Faster Force-Directed Graph Drawing with the Well-Separated Pair Decomposition

Fabian Lipp    Alexander Wolff    Johannes Zink

Julius-Maximilians-Universität Würzburg

September 24, Graph Drawing 2015
Classical force-directed algorithm

- [Eades, 1984, Fruchterman & Reingold, 1991]
- Attracting forces: $O(m)$ time with $m = \#\text{edges}$
- Repulsive forces: $\Theta(n^2)$ time with $n = \#\text{vertices}$
## Classical force-directed algorithm

- [Eades, 1984, Fruchterman & Reingold, 1991]
- **Attracting forces:** $O(m)$ time with $m = \#\text{edges}$
- **Repulsive forces:** $\Theta(n^2)$ time with $n = \#\text{vertices}$

Many speed-up techniques for force-directed algorithms known:

- Hierarchical $O(n \log n)$ force-calculation [Barnes & Hut, 1986]
- Potential-field-based multilevel algorithm [Hachul & Jünger, 2005]
- Evaluation of multilevel layout methods [Bartel et al., 2011]
Introduction

Classical force-directed algorithm

- [Eades, 1984, Fruchterman & Reingold, 1991]
- Attracting forces: $O(m)$ time with $m = \#\text{edges}$
- Repulsive forces: $\Theta(n^2)$ time with $n = \#\text{vertices}$

Many speed-up techniques for force-directed algorithms known:
- Hierarchical $O(n \log n)$ force-calculation [Barnes & Hut, 1986]
- Potential-field-based multilevel algorithm [Hachul & Jünger, 2005]
- Evaluation of multilevel layout methods [Bartel et al., 2011]

Our approach

- Using well-separated pair decomposition to speed up repulsive force calculation to $O(n \log n)$ time.
- Can be combined with other speed-up techniques.
Well-Separated Pairs

Definition

Let $s > 0$. $A$ and $B$ in $\mathbb{R}^2$ are $s$-well-separated iff there are two balls $C_A$ and $C_B$ with

- $A \subseteq C_A$ and $B \subseteq C_B$,
- radius $r := r(C_A) = r(C_B)$, and
- distance $\geq s \cdot r$.

[Callahan & Kosaraju, 1995]
Well-Separated Pairs

Definition

Let $s > 0$. $A$ and $B$ in $\mathbb{R}^2$ are $s$-well-separated iff there are two balls $C_A$ and $C_B$ with

1. $A \subseteq C_A$ and $B \subseteq C_B$,
2. radius $r := r(C_A) = r(C_B)$, and
3. distance $\geq s \cdot r$.

[Callahan & Kosaraju, 1995]
Well-Separated Pairs

Definition

Let $s > 0$. $A$ and $B$ in $\mathbb{R}^2$ are $s$-well-separated iff there are two balls $C_A$ and $C_B$ with

- $A \subseteq C_A$ and $B \subseteq C_B$,
- radius $r := r(C_A) = r(C_B)$, and
- distance $\geq s \cdot r$.

[Callahan & Kosaraju, 1995]
Well-Separated Pairs

Definition

Let $s > 0$. $A$ and $B$ in $\mathbb{R}^2$ are $s$-well-separated iff there are two balls $C_A$ and $C_B$ with

- $A \subseteq C_A$ and $B \subseteq C_B$,
- radius $r := r(C_A) = r(C_B)$, and
- distance $\geq s \cdot r$.

[Callahan & Kosaraju, 1995]
Well-Separated Pairs

Definition

Let $s > 0$. $A$ and $B$ in $\mathbb{R}^2$ are $s$-well-separated iff there are two balls $C_A$ and $C_B$ with

- $A \subseteq C_A$ and $B \subseteq C_B$,
- radius $r := r(C_A) = r(C_B)$, and
- distance $\geq s \cdot r$.

