Implementing the Projected Spatial Rich Features on a GPU

Andrew Ker

adk@cs.ox.ac.uk

Department of Computer Science

University of Oxford

SPIE/IS&T Electronic Imaging, San Francisco, 4 February 2014

Background

Features for binary classification steganalysis in raw images.

	dimension	extraction time for 1Mpix image
■ WAM [2006] *moments of noise residuals*	27	negligible
■ SPAM [2009] *co-occurrences of noise residuals*	686	0.25 s
■ SRM [2012] *co-occurrences of diverse noise residuals*	12753+	12 s
■ PSRM [2013] *histograms of randomly projected,* *diverse, noise residuals*	12870	25 m

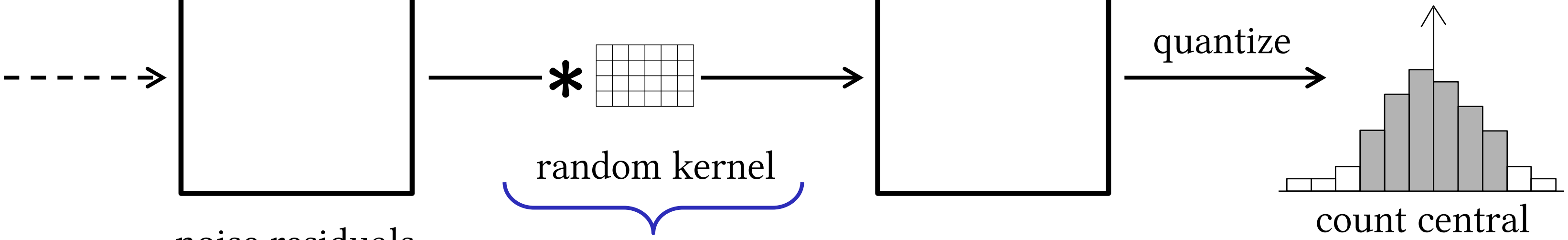
Background

Features for binary classification steganalysis in raw images.

	dimension	extraction time for 1Mpix image
▪ WAM [2006] *moments of noise residuals*	27	negligible
▪ SPAM [2009] *co-occurrences of noise residuals*	686	0.25 s
▪ SRM [2012] *co-occurrences of diverse noise residuals*	12753+	12 s
▪ PSRM [2013] *histograms of randomly projected, diverse, noise residuals*		25 m

