Aalto University
School of Science

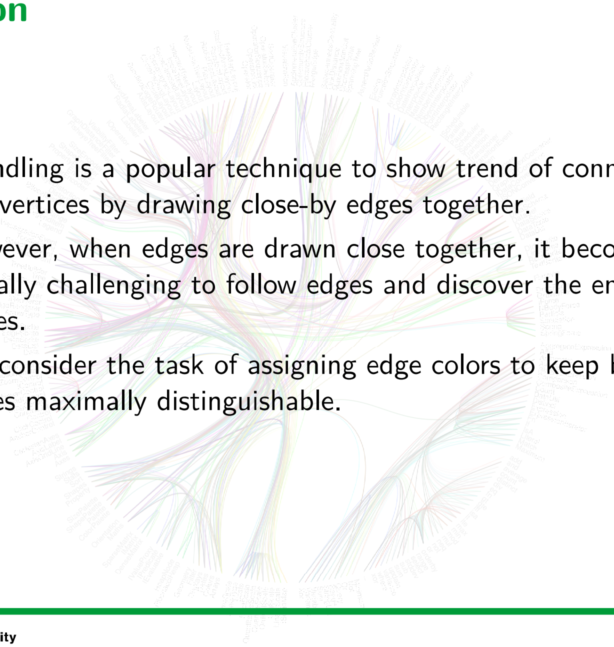
Peacock Bundles: Bundle Coloring for Graphs with Globality-Locality Trade-off

Jaakko Peltonen[1,2], Ziyuan Lin[1,2]

[1]Department of Computer Science, Aalto University, Finland
[2]School of Information Sciences, University of Tampere, Finland

September 19, 2016

Motivation



Edge bundling is a popular technique to show trend of connections between vertices by drawing close-by edges together.

- However, when edges are drawn close together, it becomes visually challenging to follow edges and discover the end nodes.
- We consider the task of assigning edge colors to keep bundled edges maximally distinguishable.

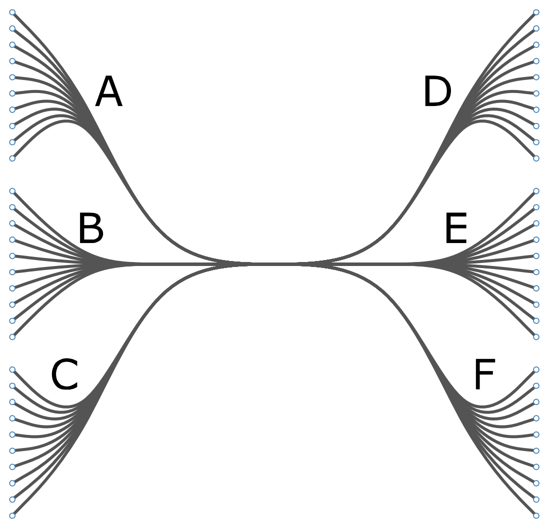
Paper summary

Given a graph with edge bundling, we optimize an edge coloring to differentiate the bundled edges, improving the ability to follow the edges:

- We start from a predefined graph layout with edge bundling;
- We use a flexible pairwise bundling definition;
- We formulate the color optimization as nonlinear dimensionality reduction;
- We also allow a user-set global-local tradeoff;
- We name the method “Peacock Bundles”, inspired by the plumage of a peacock.

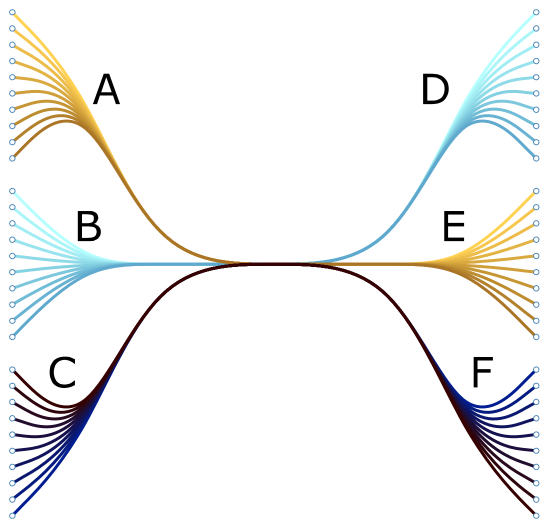
Plumage!

