GraphMaps
Browsing Large Graphs as Interactive Maps

Lev Nachmanson¹, Roman Prutkin², Bongshin Lee¹, Nathalie Henry Riche¹, Alexander E. Holroyd¹, and Xiaoji Chen³

September 23, 2015

¹ Microsoft Research, Redmond ² Karlsruhe Institute of Technology ³ Microsoft, Redmond
Why consider exploration of large graphs?

- readable drawings of large graphs still a challenge
- good and fast vertex placement algorithms are known
- displaying readable connectivity information is challenging

View of a graph *b100*: files in a C++ project.
Related approaches: aggregation and multiscale techniques

- **partition** and group nodes/edges, draw resulting smaller graph
- **drawing**: use hierarchy; coarsen, lay out, uncoarsen
- **browsing**: display a "graph of graphs"

Example from [Eades and Feng, 1997]
Related approaches: aggregation and multiscale techniques

- **partition** and group nodes/edges, draw resulting smaller graph
- **drawing**: use hierarchy; coarsen, lay out, uncoarsen
- **browsing**: display a "graph of graphs"

Example from [Eades and Feng, 1997]
Related approaches: **aggregation** and **multiscale** techniques

- **partition** and group nodes/edges, draw resulting smaller graph
- **drawing**: use hierarchy; coarsen, lay out, uncoarsen
- **browsing**: display a "graph of graphs"

example from [Eades and Feng, 1997]

example from [Auber et al., 2003]
Related approaches: aggregation and multiscale techniques

- **partition** and group nodes/edges, draw resulting smaller graph
- **drawing**: use hierarchy; coarsen, lay out, uncoarsen
- **browsing**: display a "graph of graphs"

example from [Eades and Feng, 1997]

example from [Auber et al., 2003]

[van den Elzen and van Wijk, 2014]
Related approaches: aggregation and multiscale techniques

**scale** up to millions of nodes/edges; **drawbacks:**
- misleading information on inter- and intra-connectivity
- spatial instability confuses the user

Example from [Eades and Feng, 1997]

Example from [Auber et al., 2003]

Example from [van den Elzen and van Wijk, 2014]
Related approaches: filtering techniques

- rank nodes/edges by importance
- show important nodes first

A complete and filtered view of a network. [SocialAction. Perer and Shneiderman]
Related approaches: political map metaphor

Gmap [Gansner, Hu, Kobourov 2010]. Focus on showing clusters: as countries with map-like borders and coloring; graph on top

collaboration graph, GD 1994-2004
Related approaches: political map metaphor

**Gmap** [Gansner, Hu, Kobourov 2010]. Focus on showing clusters: as countries with map-like borders and coloring; graph on top

collaboration graph, GD 1994-2004
Our inspiration: online street maps

- showing everything at all times unnecessary
- high-level graph structure should be visible
  - show important nodes and edges between them
- ability to zoom in: nodes/edges appear gradually with increasing zoom level
Our inspiration: **online street maps**

- showing everything at all times unnecessary
- high-level graph structure should be visible
  - show important nodes and edges between them
- ability to zoom in: nodes/edges appear gradually with increasing zoom level
Our inspiration: **online street maps**

- showing everything at all times unnecessary
- high-level graph structure should be visible
  - show important nodes and edges between them
- ability to zoom in: nodes/edges appear gradually with increasing zoom level

---

**Bing maps**
Our inspiration: **online street maps**

- showing everything at all times unnecessary
- high-level graph structure should be visible
  - show important nodes and edges between them
- ability to zoom in: nodes/edges appear gradually with increasing zoom level

Bing maps
Our inspiration: online street maps

- help navigate by highlighting edges
Design goals

1. No restriction on input graphs
2. Reveal key structure through zoom in/out and pan

vertices and edges of level 1
Design goals

① no restriction on input graphs
② reveal key structure through **zoom in/out** and **pan**
Design goals

1. no restriction on input graphs
2. reveal key structure through **zoom in/out** and **pan**
3. preserve **mental map** of the user
   - **edge trajectories** don’t change on zoom
Design goals

1. No restriction on input graphs
2. Reveal key structure through zoom in/out and pan
3. Preserve mental map of the user
   Edge trajectories don’t change on zoom
Design goals

1. no restriction on input graphs
2. reveal key structure through zoom in/out and pan
3. preserve mental map of the user
4. limit clutter and complexity
   bound the number of displayed geometric objects

current view

showing:
\[ \leq c_1 \] vertices
\[ \leq c_2 \] segments
Design decisions: what to do with the edges?