[Callahan & Kosaraju, 1995]
**Well-Separated Pair Decomposition (WSPD)**

**Definition**

Let $S$ be a finite set in $\mathbb{R}^2$. A WSPD for $S$ with respect to $s > 0$ is a sequence

$$\{A_1, B_1\}, \{A_2, B_2\}, \ldots, \{A_k, B_k\}$$

of $s$-well-separated pairs $A_i, B_i$ such that

- $A_i, B_i \subseteq S$, and
- $\forall a \neq b \in S$ there is exactly one $i$ with $a \in A_i$ and $b \in B_i$ or vice versa.
Definition

Let $S$ be a finite set in $\mathbb{R}^2$. A WSPD for $S$ with respect to $s > 0$ is a sequence

$$\{A_1, B_1\}, \{A_2, B_2\}, \ldots, \{A_k, B_k\}$$

of $s$-well-separated pairs $A_i, B_i$ such that

- $A_i, B_i \subseteq S$, and
- $\forall a \neq b \in S$ there is exactly one $i$ with $a \in A_i$ and $b \in B_i$ or vice versa.
Definition

Let $S$ be a finite set in $\mathbb{R}^2$. A WSPD for $S$ with respect to $s > 0$ is a sequence

$$\{A_1, B_1\}, \{A_2, B_2\}, \ldots, \{A_k, B_k\}$$

of $s$-well-separated pairs $A_i, B_i$ such that

- $A_i, B_i \subseteq S$, and
- $\forall a \neq b \in S$ there is exactly one $i$ with $a \in A_i$ and $b \in B_i$ or vice versa.
Definition

Let \( S \) be a finite set in \( \mathbb{R}^2 \). A WSPD for \( S \) with respect to \( s > 0 \) is a sequence

\[
\{A_1, B_1\}, \{A_2, B_2\}, \ldots, \{A_k, B_k\}
\]

of \( s \)-well-separated pairs \( A_i, B_i \) such that

- \( A_i, B_i \subseteq S \), and
- \( \forall a \neq b \in S \) there is exactly one \( i \) with \( a \in A_i \) and \( b \in B_i \) or vice versa.
Definition

Let $S$ be a finite set in $\mathbb{R}^2$. A WSPD for $S$ with respect to $s > 0$ is a sequence

$$\{A_1, B_1\}, \{A_2, B_2\}, \ldots, \{A_k, B_k\}$$

of $s$-well-separated pairs $A_i$, $B_i$ such that

- $A_i, B_i \subseteq S$, and
- $\forall a \neq b \in S$ there is exactly one $i$ with $a \in A_i$ and $b \in B_i$ or vice versa.
Well-Separated Pair Decomposition (WSPD)

Definition
Let $S$ be a finite set in $\mathbb{R}^2$. A WSPD for $S$ with respect to $s > 0$ is a sequence

$$\{A_1, B_1\}, \{A_2, B_2\}, \ldots, \{A_k, B_k\}$$

of $s$-well-separated pairs $A_i, B_i$ such that

- $A_i, B_i \subseteq S$, and
- $\forall a \neq b \in S$ there is exactly one $i$ with $a \in A_i$ and $b \in B_i$ or vice versa.
Well-Separated Pair Decomposition (WSPD)

**Definition**

Let $S$ be a finite set in $\mathbb{R}^2$. A WSPD for $S$ with respect to $s > 0$ is a sequence

$$\{A_1, B_1\}, \{A_2, B_2\}, \ldots, \{A_k, B_k\}$$

of $s$-well-separated pairs $A_i, B_i$ such that

- $A_i, B_i \subseteq S$, and
- $\forall a \neq b \in S$ there is exactly one $i$ with $a \in A_i$ and $b \in B_i$ or vice versa.
Computation of WSPD

Theorem

Let $S$ be a set of $n$ points in $\mathbb{R}^2$. For any $s > 0$, a WSPD consisting of $O(n)$ pairs can be computed in $O(n \log n)$ time.

