
Game Developers Conference®
February 28 - March 4, 2011
Moscone Center, San Francisco
www.GDConf.com
GDC
UBM

An Optimized Diffusion Depth Of Field Solver (DDOF)

Holger Gruen – AMD

Agenda

- Motivation
- Recap of a high-level explanation of DDOF
- Recap of earlier DDOF solvers
- A Vanilla Cyclic Reduction(CR) DDOF solver
- A DX11 optimized CR solver for DDOF
- Results

Motivation

- Solver presented at GDC 2010 [RS2010] has some weaknesses
- Great implementation but memory reqs and runtime too high for many game developers
- Looking for faster and memory efficient solver

Diffusion DOF recap 1

- DDOF is an enhanced way of blurring a picture taking an arbitrary CoC at a pixel into account
- Interprets input image as a heat distribution
- Uses the CoC at a pixel to derive a per pixel heat conductivity

CoC=Circle of Confusion

Diffusion DOF recap 2

- Blurring is done by time stepping a differential equation that models the diffusion of heat
- ADI method used to arrive at a separable solution for stepping
- Need to solve tri-diagonal linear system for each row and then each colum of the input

DDOF Tri-diagonal system

- row/col of input image
- derived from CoC at each pixel of an input row/col
- resulting blurred row/col

$$\begin{pmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & c_3 & \\ & & \ddots & \ddots & \ddots \\ 0 & & & a_n & b_n \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix}$$

Solver recap 1

- The GDC2010 solver [RS2010] is a ‚hybrid‘ solver
  - Performs three PCR steps upfront
  - Performs serial ‚Sweep‘ algorithm to solve small resulting systems
  - Check [ZCO2010] for details on other hybrid solvers

Solver recap 2

- The GDC2010 solver [RS2010] has drawbacks
  - It uses a large UAV as a RW scratch-pad to store the modified coefficients of the sweep algorithm
    - GPUs without RW cache will suffer
  - For high resolutions three PCR steps produce tri-diagonal system of substantial size
    - This means a serial (sweep) algorithm is run on a ‚big‘ system

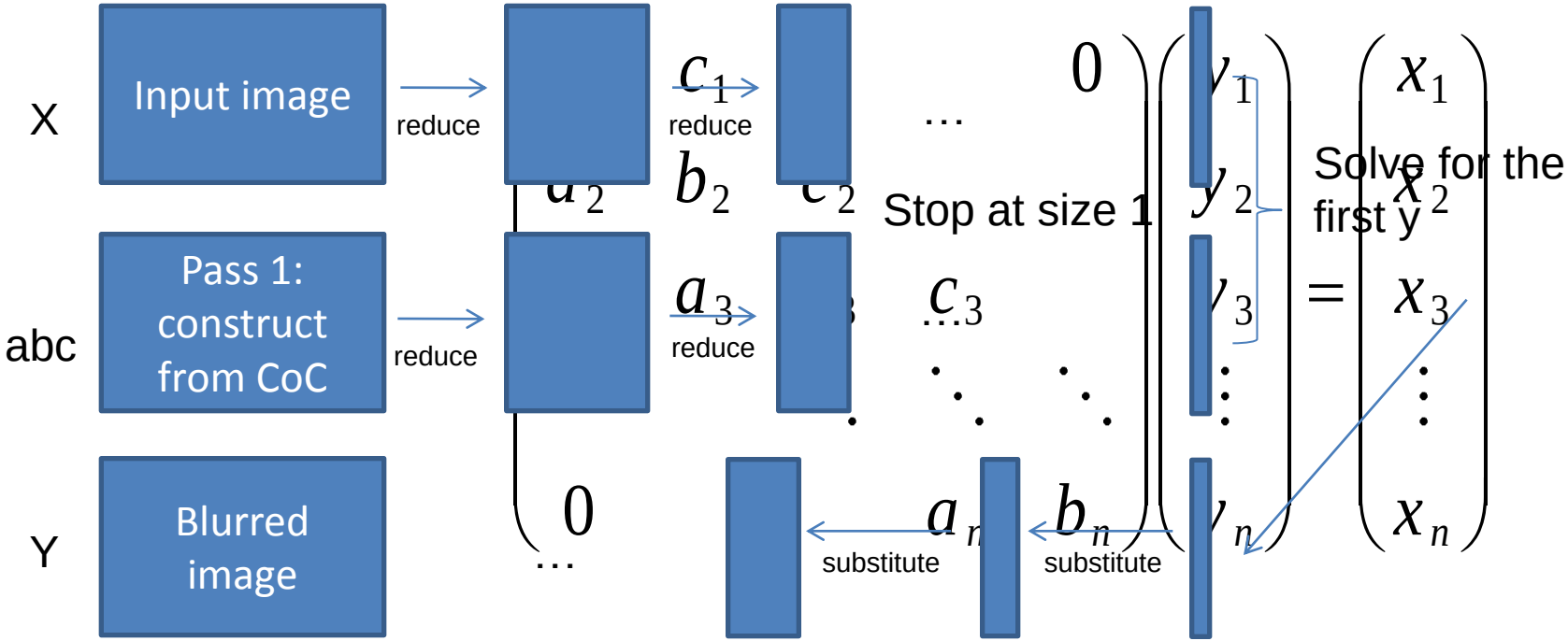
Solver recap 3

- Cyclic Reduction (CR) solver
  - Used by [Kass2006] in the original DDOF paper
  - Runs in two phases
    1. reduction phase
    2. backward substitution phase

Solver recap 4

- According to [ZCO2010]:
  - CR solver has lowest computational complexity of all solvers ☺
  - It suffers from lack of parallelism though ☹
    - At the end of the reduction phase
    - At the start of the backwards substitution phase

