A Hybrid Multiresolution Representation for Fast Tree Modeling and Rendering [1]

Ong Yongzhi

Computer Graphics Seminar, 11 December 2012

Outline

Introduction and Background

Modeling Plants and Trees using L-Systems

Solution and Algorithm
- Trunk and Branch Modeling
- Leaves and Foliage
- Combined Execution

Results and Conclusion

Outline

Introduction and Background

Modeling Plants and Trees using L-Systems

Solution and Algorithm
- Trunk and Branch Modeling
- Leaves and Foliage
- Combined Execution

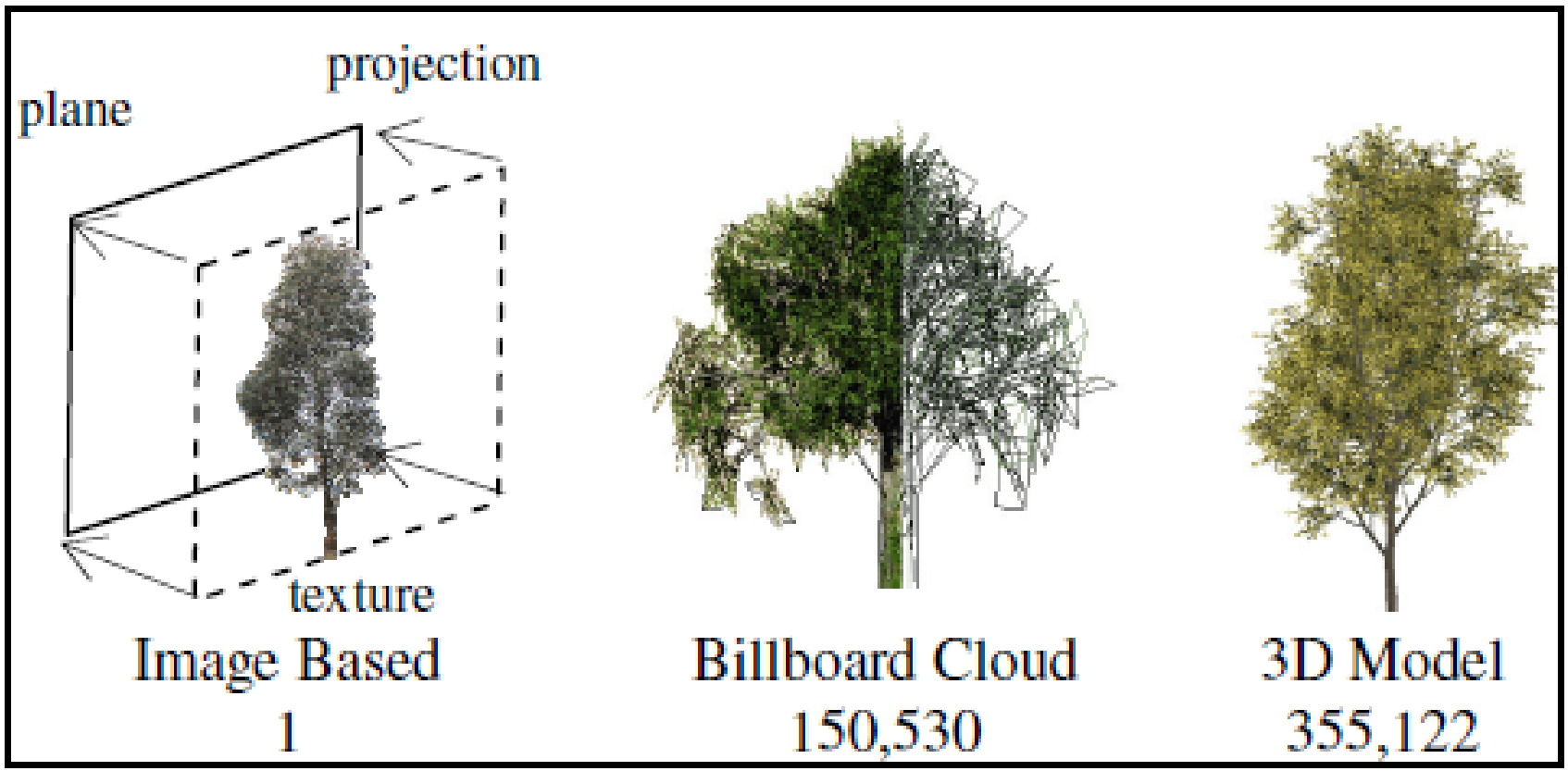
Results and Conclusion

General Overview of Paper

- Object of discussion: Plants/Trees
- Individual techniques:
  - Multiresolution - Geometry-based simplification
  - Image-based - Billboards, textures, etc.
- General disadvantages:
  - Multiresolution - Fail to capture plant/tree nature
  - Image-based - Large storage requirements and close range artifacts
- Solution:
  - Use **both** techniques at procedural level:
    Hybrid Multiresolution

Background - Plant/Tree Modeling - 1

- Texture-mapped Polygons / Billboards



[9]

Background - Plant/Tree Modeling - 2

- L-Systems [7]
- Components [2]
- Images of Real Plants/Trees [8]

Background - Level-of-Detail (LOD) - 1

- Degradation at Range and Pixel-based LODs [10]
- (Binary) Space Partitioning and Multiresolution [3]
- Cluster-based Hierarchical Polygon Decimation and Compression [11]
- Layered Depth Images [4]
- Volumetric Textures [5]
- Bidirectional Textures [6]

Background - More about Multiresolution

- Representation of objects at various LODs
- 4 Characteristics of good multiresolution models:
  - Size of model does not increase with number of LODs
  - Extraction of LODs is fast enough for interactive rendering
  - No loss of information
  - Smooth transition between LODs

Model Type Choice and Justification

- Geometry-based simplification methods fail to maintain tree/plant structure
- Long hours required for designers to build tree representations
- Huge storage requirements
- Selected Model Type:
  L-Systems:
  - Quick model generation
  - Low storage requirement
  - View direction/position independent rendering quality

Problem Division

- Trunk and branches
- Leaves or foliage

Outline

Introduction and Background

Modeling Plants and Trees using L-Systems

Solution and Algorithm
- Trunk and Branch Modeling
- Leaves and Foliage
- Combined Execution