[Callahan & Kosaraju, 1995]
Repulsive Force Calculation

\[ B \cdot F_{\text{rep}} \]

Compute forces for \( k \in O(n) \) well-separated pairs
Repulsive Force Calculation

\[ \mathbf{B} \cdot \mathbf{F}_{\text{rep}} \]

Compute forces for \( k \in \mathcal{O}(n) \) well-separated pairs
Repulsive Force Calculation

\[ \mathbf{B} \mathbf{\cdot} F_{\text{rep}} \]

Compute forces for \( k \in O(n) \) well-separated pairs.
Repulsive Force Calculation

\[ B \cdot F_{\text{rep}} \]

Compute forces for \( k \in \mathcal{O}(n) \) well-separated pairs.
Repulsive Force Calculation

\[ B \cdot F_{\text{rep}} \]

Compute forces for \( k \in O(n) \) well-separated pairs.
Repulsive Force Calculation

Compute forces for $k \in O(n)$ well-separated pairs
Repulsive Force Calculation

\[ B \cdot F_{\text{rep}} A \cdot F_{\text{rep}} \]

Compute forces for \( k \in O(n) \) well-separated pairs.
Repulsive Force Calculation

\[ \mathbf{B} \cdot \mathbf{F}_{\text{rep}} \]

Compute forces for \( k \in O(n) \) well-separated pairs
Repulsive Force Calculation

Compute forces for $\Theta(n^2)$ pairs of vertices
Repulsive Force Calculation

\[ B \cdot F_{\text{rep}} \]

Compute forces for \( k \in \Omega(n) \) well-separated pairs.
Repulsive Force Calculation

\[ \mathbf{B} \cdot \mathbf{F}_{\text{rep}} \]

Compute forces for \( k \in O(n) \) well-separated pairs.
Repulsive Force Calculation

$$B \cdot F_{rep}$$

Compute forces for \( k \in O(n) \) well-separated pairs
Repulsive Force Calculation

\[ B \cdot F_{\text{rep}} \]

Compute forces for \( k \in O(n) \) well-separated pairs.
Repulsive Force Calculation

Compute forces for $k \in \mathcal{O}(n)$ well-separated pairs
Repulsive Force Calculation

\[ |B| \cdot F_{\text{rep}} \]

\[ |A| \cdot F_{\text{rep}} \]
Repulsive Force Calculation

\[ B \cdot F_{\text{rep}} A \cdot F_{\text{rep}} \]

Compute forces for \( k \in O(n) \) well-separated pairs.
Repulsive Force Calculation

\[ B \cdot F_{\text{rep}} \]

Compute forces for \( k \in \mathcal{O}(n) \) well-separated pairs
Repulsive Force Calculation

\[ \mathbf{B} \cdot \mathbf{F}_{\text{rep}} \]

Compute forces for \( k \in \Omega(n) \) well-separated pairs
Compute forces for $k \in O(n)$ well-separated pairs
Using the Split Tree
Using the Split Tree
Using the Split Tree
Using the Split Tree
Using the Split Tree
Using the Split Tree
### Running time

<table>
<thead>
<tr>
<th>Task</th>
<th>Time Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute the WSPD using a split tree</td>
<td>$O(n \log n)$</td>
</tr>
<tr>
<td>Compute forces for every pair</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>Propagate forces in the split tree</td>
<td>$O(n)$</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>$O(n \log n)$</td>
</tr>
</tbody>
</table>
Experiments

Implementation

- in Java
- based on FRLayout in Jung (Java Universal Network/Graph Framework)

Comparison to similar speed-up techniques:

- Quadtree based [Barnes & Hut, 1986]
- Grid variant [Fruchterman & Reingold, 1991]
Experiments: Running time

![Graph showing running time vs number of vertices for different layouts: FRLayout (JUNG), FR+Quad, Θ = 1.0, FR+Grid, FR+WSPD, s = 0.1]
Experiments: Quality