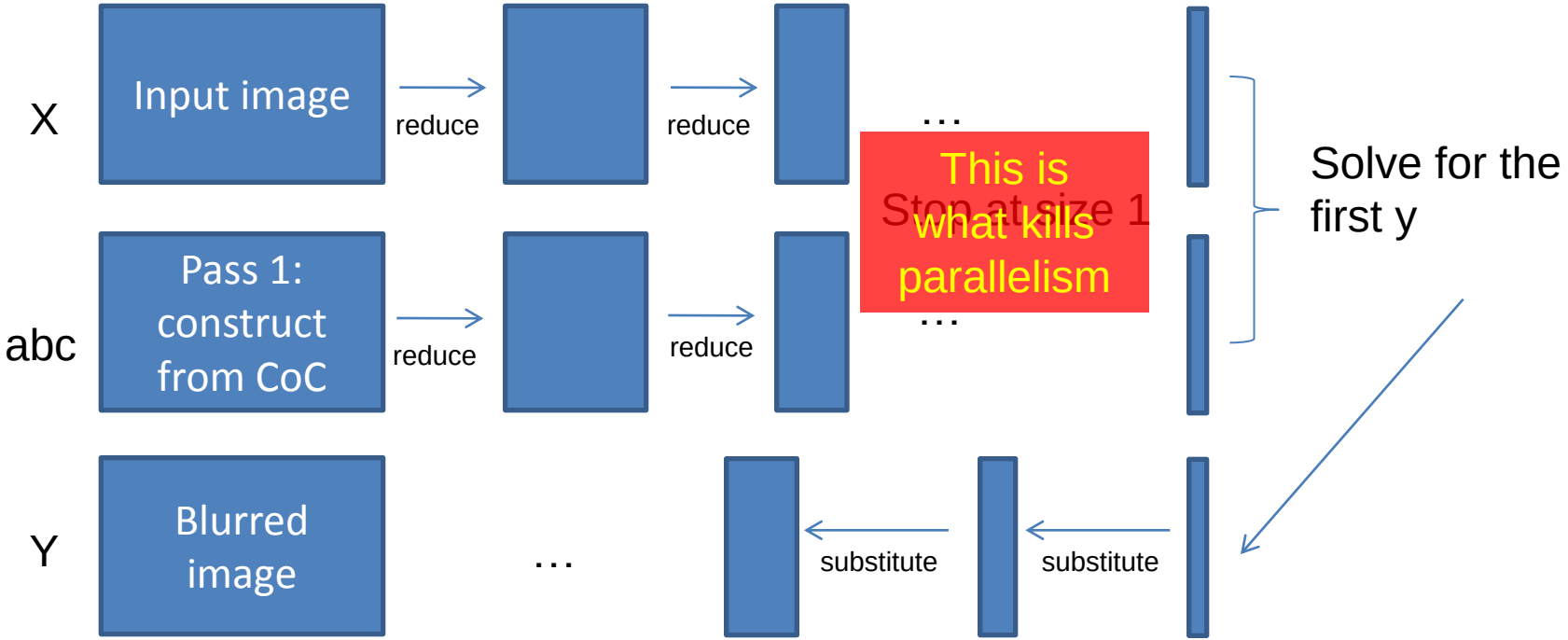
Passes of a Vanilla CR Solver



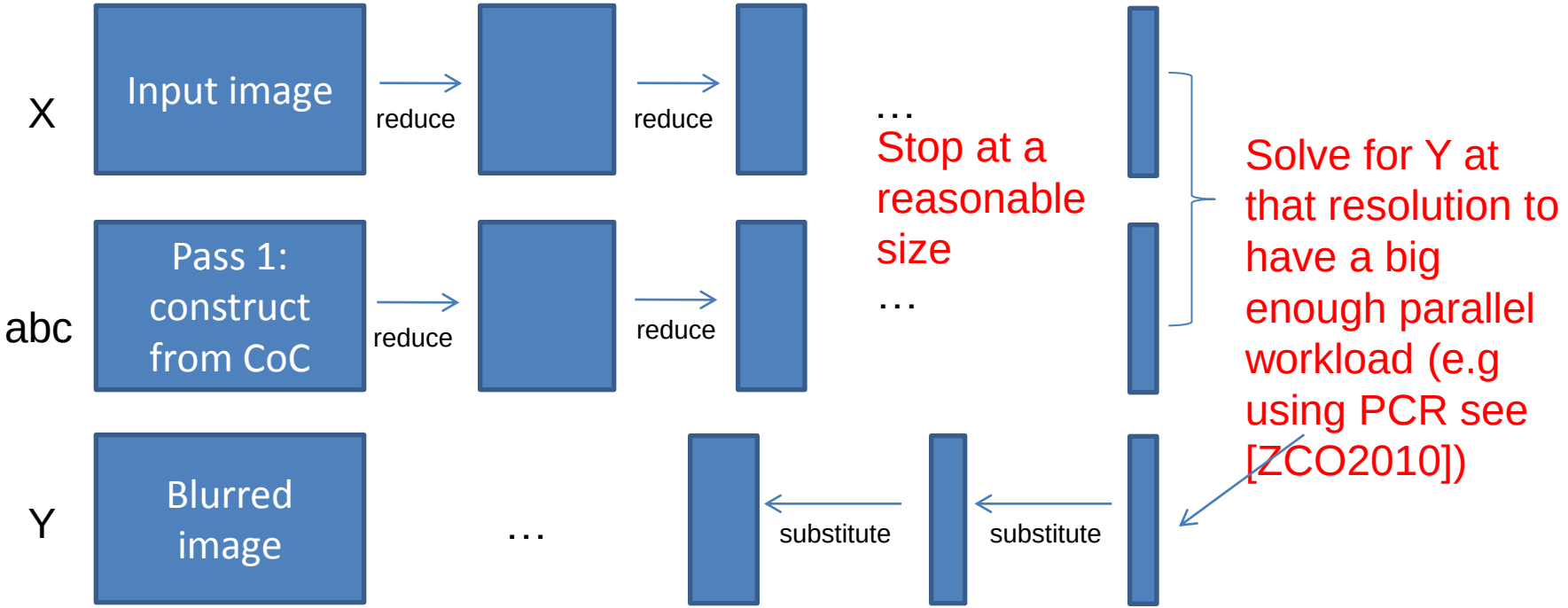
Vanilla Solver Results

- Higher performance than reported in [Bavoil2010] ☺ (~6 ms vs. ~8ms at 1600x1200)
- Memory footprint prohibitively high ☹
  - >200 MB at 1600x1200
- Need an answer to tackling the lack of parallelism problem – answer given in [ZCO2010]

