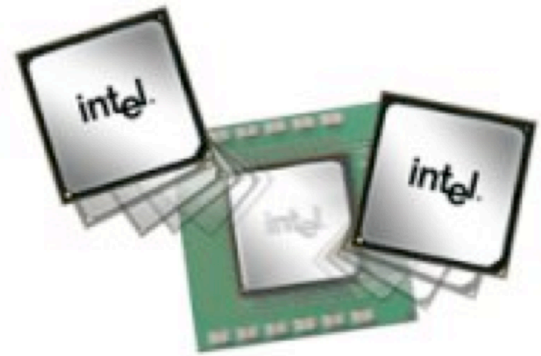
GPGPU: What's Next?

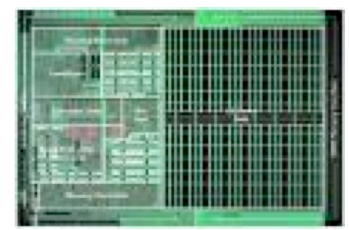
John Owens
Assistant Professor, Electrical and Computer Engineering
Institute for Data Analysis and Visualization
University of California, Davis

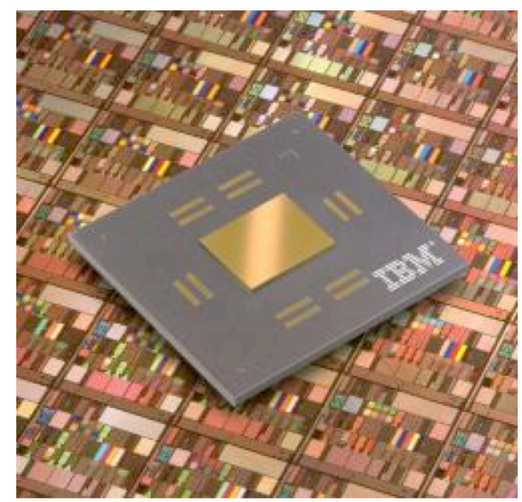
Parallel Processing Has Arrived

- Major vendors supporting multicore
  - Intel, AMD



- Excitement about IBM Cell
- Hardware support for threads
- GPGPU





What is the future of the GPU?

The GPU, Pro and Con

Historical performance increases

Track record of innovation

Established programming model

Market penetration

"How do I get started with GPGPU?"

"Now that I've started, which tools are best?"

Requires *abstractions*:

- Computation
- Data storage/access
- Communication
- Compatibility/ interoperability

The Real Challenge: Programming Systems

Programming Model	High-Level Abstractions/ Libraries	Low-Level Languages	Compilers

Performance Analysis Tools

Docs

CPU	GPU
Scalar	Stream? Data-Parallel?
STL, GNU SL, MPI, …	Brook, sh, Scout, Glift
C, Fortran, …	GLSL, Cg, HLSL, …
gcc, vendor-specific, …	Vendor-specific
gdb, vtune, Purify, …	Shadesmith, NVPerfHUD, GQL
Lots	Little
→ *applications*	→ *kernels*

The Real Challenge: Programming Systems

Programming Model	High-Level Abstractions/ Libraries	Low-Level Languages	Compilers

Performance Analysis Tools

Docs

CPU	GPU
Scalar	Stream? Data-Parallel?
STL, GNU SL, MPI, …	Brook, sh, Scout, Glift
C, Fortran, …	GLSL, Cg, HLSL, …
gcc, vendor-specific, …	Vendor-specific
gdb, vtune, Purify, …	Shade
Lots	Little
→ *applications*	→ *kernels*

Great start … but learn from past and continue to diversify

Glift: Data Structures for GPUs

Virtual representation of memory: N-D array, stack, hash table, queue, …
Abstractions provided by library (STL, Boost, Glift …)
Physical representation of memory: 1D array

Glift: data structures for GPUs

- User programs in virtual domain, Glift translates into physical domain
- Glift provides an abstraction for N-D point-addressable grids
- Encompasses filtered N-D texture lookups, page tables, trees, …
- Iterators separate *algorithms* from *data structures*