Conceptual demonstration



Plain gray coloring. We are not able to see how the vertices in groups A/B/C connect to the vertices in groups D/E/F.

Conceptual demonstration



It is revealed that nodes in group A connect to nodes in group E in order, also B to D in order, but C to F in reverse order.

Presentation Outline

1. flexible model of bundles by pairwise bundling detection
2. quantifying edge dissimilarity from origins and destinations
3. color optimization as dimensionality reduction
4. local color range normalization
5. local to global differentiation
6. experiment results

Peacock bundles: pairwise bundling detection

Why pairwise bundling detection instead of some global partitioning of edges into subsets that form bundles?

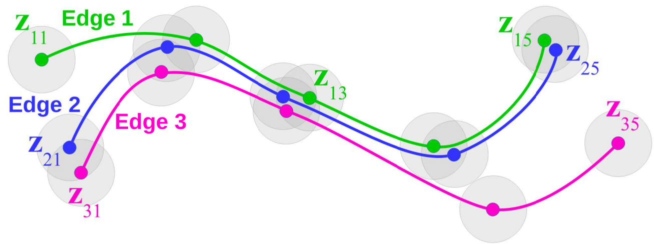
- Bundles are typically not clearly defined: an edge may belong to a bundle at some part of its curve and join another bundle in another part of its curve.

Let each curve i be generated by C_i “control points”, which can be

- control points on a spline;
- midpoints of piecewise linear curves;

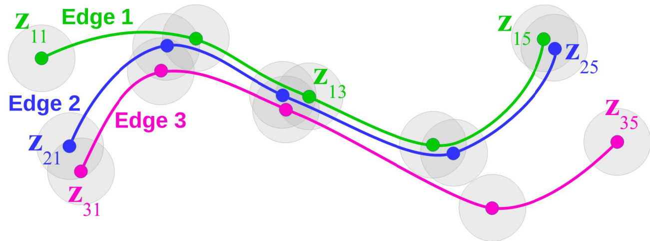
We will define a variable B_{ij} in $[0, 1]$ denoting whether edge i is bundled together with edge j.

Peacock bundles: pairwise bundling detection (local)



In plain language: two edges are considered as bundled ($B_{ij} = 1$) if, there exist two "long enough" segments on these two edges, that are "close enough" to each other within the segments.

Peacock bundles: pairwise bundling detection (local)



$$B_{ij} = \max_{r_0=1,\ldots,C_i-K_{ij}+1} \prod_{r=r_0}^{r_0+K_{ij}-1} 1\left(\min_{s=1,\ldots,C_j} d(\mathbf{z}_{ir}, \mathbf{z}_{js}) \leq T\right)$$

(equation explained in the next slide)

Peacock bundles: pairwise bundling detection

$$B_{ij} = \max_{r_0=1,\dots,C_i-K_{ij}+1} \prod_{r=r_0}^{r_0+K_{ij}-1} 1\left(\min_{s=1,\dots,C_j} d(\mathbf{z}_{ir}, \mathbf{z}_{js}) \leq T\right)$$

where

r : indices of consecutive control points in edge i

$1(\cdot)$: 1 if the statement inside is true and 0 otherwise

$\mathbf{z}_{i1}, \dots, \mathbf{z}_{iC_i}$: on-screen coordinates of the C_i control points

$d(\cdot,\cdot)$: the Euclidean distance between two control points

T : a distance threshold

Set $K_{ij} = \max(1, \lfloor \max(C_i, C_j) K_{min} \rfloor)$, where $K_{min} \in (0, 1]$ is a desired fraction. In experiments we choose $K_{min} = 0.4$.

Presentation Outline

1. ✓flexible model of bundles by pairwise bundling detection
2. quantifying edge dissimilarity from origins and destinations
3. color optimization as dimensionality reduction
4. local color range normalization
5. local to global differentiation
6. experiment results

Peacock bundles: quantifying edge dissimilarity

Motivation: there are too many bundled edges to separate their colors all equally much.
Idea:

1. if two bundled edges start and end close-by in the layout, confusing them is not so harmful (at least if the layout is meaningful);
2. bundled edges with distant end-nodes are harmful to confuse;
3. make color separation correspond to end-node difference.