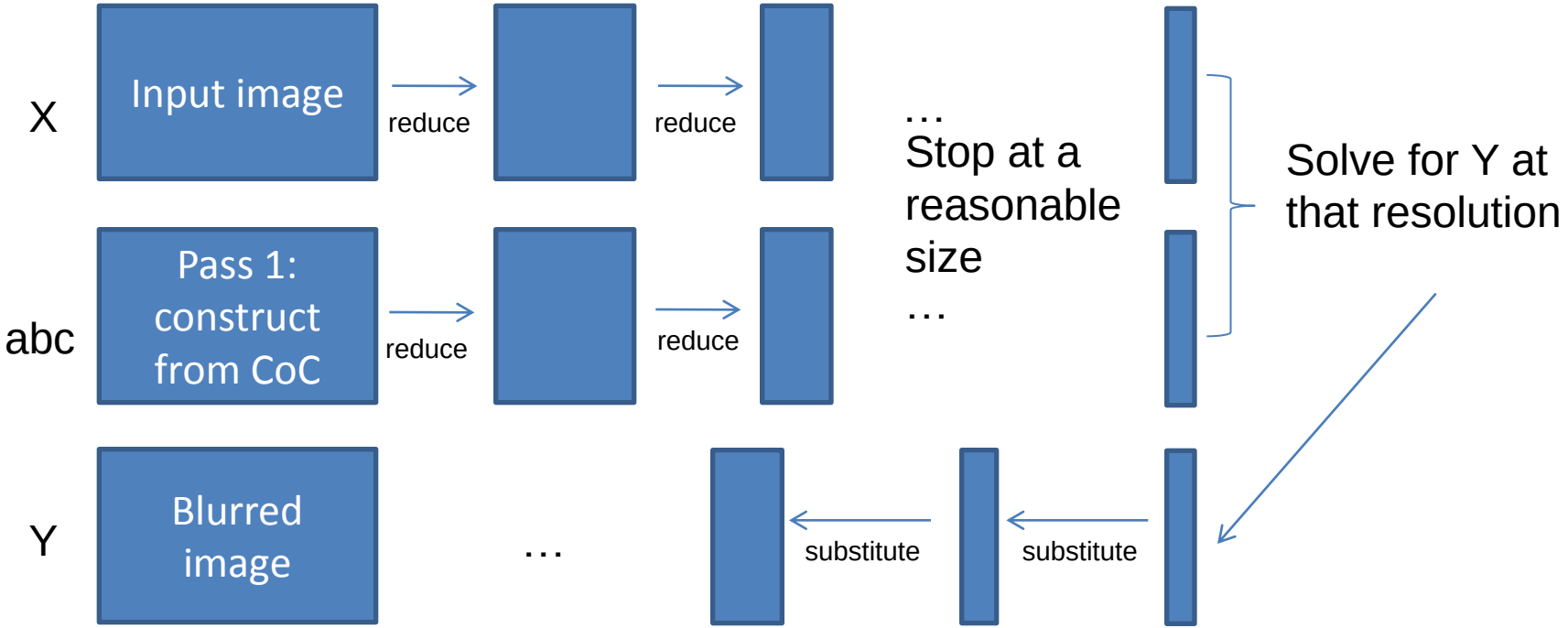
Vanilla CR Solver



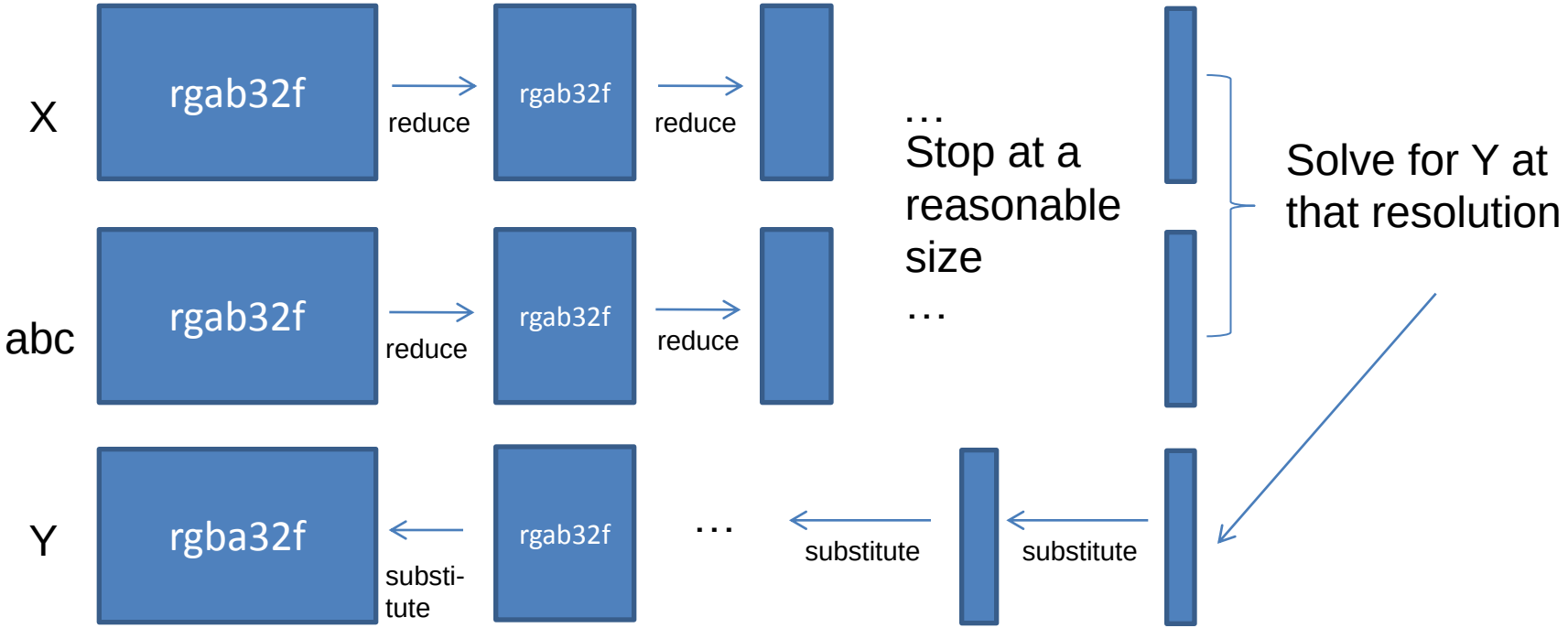
Keeping the parallelism high



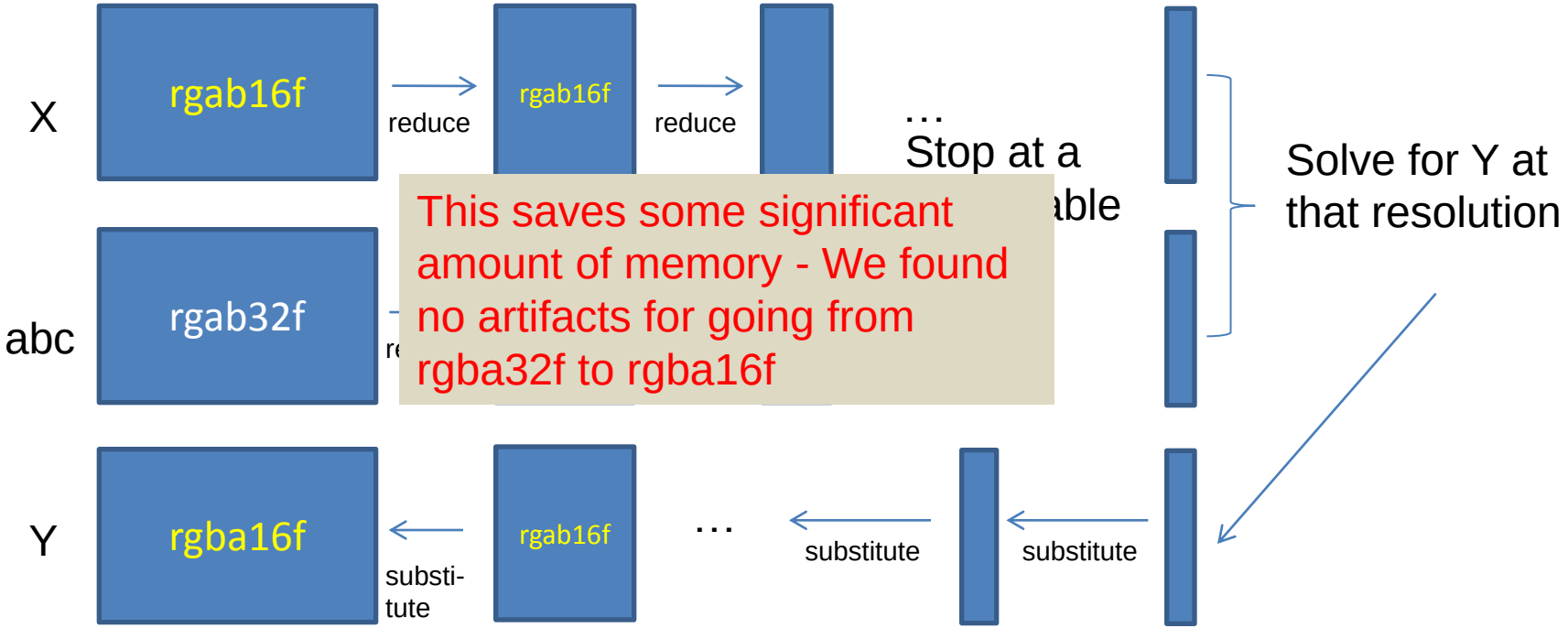
Memory Optimizations 1



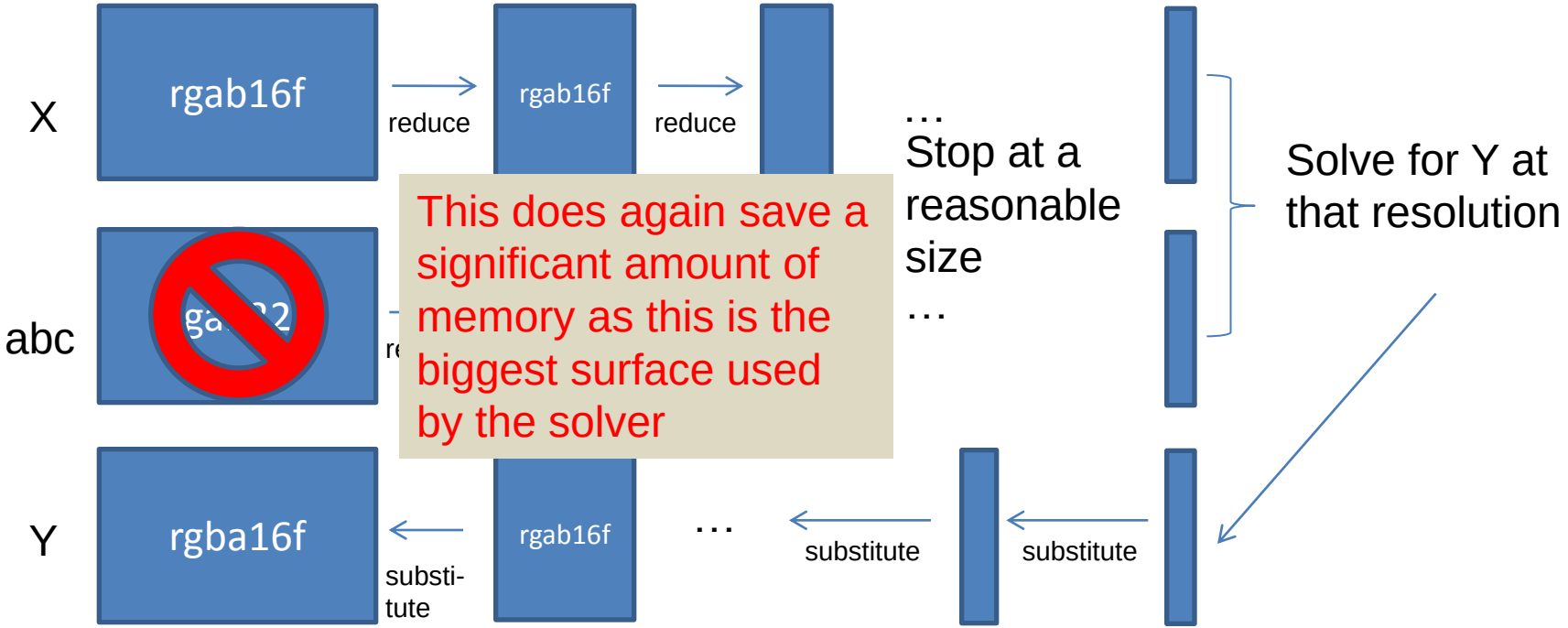
