SIEMENS

The Texture Unit as Performance Booster in CT-Reconstruction

Dr.-Ing. Karl Schwarz

Siemens AG

Healthcare

Computed Tomography

H IM CR R&D SE

Forchheim

Germany

Computed Tomography

CT milestones:

from 1895

- Discovery of X-rays
- Radon transformation
- Axial CT
- Spiral CT
- Multislice CT
- 3D-Backprojection
- Dual Tube
- Iterative Reconstruction

until today

Somatom Definition Flash

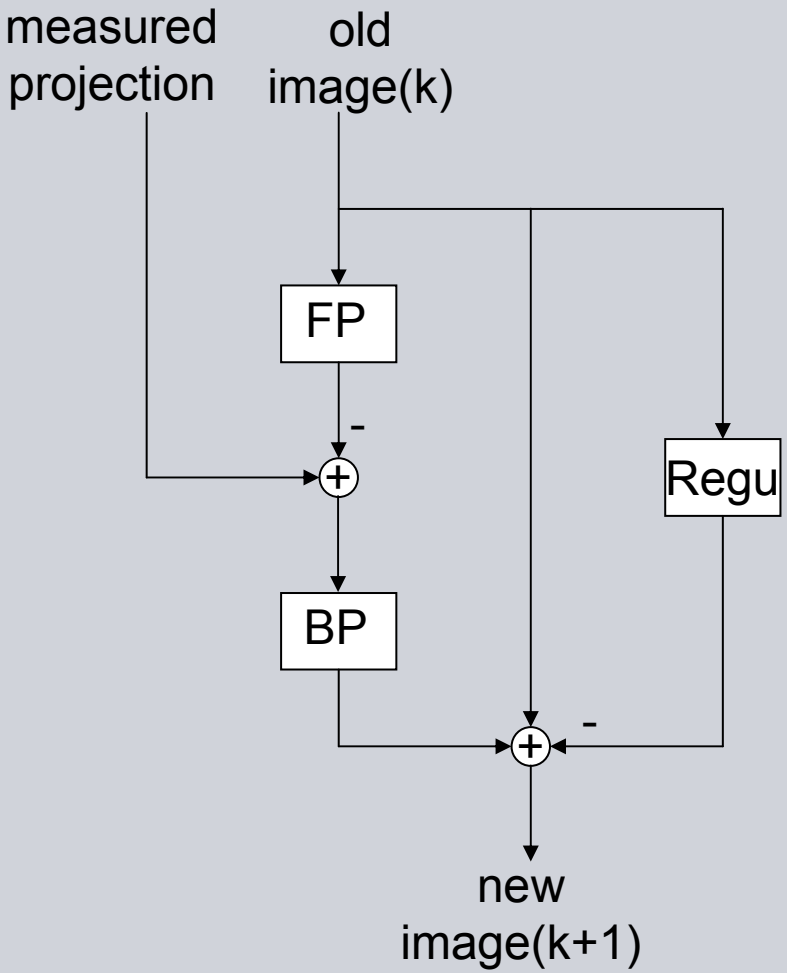
SIEMENS

The Current Challenge in CT: Iterative Reconstruction

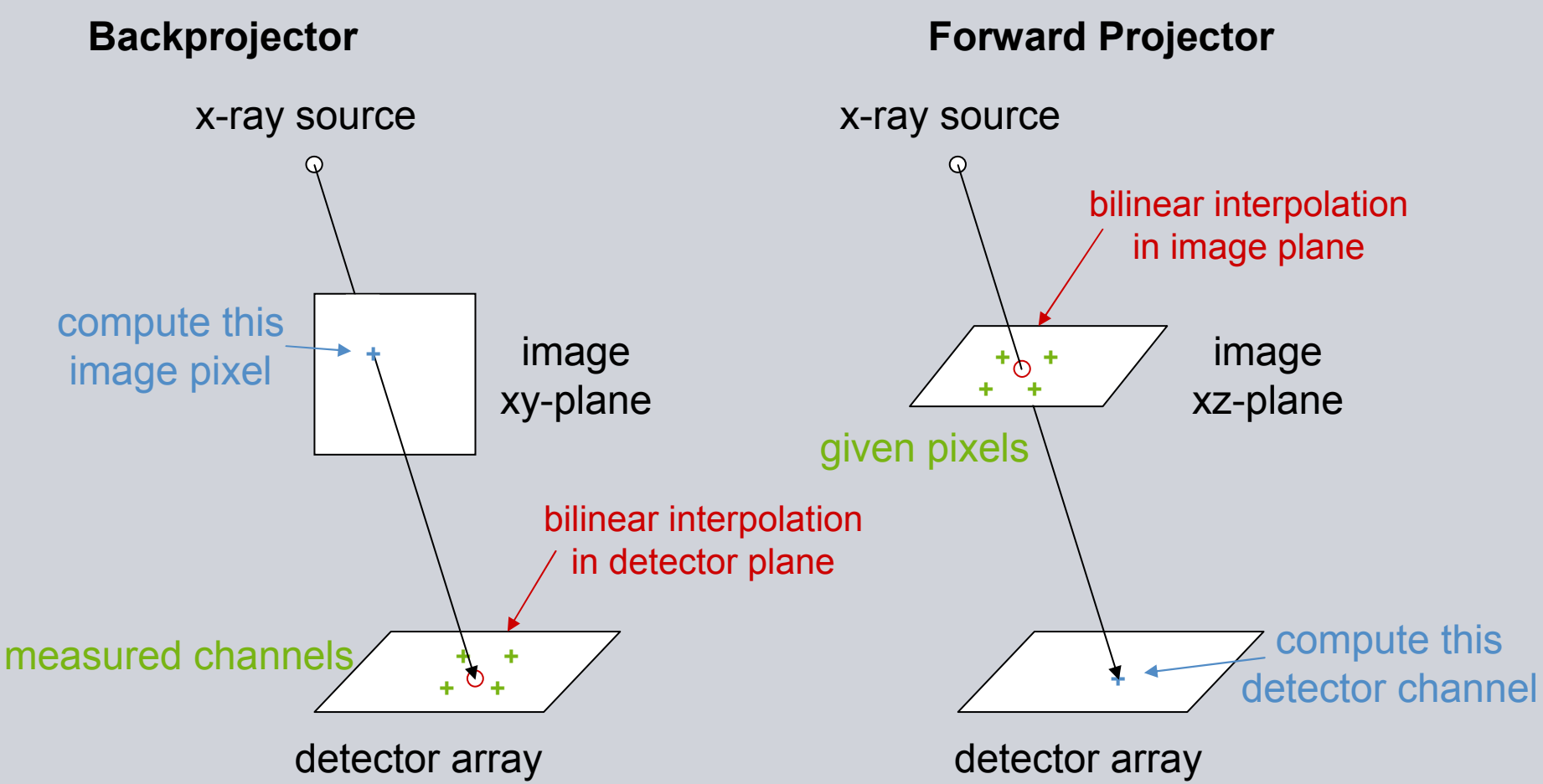
Main Building Blocks in Iterative Reconstruction :

- Backprojector (BP)
  - computes the image
- Forward Projector (FP)
  - simulates the measurement
- Regularization Filter (Regu)
  - noise reduction

One Iteration Step



Bilinear Interpolation



SIEMENS

Subtitle :

“With a Little Help from” [1]

the Texture Unit

[1] Beatles: 2nd title on album “Sgt. Pepper’s Lonely Hearts Club Band”, Abbey Road Studios, London, 1967

SIEMENS

Texture Features

linear interpolation :	**tex1d**(vector, x)	= 1 clock cycle
bilinear interpolation :	**tex2d**(matrix, x, y)	= 1 clock cycle
trilinear interpolation :	**tex3d**(volume, x, y, z)	= 3 clock cycles

- nearest neighbour access or interpolation
- border handling (clamp or wrap)
- format conversion (short to float, half to float)
- separate texture cache

SIEMENS

Peak Performance Comparison

	Tesla C1060	**Tesla C2050**	speedup %
Shader	240 x 1.296 GHz	448 x 1.15 GHz	
GCycles/sec	311	515	+ 65
GFlops*	622	1030	
Texture	80 x 0.61 GHz	56 x 0.575 GHz	
GTexel/sec	48.8	32.2	- 34
GOps**	976	644	
Sum GOps	1598	1674	+ 5
Texture/Shader Cycle Ratio	1 / 6.4	1 / 16	- 60