Results and Conclusion

Parametric L-Systems - 1

Example

Axiom :

```
A(length)
```

Rule 1 :

```
A(l):  itNum < maxIt -> B(l) [A(l/2) A(l/2)]
```

Rule 2 :

```
A(l):  itNum = maxIt -> B(l)
```

Parametric L-Systems - 2

Example

```
Axiom :  A(1)
itNum = 0
maxIt = 2

Output Chain:
A(1)
```

Parametric L-Systems - 3

Example

Rule 1 :

```
A(l):  itNum < maxIt -> B(l) [A(l/2) A(l/2)]
```

itNum = 1

maxIt = 3

Output Chain (Before)

```
A(1)
```

Output Chain:

```
B(1) [A(0.5) A(0.5)]
```

Parametric L-Systems - 4

Example

Rule 1 :

```
A(l):  itNum < maxIt -> B(l) [A(l/2) A(l/2)]
```

itNum = 2

maxIt = 3

Output Chain (Before)

```
B(1) [A(0.5) A(0.5)]
```

Output Chain:

```
B(1) [B(0.5) [A(0.25) A(0.25)]
B(0.5) [A(0.25) A(0.25)]]
```

Parametric L-Systems - 5

Example

Rule 2 :

```
A(l):  itNum = maxIt -> B(l)
```

itNum = 3

maxIt = 3

Output Chain (Before)

```
B(1) [B(0.5) [A(0.25) A(0.25)]
B(0.5) [A(0.25) A(0.25)]]
```

Output Chain:

```
B(1) [B(0.5) [B(0.25) B(0.25)]
B(0.5) [B(0.25) B(0.25)]]
```

Turtle Interpretation - 1

Example

Interpretation: F = Forward, R = Rotate

```
F(1) [R(90) F(1)] F(2)
```

Turtle Interpretation - 2

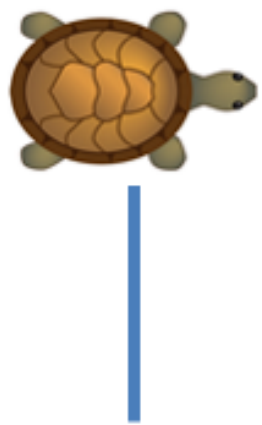
Example

```
F(1) [ R(90) F(1)] F(2)
```

Turtle Interpretation - 3

Example

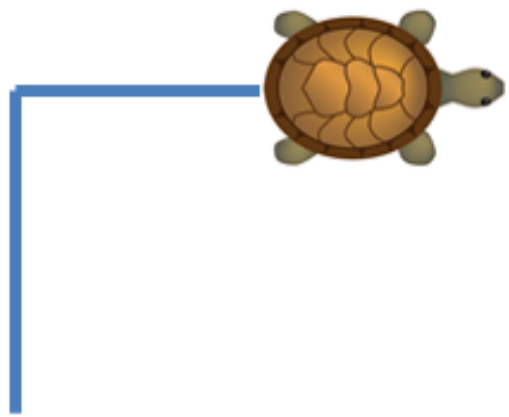
```
F(1) [ R(90) F(1)] F(2)
```



Turtle Interpretation - 4

Example

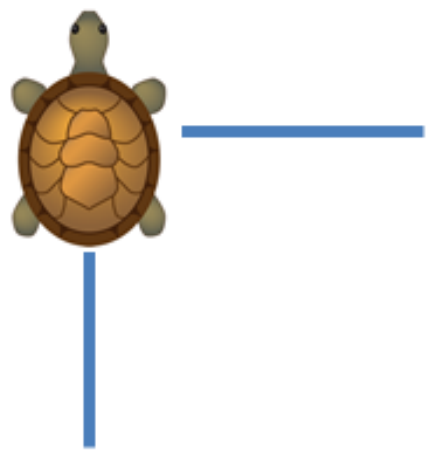
```
F(1) [ R(90) F(1)] F(2)
```



Turtle Interpretation - 5

Example

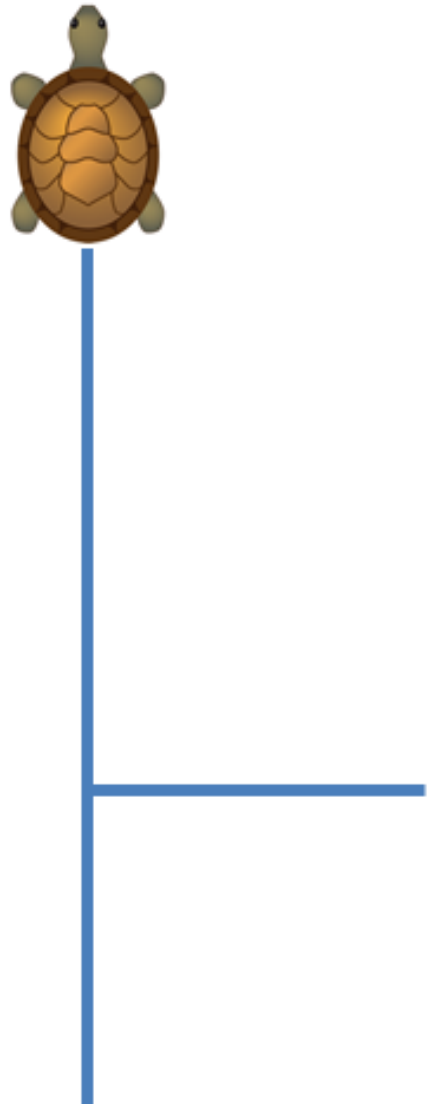
```
F(1) [ R(90) F(1)] F(2)
```



Turtle Interpretation - 6

Example

```
F(1) [ R(90) F(1)] F(2)
```



Outline

Introduction and Background

Modeling Plants and Trees using L-Systems

Solution and Algorithm
- Trunk and Branch Modeling
- Leaves and Foliage
- Combined Execution

Results and Conclusion

Outline

Introduction and Background

Modeling Plants and Trees using L-Systems

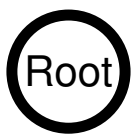
Solution and Algorithm
- Trunk and Branch Modeling
- Leaves and Foliage
- Combined Execution

Results and Conclusion

Step 1 - Tree Abstract Data Type (tADT) - 1

Example

```
Output Chain from L-System:
C1 [ C2 [ C3 ] [ C4 ] ] [ C5 ]
```



Step 1 - Tree Abstract Data Type (tADT) - 2

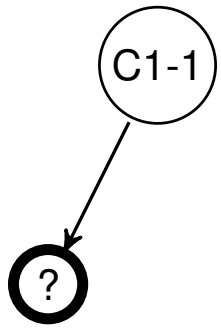
Example

```
Output Chain from L-System:
C1 [ C2 [ C3 ] [ C4 ] ] [ C5 ]
```

Step 1 - Tree Abstract Data Type (tADT) - 3

Example

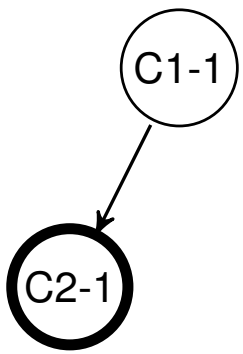
```
Output Chain from L-System:
C1 [ C2 [ C3 ] [ C4 ] ] [ C5 ]
```



Step 1 - Tree Abstract Data Type (tADT) - 4

Example

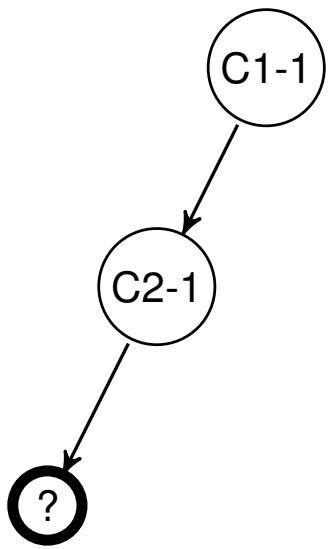
```
Output Chain from L-System:
C1 [ C2 [ C3 ] [ C4 ] ] [ C5 ]
```



