**TreeJuxtaposer:
Scalable Tree Comparison
using Focus+Context
with Guaranteed Visibility**

Tamara Munzner
Univ. British Columbia

François Guimbretière
Univ. Maryland College Park

Serdar Taşiran
Koç University

Li Zhang, Yunhong Zhou
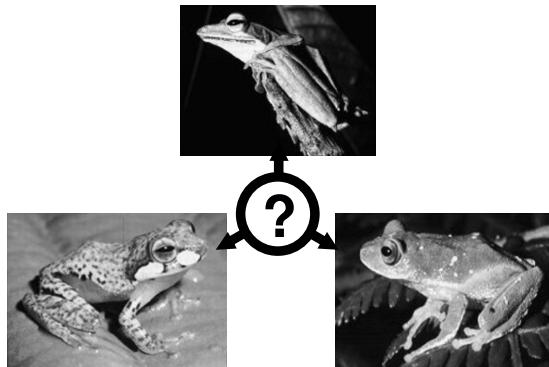Hewlett Packard Systems
Research Center
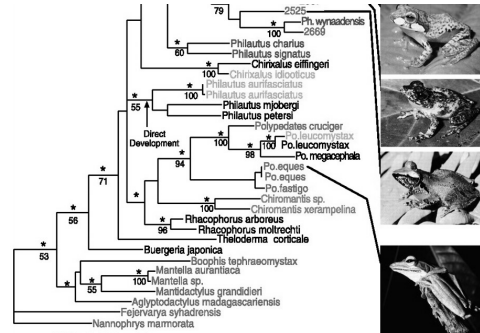
1

Tree comparison

- Active area: hierarchy visualization
  - previous work: browsing
  - comparison still open problem

- Bioinformatics application
  - phylogenetic trees reconstructed from DNA
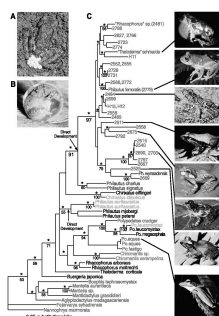
2

Inferring species relationships

Phylogenetic tree

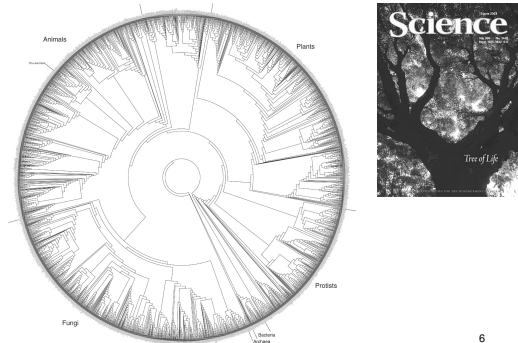M Meegaskumbura et al., Science 298:379 (2002)

4

Phylogenetic tree

M Meegaskumbura et al., Science 298:379 (2002)
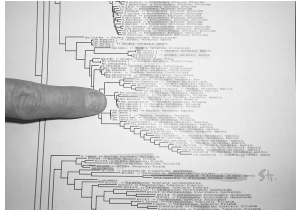
5

Tree of Life: 10M species

David Hillis, Science 300:1687 (2003)

6

Comparing trees: current practice

Will Fischer, postdoc with David Hillis at UT-Austin

7

Biologists' requirements

- Reliable detection of structural differences
  - rapid identification of interesting spots
- Analysis of differences in context
  - mostly side by side comparison
- Manipulation of increasingly larger trees

- Support for multiple platforms

8

TreeJuxtaposer contributions

- Interactive tree comparison system
  - automatic detection of structural differences
    - sub-quadratic preprocessing
  - efficient Focus+Context navigation and layout
    - merge overview and detail in single view
  - guaranteed visibility under extreme distortion

- Scalable
  - dataset size: handles 280K – 500K nodes
  - display size: handles 3800x2400 display