1. no restriction on input graphs
2. overview through zoom and pan
3. preserve mental map of the user
4. limit clutter and complexity

Decision: use edge bundling using polygonal mesh

Problems: low-level connectivity lost inside bundles

no guarantees (unlike, e.g., for confluent drawings)

edge between A, B

no edge between C, D

some edges between \{A, B\} and \{C, D\}

should suffice for an overview

for more details: highlight incident edges
Design decisions: what to do with the edges?

1. no restriction on input graphs
2. overview through zoom and pan
3. preserve mental map of the user
4. limit clutter and complexity

Decision: use edge bundling using polygonal mesh

Problems: low-level connectivity lost inside bundles
no guarantees (unlike, e.g., for confluent drawings)

should suffice for an overview
for more details: highlight incident edges
Design decisions: what to do with the **edges**?

1. no restriction on input graphs  
2. overview through **zoom** and **pan**  
3. preserve **mental map** of the user  
4. limit **clutter** and **complexity**

**Decision**: use **edge bundling** using polygonal mesh  
**Problems**: low-level connectivity lost inside bundles  
no guarantees (unlike, e.g., for **confluent** drawings)

should suffice for an overview  
for more details: highlight incident edges
Design decisions: what to do with the edges?

1. no restriction on input graphs  2. overview through zoom and pan
3. preserve mental map of the user  4. limit clutter and complexity

Decision: use edge bundling using polygonal mesh

Problems: low-level connectivity lost inside bundles
no guarantees (unlike, e.g., for confluent drawings)

compute set of rails
assignment: edge $e \mapsto$ set of rails forming $e$

should suffice for an overview
for more details: highlight incident edges
Design decisions: what to do with the edges?

1. no restriction on input graphs
2. overview through zoom and pan
3. preserve mental map of the user
4. limit clutter and complexity

Decision: use edge bundling using polygonal mesh

Problems: low-level connectivity lost inside bundles
no guarantees (unlike, e.g., for confluent drawings)

should suffice for an overview
for more details: highlight incident edges

compute set of rails assignment: edge $e \mapsto$ set of rails forming $e$
Design decisions: what to do with the edges?

1. no restriction on input graphs  
2. overview through zoom and pan  
3. preserve mental map of the user  
4. limit clutter and complexity

**Decision:** use edge bundling using polygonal mesh

**Problems:** low-level connectivity lost inside bundles  
no guarantees (unlike, e.g., for confluent drawings)

Compute set of rails  
assignment: edge $e \mapsto$ set of rails forming $e$

should suffice for an overview  
for more details: highlight incident edges
Design decisions: what to do with the edges?

1. no restriction on input graphs  
2. overview through zoom and pan  
3. preserve mental map of the user  
4. limit clutter and complexity

Decision: use edge bundling using polygonal mesh

Problems: low-level connectivity lost inside bundles

no guarantees (unlike, e.g., for confluent drawings)

should suffice for an overview

for more details: highlight incident edges

compute set of rails

assignment: edge $e \mapsto$ set of rails forming $e$
Design decisions: what to do with the edges?

1. no restriction on input graphs  
2. overview through zoom and pan  
3. preserve mental map of the user  
4. limit clutter and complexity

Decision: use edge bundling using polygonal mesh

Problems: low-level connectivity lost inside bundles  
no guarantees (unlike, e.g., for confluent drawings)

should suffice for an overview  
for more details: highlight incident edges

compute set of rails  
assignment: edge $e \mapsto$ set of rails forming $e$
Design decisions: what to do with the edges?

1. no restriction on input graphs  
2. overview through zoom and pan  
3. preserve mental map of the user  
4. limit clutter and complexity

Decision: use edge bundling using polygonal mesh

Problems: low-level connectivity lost inside bundles
no guarantees (unlike, e.g., for confluent drawings)

should suffice for an overview
for more details: highlight incident edges
Design decisions: choosing visible level of detail?