Step 1 - Tree Abstract Data Type (tADT) - 5

Example

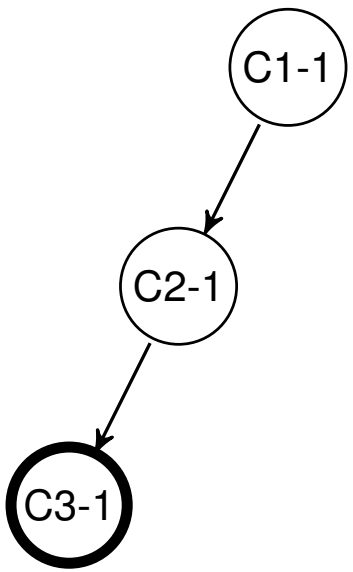
```
Output Chain from L-System:
C1 [ C2 [ C3 ] [ C4 ] ] [ C5 ]
```



Step 1 - Tree Abstract Data Type (tADT) - 6

Example

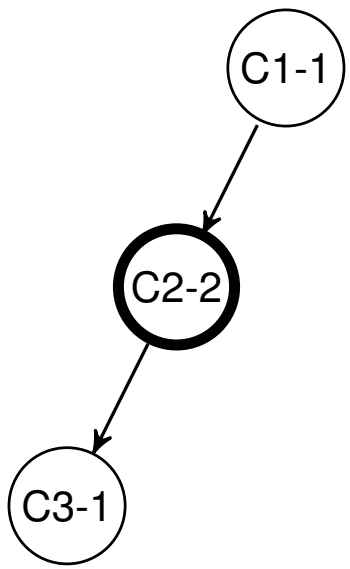
```
Output Chain from L-System:
C1 [ C2 [ C3 ] [ C4 ] ] [ C5 ]
```



Step 1 - Tree Abstract Data Type (tADT) - 7

Example

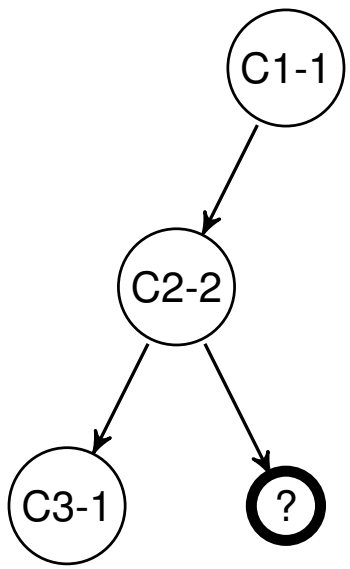
```
Output Chain from L-System:
C1 [ C2 [ C3 ] [ C4 ] ] [ C5 ]
```



Step 1 - Tree Abstract Data Type (tADT) - 8

Example

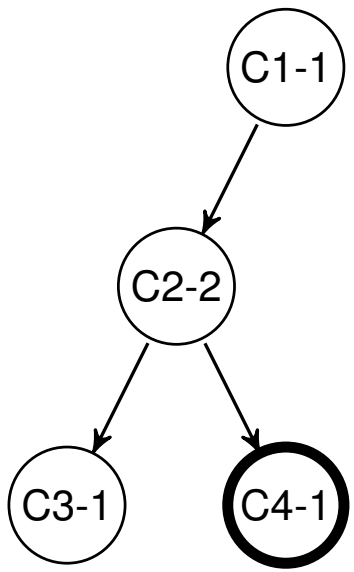
```
Output Chain from L-System:
C1 [ C2 [ C3 ] [ C4 ] ] [ C5 ]
```



Step 1 - Tree Abstract Data Type (tADT) - 9

Example

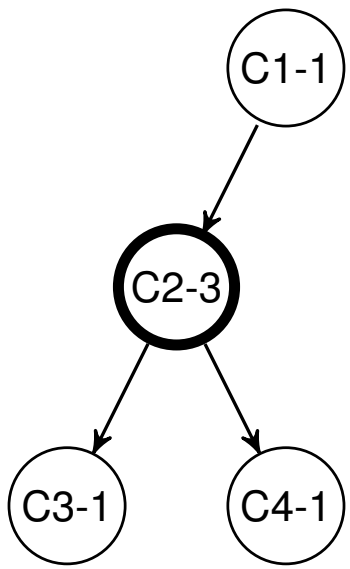
```
Output Chain from L-System:
C1 [ C2 [ C3 ] [ C4 ] ] [ C5 ]
```



Step 1 - Tree Abstract Data Type (tADT) - 10

Example

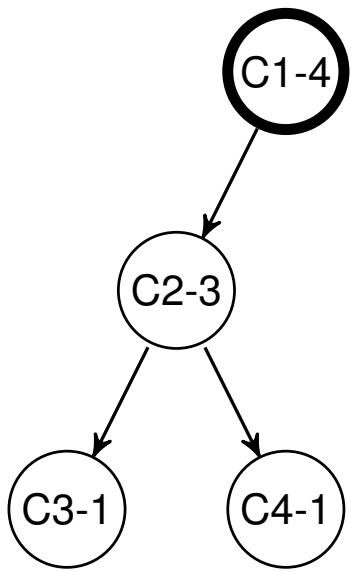
```
Output Chain from L-System:
C1 [ C2 [ C3 ] [ C4 ] ] [ C5 ]
```



Step 1 - Tree Abstract Data Type (tADT) - 11

Example

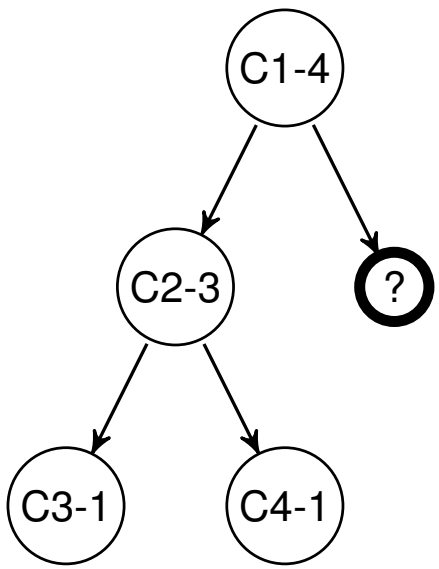
```
Output Chain from L-System:
C1 [ C2 [ C3 ] [ C4 ] ] [ C5 ]
```