An experiment with 1 million images takes 50 years ☹

Projected residuals



noise residuals

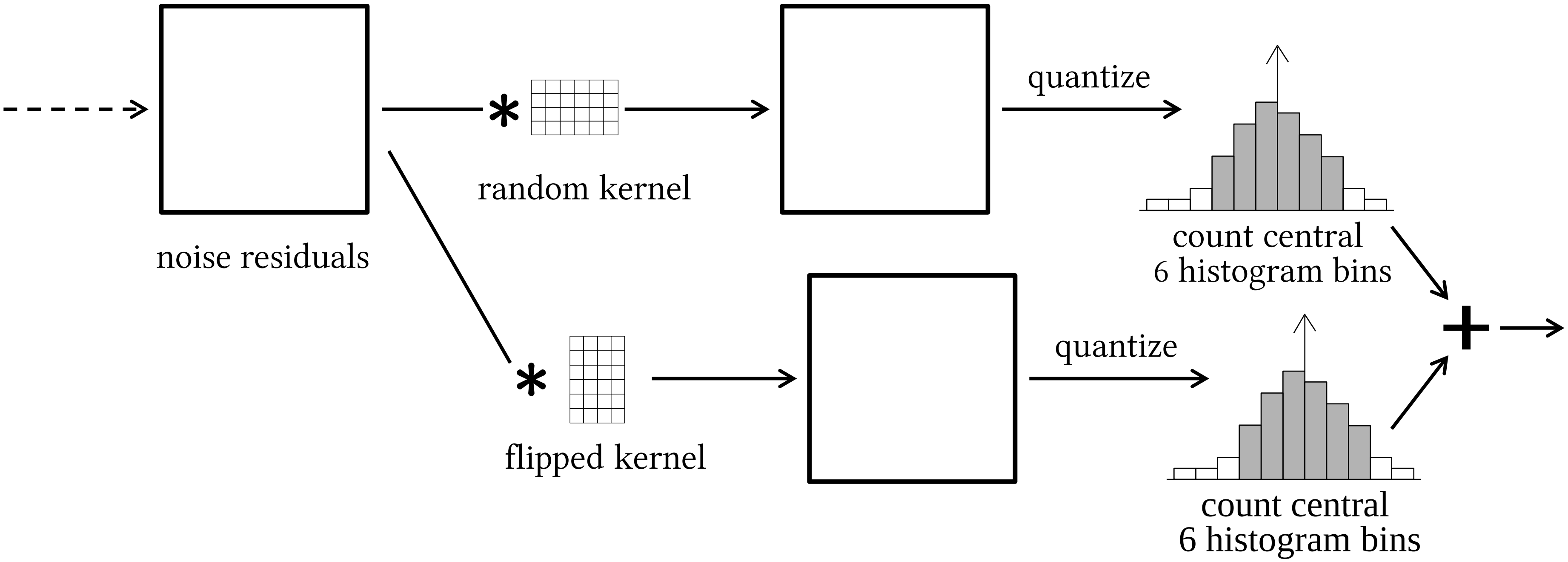
random kernel

quantize

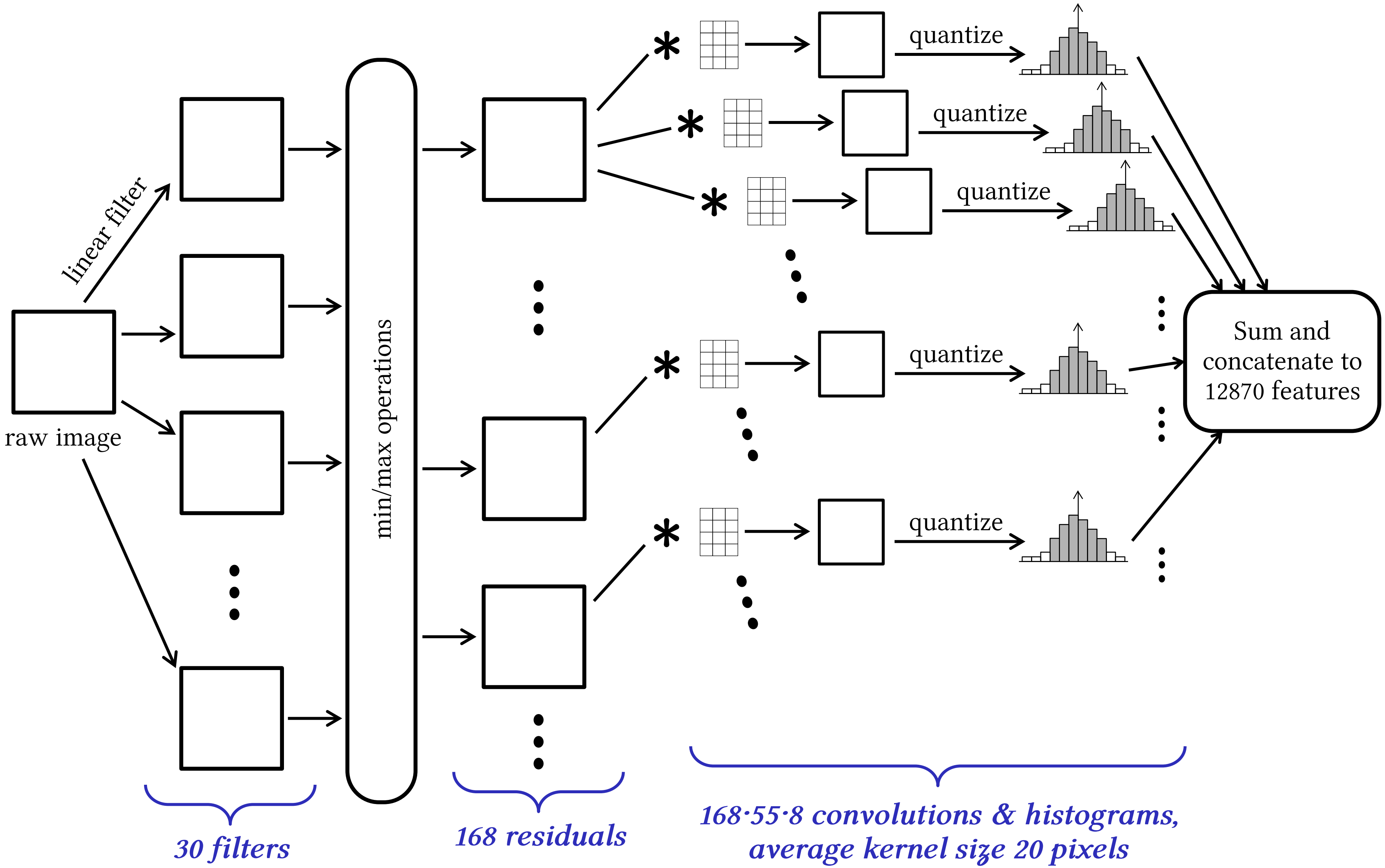
count central
6 histogram bins

- *Width, height uniform on* {1,…,8}