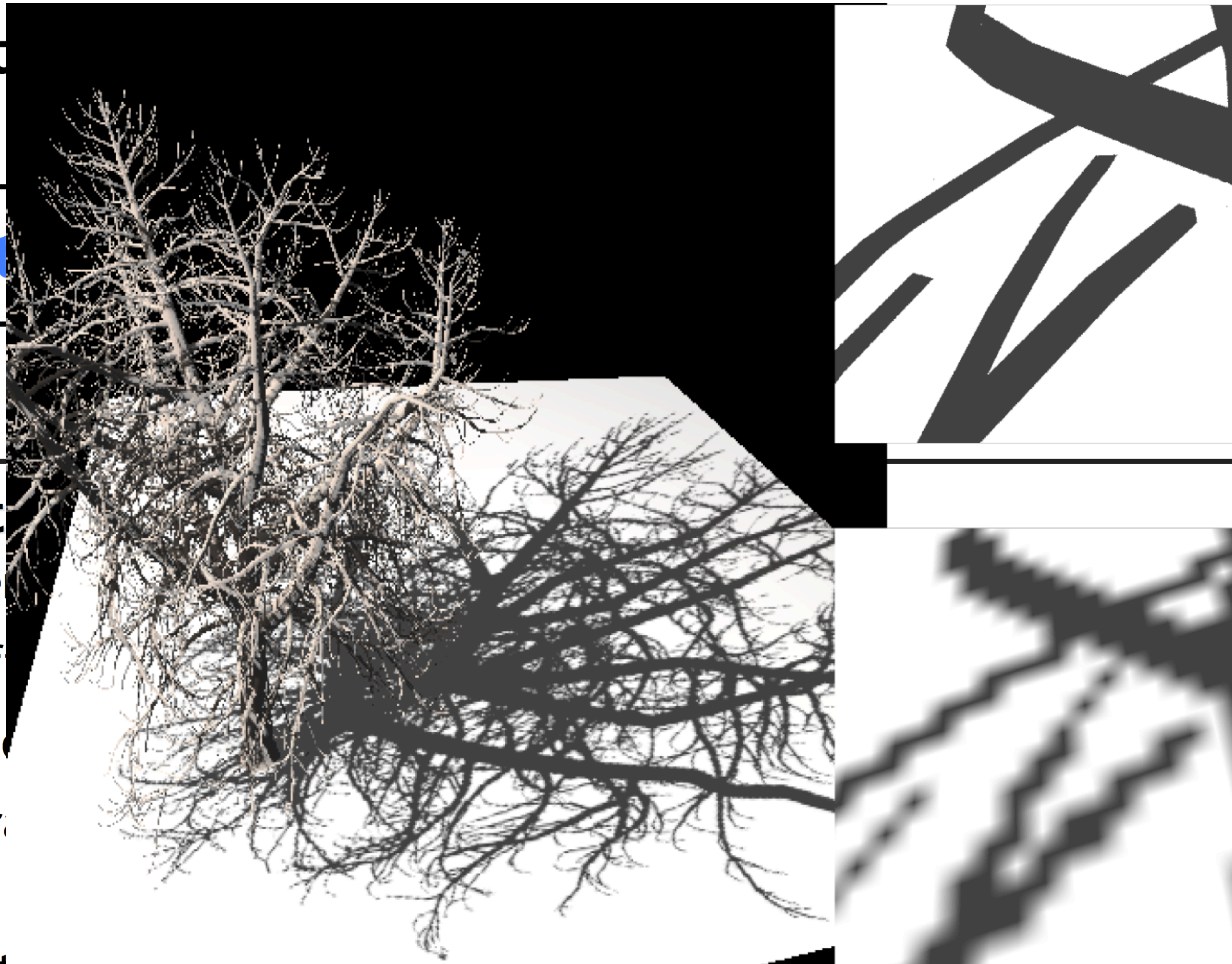
Future:

- Bottom up: space-partitioning structures (kd trees), connectivity, …
- Top down: STL-type structures (lists, sets, associative arra

UCDAVIS

Glift: Data Structures for GPUs

Virt … tack,
Abst … lift …)
… ay



Glift: dat

- Use … omain
- Glif
- Enc … .
- Iter

Future:

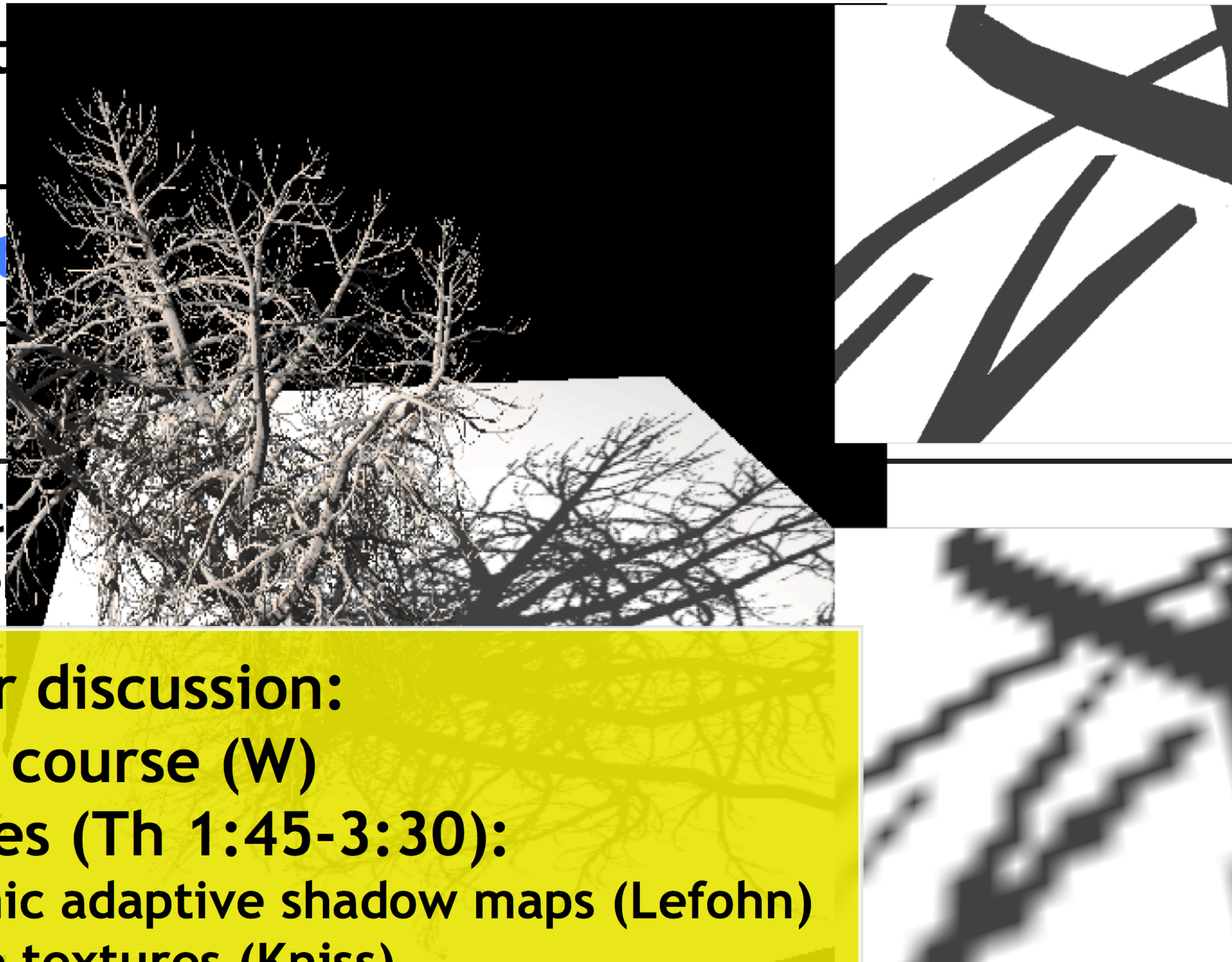
- Bottom up: space-partitioning structures (kd trees), connectivity, …
- Top down: STL-type structures (lists, sets, associative arra

Glift: Data Structures for GPUs

Virt… …tack,
Abst… …lift …)
…ay

Glift: dat…

- Use… …omain



Further discussion:
GPGPU course (W)
Sketches (Th 1:45-3:30):
> Dynamic adaptive shadow maps (Lefohn)
> Octree textures (Kniss)
Upcoming TOG paper (Lefohn et al.)

…trees), connectivity, …

…sociative arra

UC DAVIS

Communicating Beyond a Single GPU

Communication:

- Multi-GPU systems (single CPU or clusters)
- Dynamic partitions of work between CPUs and GPUs

Compatibility and Interoperability

- Between driver versions
- Between different cards, same vendor
- Between vendors
- Between GPU, CPU, new processors …

Rob Pike on Languages

Conclusion

A highly parallel language used by non-experts.

Power of notation

Good:

make it easier to express yourself

Better:

hide stuff you don't care about

Best:

hide stuff you do care about

Give the language a purpose.

Rob Pike on Languages

Conclusion

A highly parallel language used by non-experts.

Power of notation

Good:

make it easier to express yourself

Better:

hide stuff you don't care about

Best:

hide stuff you do care about

Give the language a purpose.

Exposing Parallelism

Control Flow

Data Locality

Synchronization