Step 1 - Tree Abstract Data Type (tADT) - 12

Example

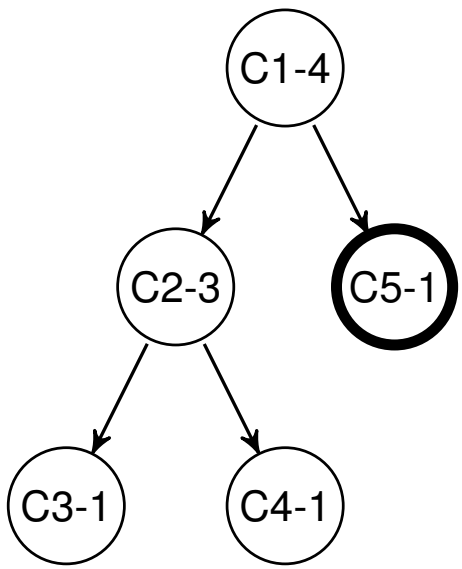
```
Output Chain from L-System:
C1 [ C2 [ C3 ] [ C4 ] ] [ C5 ]
```



Step 1 - Tree Abstract Data Type (tADT) - 13

Example

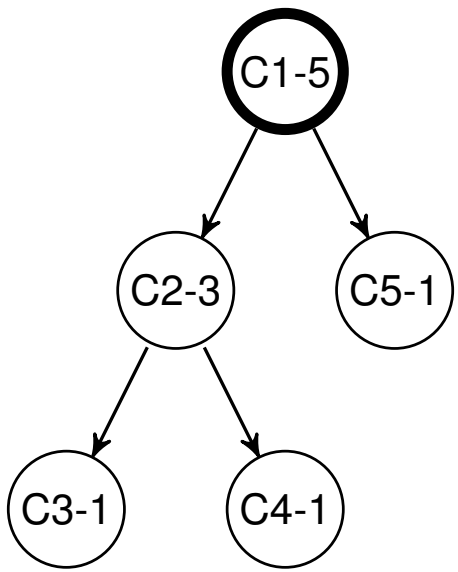
```
Output Chain from L-System:
C1 [ C2 [ C3 ] [ C4 ] ] [ C5 ]
```



Step 1 - Tree Abstract Data Type (tADT) - 14

Example

```
Output Chain from L-System:
C1 [ C2 [ C3 ] [ C4 ] ] [ C5 ]
```



Step 2 - Metric Selection

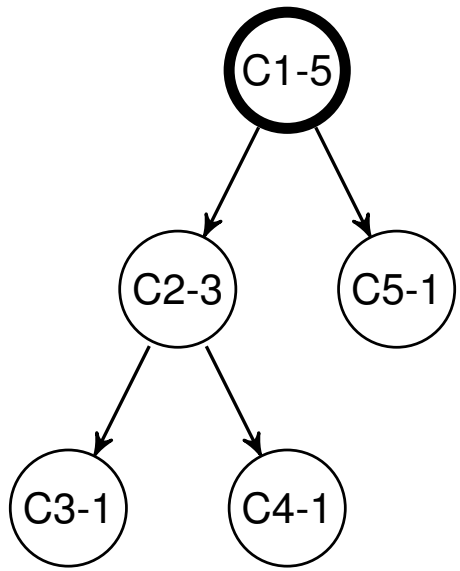
- Number of children
- Number of descendents
- Longest path to a leaf node
- **Branching length** (accumulated in each node in during tADT construction)

Step 3 - Multiresolution Chain - 1

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

Example

`tADT:`



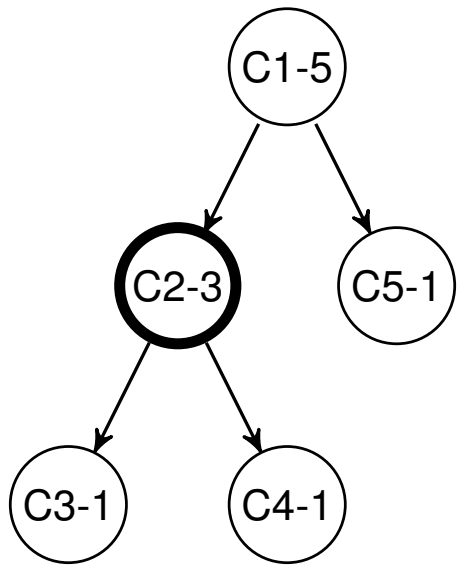
C1 SAVE(C1)

Step 3 - Multiresolution Chain - 2

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

Example

`tADT:`



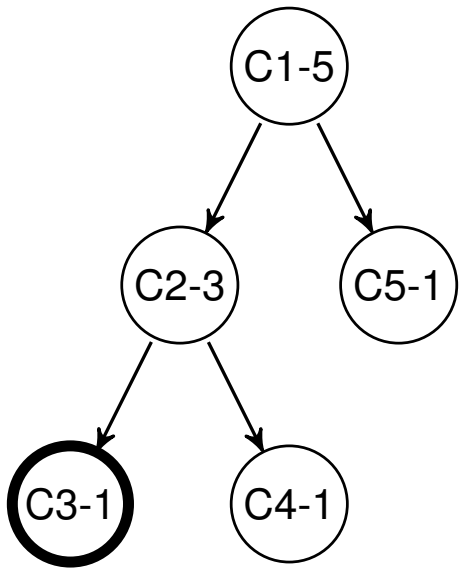
C1 SAVE(C1) C2 SAVE(C2)