9

TreeJuxtaposer video

- Platforms shown
  - java 1.4, GL4Java 2.7 bindings for OpenGL
  - Windows
    - 2.4 GHz P3, nVidia Quadro4 700XGL
    - 1.1GB java heap
    - window sizes 1280x1024, 3800x2400
  - Linux
    - 3.1 GHz P4, nVidia GeForce FX 5800 Ultra
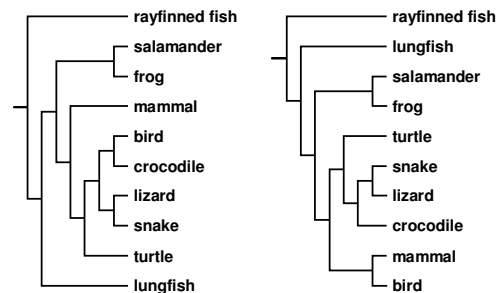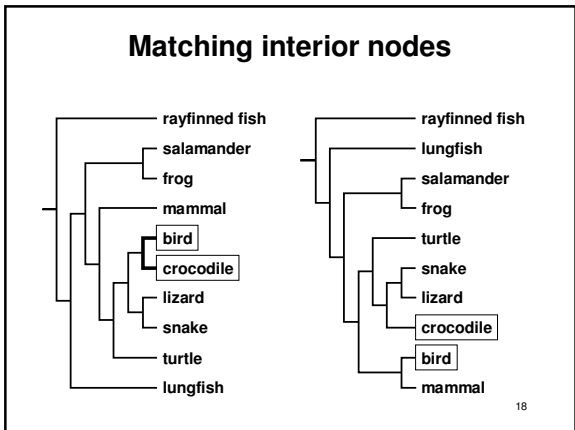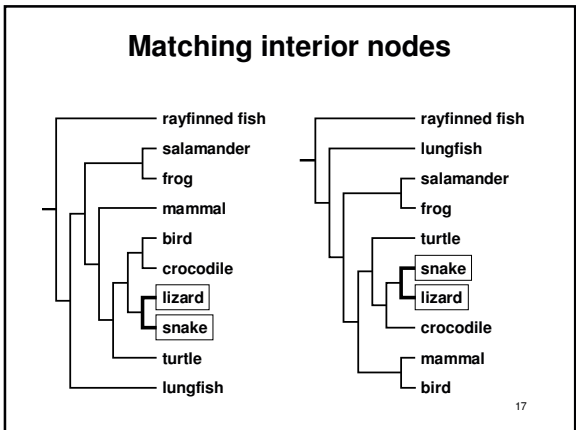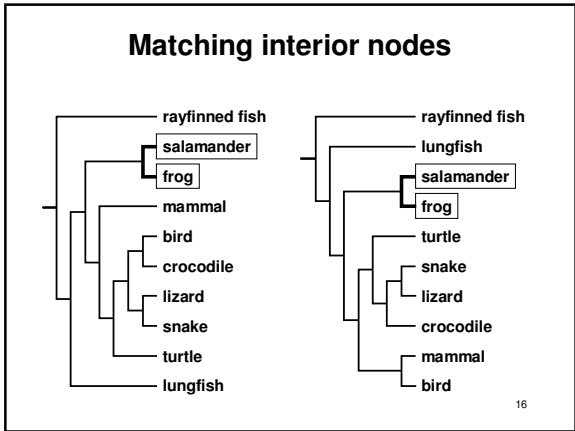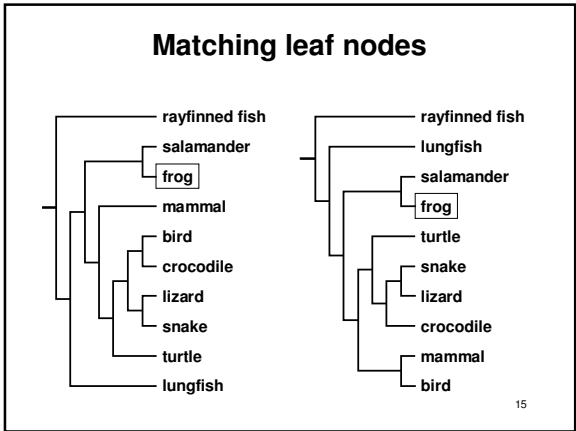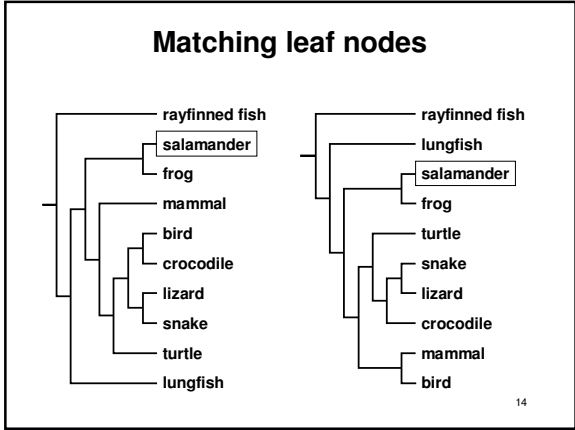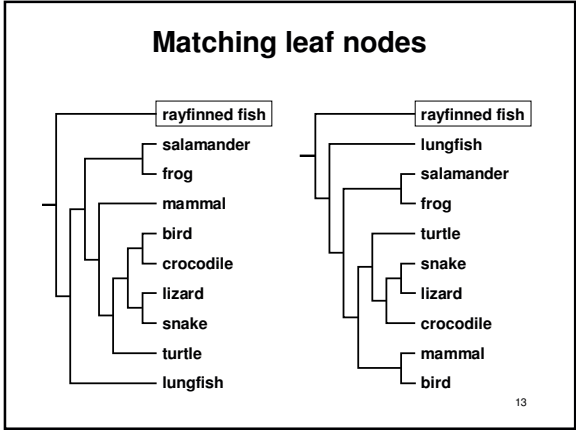   - 1.7GB java heap
    - window size 800x600