Peacock bundles: quantifying edge dissimilarity

Denote the two on-screen node layout coordinates of edge i by $\mathbf{v}_i^1$ and $\mathbf{v}_i^2$. We define

$$d_{ij}^{\text{original}} = \min(\|\mathbf{v}_i^1 - \mathbf{v}_j^1\| + \|\mathbf{v}_i^2 - \mathbf{v}_j^2\|,\ \|\mathbf{v}_i^1 - \mathbf{v}_j^2\| + \|\mathbf{v}_i^2 - \mathbf{v}_j^1\|)$$

Peacock bundles: color separation as dimensionality reduction

Dimensionality reduction task: optimize RGB colors so that for each edge pair, their dissimilarity corresponds to end-node dissimilarity;
Only bundled edges ($B_{ij} = 1$) need color separation;
Denote the low-dimensional output features for edge i as a vector $\mathbf{y}_i = [y_{i1}, \ldots, y_{iq}]$ where $q \in \{1, 2, 3\}$ is the output dimensionality.
We optimize

$$\min_{\{\mathbf{y}_1, \ldots, \mathbf{y}_M\}} \sum_i \sum_j B_{ij} (d_{ij}^{original} - d^{out}(\mathbf{y}_i, \mathbf{y}_j))^2$$

to obtain the to-be-normalized output features $\{\mathbf{y}_i\}_{i=1}^{M}$ for the M edges.

Presentation Outline

1. ✓flexible model of bundles by pairwise bundling detection
2. ✓quantifying edge dissimilarity from origins and destinations
3. ✓color optimization as dimensionality reduction
4. local color range normalization
5. local to global differentiation
6. experiment results

Peacock bundles: local color range normalization

Dimensionality reduction does not guarantee maximal use of color range. We create a local, within-bundle normalization, which uses colors more efficiently than global normalization.
For each edge i, take the color coordinate matrix Y^i containing the colors of i and its bundled edges, and then affinely transform Y^i to the maximal color range, and store the color for edge i only.
Repeat for other edges.

Peacock bundles: from local to global color differentiation

$$\min_{\{\mathbf{y}_1,\ldots,\mathbf{y}_M\}} \sum_i \sum_j B_{ij}(d_{ij}^{original} - d^{out}(\mathbf{y}_i, \mathbf{y}_j))^2$$

B_{ij} was defined as binary: $B_{ij} = 1$ if edge i and edge j are bundled, and 0 otherwise.

Modify it to: $B_{ij} = 1$ if edge i and edge j are bundled, and ϵ otherwise, where $\epsilon > 0$ is a user-set parameter for the global-local tradeoff (in experiments we choose $\epsilon = 0.001$).

- $\epsilon = 0$: pure local
- $\epsilon = 1$: pure global

Presentation Outline

1. ✓flexible model of bundles by pairwise bundling detection
2. ✓quantifying edge dissimilarity from origins and destinations
3. ✓color optimization as dimensionality reduction
4. ✓local color range normalization
5. ✓local to global differentiation
6. experiment results

Experiments: the “Radial” graph

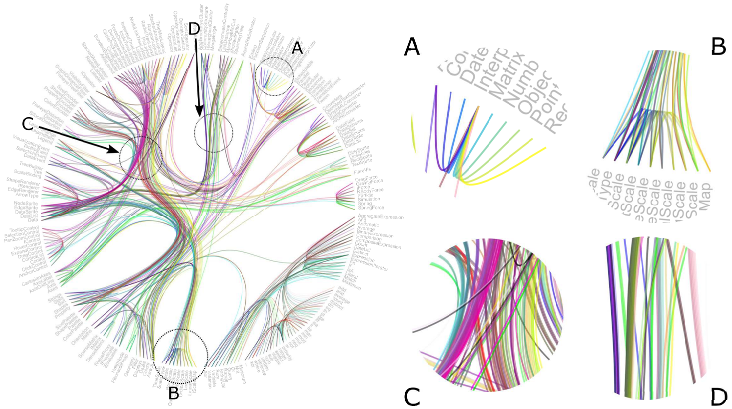


Figure: A graph of software package class hierarchy with hierarchical edge bundling. Edges are locally differentiated and the colors show a linear gradient spanning a full color range (e.g., yellow-red-blue).