Step 3 - Multiresolution Chain - 3

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

Example

`tADT:`



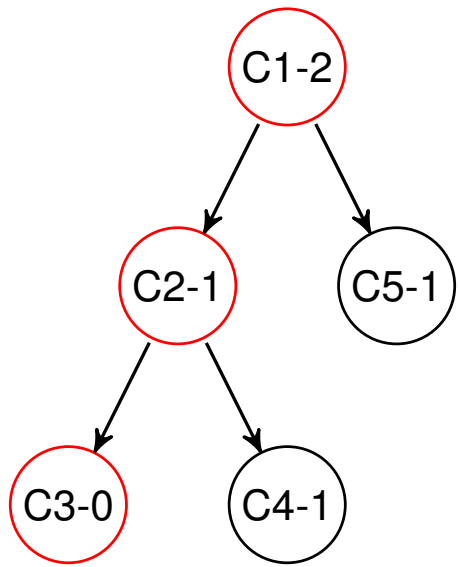
C1 SAVE(C1) C2 SAVE(C2) C3

Step 3 - Multiresolution Chain - 4

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

Example

`tADT:`



Branching length update

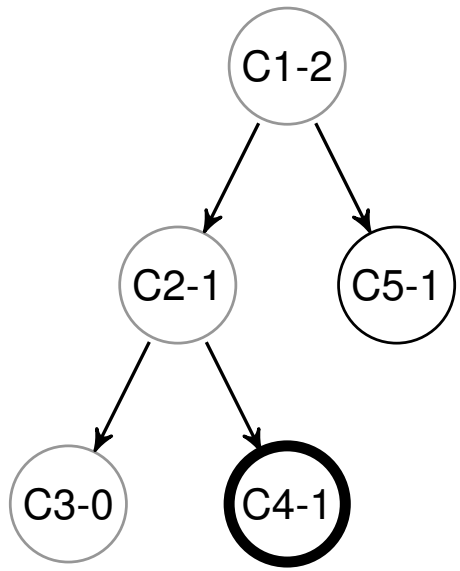
C1 SAVE(C1) C2 SAVE(C2) C3

Step 3 - Multiresolution Chain - 5

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

Example

`tADT:`



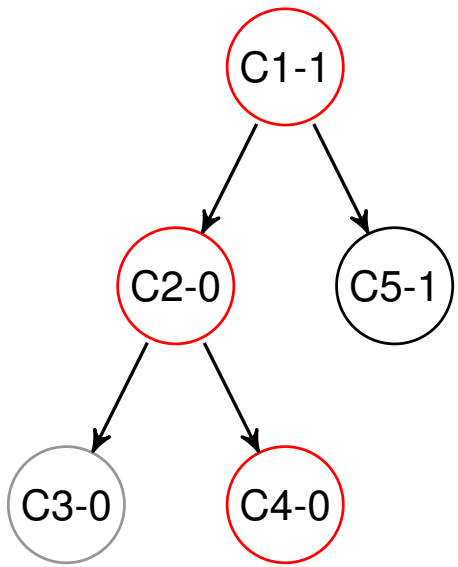
C1 SAVE(C1) C2 SAVE(C2) C3 RESTORE(C2) C4

Step 3 - Multiresolution Chain - 6

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

Example

`tADT:`



Branching length update

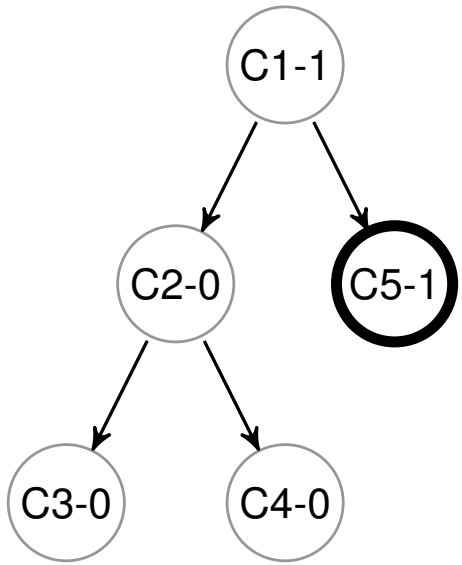
C1 SAVE(C1) C2 SAVE(C2) C3 RESTORE(C2) C4

Step 3 - Multiresolution Chain - 7

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

Example

`tADT:`



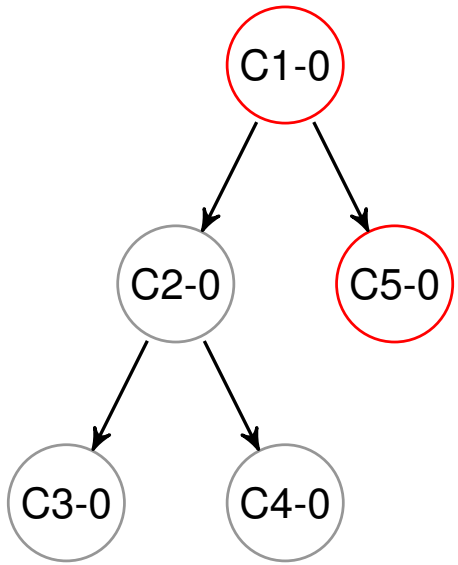
C1 SAVE(C1) C2 SAVE(C2) C3 RESTORE(C2) C4 RESTORE(C1) C5

Step 3 - Multiresolution Chain - 8

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

Example

`tADT:`



Branching length update

C1 SAVE(C1) C2 SAVE(C2) C3 RESTORE(C2) C4 RESTORE(C1) C5

Outline

Introduction and Background

Modeling Plants and Trees using L-Systems

Solution and Algorithm
- Trunk and Branch Modeling
- Leaves and Foliage
- Combined Execution

Results and Conclusion

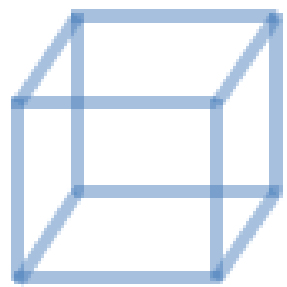
Step 4 - Bounding Box Hierarchy - 1

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

Example

C1 and C2 are turtle commands (orientated bounding boxes)
C3, C4 and C5 are geometry modules

```
Output Chain from L-System:
C1[C2[C3(leaf)][C4(leaf)]][C5(leaf)]
```

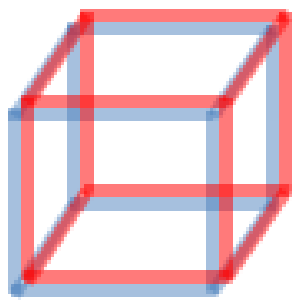


Step 4 - Bounding Box Hierarchy - 2

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

Example

```
Output Chain from L-System:
C1[C2[C3(leaf)][C4(leaf)]][C5(leaf)]
```

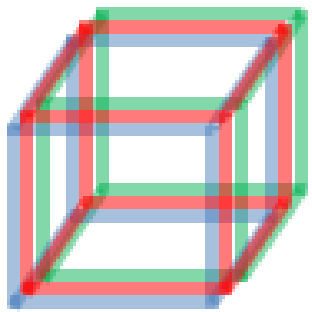


Step 4 - Bounding Box Hierarchy - 3

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

Example

```
Output Chain from L-System:
C1[C2[C3(leaf)][C4(leaf)]][C5(leaf)]
```

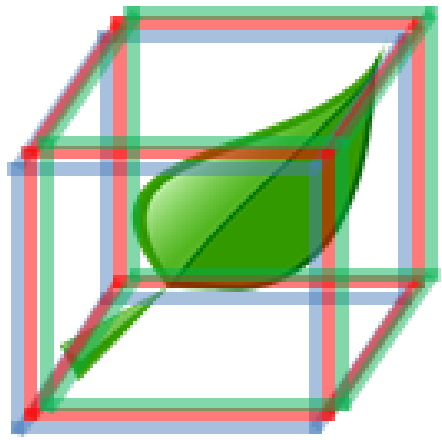


Step 4 - Bounding Box Hierarchy - 4

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

Example

```
Output Chain from L-System:
C1[C2[C3(leaf)][C4(leaf)]][C5(leaf)]
```

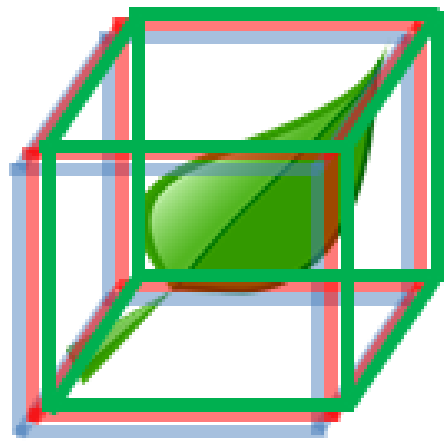


Step 4 - Bounding Box Hierarchy - 5

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

Example

```
Output Chain from L-System:
C1[C2[C3(leaf)][C4(leaf)]][C5(leaf)]
```