Comparison of theoretical peak performance

* one fused multiply add per shader clock cycle

** 1 Texel = 20 operations

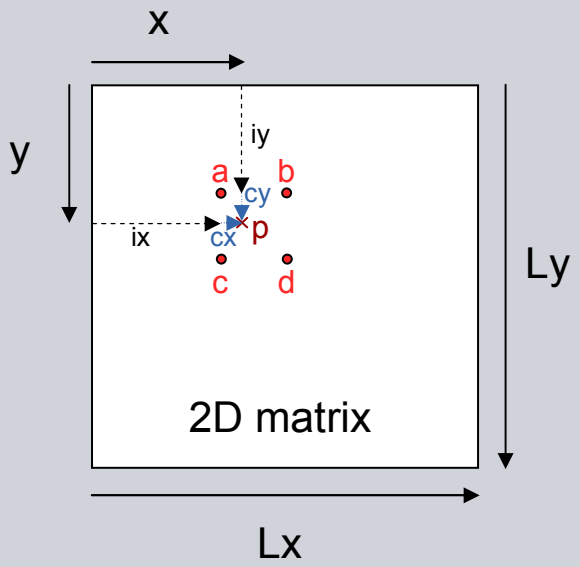
SIEMENS

Bilinear Interpolation

CUDA:

p = tex2D(matrix,x+0.5f,y+0.5f);

Conventional C:	operations	moves
ix = (int) x;	1	
cx = x - (float)ix;	2	
iy = (int) y;	1	
cy = y - (float)iy;	2	
addr = iy * Lx + ix;	2	
a = matrix[addr];		1
b = matrix[addr + 1];	1	1
addr = addr + Lx;	1	
c = matrix[addr];		1
d = matrix[addr + 1];	1	1
g = a + cx * (b - a);	3	
h = c + cx * (d - c);	3	
p = g + cy * (h - g);	3	
sum:	**20**	**4**

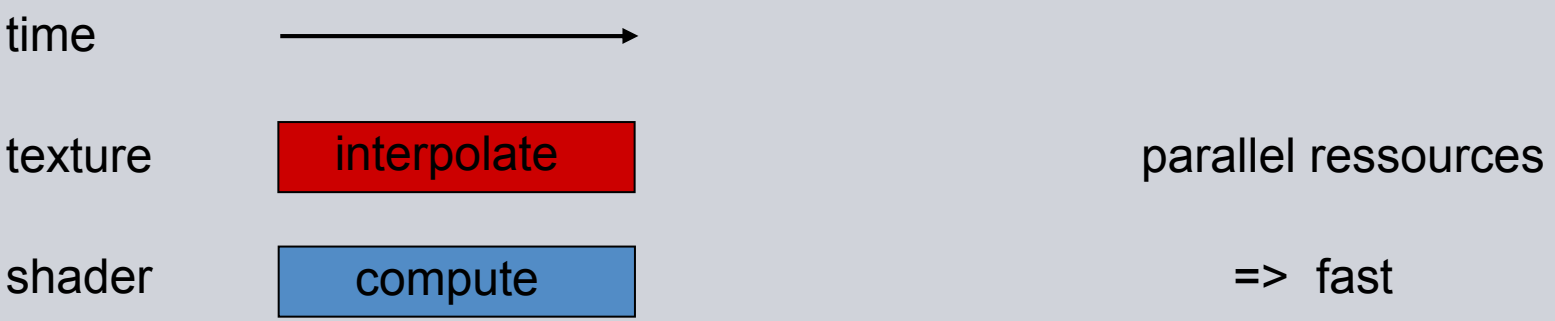


rewrite linear interpolation:
a*(1-c) + b*c = a + c*(b-a)

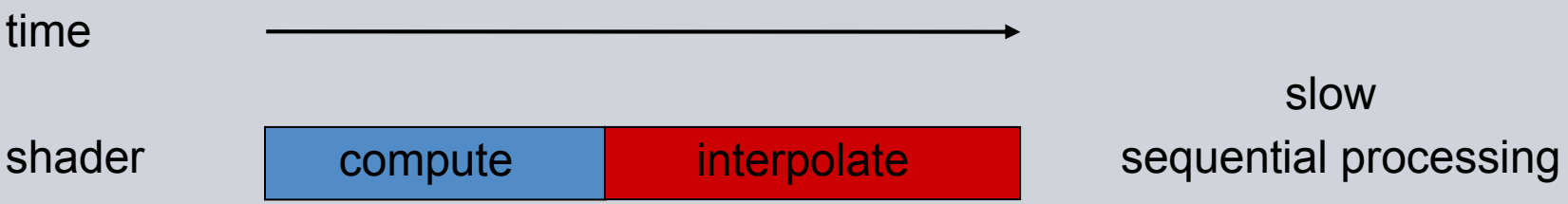
SIEMENS

Texture Discussion

Textures and shader work in parallel:



If all work is done only by shader (no textures used):