Number of edge crossings vs. Standard deviation of edge length

- FRLayout (JUNG)
- FR+Quad, $\Theta = 1.0$
- FR+Grid
- FR+WSPD, $s = 0.1$

Better performance is indicated by a lower standard deviation of edge length.
Experiments: Quality

![Graph showing the relationship between the number of edge crossings and the standard deviation of edge length for different FR related layout algorithms. The graph indicates that FRLayout (JUNG) and FR+Quad, Θ = 1.0 have a lower number of edge crossings compared to FR+Grid and FR+WSPD, s = 0.1.](image)

- FRLayout (JUNG)
- FR+Quad, Θ = 1.0
- FR+Grid
- FR+WSPD, s = 0.1
Experiments: Quality

Number of edge crossings vs. Standard deviation of edge length

- **FRLayout (JUNG)**
- **FR+Quad, \( \Theta = 1.0 \)**
- **FR+Grid**
- **FR+WSPD, \( s = 0.1 \)**

Fabian Lipp, Alexander Wolff, Johannes Zink | Force-Directed Graph Drawing with WSPD
**Experiments: Quality**

The graph shows the relationship between the number of edge crossings and the standard deviation of edge length for different layout algorithms.

- **FRLayout (JUNG)**
- **FR+Quad, Θ = 1.0**
- **FR+Grid**
- **FR+WSPD, s = 0.1**

The data points are color-coded accordingly, with different markers representing each algorithm.
Experiments: Quality

Number of edge crossings

Standard deviation of edge length

- FRLayout (JUNG)
- FR+Quad, $\Theta = 1.0$
- FR+Grid
- FR+WSPD, $s = 0.1$
Summary

- Asymptotic running time of $O(n \log n)$ for each iteration
- Notably faster than unmodified algorithm
- Similar to unmodified algorithm in terms of quality
- Better quality than Grid variant of Fruchterman & Reingold
Summary

- Asymptotic running time of $O(n \log n)$ for each iteration
- Notably faster than unmodified algorithm
- Similar to unmodified algorithm in terms of quality
- Better quality than Grid variant of Fruchterman & Reingold

Demonstration during the poster session this afternoon
Summary

- Asymptotic running time of $O(n \log n)$ for each iteration
- Notably faster than unmodified algorithm
- Similar to unmodified algorithm in terms of quality
- Better quality than Grid variant of Fruchterman & Reingold

Demonstration during the poster session this afternoon

Thank you!
APPENDIX
Example drawings

FRLayout (JUNG)

FR+WSPD, $s = 0.1$
Example drawings

FRLayout (JUNG)

FR+WSPD, \( s = 0.1 \)
Experiments: Running time on large graphs

![Graph showing running time vs number of vertices for different layouts.]

- **FRLayout (JUNG)**
- **FR+Quad, \( \Theta = 1.0 \)**
- **FR+Grid**
- **FR+WSPD, \( s = 0.1 \)**
Experiments: Running times compared to OGDF

![Graph showing running times vs number of vertices](image)

- **FRLayout (JUNG)**
- **FRExact (OGDF)**
- **FR+Grid**
- **FM^3 (OGDF)**
- **FR+Quad, \( \Theta = 1.0 \)**
- **FM^3 single (OGDF)**
- **FR+WSPD, \( s = 0.1 \)**
- **GEM (OGDF)**
Experiments: Running times compared to OGDF

![Graph showing running times for different algorithms]

- **FRLayout (JUNG)**
- **FRExact (OGDF)**
- **FR+Grid**
- **FM³ (OGDF)**
- **FR+Quad, Θ = 1.0**
- **FM³ single (OGDF)**
- **FR+WSPD, s = 0.1**
- **GEM (OGDF)**

Number of vertices vs. Running time in seconds
Experiments: Quality compared to OGDF

![Scatter plot showing standard deviation of edge length versus number of edge crossings for different methods.]

- FR+WSPD, $s = 0.1$
- FM³ (OGDF)
- GEM (OGDF)
- FREExact (OGDF)
- FM³ single (OGDF)