- *Entries Gaussian, scaled to unit norm*

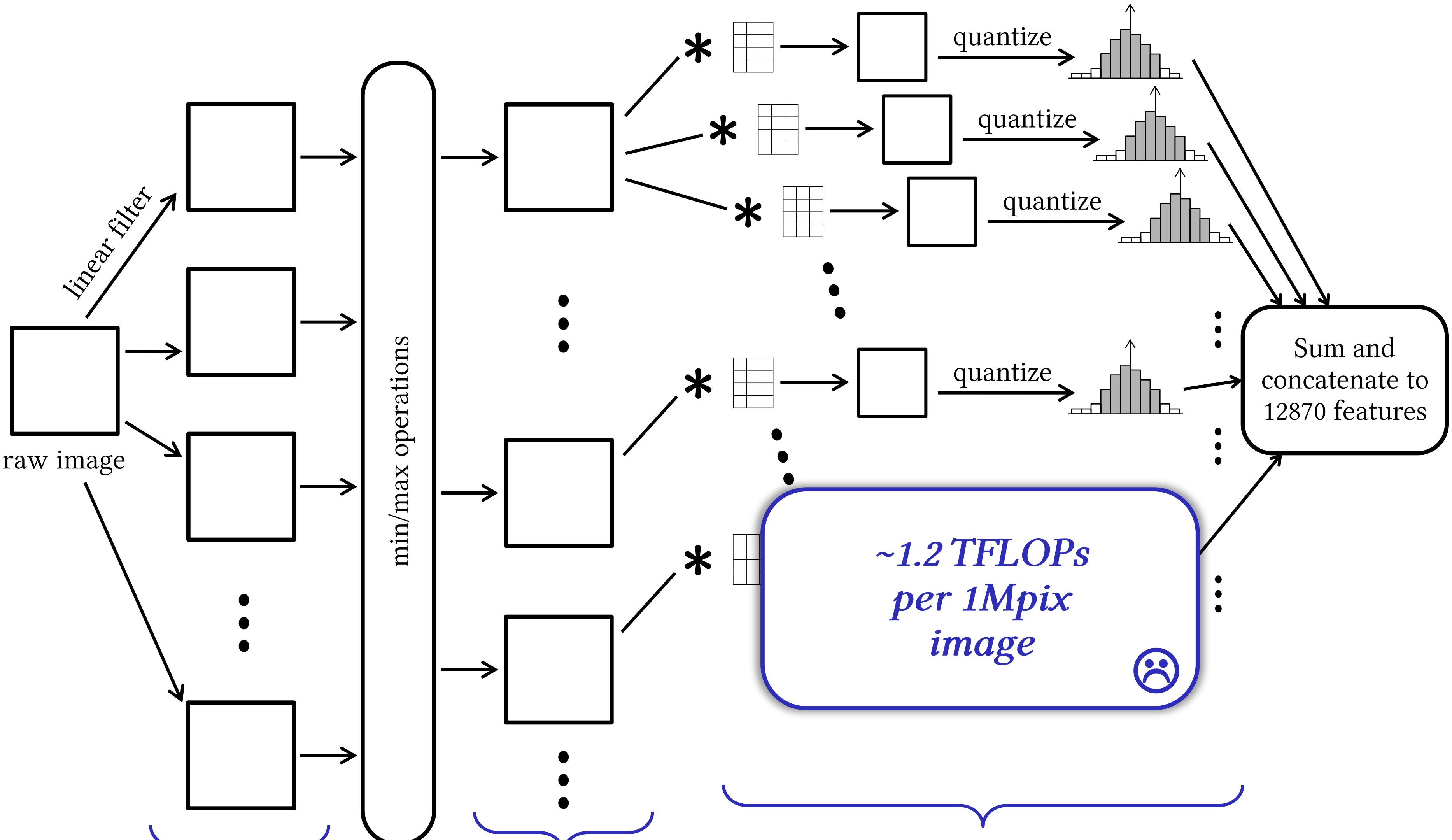
Projected residuals



PSRM features


linear filter
raw image
min/max operations
*
quantize
Sum and concatenate to 12870 features
30 filters
168 residuals
168·55·8 convolutions & histograms, average kernel size 20 pixels