SIEMENS

Texture Discussion (GF100)

Additional advantages of textures:

- **separate texture cache** for every SM (Streaming Multiprocessor)
- **border handling** (clamp, wrap)
- **format conversion** (16 bit half float to 32 bit float, 16 bit short to 32 bit float)

Handicaps of textures:

- precision of interpolation coefficients: 8 bit resolution
- limited texture resources => multiple textures, texture atlas
- +0.5 in texture argument: tex2D(matrix,x+0.5f,y+0.5f);

Optimization Hints

Texture cache access basics (GF100):

- 4 subsequent threads access the texture memory at the same time
- only one cache line is transferred to this texture block per clock cycle
- a texture cache line is 32 bytes x 4

Use a more local texture memory access pattern:

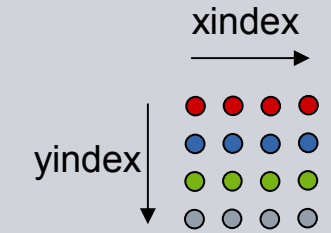
SIEMENS

Optimization Hints

Resort thread coordinates at the beginning of kernel:

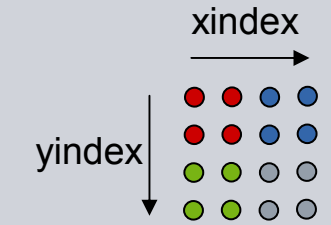
Before:

```
xindex = (blockIdx.x * blockDim.x) + threadIdx.x;
yindex = (blockIdx.y * blockDim.y) + threadIdx.y;
```



After:

```
x = threadIdx.x;
y = threadIdx.y;
x1 = (x>>1)&1;
y1 = y&1;
x = (x & ~2) | (y1 << 1);
y = (y & ~1) | x1;
xindex = (blockIdx.x * blockDim.x) + x;
yindex = (blockIdx.y * blockDim.y) + y;
```



more local texture access, but
storage of results is no longer coalescent

SIEMENS

Optimization Hints

If possible:

Reduce input data wordlength from 4 to 2 Byte.

from float to half float or
from float to short

- reduces main memory size by 2
- reduces necessary load bandwidth from memory by 2
- doubles content of the texture cache
- doubles content of one cache line (64 words against 32 words)

SIEMENS

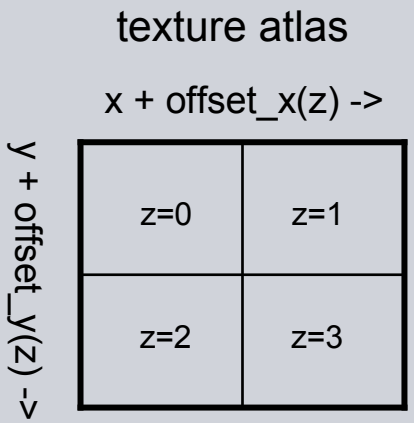
Texture Atlas

Shortcoming until CUDA 3.2

Currently we have to work with a **texture atlas**

tex2D(tex_atlas,x+**offset_x(z)**,y+**offset_y(z)**)

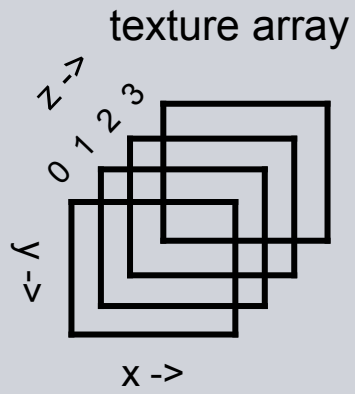
But we need **vectors of textures** to simplify handling



Solved with CUDA 4.0 : layered textures

tex2DLayered(tex_array,x,y,**z**)

z-dimension is integer index (up to 2^11),
similar to 3D texture,
but with bilinear interpolation only.



SIEMENS

Texture Precision (GF100)

tex2D(matrix,x,y)

Maximum 2D texture width and height is 65535 x 65536 (=2^{16})

Interpolation coefficient is a fraction between [0 , 1[.

The interpolation coefficient is represented with 8 bit.