1. no restriction on input graphs
2. overview through **zoom** and **pan**
3. preserve mental map of the user
4. limit clutter and complexity

**Node layout**: MDS, force-directed, . . . , user input

**Node ranking**: sorted by decreasing importance using betweenness centrality, PageRank, degree, . . . , user input

\[
\text{Sorted} V = (\text{Seattle}, \text{Tacoma}, \text{Redmond}, \ldots)
\]

assign **zoom levels**:

\[ v \in V \mapsto z(v) \in \{1.0, 2.0, 4.0, 8.0, \ldots\} \]

show \( v \) iff \( z(v) \leq \text{current zoom} \)

**motivation**: on maps, never show Redmond without Seattle if both inside view
Demo

- *Composers* graph
- largest connected component: 2743 nodes, 13769 edges
- vertices: Wikipedia articles about composers
- edges: links between these articles
- benchmark graph from the **Graph Drawing 2011** contest
Algorithm overview: **zoom level** assignment

1. No restriction on input graphs  
2. Overview through **zoom** and **pan**  
3. Preserve **mental map** of the user  
4. Limit **clutter** and **complexity**

**Initial node layout**

**Node ranking**
Algorithm overview: zoom level assignment

1. No restriction on input graphs
2. Overview through zoom and pan
3. Preserve mental map of the user
4. Limit clutter and complexity

Initial node layout

Node ranking

Pick candidate nodes for current level
Algorithm overview: zoom level assignment

1. no restriction on input graphs  
2. overview through zoom and pan 
3. preserve mental map of the user  
4. limit clutter and complexity

initial node layout  →  pick candidate nodes for current level  →  generate mesh for edge routing

node ranking
Algorithm overview: zoom level assignment

1. no restriction on input graphs
2. overview through zoom and pan
3. preserve mental map of the user
4. limit clutter and complexity

Initial node layout → pick candidate nodes for current level → generate mesh for edge routing → try adding next node+edges

Node ranking
Algorithm overview: **zoom level** assignment

1. no restriction on input graphs  
2. overview through **zoom** and **pan**  
3. preserve **mental map** of the user  
4. limit **clutter** and **complexity**

- initial node layout
- node ranking
- pick **candidate nodes** for current **level**
- generate mesh for edge routing
- try adding **next node+edges**
- cell quotas exceeded?
  - no
  - confirm
Algorithm overview: zoom level assignment

1. no restriction on input graphs  
2. overview through zoom and pan  
3. preserve mental map of the user  
4. limit clutter and complexity

initial node layout

pick candidate nodes for current level

generate mesh for edge routing

try adding next node+edges

cell quotas exceeded?

yes → subdivide cells go to next level

no → confirm

node ranking
Algorithm overview: zoom level assignment

1. no restriction on input graphs
2. overview through zoom and pan
3. preserve mental map of the user
4. limit clutter and complexity

initial node layout

pick candidate nodes for current level

generate mesh for edge routing

node ranking

maintain edge trajectories on zoom

R. Prutkin: Browsing large graphs as interactive maps
Algorithm overview: zoom level assignment

1. no restriction on input graphs
2. overview through zoom and pan
3. preserve mental map of the user
4. limit clutter and complexity

- initial node layout
- node ranking
- pick candidate nodes for current level
- generate mesh for edge routing
- maintain edge trajectories on zoom
Method description: **zoom level** assignment

1. no restriction on input graphs
2. overview through **zoom** and **pan**
3. preserve **mental map** of the user
4. limit **clutter** and **complexity**

maintain **node quota** $Q_N$ and **rail quota** $Q_R$ in each **cell**

graph with no edges (for presentation)

$Q_N = 20$;
Method description: **zoom level** assignment

1. no restriction on input graphs    2. overview through **zoom** and **pan**
3. preserve **mental map** of the user    4. limit **clutter** and **complexity**

Maintain **node quota** $Q_N$ and **rail quota** $Q_R$ in each **cell**

Graph with no edges (for presentation)

$Q_N = 20$; filling **level** 1.0

Set $z(v) = 1.0$
Method description: zoom level assignment

1. no restriction on input graphs  
2. overview through zoom and pan  
3. preserve mental map of the user  
4. limit clutter and complexity

Maintain node quota $Q_N$ and rail quota $Q_R$ in each cell.