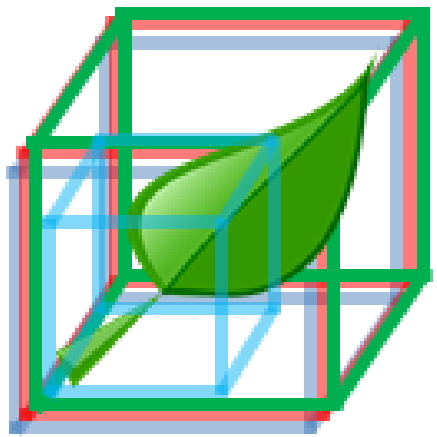


Step 4 - Bounding Box Hierarchy - 6

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

Example

```
Output Chain from L-System:
C1[C2[C3(leaf)][C4(leaf)]][C5(leaf)]
```

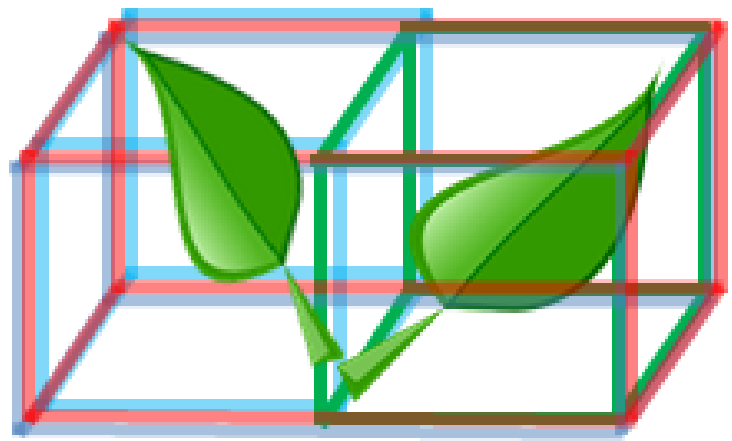


Step 4 - Bounding Box Hierarchy - 7

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

Example

```
Output Chain from L-System:
C1[C2[C3(leaf)][C4(leaf)]][C5(leaf)]
```

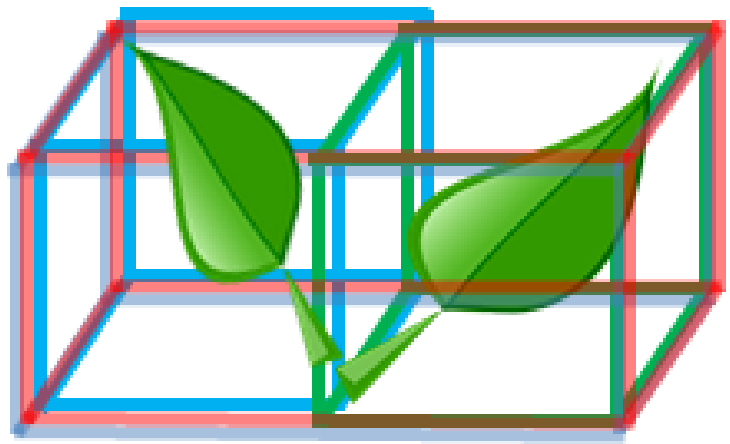


Step 4 - Bounding Box Hierarchy - 8

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

Example

```
Output Chain from L-System:
C1[C2[C3(leaf)][C4(leaf)]][C5(leaf)]
```

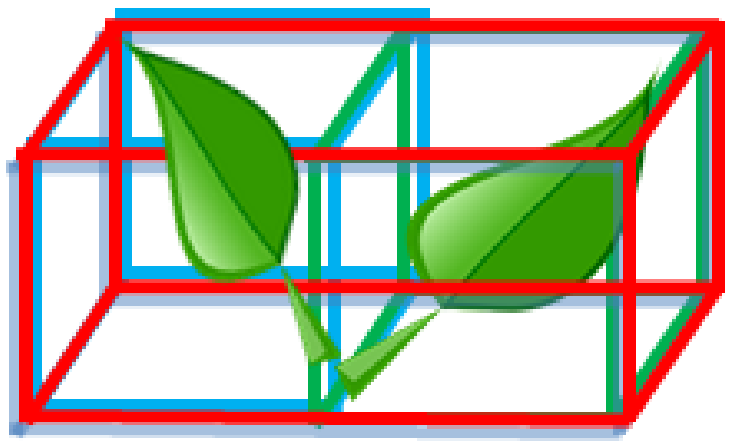


Step 4 - Bounding Box Hierarchy - 9

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

Example

```
Output Chain from L-System:
C1[C2[C3(leaf)][C4(leaf)][C5(leaf)]
```

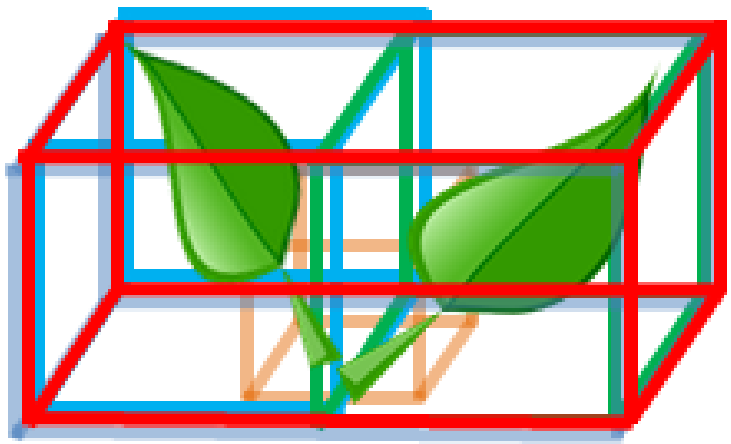


Step 4 - Bounding Box Hierarchy - 10

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

Example

```
Output Chain from L-System:
C1[C2[C3(leaf)][C4(leaf)]][C5(leaf)]
```