PSRM features



raw image

linear filter

min/max operations

quantize

quantize

quantize

quantize

Sum and concatenate to 12870 features

~1.2 TFLOPs per 1Mpix image ☹

30 filters

168 residuals

168·55·8 convolutions & histograms, average kernel size 20 pixels

GPU architecture

We target the NVIDIA *Tesla K20* card (GK110 GPU):

- Costs $2800.
- ***CUDA*** programming language.

- Execution in ***warps***, 32 simultaneous identical instructions per multiprocessor (MP).
- Communicating warps grouped in ***blocks***.
- Blocks interleaved concurrently on 78 MPs.

2496 FP processors: ~3.52TFLOP/s.

… but memory bandwidth & latency is limiting.

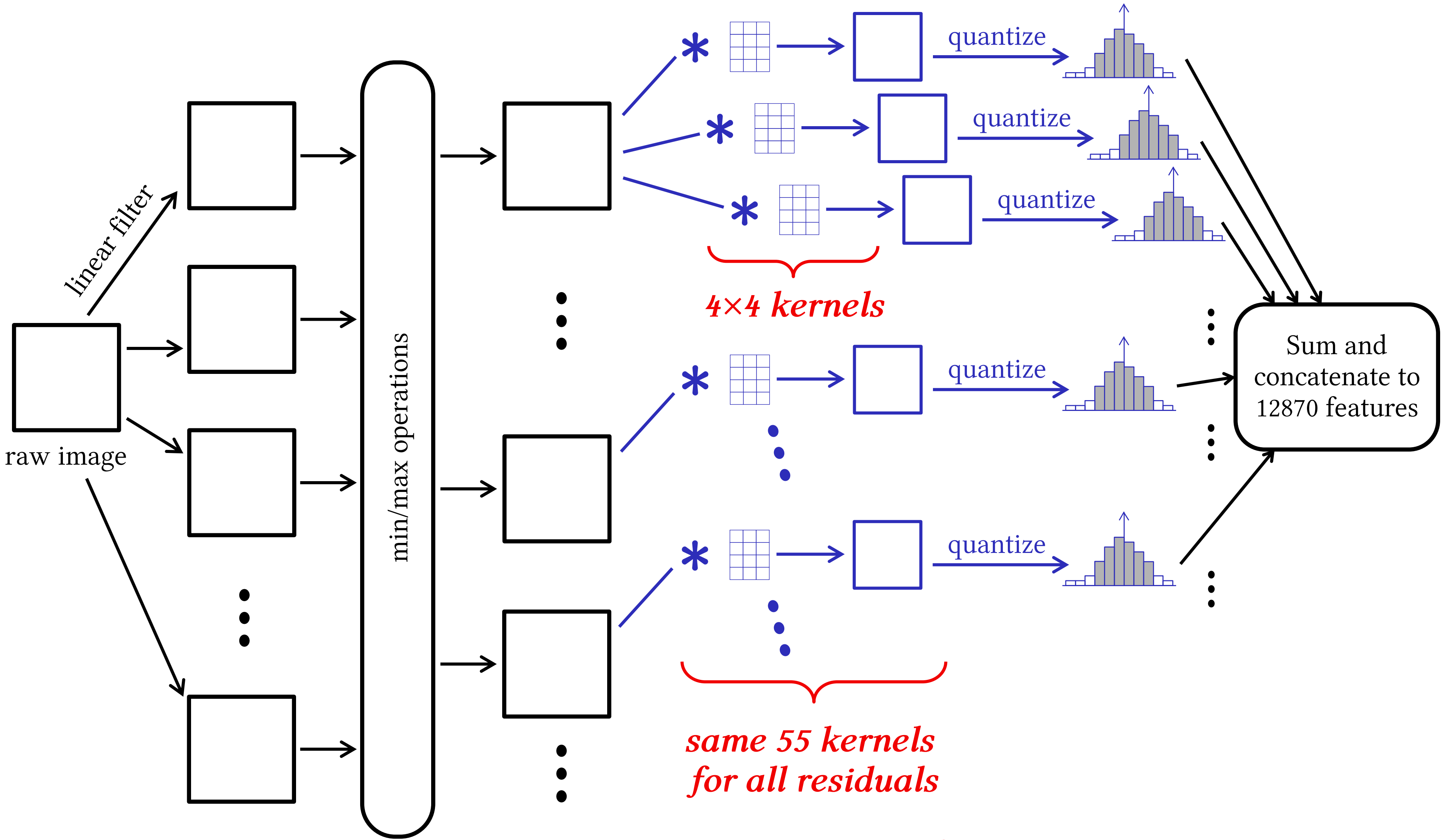
GPU architecture

	latency	**size**
Registers	zero	64K words per MP
Shared memory	~ 10 cycles	~ 48KB *for all concurrent blocks*
Global memory	~ 200 cycles	~ 5GB

Global access latency hidden by concurrently-running blocks (with immediate context switching).

... parallelism vs register exhaustion.