10

Outline

- Application domain: evolutionary trees
- Demonstration
- Computing structural differences
- Guaranteed visibility of marked areas
- Results and conclusions

11

Comparing tree

12

2

Matching leaf nodes

rayfinned fish
salamander
frog
mammal
bird
crocodile
lizard
snake
turtle
lungfish

rayfinned fish
lungfish
salamander
frog
turtle
snake
lizard
crocodile
mammal
bird

13

Matching leaf nodes

rayfinned fish
salamander
frog
mammal
bird
crocodile
lizard
snake
turtle
lungfish

rayfinned fish
lungfish
salamander
frog
turtle
snake
lizard
crocodile
mammal
bird

14

Matching leaf nodes

rayfinned fish
salamander
frog
mammal
bird
crocodile
lizard
snake
turtle
lungfish

rayfinned fish
lungfish
salamander
frog
turtle
snake
lizard
crocodile
mammal
bird

15

Matching interior nodes

rayfinned fish
salamander
frog
mammal
bird
crocodile
lizard
snake
turtle
lungfish

rayfinned fish
lungfish
salamander
frog
turtle
snake
lizard
crocodile
mammal
bird

16

Matching interior nodes

rayfinned fish
salamander
frog
mammal
bird
crocodile
lizard
snake
turtle
lungfish

rayfinned fish
lungfish
salamander
frog
turtle
snake
lizard
crocodile
mammal
bird

17

Matching interior nodes

rayfinned fish
salamander
frog
mammal
bird
crocodile
lizard
snake
turtle
lungfish

rayfinned fish
lungfish
salamander
frog
turtle
snake
lizard
crocodile
bird
mammal

18

3

Matching interior nodes

rayfinned fish
salamander
frog
mammal
? bird
crocodile
lizard
snake
turtle
lungfish

rayfinned fish
lungfish
salamander
frog
turtle
snake
lizard
crocodile
mammal
bird

19

Previous work

- Tree comparison
  - RF distance [Robinson and Foulds 81]
  - perfect node matching [Day 85]
  - creation/deletion [Chi and Card 99]
  - leaves only [Graham and Kennedy 01]

