Computer-Generated Pen-and-Ink Illustration of Trees
Oliver Deussen and Thomas Strothotte

Presentation by
Dmytro Prykhodko

Overview
- Introduction and Goal
- Related work
- Traditional approach
- Illustration of trees
- Results
- Conclusions

Introduction
- Method for automatically rendering pen-and-ink illustrations of trees
- Input is detailed tree models consisting of tree skeleton and leaves

Goal
- To provide the user with a transition from a tree illustration with a realistic plant-specific look to an abstract representation consisting of only a few strokes

Related Work
- C. I. Yessios - uses abstract plant symbols in 2D drafting system, 1979
- A.R. Smith - “cartoon tree” with small disks representing bunches of leaves, 1984
- Kowalski et al. - Art-based rendering, 1999

Traditional Approach
- The tree skeleton is usually drawn up to the second branching level, by silhouette lines and crosshatching on the stem surface
Traditional Approach

- Shape of the foliage is presented by
  - an abstract outline, or
  - a collection of many small objects

Illustration of Trees

- Create detailed tree model using conventional modeling program (xfrog)

Generating Illustrations

- Trunk and branches are drawn using NPR
- Foliage is rendered by representing each leaf by a disk facing the viewer
- Optionally, cast shadows

Drawing the Tree Skeleton

- Tree skeleton consists of cylinders each representing a branch
- Outline generated by analytical silhouette algorithms
Drawing the Tree Skeleton
- Crosshatches placed with “Difference Image Algorithm” or variant of Floyd Steinberg method
- Trunk and main branches of Tree I

Drawing the Foliage
- Each leaf is represented by the outline of abstract drawing primitive
- Position is determined by the 3D leaf position and the size is controlled by the user

Algorithm to draw the foliage
- Draw the leaves as solids and get depth buffer
- For each pixel, determine how far it is in front of its neighbors, and store it in a separate buffer
- Use that data to draw the leaves

Depth in camera coordinate system
\[ z = \frac{z_1 z_0 (d_1 - d_0)}{d - (z_1 + z_0) (d_1 - d_0) - (d_1 + d_0)} \]
- \( d_0 \) & \( d_1 \) = min and max values represented in the depth buffer
- \( z_0 \) & \( z_1 \) = corresponding depth values of near and far clipping planes in camera projection
- \( d \) = depth value [0..1]

Depth differences can be computed for camera coordinates or for depth buffer
- Depth difference threshold is determined by a percentage of the depth range of the tree

Changing ratio of \( z_1 \) to \( z_0 \) alters non-linearity
- More detail up front and less in the back because depth buffer has non-linear characteristic
- Changing ratio of \( z_1 \) to \( z_0 \) alters non-linearity

Level of Detail
- Primitive sizes and threshold are constant
- Primitive sizes are enlarged with distance
A number of drawing primitives can be used to represent leaves.

- Leaves are rendered using interpolated polygons from nine given samples.
- Shadow drawn in black, threshold=100.
- Shadow represented by detail, threshold=6000.

For generating high quality images:
- Bitmaps of the stem and foliage are vectorized
  1) least squares fitting (determine global vectors)
  2) index buffer (stores primitive IDs at each pixel as a color value)
- Polygons drawn by spline interpolation.

This tree is drawn using view-facing elliptic primitives of random orientation.
- Only the lower part of each ellipse is drawn.
- Threshold = 100.
- 13,200 particles.

16,200 particles (originally 200,000 leaves).

90,000 particles + 23,000 for the ground.

Real leaves.
Conclusion

- Trunk and branches are represented by silhouette lines augmented by crosshatching in dark areas.
- Foliage is drawn by using abstract primitives that represent leaves.
- Depth differences are used to determine what part of the primitives are drawn.

Future Work

- Instead of shadow buffer, maybe crosshatch the leaves to suggest shadow.
- Reduce amount of geometric data. A continuous level-of-detail algorithm for trees will improve performance.
- Add new styles and colored versions for cartoons application.

The End