Graph with no edges (for presentation)

$Q_N = 20$; filling level 1.0

Set $z(v) = 1.0$
Method description: zoom level assignment

① no restriction on input graphs  ② overview through zoom and pan  ③ preserve mental map of the user  ④ limit clutter and complexity

maintain node quota $Q_N$ and rail quota $Q_R$ in each cell

graph with no edges (for presentation)

$Q_N = 20$; filling level $1.0$

set $z(v) = 1.0$
Method description: **zoom level** assignment

1. no restriction on input graphs  
2. overview through **zoom** and **pan**  
3. preserve **mental map** of the user  
4. limit **clutter** and **complexity**

Maintain **node quota** $Q_N$ and **rail quota** $Q_R$ in each **cell**

Graph with no edges (for presentation)

$Q_N = 20$; **level** 1.0 full
Method description: zoom level assignment

1. no restriction on input graphs  
2. overview through zoom and pan  
3. preserve mental map of the user  
4. limit clutter and complexity  

maintain node quota $Q_N$ and rail quota $Q_R$ in each cell

graph with no edges (for presentation)

$Q_N = 20$; level 1.0 full
Method description: zoom level assignment

1. no restriction on input graphs  
2. overview through zoom and pan  
3. preserve mental map of the user  
4. limit clutter and complexity

Maintain node quota $Q_N$ and rail quota $Q_R$ in each cell

Graph with no edges (for presentation)

$Q_N = 20$; level 1.0 full subdivide cells
Method description: **zoom level** assignment

1. no restriction on input graphs  
2. overview through **zoom** and **pan**  
3. preserve **mental map** of the user  
4. limit **clutter** and **complexity**

maintain **node quota** $Q_N$ and **rail quota** $Q_R$ in each **cell**

Graph with no edges (for presentation)

$Q_N = 20$; filling **level** 2.0

Set $z(v) = 2.0$
Method description: zoom level assignment

1. no restriction on input graphs  
2. overview through zoom and pan  
3. preserve mental map of the user  
4. limit clutter and complexity

Maintain node quota $Q_N$ and rail quota $Q_R$ in each cell.

Graph with no edges (for presentation)

$Q_N = 20$; filling level 2.0

Set $z(v) = 2.0$
Method description: zoom level assignment

1. no restriction on input graphs
2. overview through zoom and pan
3. preserve mental map of the user
4. limit clutter and complexity

Maintain node quota $Q_N$ and rail quota $Q_R$ in each cell

Graph with no edges (for presentation)

$Q_N = 20$; filling level 2.0

Set $z(v) = 2.0$ for all nodes added.
Method description: zoom level assignment

1. no restriction on input graphs
2. overview through zoom and pan
3. preserve mental map of the user
4. limit clutter and complexity

Maintain node quota $Q_N$ and rail quota $Q_R$ in each cell

graph with no edges (for presentation)

$Q_N = 20$; filling level 2.0
set $z(v) = 2.0$
all nodes added
Method description: zoom level selection

1. no restriction on input graphs  
2. overview through zoom and pan  
3. preserve mental map of the user  
4. limit clutter and complexity

maintain node quota \( Q_N \) and rail quota \( Q_R \) in each cell

graph with no edges (for presentation)

\[ Q_N = 20; \]

view rectangle \( R \) has zoom level

\[ z_R = \min\left\{ \frac{w_{\text{total}}}{w_R}, \frac{h_{\text{total}}}{h_R} \right\} \approx 1.3 \]

show \( v \) iff \( z(v) \leq z_R \)
Method description: **zoom level** assignment

1. no restriction on input graphs
2. overview through **zoom** and **pan**
3. preserve **mental map** of the user
4. limit **clutter** and **complexity**

Maintain **node quota** $Q_N$ and **rail quota** $Q_R$ in each cell

Graph with no edges (for presentation)

$Q_N = 20$;
view rectangle $R$
has **zoom level**

$z_R = \min\left\{ \frac{w_{total}}{w_R}, \frac{h_{total}}{h_R} \right\}$

$\approx 1.3$

Show $v$ iff $z(v) \leq z_R$

At most $4Q_N$ visible
Method description: **zoom level** assignment with **rails**

1. no restriction on input graphs  
2. overview through **zoom** and **pan**  
3. preserve **mental map** of the user  
4. limit **clutter** and **complexity**

Maintain **node quota** $Q_N$ and **rail quota** $Q_R$ in each **cell**

Graph with edges $Q_N = 20$, $Q_R = 45$ filling **level** 1.0
Method description: **zoom level** assignment with **rails**

1. No restriction on input graphs  
2. Overview through **zoom** and **pan**  
3. Preserve **mental map** of the user  
4. Limit **clutter** and **complexity**

Maintain **node quota** $Q_N$ and **rail quota** $Q_R$ in each **cell**

Graph with edges $Q_N = 20$, $Q_R = 45$

Filling level 1.0

Use as input for mesh generator **triangle** [J. R. Shewchuk]
Method description: zoom level assignment with rails

1. no restriction on input graphs
2. overview through zoom and pan
3. preserve mental map of the user
4. limit clutter and complexity

Maintain node quota $Q_N$ and rail quota $Q_R$ in each cell.