20

Similarity score: $S(\mathbf{m},\mathbf{n})$

T_1

A
B
C
D
E
m — F

T_2

A
C
B
D
F
n — E

$L(\text{m}) = \{\text{E},\text{F}\}$ $L(\text{n}) = \{\text{D},\text{E},\text{F}\}$

$$S(\text{m},\text{n}) = \frac{|L(\text{m}) \cap L(\text{n})|}{|L(\text{m}) \cup L(\text{n})|} = \frac{|\{\text{E},\text{F}\}|}{|\{\text{D},\text{E},\text{F}\}|} = \frac{2}{3}$$

21

Best corresponding node

T_1

A
B
C
D
E
m — F

T_2

0 — A
0 — C
0 — B
0 — D
1/2 — F
1/2 — E

0
0
2/6
1/3
2/3

BCN(m) = n

- BCN(m) = argmax$_{v \in T_2} (S(\text{m}, v))$
  - computable in $O(n \log^2 n)$
  - linked highlighting

22

Marking structural differences

T_1

A
B
C
D
E
m — F

T_2

A
C
B
D
F
n — E

- Nodes for which $S(v, \text{BCN}(v)) \neq 1$
  - Matches intuition

23

Outline

- Application domain: evolutionary trees
- Demonstration
- Computing structural differences
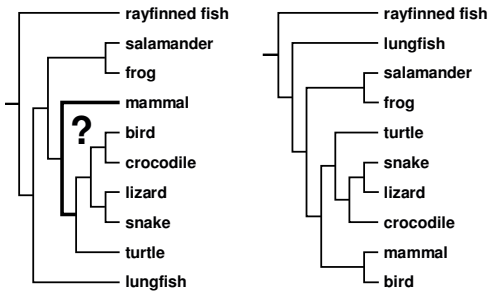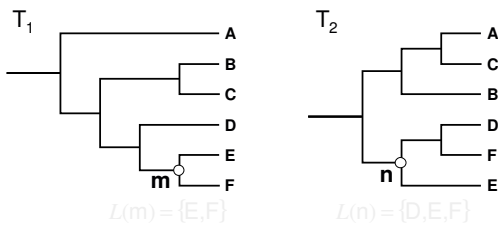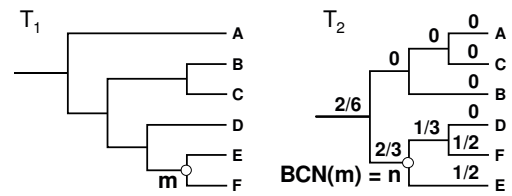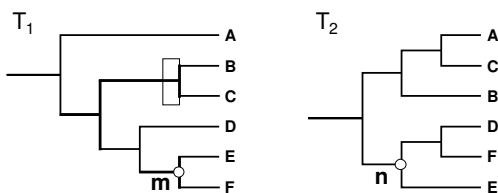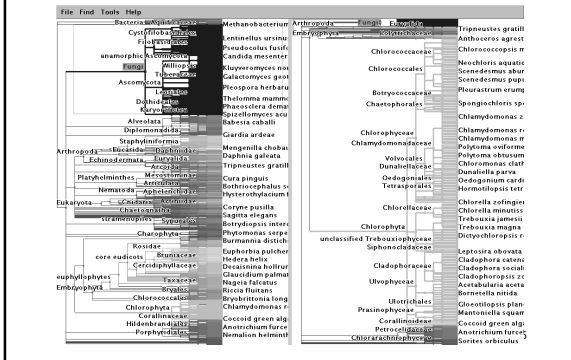- Guaranteed visibility of marked areas
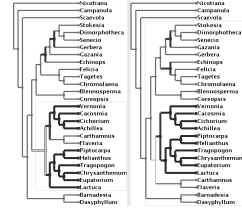- Results and conclusions

24

Guaranteed mark visibility

Marks

- Region of interest shown with color highlight
  - structural difference
  - search results
  - user-specified

- Purpose
  - guide navigation
  - provide landmarks
  - subtree contiguity check

26

Guaranteed visibility of marks

- How can a mark disappear?