GPU-PSRM features

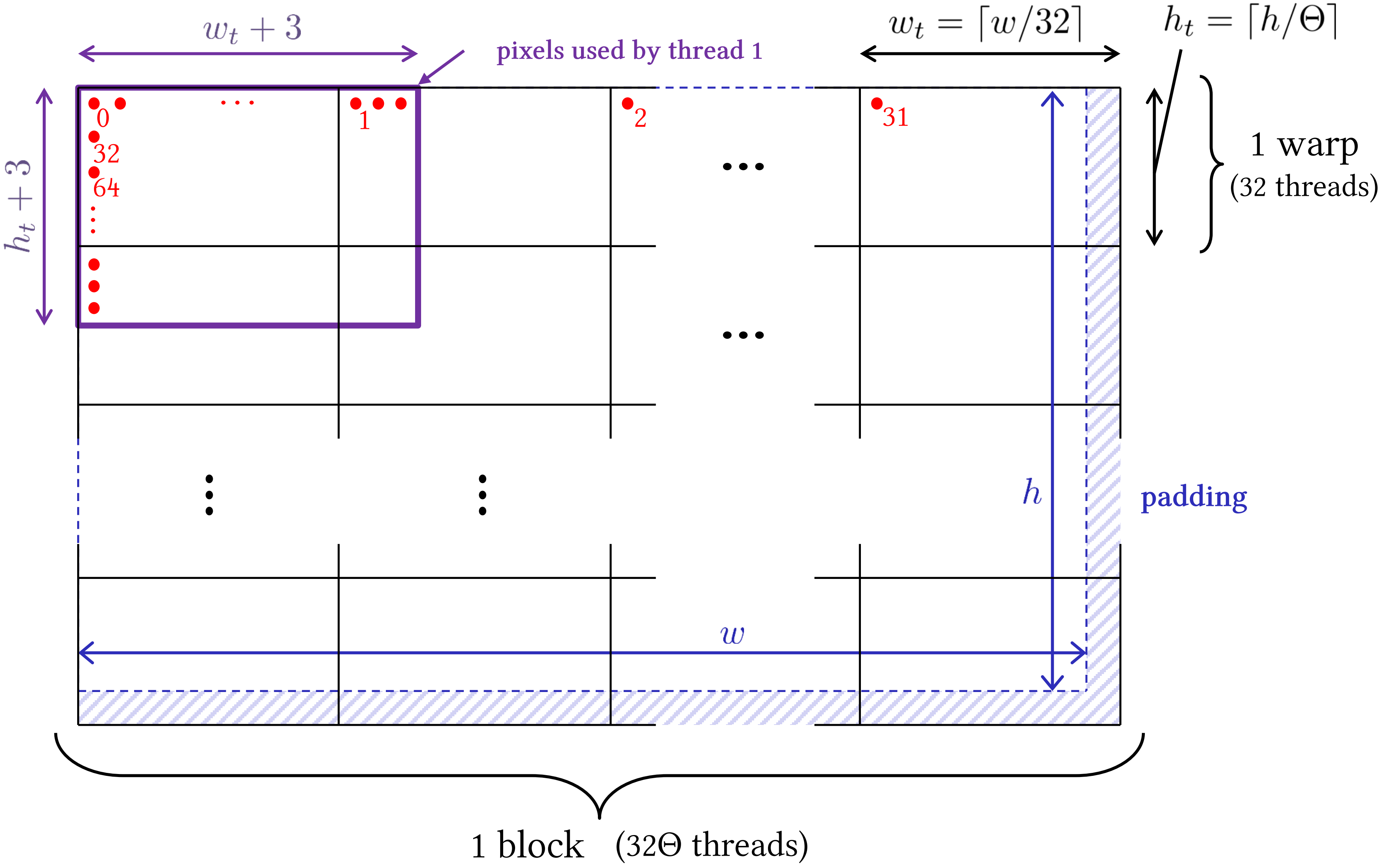


4×4 kernels

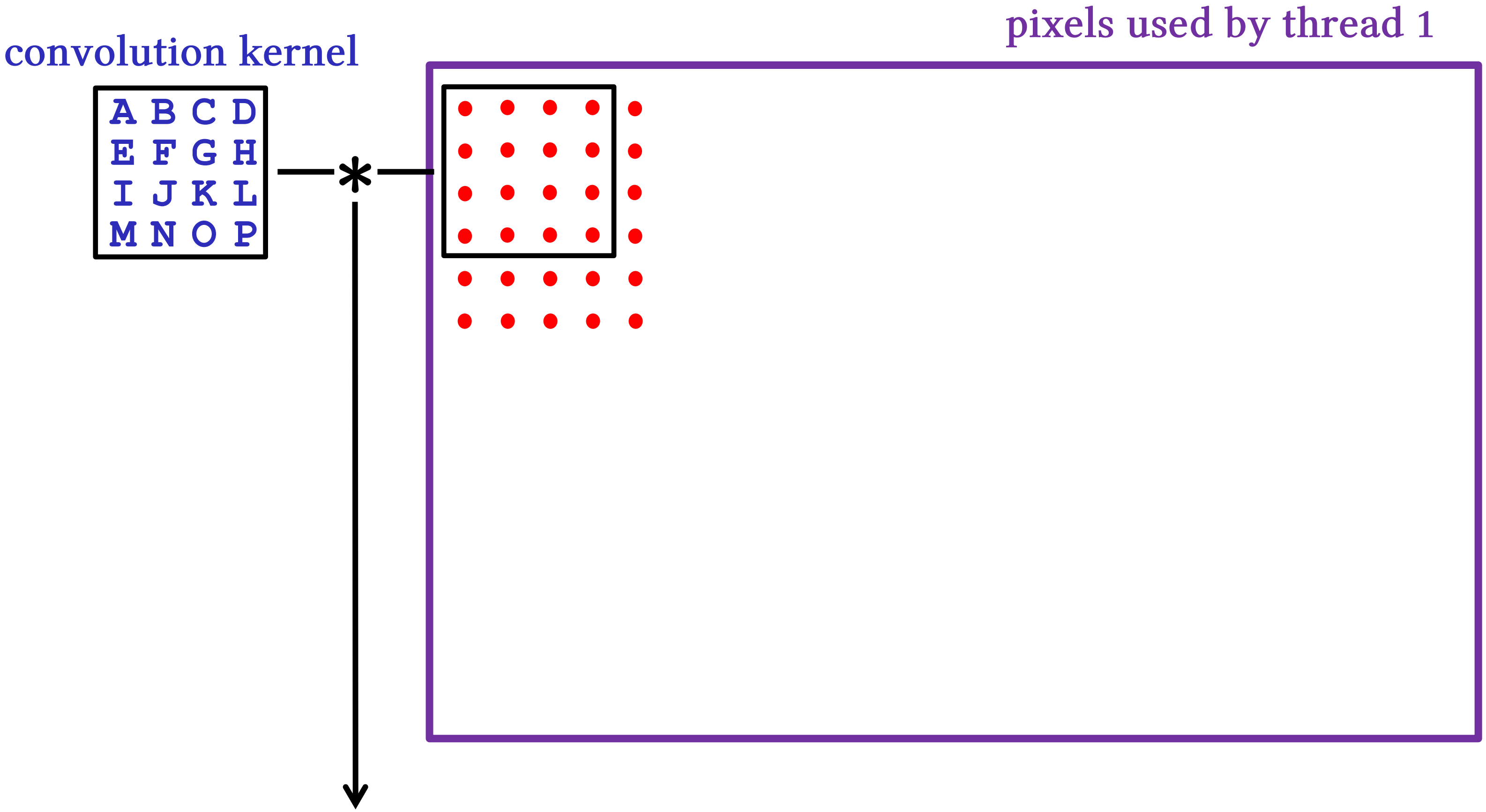
same 55 kernels for all residuals

... also consider fewer projections per residual

Tiles


$w_t + 3$
pixels used by thread 1
$w_t = \lceil w/32 \rceil$
$h_t = \lceil h/\Theta \rceil$
$h_t + 3$
0
1
2
31
32
64
1 warp
(32 threads)
h
padding
w
1 block (32Θ threads)