Memory Optimizations 1



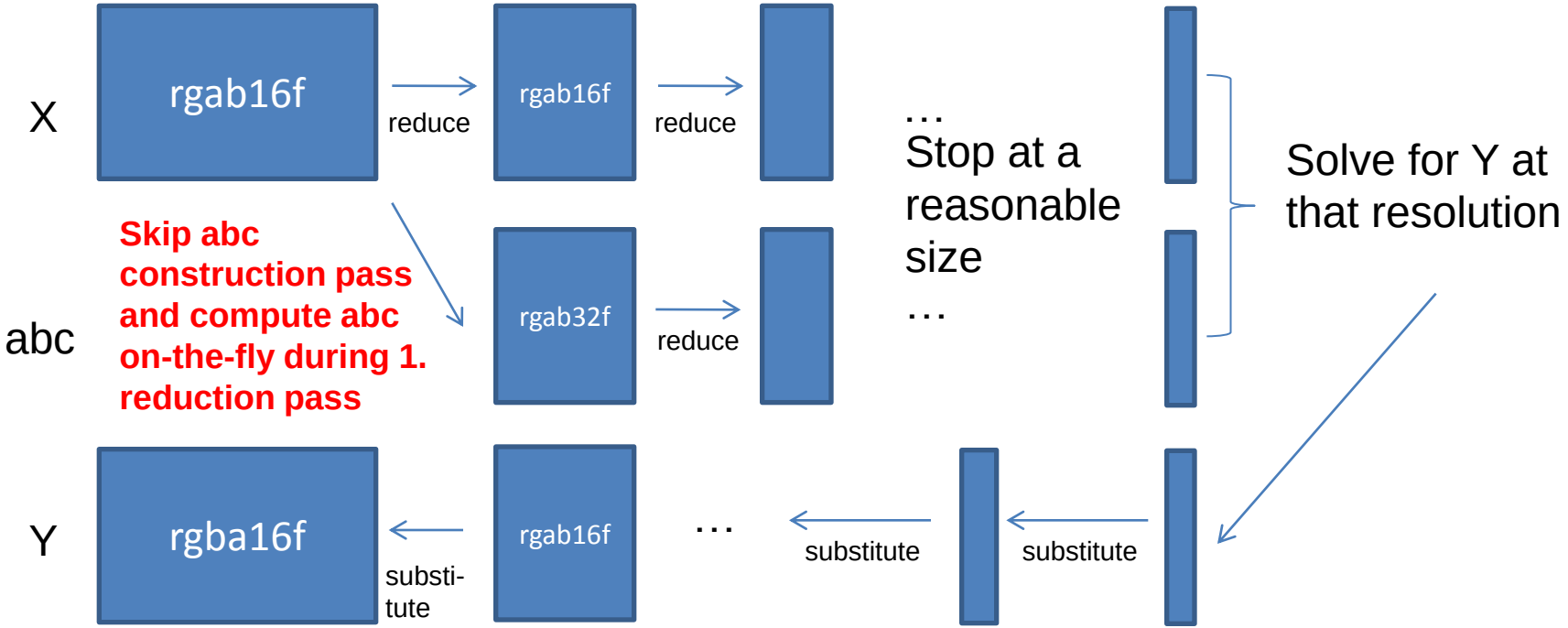
Memory Optimizations 1



Memory Optimizations 2



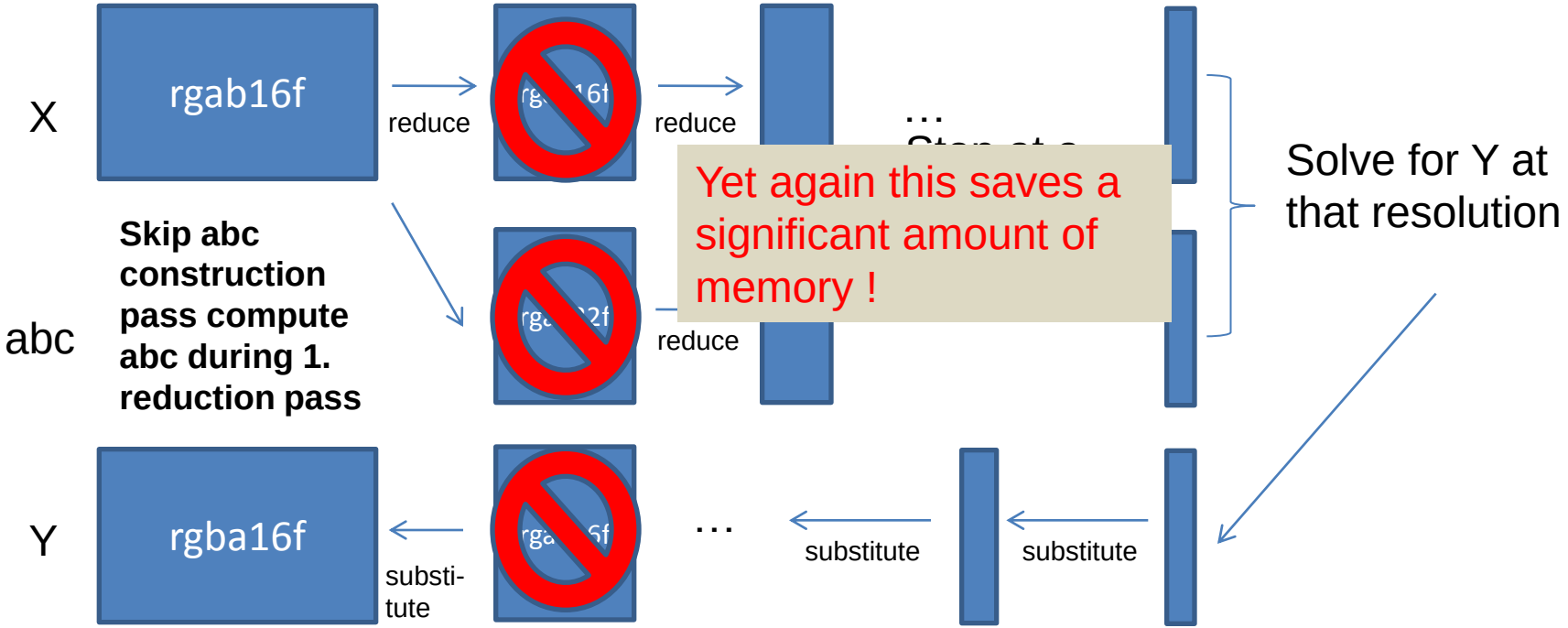
Memory Optimizations 2



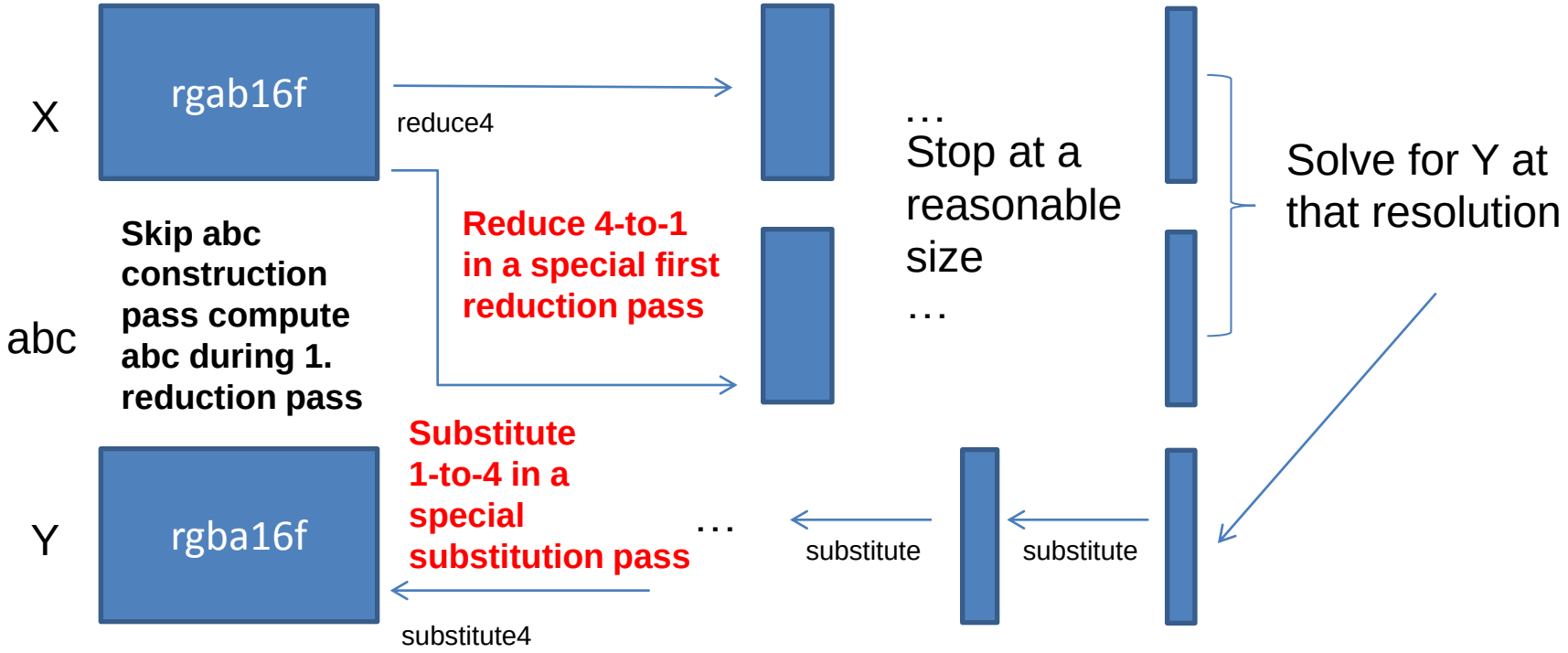
Intermediate Results 1600x1200

Solver	Time in ms		Memory in Megabytes
	HD5870	GTX480	
GDC2010 hybrid solver on GTX480	~8.5	8.00 [Bavoil 2010]	~117 (guesstimate)
Standard Solver (already skips high res abc construction)	3.66	3.33	~132