27

Guaranteed visibility of marks

- How can a mark disappear?
  - moving outside the frustum

28

Guaranteed visibility of marks

- How can a mark disappear?
  - moving outside the frustum

- Solutions
  - choose global Focus+Context navigation
    - "tacked down" borders

29

Focus+Context previous work

- combine overview and detail into single view
- Focus+Context
  - large tree browsing
    - Cone Trees [Robertson et al 91]
    - Hyperbolic Trees [Lamping et al], H3 [Munzner 97]
    - SpaceTree [Plaisant et al 02]
    - DOI Trees [Card and Nation 02]
  - global
    - Document Lens [Robertson and Mackinlay 93]
    - Rubber Sheets [Sarkar et al 93]
- our contribution
  - scalability, guaranteed visibility

30

Guaranteed visibility of marks

- How can a mark disappear?
  – moving outside the frustum

- Solutions
  – choose global Focus+Context navigation
    • "tacked down" borders

31

Guaranteed visibility of marks

- How can a mark disappear?
  – moving outside the frustum
  – occlusion

- Solutions
  – choose global Focus+Context navigation
    • "tacked down" borders

32

Guaranteed visibility of marks

- How can a mark disappear?
  – moving outside the frustum
  – occlusion

- Solutions
  – choose global Focus+Context navigation
    • "tacked down" borders
  – choose 2D layout

33

Guaranteed visibility of marks

- How can a mark disappear?
  – moving outside the frustum
  – occlusion
  – culling at subpixel sizes

- Solutions
  – choose global Focus+Context navigation
    • "tacked down" borders
  – choose 2D layout

34

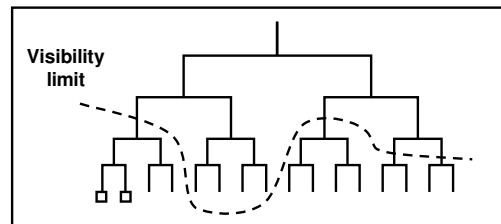**Guaranteed visibility of marks**

- How can a mark disappear?
  – moving outside the frustum
  – occlusion
  – culling at subpixel sizes

- Solutions
  – choose global Focus+Context navigation
    • "tacked down" borders
  – choose 2D layout
  – develop efficient check for marks when culling
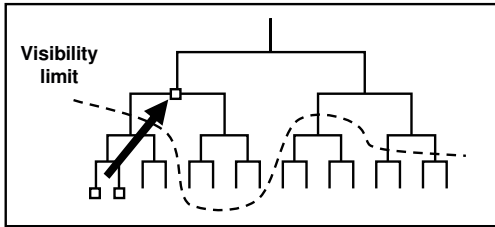
35

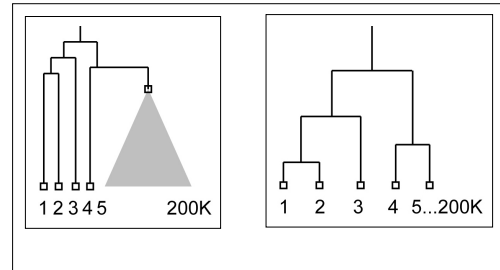**Preserving marks while culling**

- Show mark at unculled node

Visibility limit

36

6

Preserving marks while culling

- Show mark at unculled node

Visibility limit

Mark preservation strategies

- Compress large subtree to small spatial area

1 2 3 4 5   200K   1   2   3   4   5...200K

User selects nodes [135,199995]

- Propagation : cost depends on total nodes
- Precomputation: cost depends visible nodes

[1, 200K]
[1, 3]
[1, 2]
[4, 200K]
[5, 200K]