Experiments: the “Treemap” graph

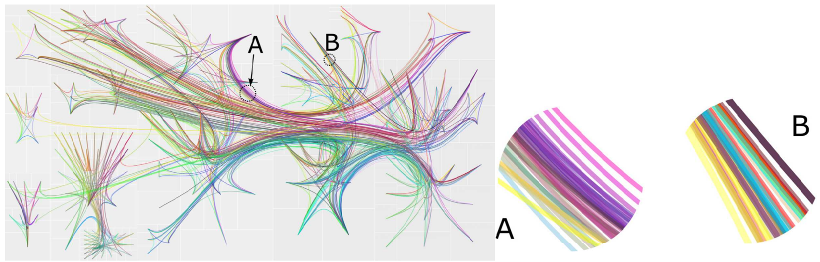


Figure: Same data but with hierarchical edge bundling for treemap. Besides the zoomed-out parts, full-color-range linear gradients can also be seen in the “claws” parts.

Experiments: the “Jane Austen” graph

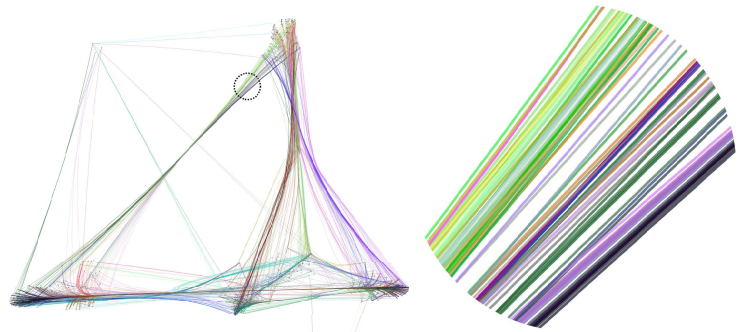


Figure: A graph of consecutive word-to-word appearances in novels of Jane Austen with force-directed edge bundling. Besides the zoomed-out part, the right half of the graph spans a red-green-blue color range since they are considered bundled at the top right part.

Experiments: the "Football" graph

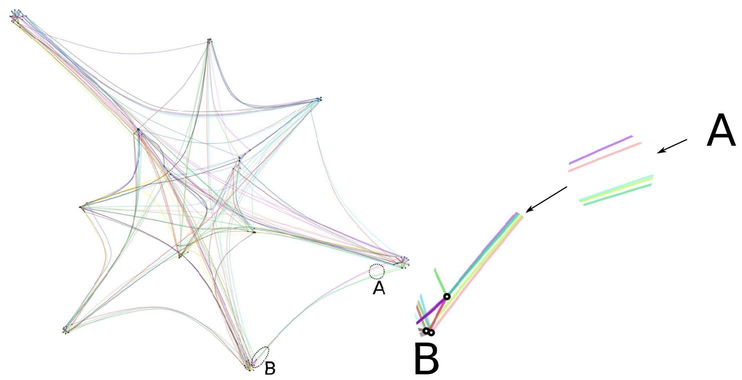


Figure: A graph of matches between US college football teams with force-directed edge bundling.

Experiments: the global side ($\epsilon = 1$)

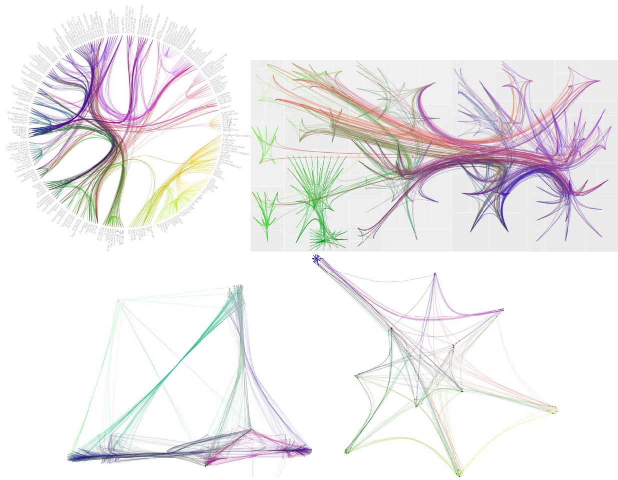


Figure: Bundles are more differentiated.

Summary

We introduced “peacock bundles”, an edge coloring algorithm for graphs with edge bundling. Colors are optimized both to preserve differences between bundle locations, and differentiate edges within bundles. The algorithm is based on dimensionality reduction and works with a flexible, local detection of pairwise bundles instead of requiring a predefined partitioning of edges into bundles.