Efficient Distance Field Computation using Cluster Trees

Elena Jakubiak, Tufts University Department of Computer Science • Sarah Friaken, Tufts University Department of Computer Science • Ronald Perry, Mitsubishi Electric Research Laboratories

The Abstract
Hierarchical data structures provide a means for organizing data for efficient processing. Most spatial data structures are optimized for performing queries, such as intersection and containment testing, on large, often dynamic, data sets. Set up time and complexity of these structures can limit their value for small, static data sets. We have developed a spatial data structure and associated query algorithm, dubbed Cluster Trees, for providing efficient minimum distance queries from an arbitrary point in space to a small set of Bezier curves. Cluster Trees have been used to significantly reduce the time required to generate an Adaptively Sampled Distance Field (ADF) representing a character glyph such as the letter ‘A’ from a standard outline representation consisting of lines and Bezier curves.

The Background
Adaptively Sampled Distance Fields (ADFs) provide many advantages over traditional methods, such as outlines, for representing and rendering fonts.

The Motivation
Computing the distance field of an outline font requires many costly queries.

A Solution
Eliminate unnecessary distance computations using a spatial hierarchy of the axis-aligned bounding boxes of the glyph’s Bezier curves.

Building a Cluster Tree
Organize the axis-aligned bounding boxes into a hierarchical spatial data structure.

Querying a Cluster Tree
Compute the minimum distance from a given point to the glyph’s outline. The minimum distance is less than the distance from the point to the node at the head of the list. Here, k is a leaf node containing a Bezier curve segment δc_k. The minimum distance is updated to dist(δc_k, δS). Since dist(δc_k, δS) < dist(δc_k, δS), the query is complete.

The Conclusion
Cluster Trees were used to generate ADFs from several standard outline-based font representations. On average, querying the cluster tree required 24 distance-to-boundary queries to compute the minimum distance and less than three distance-to-quadratic Bezier curve computations to compute the distance from a point to a quadratic Bezier curve takes approximately 20 times as long as computing the distance from a point to an axis-aligned bounding box. Hence, for an average glyph containing 40 Bezier curve segments, using Cluster Trees resulted in a 10X reduction in computation during querying. Total ADF generation times, which included building a Cluster Tree for each glyph, were reduced on average by 28 to 49%.

Building a Cluster Tree

Querying a Cluster Tree

The Conclusion

Cluster Trees were used to generate ADFs from several standard outline-based font representations. On average, querying the cluster tree required 24 distance-to-boundary queries to compute the minimum distance and less than three distance-to-quadratic Bezier curve computations to compute the distance from a point to a quadratic Bezier curve takes approximately 20 times as long as computing the distance from a point to an axis-aligned bounding box. Hence, for an average glyph containing 40 Bezier curve segments, using Cluster Trees resulted in a 10X reduction in computation during querying. Total ADF generation times, which included building a Cluster Tree for each glyph, were reduced on average by 28 to 49%.

Building a Cluster Tree
Organize the axis-aligned bounding boxes into a hierarchical spatial data structure.

Querying a Cluster Tree
Compute the minimum distance from a given point to the glyph’s outline. The minimum distance is less than the distance from the point to the node at the head of the list. Here, k is a leaf node containing a Bezier curve segment δc_k. The minimum distance is updated to dist(δc_k, δS). Since dist(δc_k, δS) < dist(δc_k, δS), the query is complete.

The Conclusion
Cluster Trees were used to generate ADFs from several standard outline-based font representations. On average, querying the cluster tree required 24 distance-to-boundary queries to compute the minimum distance and less than three distance-to-quadratic Bezier curve computations to compute the distance from a point to a quadratic Bezier curve takes approximately 20 times as long as computing the distance from a point to an axis-aligned bounding box. Hence, for an average glyph containing 40 Bezier curve segments, using Cluster Trees resulted in a 10X reduction in computation during querying. Total ADF generation times, which included building a Cluster Tree for each glyph, were reduced on average by 28 to 49%.

Building a Cluster Tree

Querying a Cluster Tree

The Conclusion
Cluster Trees were used to generate ADFs from several standard outline-based font representations. On average, querying the cluster tree required 24 distance-to-boundary queries to compute the minimum distance and less than three distance-to-quadratic Bezier curve computations to compute the distance from a point to a quadratic Bezier curve takes approximately 20 times as long as computing the distance from a point to an axis-aligned bounding box. Hence, for an average glyph containing 40 Bezier curve segments, using Cluster Trees resulted in a 10X reduction in computation during querying. Total ADF generation times, which included building a Cluster Tree for each glyph, were reduced on average by 28 to 49%.

Building a Cluster Tree
Organize the axis-aligned bounding boxes into a hierarchical spatial data structure.

Querying a Cluster Tree
Compute the minimum distance from a given point to the glyph’s outline. The minimum distance is less than the distance from the point to the node at the head of the list. Here, k is a leaf node containing a Bezier curve segment δc_k. The minimum distance is updated to dist(δc_k, δS). Since dist(δc_k, δS) < dist(δc_k, δS), the query is complete.

The Conclusion
Cluster Trees were used to generate ADFs from several standard outline-based font representations. On average, querying the cluster tree required 24 distance-to-boundary queries to compute the minimum distance and less than three distance-to-quadratic Bezier curve computations to compute the distance from a point to a quadratic Bezier curve takes approximately 20 times as long as computing the distance from a point to an axis-aligned bounding box. Hence, for an average glyph containing 40 Bezier curve segments, using Cluster Trees resulted in a 10X reduction in computation during querying. Total ADF generation times, which included building a Cluster Tree for each glyph, were reduced on average by 28 to 49%.

Building a Cluster Tree
Organize the axis-aligned bounding boxes into a hierarchical spatial data structure.

Querying a Cluster Tree
Compute the minimum distance from a given point to the glyph’s outline. The minimum distance is less than the distance from the point to the node at the head of the list. Here, k is a leaf node containing a Bezier curve segment δc_k. The minimum distance is updated to dist(δc_k, δS). Since dist(δc_k, δS) < dist(δc_k, δS), the query is complete.

The Conclusion
Cluster Trees were used to generate ADFs from several standard outline-based font representations. On average, querying the cluster tree required 24 distance-to-boundary queries to compute the minimum distance and less than three distance-to-quadratic Bezier curve computations to compute the distance from a point to a quadratic Bezier curve takes approximately 20 times as long as computing the distance from a point to an axis-aligned bounding box. Hence, for an average glyph containing 40 Bezier curve segments, using Cluster Trees resulted in a 10X reduction in computation during querying. Total ADF generation times, which included building a Cluster Tree for each glyph, were reduced on average by 28 to 49%.

Building a Cluster Tree

Querying a Cluster Tree

The Conclusion
Cluster Trees were used to generate ADFs from several standard outline-based font representations. On average, querying the cluster tree required 24 distance-to-boundary queries to compute the minimum distance and less than three distance-to-quadratic Bezier curve computations to compute the distance from a point to a quadratic Bezier curve takes approximately 20 times as long as computing the distance from a point to an axis-aligned bounding box. Hence, for an average glyph containing 40 Bezier curve segments, using Cluster Trees resulted in a 10X reduction in computation during querying. Total ADF generation times, which included building a Cluster Tree for each glyph, were reduced on average by 28 to 49%.