1 2 3 4   1 2 3 4

5   200K   5   200K
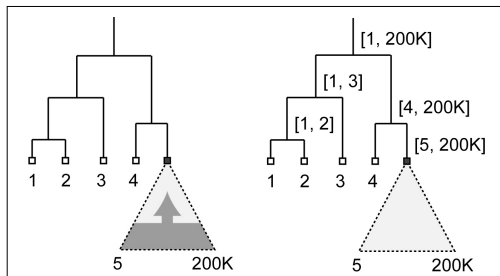
Marks and linked highlighting

- Also check for linked marks from other tree

  – check if best match for node is marked
    - up to $O(n)$ to look up each node in range

  – intersect node ranges between trees
    - reduces to point in polygon test, $O(\log^2 n)$

Storing topological ranges

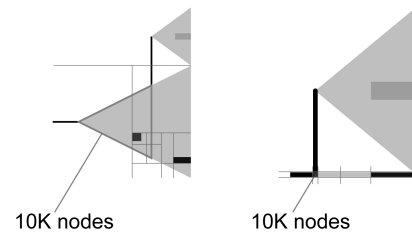- At each node, store range of subtree beneath
  – range stored doesn't match spatial range needed

500X500 pixels

1x1 pixel

10K nodes   10K nodes
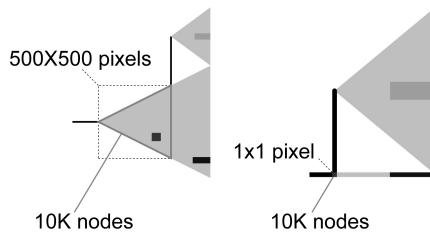
Storing spatial ranges

- At each box, store range of objects inside

10K nodes   10K nodes

Spatial range solution

- Recursive spatial subdivision
  - quadtree
  - store range of objects enclosed for each cell
  - quick check: spatial range vs. selection range

- Extending quadtrees to Focus+Context
  - quadtree cells also "painted on rubber sheet"
  - efficient O(log n) update when stretch/shrink
    - details in paper

43

Guaranteed visibility previous work

- Visibility of abstract information

  - Effective view navigation [Furnas 97]

  - Critical zones [Jul and Furnas 98]

44

Outline

- Application domain: evolutionary trees
- Demonstration
- Computing structural differences
- Guaranteed visibility of marked areas
- Results and conclusions

45

Difference computation

- Powerful and totally automatic
  - leads users to important locations
  - efficient algorithms: 7s for 2x140K nodes
  - matches intuition
    - UT-Austin Biology Lab, several others

- Challenges
  - memory footprint
  - handling weighted edges

46

Guaranteed visibility

- Relief from exhaustive exploration
  - missed marks lead to false conclusions
  - hard to determine completion
  - tedious, error-prone

- Compelling reason for Focus+Context
  - controversy: does distortion help or hurt?
  - strong rationale for comparison

47

Guaranteed visibility challenges

- Integration with progressive rendering
  - might lose context during motion
  - need several seeds for rendering queue
    - focus point
    - marked items
  - up to empirical cutoff, no guarantees
- Constraint to fit everything in frustum
  - instead could show indirectly

48

Future Work

- Adoption
  - open-source release
  - tighter integration with biology tools
  - broad range of application domains
- Detectability vs. visibility
  - display resolution, surrounding colors
- Extend difference computation
  - weighted trees
  - graphs

49

Conclusion

- First interactive tree comparison system
  - automatic structural difference computation
  - guaranteed visibility of marked areas
- Scalable to large datasets
  - 250,000 to 500,000 total nodes
  - all preprocessing subquadratic
  - all realtime rendering sublinear
- Techniques broadly applicable
  - not limited to biological trees

50

Acknowledgments

- Biologists
  - David Hillis, Bob Jensen, Will Fischer, Derrick Zwickl
- Computer scientists
  - Nina Amenta, Katherine St. John
- Partial funding
  - NSF/DEB-0121682

- Talk preparation
  - Mary Czerwinski, Pat Hanrahan, George Robertson, Chris Stolte, Diane Tang, Gina Venolia

51