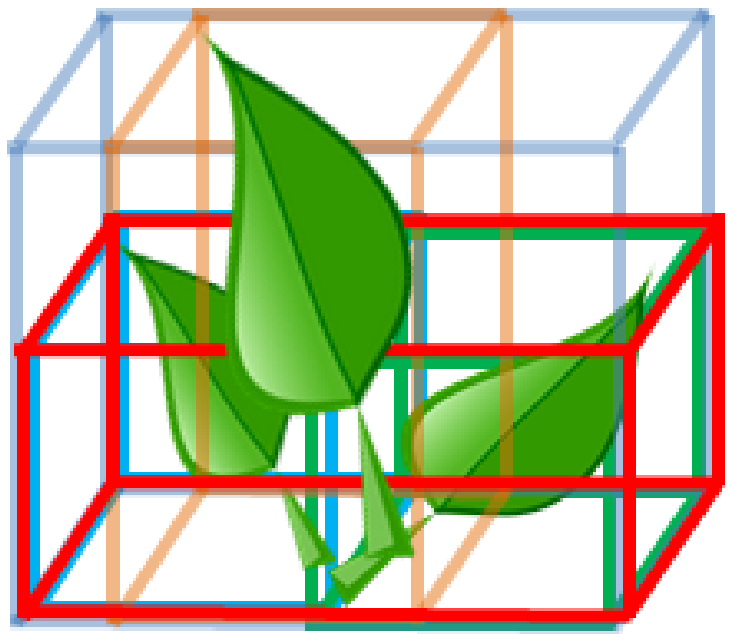


Step 4 - Bounding Box Hierarchy - 11

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

Example

```
Output Chain from L-System:
C1[C2[C3(leaf)][C4(leaf)][C5(leaf)]
```

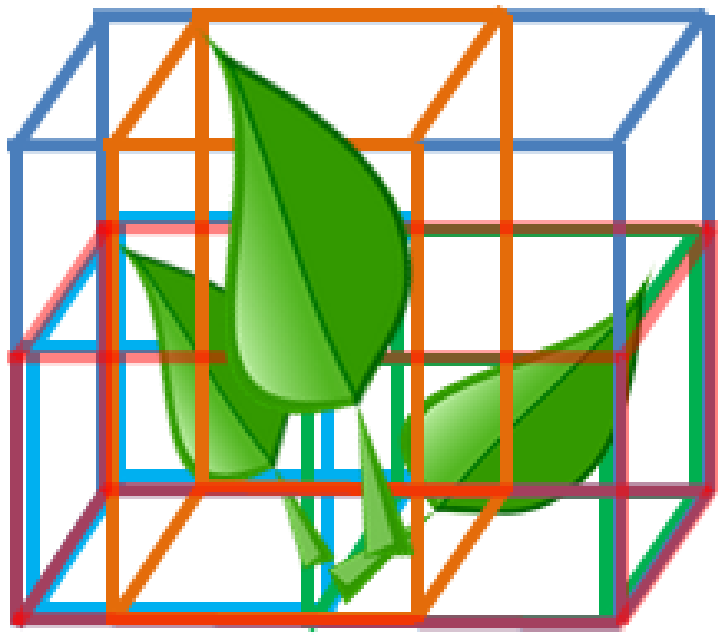


Step 4 - Bounding Box Hierarchy - 12

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

Example

```
Output Chain from L-System:
C1[C2[C3(leaf)][C4(leaf)][C5(leaf)]
```

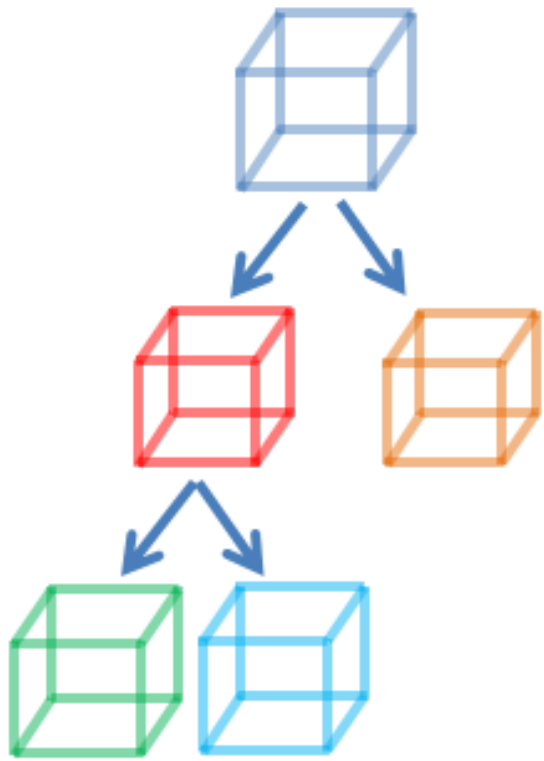


Step 4 - Bounding Box Hierarchy

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

Example

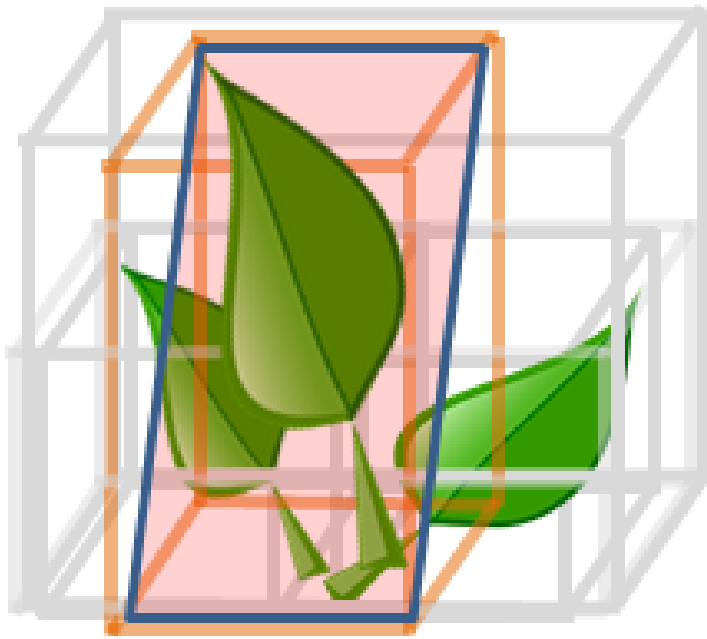
```
Output Chain from L-System:
C1[C2[C3(leaf)][C4(leaf)]][C5(leaf)]
```



Step 5 - Pre-computed Textures

L-System Chain -> tADT -> Multiresolution Chain -> BBox -> Textures

- Textures of 6 diagonal planes for **each** bounding box
- Size: 128x128 pixels (constant for all levels of detail)
- If ratio(current bbox volume:root bbox volume) < threshold, stop computing textures



Outline

Introduction and Background

Modeling Plants and Trees using L-Systems

Solution and Algorithm
- Trunk and Branch Modeling
- Leaves and Foliage
- Combined Execution