Graph with edges

$Q_N = 20, Q_R = 45$

Filling level 1.0

Use as input for mesh generator

Triangle [J. R. Shewchuk]
Method description: zoom level assignment with rails

1. no restriction on input graphs
2. overview through zoom and pan
3. preserve mental map of the user
4. limit clutter and complexity

Maintain node quota $Q_N$ and rail quota $Q_R$ in each cell graph with edges $Q_N = 20$, $Q_R = 45$ filling level 1.0

Use as input for mesh generator triangle [J. R. Shewchuk]
Method description: **zoom level** assignment with **rails**

1. no restriction on input graphs  
2. overview through **zoom** and **pan**  
3. preserve **mental map** of the user  
4. limit **clutter** and **complexity**  

**Maintain node quota** $Q_N$ and **rail quota** $Q_R$ in each **cell**

Graph with edges

$Q_N = 20, Q_R = 45$

filling **level** 1.0

use as input for

**mesh generator**

**triangle** [J. R. Shewchuk]
Method description: zoom level assignment with rails

1. no restriction on input graphs  
2. overview through zoom and pan  
3. preserve mental map of the user  
4. limit clutter and complexity

Maintain node quota $Q_N$ and rail quota $Q_R$ in each cell.

Graph with edges $Q_N = 20, Q_R = 45$ filling level 1.0

Use as input for mesh generator triangle [J. R. Shewchuk]
Method description: \textbf{zoom level} assignment with \textbf{rails}

1. no restriction on input graphs
2. overview through \textbf{zoom} and \textbf{pan}
3. preserve \textbf{mental map} of the user
4. limit \textbf{clutter} and \textbf{complexity}

Maintain \textbf{node quota} $Q_N$ and \textbf{rail quota} $Q_R$ in each \textbf{cell}

Graph with \textbf{edges}

$Q_N = 20$, $Q_R = 45$

filling \textbf{level} 1.0

Use as input for \textbf{mesh generator} \textbf{triangle} [J. R. Shewchuk]
Method description: **zoom level** assignment with **rails**

① no restriction on input graphs  ② overview through **zoom** and **pan**
③ preserve **mental map** of the user  ④ limit **clutter** and **complexity**

Maintain **node quota** $Q_N$ and **rail quota** $Q_R$ in each **cell**

Graph with edges $Q_N = 20$, $Q_R = 45$ filling **level** 1.0

Use as input for **mesh generator**

**triangle** [J. R. Shewchuk]
Method description: zoom level assignment with rails

1. no restriction on input graphs  
2. overview through zoom and pan  
3. preserve mental map of the user  
4. limit clutter and complexity

Maintain node quota $Q_N$ and rail quota $Q_R$ in each cell

Graph with edges

$Q_N = 20, \quad Q_R = 45$

Filling level 1.0

Added nodes 0…19

All edges in between

Per cell: \( \leq 20 \) nodes, \( \leq 45 \) rails
Method description: **zoom level** assignment with **rails**

1. no restriction on input graphs  
2. overview through **zoom** and **pan**  
3. preserve **mental map** of the user  
4. limit **clutter** and **complexity**  

Maintain **node quota** $Q_N$ and **rail quota** $Q_R$ in each **cell**

Graph with edges  
$Q_N = 20$, $Q_R = 45$  
filling **level** 1.0

Added nodes 0 . . . 19  
All edges in between

Subdivide cells per cell: ≤ **20 nodes**,  
≤ **45 rails**
Method description: **zoom level** assignment with **rails**

1. no restriction on input graphs
2. overview through **zoom** and **pan**
3. preserve **mental map** of the user
4. limit **clutter** and **complexity**

Maintain **node quota** $Q_N$ and **rail quota** $Q_R$ in each **cell**

Graph with edges
$Q_N = 20$, $Q_R = 45$
filling **level** 2.0
added nodes 20 … 46

