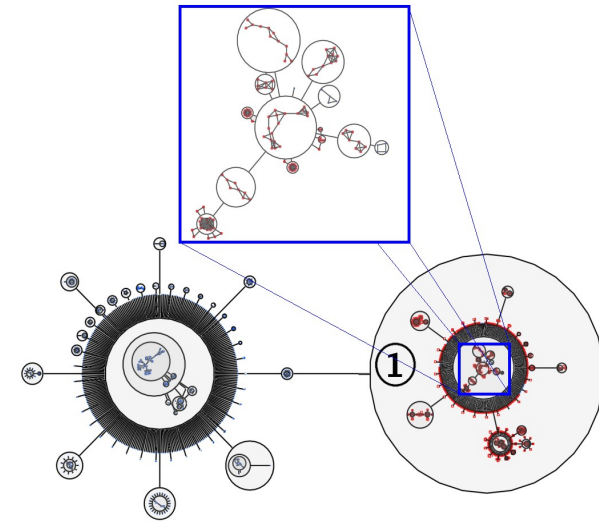
Multiple Views



MCL clustering



Strength Clustering



Coreness based

UBC website

Modifying the *LOD*

- In previous work: *Expand/Collapse* Operation

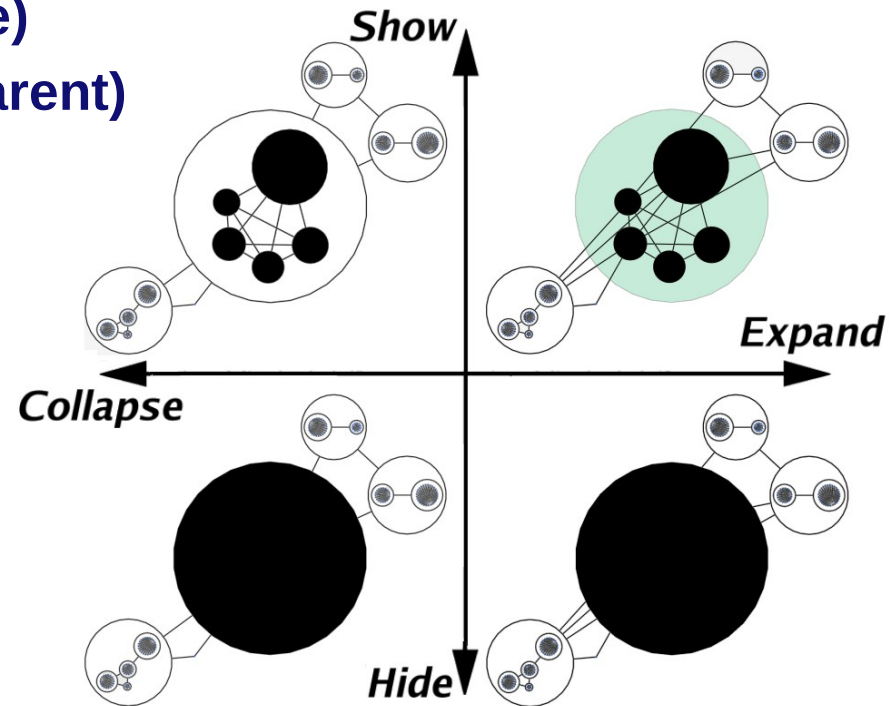
Modifying the *LOD*

- In previous work: *Expand/Collapse* Operation
- We consider 2 Operations:
 - *Expand/Collapse*
 - *Show/Hide* (opaque/transparent)

Modifying the *LOD*

A metanode can have four states

- *Collapsed and Hidden* (opaque)
- *Collapsed and Shown* (transparent)
- *Expanded and Hidden*
- *Expanded and Shown*