Memory Optimizations 3



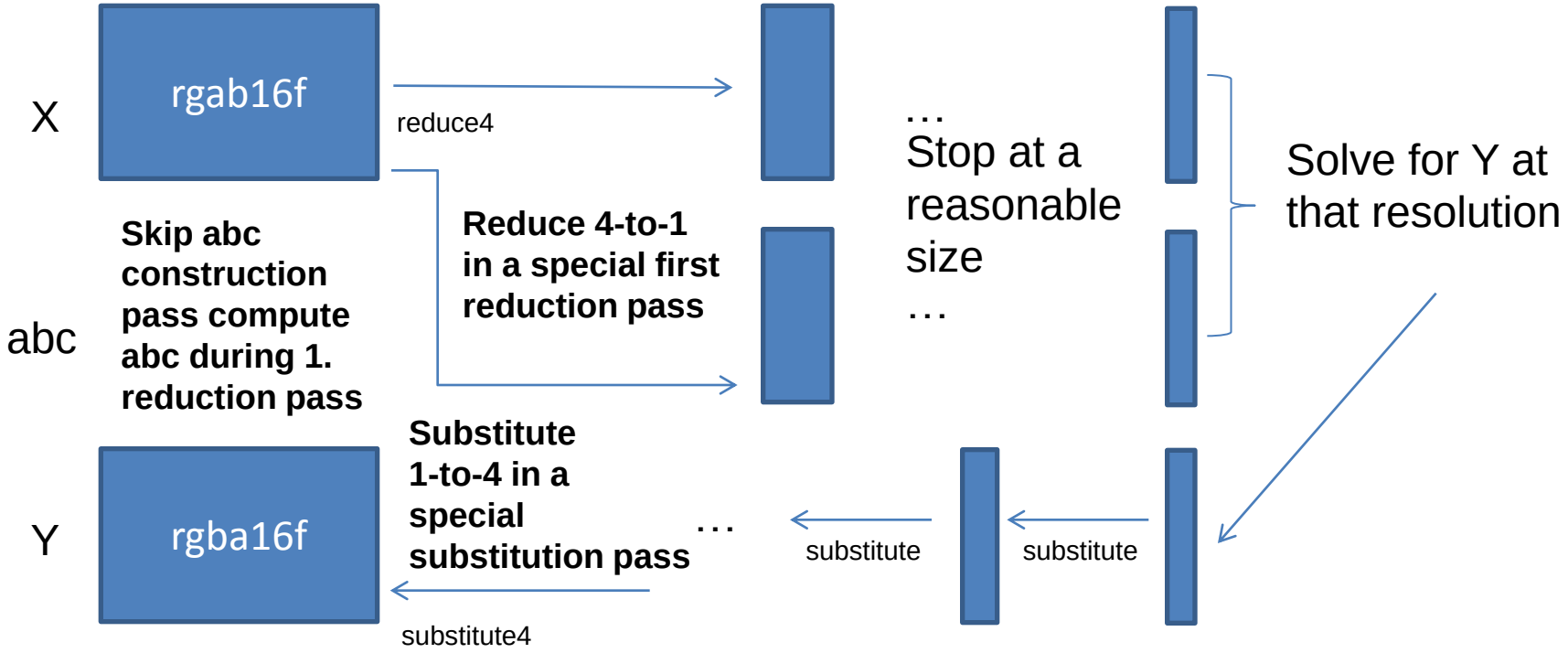
Memory Optimizations 3



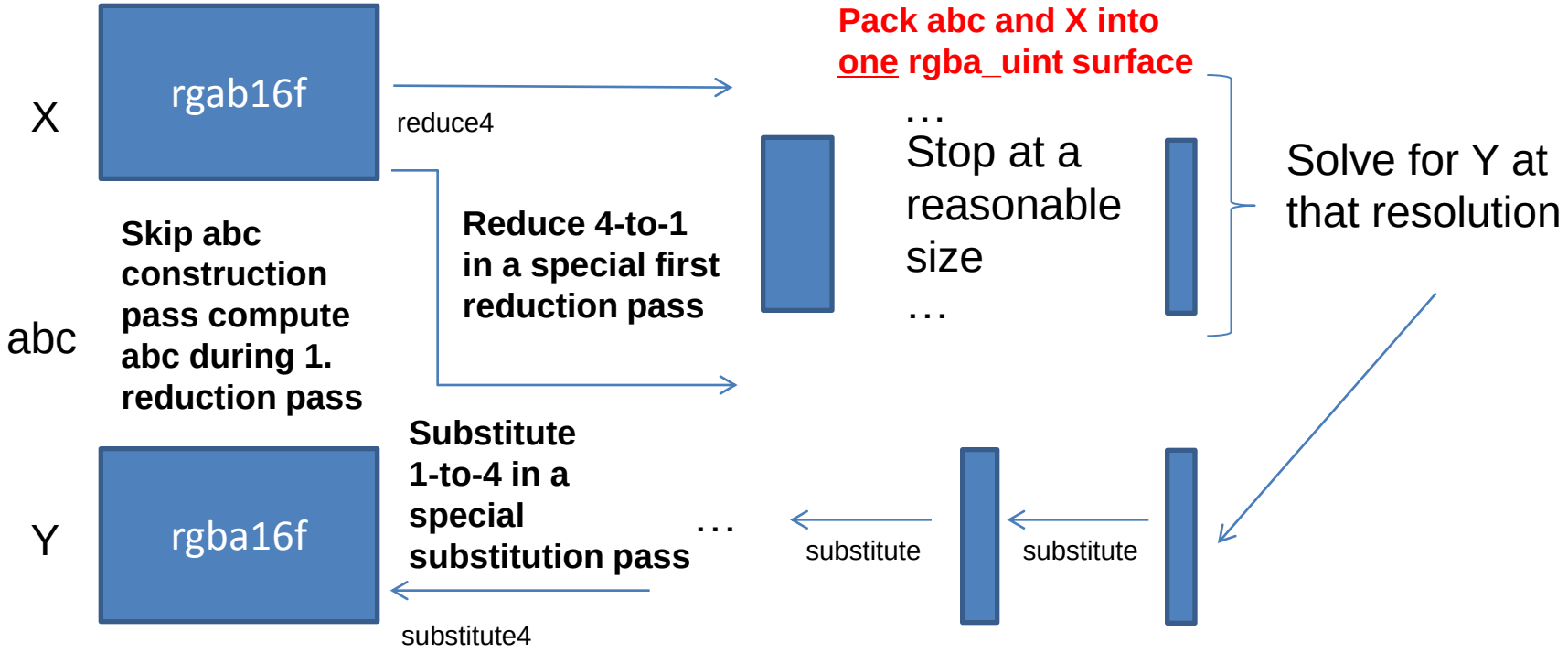
Intermediate Results 1600x1200

Solver	Time in ms		Memory in Megabytes
	HD5870	GTX480	
GDC2010 hybrid solver on GTX480	~8.5	8.00 [Bavoil 2010]	~117 (guesstimate)
Standard Solver (already skips high res abc construction)	3.66	3.33	~132
4–to-1 Reduction	2.87	3.32	~73