Subdivide cells
per cell: $\leq 20$ nodes, $\leq 45$ rails
Method description: **zoom level** assignment with **rails**

1. no restriction on input graphs  
2. overview through **zoom** and **pan**  
3. preserve **mental map** of the user  
4. limit **clutter** and **complexity**

Maintain **node quota** $Q_N$ and **rail quota** $Q_R$ in each **cell**

- Graph with edges $Q_N = 20$, $Q_R = 45$
- Filling level 2.0
- Added nodes 20...46
- Use added nodes and old rails as input
Method description: **zoom level** assignment with **rails**

1. no restriction on input graphs  
2. overview through **zoom** and **pan**  
3. preserve **mental map** of the user  
4. limit **clutter** and **complexity**

 maintain **node quota** $Q_N$ and **rail quota** $Q_R$ in each **cell**

Graph with edges $Q_N = 20$, $Q_R = 45$
filling **level** 2.0

added nodes 20 ... 46

use **added nodes**
and **old rails** as input
Method description: **zoom level** assignment with **rails**

1. no restriction on input graphs  
2. overview through **zoom** and **pan**  
3. preserve **mental map** of the user  
4. limit **clutter** and **complexity**

**Maintain node quota** $Q_N$ and **rail quota** $Q_R$ in each **cell**

Graph with edges $Q_N = 20$, $Q_R = 45$

Filling level 2.0

Added nodes 0 . . . 36

All edges in between

Per cell: $\leq 20$ nodes, $\leq 45$ rails
Method description: zoom level assignment with rails

1. no restriction on input graphs  2. overview through zoom and pan
3. preserve mental map of the user  4. limit clutter and complexity

Maintain node quota $Q_N$ and rail quota $Q_R$ in each cell.

Graph with edges $Q_N = 20$, $Q_R = 45$

Filling level 2.0

Added nodes 0 ... 36

All edges in between

Per cell: $\leq 20$ nodes, $\leq 45$ rails
Method description: **zoom level** assignment with **rails**

1. no restriction on input graphs
2. overview through **zoom** and **pan**
3. preserve **mental map** of the user
4. limit **clutter** and **complexity**

Maintain **node quota** $Q_N$ and **rail quota** $Q_R$ in each **cell**

Graph with edges $Q_N = 20$, $Q_R = 45$

Filling level 2.0

Added nodes 0 . . . 36
All edges in between

Per cell: $\leq 20$ nodes, $\leq 45$ rails
Method description: **zoom level** assignment with **rails**

1. No restriction on input graphs  
2. Overview through **zoom** and **pan**  
3. Preserve **mental map** of the user  
4. Limit **clutter** and **complexity**  

Maintain **node quota** $Q_N$ and **rail quota** $Q_R$ in each **cell**

Graph with edges

$Q_N = 20$, $Q_R = 45$

Filling level 4.0

Added nodes 0...36

All edges in between

Subdivide cells

Per cell: $\leq 20$ nodes, $\leq 45$ rails
Method description: **zoom level** assignment with **rails**

1. **no restriction on input graphs**  
2. **overview through zoom and pan**  
3. **preserve mental map** of the user  
4. **limit clutter and complexity**

**maintain node quota** $Q_N$ and **rail quota** $Q_R$ in each cell

Graph with edges  
$Q_N = 20, Q_R = 45$  
filling **level** 4.0  
added nodes 0 \ldots 46

**subdivide cells**  
per cell: $\leq 20$ nodes, $\leq 45$ rails
Method description: **zoom level** assignment with **rails**

1. no restriction on input graphs  
2. overview through **zoom** and **pan**  
3. preserve **mental map** of the user  
4. limit **clutter** and **complexity**

Maintain **node quota** $Q_N$ and **rail quota** $Q_R$ in each **cell**

Graph with edges $Q_N = 20, Q_R = 45$

Filling **level** 4.0

Added nodes 0 ... 46

Use **added nodes** and **old rails** as input
Method description: **zoom level** assignment with **rails**

1. no restriction on input graphs  
2. overview through **zoom** and **pan**  
3. preserve **mental map** of the user  
4. limit **clutter** and **complexity**

Maintain **node quota** $Q_N$ and **rail quota** $Q_R$ in each **cell**

Graph with edges $Q_N = 20, Q_R = 45$

Filling **level** 4.0

Added nodes $0 \ldots 46$

Use added nodes and **old rails** as input
Method description: **zoom level** assignment with rails