Simple and Thorough Interaction

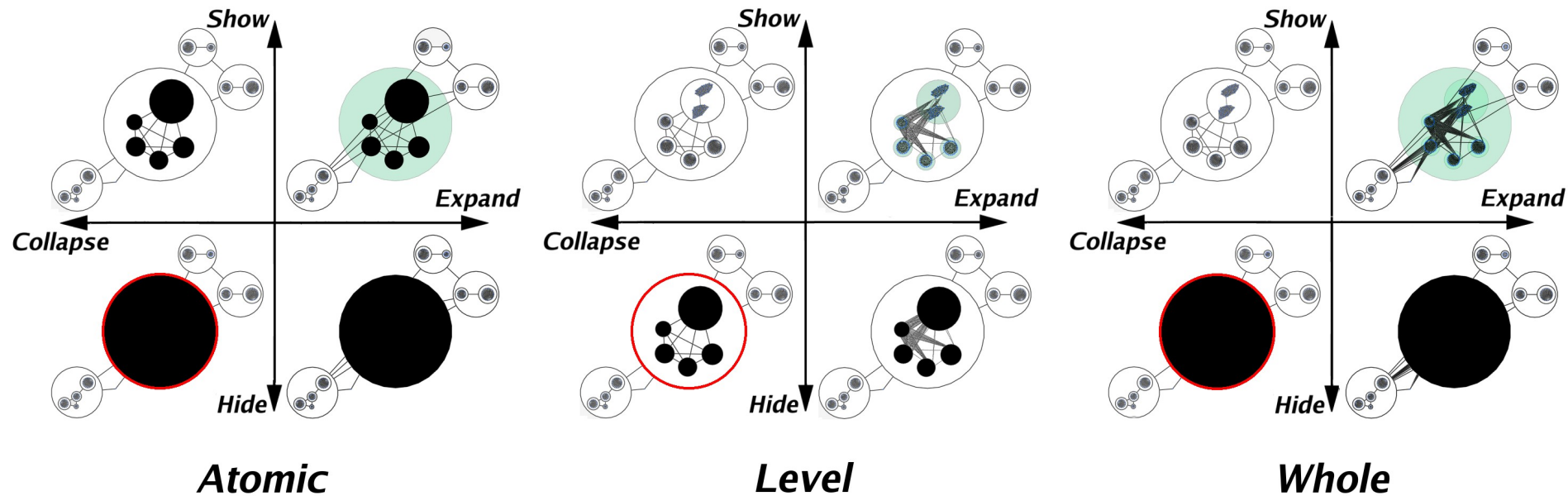
Defined by three facets:

- **Operation:** *Expand/Collapse* and *Show/Hide*
- **Mode:** *More Detail/Less Detail*
- **Scope:** **Set of (meta)nodes on which the operation is applied**

Defining the different scopes according to the user interest

- ***Atomic:*** user interested in the clicked metanode
- ***Level:*** user interested in the sub-hierarchy rooted on the clicked metanode
- ***Whole:*** user interested in the metanodes of the sub-hierarchy rooted on the “clicked” metanode

Simple and Thorough Interaction: Scope facet



Different scopes of the Scope facet

Simple and Thorough Interaction

- **3 Facets: *Operation, Mode, Scope***
- ***Few distinct values per facet***

Simple and Thorough Interaction

- **3 Facets: *Operation, Mode, Scope***
- ***Few distinct values per facet***

⇒ ***Minimal User Interface:***

- ***2 buttons ~ Expand/Collapse and Show/Hide***
- ***Left/Right clicks ~ More Detail/Less Detail***
- ***1 slider ~ Atomic/Level/Whole***

Simple and Thorough Interaction

- **3 Facets: *Operation, Mode, Scope***
- ***Few distinct values per facet***

⇒ ***Minimal User Interface:***

- **3 buttons ~ *Expand/Collapse* and *Show/Hide* and **Selection****
- **Left/Right clicks ~ *More Detail/Less Detail***
- **1 slider ~ *Atomic/Level/Whole***

Outline

- I. Introduction
- II. Multiscale force directed algorithms
- III. Compound Graph Visualization
- IV. Fitting the Level Of Detail with the user interest**
- V. Conclusion and Future Work

Outline

- I. Introduction
- II. Multiscale force directed algorithms
- III. Compound Graph Visualization
- IV. Fitting the Level Of Detail with the user interest
- V. Conclusion and Future Work**

Conclusion

- **New compound graph visualization combining both classical and multilevel visualizations**
- **Simple and thorough interaction system allowing to modify freely the *LOD***
- ***ViiFit Software***
 - *Multiple clustering algorithms*
 - *Multiple views*
 - *Linking and Brushing*
 - *...*