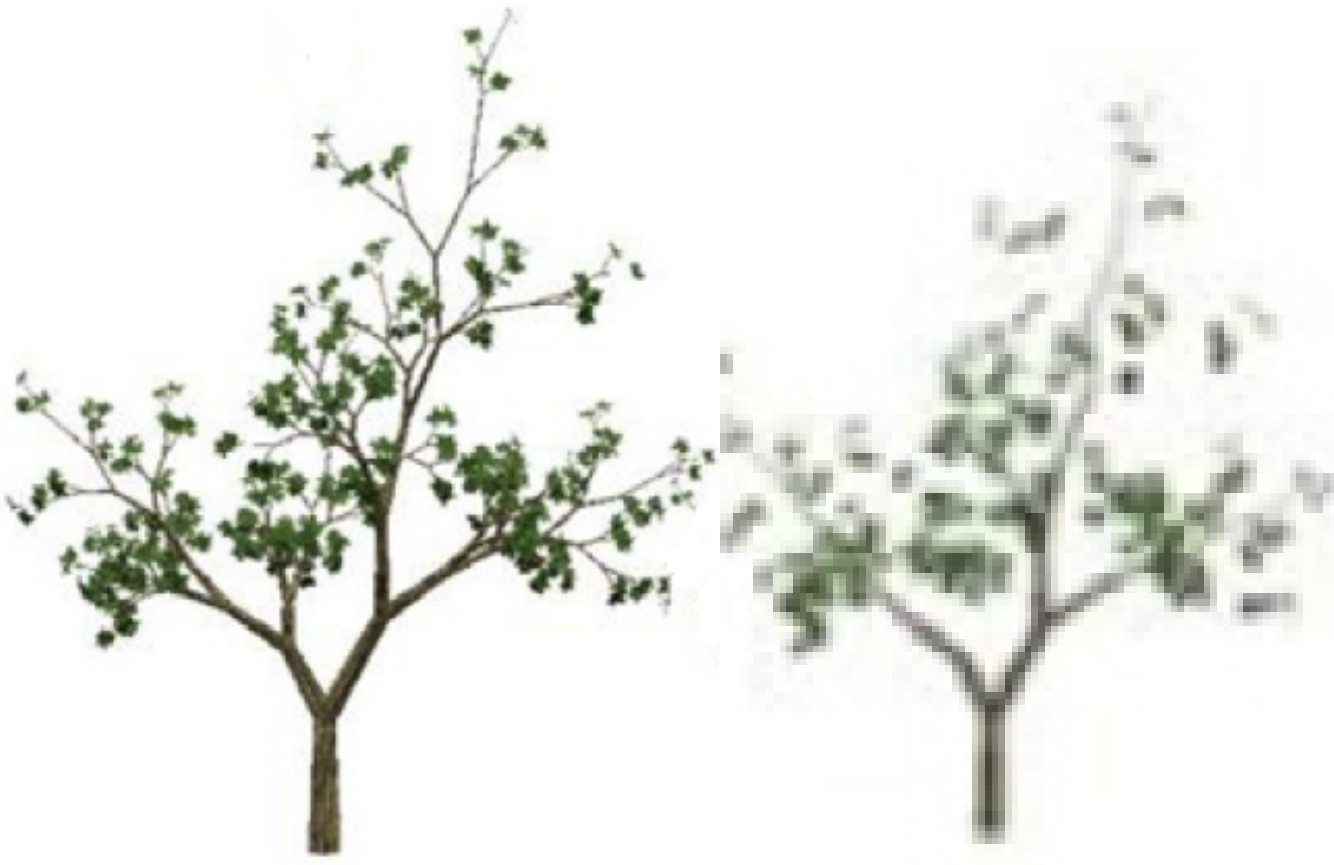
Results and Conclusion

LOD Extraction

- Each **RESTORE** point on the `Multiresolution Chain` is the next LOD.
- Each **Level** of the `Bounding Box Hierarchy` is the next LOD.
- Progressively traverse down `Multiresolution Chain` and `Bounding Box Hierarchy` until desired LOD and render.
- A `Multiresolution Chain` is interpreted as in turtle interpretation. Geometry-representing modules are rendered.

Visual Results - Branches

Visual Results - Branches and Leaves



Outline

Introduction and Background

Modeling Plants and Trees using L-Systems

Solution and Algorithm
- Trunk and Branch Modeling
- Leaves and Foliage
- Combined Execution

Results and Conclusion

Results and Comparison

- 100 Trees:
  - Geometric Model: 3 fps (min)
  - Hybrid Multiresolution Model: 69 fps (min)
- 2000 Trees:
  - Geometric Model: 0.1 fps (min)
  - Hybrid Multiresolution Model: 6 fps (min)

Potential Improvements

- Smooth transitions between LODs
- Reduce memory required for textures
- Wind movements

Thank you!

Bibliography I

[1] L. Javier, C. Emilio, H. Jose Luis, and V. Roberto.
A Hybrid Mutiresolution Representation for Fast Tree Modeling and Rendering.
Procedia Computer Science, 1(1):485–494, 2010.

[2] B. Lintermann and O. Deussen.
Interactive Modeling of Plants.
IEEE Computer Graphics and Applications, 19(1):56–65, 1999.

[3] D. Marshall, D. S. Fussell, and A. T. Campbell.
Multiresolution Rendering of Complex Botanical Scenes.
In Wayne A. Davis, Marilyn M. Mantei, and R. Victor Klassen, editors, *Graphics Interface*, pages 97–104. Canadian Human-Computer Communications Society, 1997.

Bibliography II

[4] N. Max.
Hierarchical rendering of trees from precomputed multi-layer z-buffers.
In *Proceedings of the Eurographics Workshop on Rendering Techniques '96*, pages 165–174, London, UK, UK, 1996. Springer-Verlag.

[5] A. Meyer and F. Neyret.
Interactive Volumetric Textures.
In George Drettakis and Nelson Max, editors, *Eurographics Workshop on Rendering Techniques (Rendering Techniques'98)*, pages 157–168, Vienna, Autriche, 1998. Eurographics, Springer Wein.

Bibliography III

[6] A. Meyer, F. Neyret, and P. Poulin.
Interactive Rendering of Trees with Shading and Shadows.

In Steven J. Gortler and Karol Myszkowski, editors, *12th Eurographics Workshop on Rendering Techniques*, pages 183–196, London, Royaume-Uni, 2001. Springer.

[7] P. Prusinkiewicz, A. Lindenmayer, and J. Hanan.
Development Models of Herbaceous Plants for Computer Imagery Purposes.
SIGGRAPH Computer Graphics, 22(4):141–150, 1988.

[8] L. Quan, P. Tan, G. Zeng, L. Yuan, J. Wang, and S. B. Kang.
Image-based Plant Modeling.
ACM Trans. Graph., 25(3):599–604, July 2006.

Bibliography IV

[9] H. Romeijn, F. Sheth, and C.J. Pettit.
An Evaluative Review of Simulated Dynamic Smart 3D Objects.
ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, 1(4):125–130, 2010.

[10] J. Weber and J. Penn.
Creation and Rendering of Realistic Trees.
In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 119–128, New York, NY, USA, 1995. ACM.

Bibliography V

[11] X. Zhang, F. Blaise, and M. Jaeger.
Multiresolution Plant Models with Complex Organs.
In *Proceedings of the 2006 ACM International Conference on Virtual Reality Continuum and its Applications*, VRCIA '06, pages 331–334, New York, NY, USA, 2006. ACM.

Tree Abstract Data Type (tADT) - Problem?

Example

```
Output Chain from L-System:
C1 [ C2 [ C3 ] [ C4 ] ] [ C5 ] C6
```