32 bit floating point number representation according IEEE 754:

$x = (-1)^S 2^{E-127} (1.M)$

x = 33000.1 = 0x 4700 E81A = 0100 0111 0000 0000 1110 1000 0001 1010

sign exponent mantissa

S = 0 , E = 0x8E = 142 , M = 0x00E81A = 59418

1.M = 0x80E81A * 2^{-23} = 8448026 * 2^{-23}

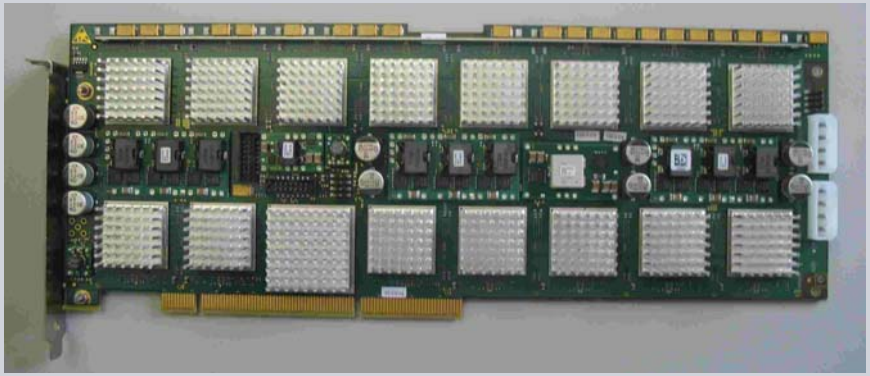
floor(1.M * 2^{15}) = 0x80E8 = 33000 = index

frac(1.M * 2^{15}) = 0x1A * 2^{-8} = 0.1015625 = coefficient

There are only 8 bit remaining for the coefficient !

SIEMENS

Mind the Fermi Gap

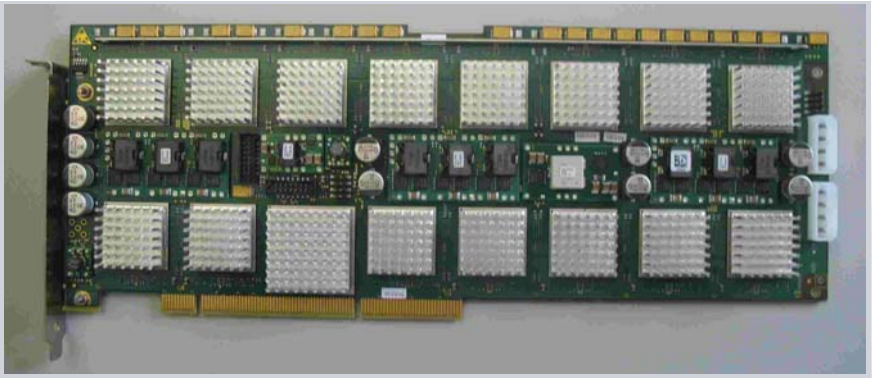


	CBR	Tesla C2050	increase %
Memory / GBytes	8	3	- 62
GTexel / sec	76.3	32.2	- 58
images / sec	40	25	- 37
power / W	200	238	+ 19
year	2006	2010	+ 0.2
technology	FPGA	GPU	
manufacturer	Siemens	Nvidia	

3D-Backprojector comparison: CBR versus Tesla C2050

SIEMENS

Solve the Fermi Gap



	1 x CBR	3 x Tesla C2050	increase %
Memory / GBytes	8	9	+ 13
GTexel / sec	76.3	96.6	+ 27
images / sec	40	64	+ 60
power / W	200	714	+ 257
year	2006	2010	+ 0.2
technology	FPGA	GPU	
manufacturer	Siemens	Nvidia	

3D-Backprojector comparison: 1 x CBR versus 3 x Tesla C2050

SIEMENS

Motivation

A GPU offers:

- programmable in software
- can run several algorithms
- low development costs
- fast development cycles for algorithms
- free performance upgrade with next GPU generation

SIEMENS

Accelerator Technologies

- **ASIC** : highest performance
  very expensive development
  longest development cycle
  no flexibility
- **FPGA** : high performance
  expensive development
  long development cycle
  programmable in firmware
- **DSP** : programmable in software
  low power consumption
- **CPU** : mainstream
  programmable with all kind of languages and tools
- **GPU** : fastest software programmable solution
  programmable in CUDA and OpenCL

SIEMENS

Take Home Messages

- Medical imaging depends on bilinear interpolations.
  => Textures

- Texture units are powerful hardware resources.
  Therefore use them !

- Texture units can deliver a performance increase by a factor ≥ 2 !

Computed Tomography
SOMATOM Definition Flash

SIEMENS

Thank You

Courtesy of: Friedrich-Alexander University Erlangen-Nuremberg - Institute of Medical Physics / Erlangen, Germany