Easy binary tree drawing

\[x\text{-coord} = \text{in-order number}\]
\[y\text{-coord} = -\text{depth}\]
Tidier binary tree drawing [Reingold & Tilford 81]

1. Construct a tidy drawing of left subtree
2. Construct a tidy drawing of right subtree
3. Shift them horizontally to create a gap of 2
4. Shift them down by 1
5. Center the root (on $x = 0$) between its children
Tidier binary tree drawing [Reingold & Tilford 81]

1. Construct a tidy drawing of left subtree
2. Construct a tidy drawing of right subtree
3. Shift them horizontally to create a gap of 2
4. Shift them down by 1
5. Center the root (on $x = 0$) between its children
Tidier binary tree drawing [Reingold & Tilford 81]

1. Construct a tidy drawing of left subtree
2. Construct a tidy drawing of right subtree
3. Shift them horizontally to create a gap of 2
4. Shift them down by 1
5. Center the root (on $x = 0$) between its children
Tidier binary tree drawing [Reingold & Tilford 81]

1. Construct a tidy drawing of left subtree
2. Construct a tidy drawing of right subtree
3. Shift them horizontally to create a gap of 2
4. Shift them down by 1
5. Center the root (on $x = 0$) between its children
Tidier binary tree drawing [Reingold & Tilford 81]

3. Shift them horizontally to create a gap of 2

\[
\tilde{L} = [(-1, -1), (+1, +1), (-1, +1)]
\]

\[
\tilde{R} = [(-1, +1), (-1, -1)]
\]

\[
\tilde{T} = [(-2, +2), (-1, +1), (+1, -1), (-1, -3)]
\]
Recursive Winding [Chan et al. 97]

HV-drawing

Horizontal

Vertical
Recursive Winding [Chan et al. 97]

Right-heavy H-drawing

Always horizontal but bigger subtree to right

1. Height is at most $\log n$
2. Width is at most $n - 1$
3. Aspect ratio is bad: $\Omega(n/\log n)$ (not order preserving)
Recursive Winding [Chan et al. 97]

Choose $k$ so that:

1. $\ell_1 + \ell_2 + \ldots + \ell_{k-1} < A$
2. $\ell_1 + \ell_2 + \ldots + \ell_{k-1} + \ell' \geq A$

so $\ell' \leq \ell'' \leq \ell - A$. 
Recursive Winding [Chan et al. 97]

\[ k = 1 \]

\[ k = 2 \]
Recursive Winding [Chan et al. 97]

\[ k > 2 \]

\[ T_1 \quad T_2 \quad \ldots \quad T_{k-2} \quad T_{k-1} \]

\[ L \quad R \]
Rings [Teoh & Ma 02]
Let \( f(k) = 1 - \frac{\text{red area}}{\text{total area}} \).

1. Sort children by their number of kids.
2. Pick \( k \) so number of kids of top \( k \) children \( \approx f(k) \) fraction of all grandkids.
3. Recursively draw top \( k \) child-subtrees in blue circles.
4. Draw remaining child-subtrees similarly in rings within red circle.
Rings [Teoh & Ma 02]

Rotate to minimize overlap.
Hyperbolic trees [Lamping, Rao & Pirolli 95]