1. no restriction on input graphs  
2. overview through **zoom** and **pan**  
3. preserve **mental map** of the user  
4. limit **clutter** and **complexity**

maintain **node quota** $Q_N$ and **rail quota** $Q_R$ in each cell

graph with edges

$Q_N = 20$, $Q_R = 45$

filling **level** 4.0

added nodes 0 ... 46

all edges in between

per cell: $\leq 20$ nodes, $\leq 45$ rails
Method description: zoom level assignment with rails

1. no restriction on input graphs
2. overview through zoom and pan
3. preserve mental map of the user
4. limit clutter and complexity

Maintain node quota $Q_N$ and rail quota $Q_R$ in each cell:

Graph with edges $Q_N = 20$, $Q_R = 45$

Filling level 4.0

Added nodes 0...46

All edges in between

Per cell: \( \leq 20 \) nodes, \( \leq 45 \) rails
More details

remove overlaps
More details

remove overlaps
More details

remove overlaps
More details

remove overlaps

pre-rendered bitmap tiles for orientation
More details

remove overlaps

pre-rendered bitmap tiles for orientation

node labeling
More details

- remove overlaps
- pre-rendered bitmap tiles for orientation
- node labeling
- interactions
More details: see paper

remove overlaps

pre-rendered bitmap tiles for orientation

node labeling

interactions
Conclusion

- introduced new approach of browsing large graphs
- provide a familiar browsing experience
- preserve mental map
- bounds on number of visible objects
- scalability, reduced clutter

part of MSAGL, available at github.com/Microsoft/automatic-graph-layout
Conclusion

- introduced new approach of browsing large graphs
- provide a familiar browsing experience
- preserve mental map
- bounds on number of visible objects
- scalability, reduced clutter

Future work

- find suitable initial node layout
- replace mesh generator
- prove guarantees on feature size and number of levels
- improve node labeling
- user studies

part of MSAGL, available at github.com/Microsoft/automatic-graph-layout
Conclusion

- introduced new approach of browsing large graphs
- provide a familiar browsing experience
- preserve mental map
- bounds on number of visible objects
- scalability, reduced clutter

Future work

- find suitable initial node layout
- replace mesh generator
- prove guarantees on feature size and number of levels
- improve node labeling
- user studies

part of MSAGL, available at github.com/Microsoft/automatic-graph-layout

Future work

- find suitable initial node layout
- replace mesh generator
- prove guarantees on feature size and number of levels
- improve node labeling
- user studies

Future work

- find suitable initial node layout
- replace mesh generator
- prove guarantees on feature size and number of levels
- improve node labeling
- user studies

Future work

- find suitable initial node layout
- replace mesh generator
- prove guarantees on feature size and number of levels
- improve node labeling
- user studies

Future work

- find suitable initial node layout
- replace mesh generator
- prove guarantees on feature size and number of levels
- improve node labeling
- user studies

Future work

- find suitable initial node layout
- replace mesh generator
- prove guarantees on feature size and number of levels
- improve node labeling
- user studies

Future work

- find suitable initial node layout
- replace mesh generator
- prove guarantees on feature size and number of levels
- improve node labeling
- user studies

Future work

- find suitable initial node layout
- replace mesh generator
- prove guarantees on feature size and number of levels
- improve node labeling
- user studies

Future work

- find suitable initial node layout
- replace mesh generator
- prove guarantees on feature size and number of levels
- improve node labeling
- user studies

Future work

- find suitable initial node layout
- replace mesh generator
- prove guarantees on feature size and number of levels
- improve node labeling
- user studies

Future work

- find suitable initial node layout
- replace mesh generator
- prove guarantees on feature size and number of levels
- improve node labeling
- user studies

Future work

- find suitable initial node layout
- replace mesh generator
- prove guarantees on feature size and number of levels
- improve node labeling
- user studies

Future work

- find suitable initial node layout
- replace mesh generator
- prove guarantees on feature size and number of levels
- improve node labeling
- user studies

Future work

- find suitable initial node layout
- replace mesh generator
- prove guarantees on feature size and number of levels
- improve node labeling
- user studies

Future work

- find suitable initial node layout
- replace mesh generator
- prove guarantees on feature size and number of levels
- improve node labeling
- user studies