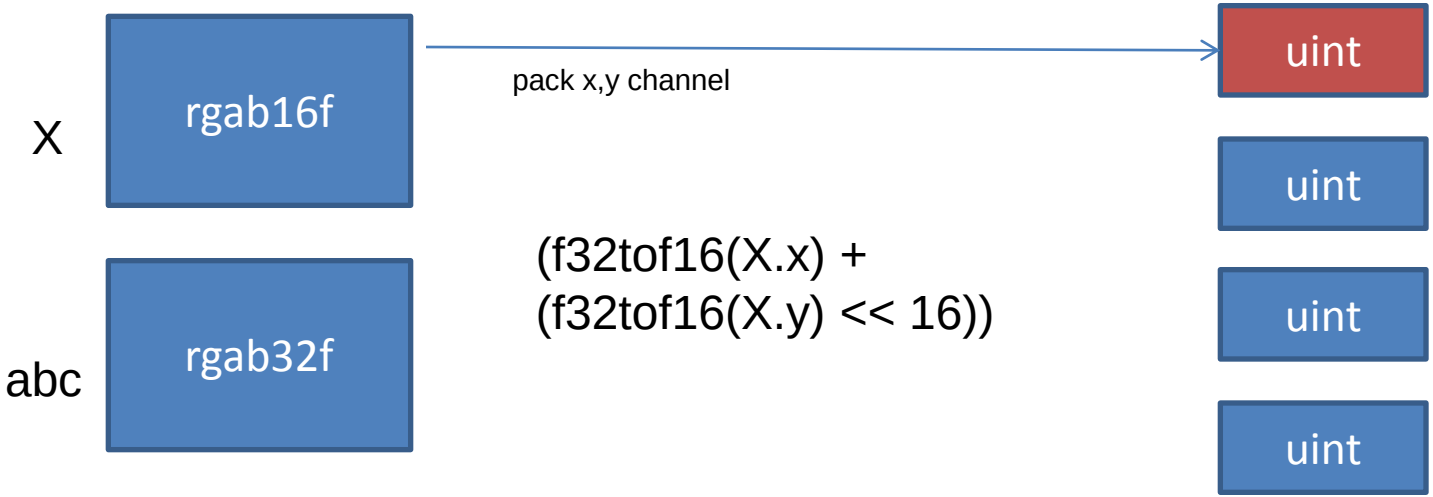
DX11 Memory Optimizations 1



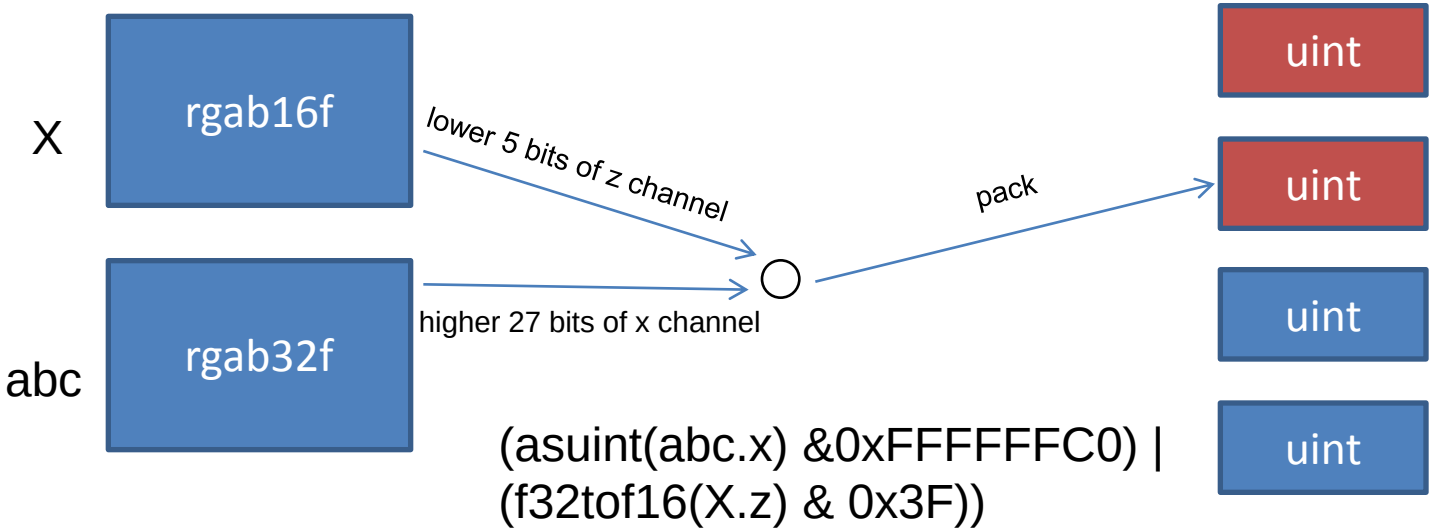
DX11 Memory Optimizations 1



Using SM5 for data packing

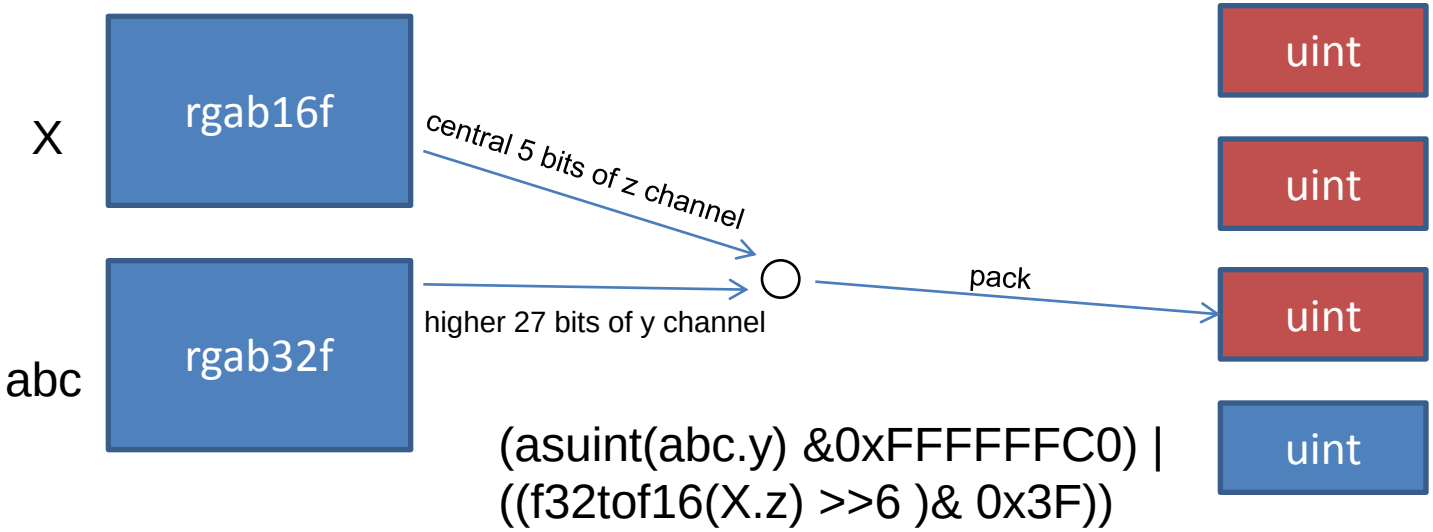


Using SM5 for data packing



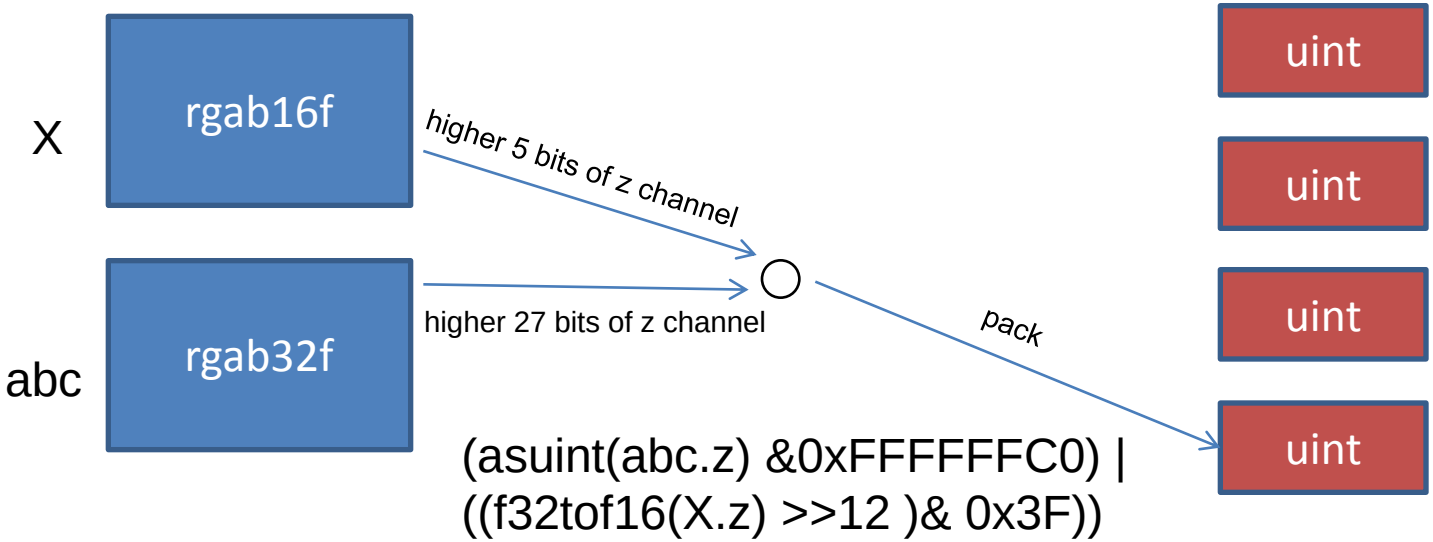
Steal 6 lowest mantissa bits of abc.x to store some bits of X.z

Using SM5 for data packing



Steal 6 lowest mantissa bits of abc.y to store some bits of X.z

SM5 Memory Optimizations 1



Steal 6 lowest mantissa bits of abc.z to store some bits of X.z

Sample Screenshot

Abs(Packed-Unpacked) x 255.0f

DX11 Memory Optimizations 2

- Solver does a horizonal and vertical pass
- Chain of lower res RTs needs to be there twice
  - Horizontal reduction/substitution chain
  - Vertical reduction/substitution chain
- How can DX11 help?

DX11 Memory Optimizations 2

- UAVs allow us to reuse data of the horizontal chain for the vertical chain
  - A proof of concept implementation shows that this works nicely but impacts the runtime significantly
    - ~40% lower fps
  - Stayed with RTs as memory was already quite low
  - Use only if you are really concerned about memory

Final Results 1600x1200

Solver	Time in ms		Memory in Megabytes
	HD5870	GTX480	
GDC2010 hybrid solver on GTX480	~8.5	8.00 [Bavoil 2010]	~117 (guesstimate,)
Standard Solver (already skips high res abc construction)	3.66	3.33	~132
4–to-1 Reduction	2.87	3.32	~73
4-to-1 Reduction + SM5 Packing	2.75	3.14	~58

Future Work

- Look into CS acceleration of the solver
  - 4-to-1 reduction pass
  - 1-to-4 substitution pass
- Look into using heat diffusion for other effects
  - e.g. Motion blur

Conclusion

- Optimized CR solver is fast and mem-efficient
  - Used in Dragon Age 2
  - 4aGames considering its use for new projects
  - Detailed description in ‚Game Engine Gems 2‘
- Mail me (holger.gruen@amd.com) if you want access to the sources

References

- [Kass2006] “Interactive depth of field using simulated diffusion on a GPU” Michael Kass, Pixar Animation studios, Pixar technical memo #06-01
- [ZCO2010] “Fast Tridiagonal Solvers on the GPU” Y. Zhang, J. Cohen, J. D. Owens, PPoPP 2010
- [RS2010] “DX11 Effects in Metro 2033: The Last Refuge” A. Rege, O. Shishkovtsov, GDC 2010
- [Bavoil2010] „Modern Real-Time Rendering Techniques“, L. Bavoil, FGO2010

Backup

Results 1920x1200

Solver	Time in ms		Memory in Megabytes
	HD5870	GTX480	
Standard Solver (already skips high res abc construction)	4.31	4.03	~158
4–to-1 Reduction	3.36	4.02	~88
4-to-1 Reduction + SM5 Packing	3.23	3.79	~70