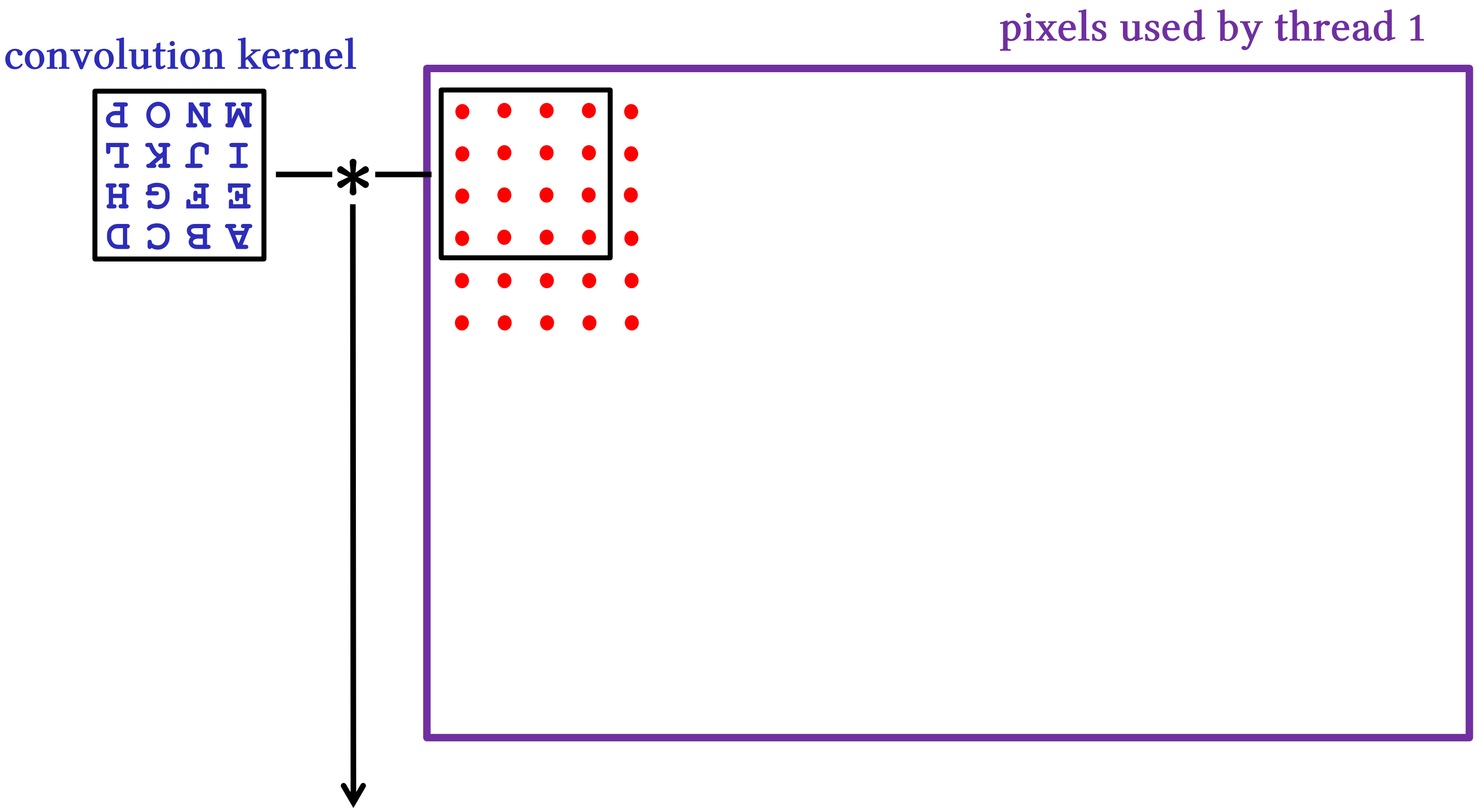
One thread



A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

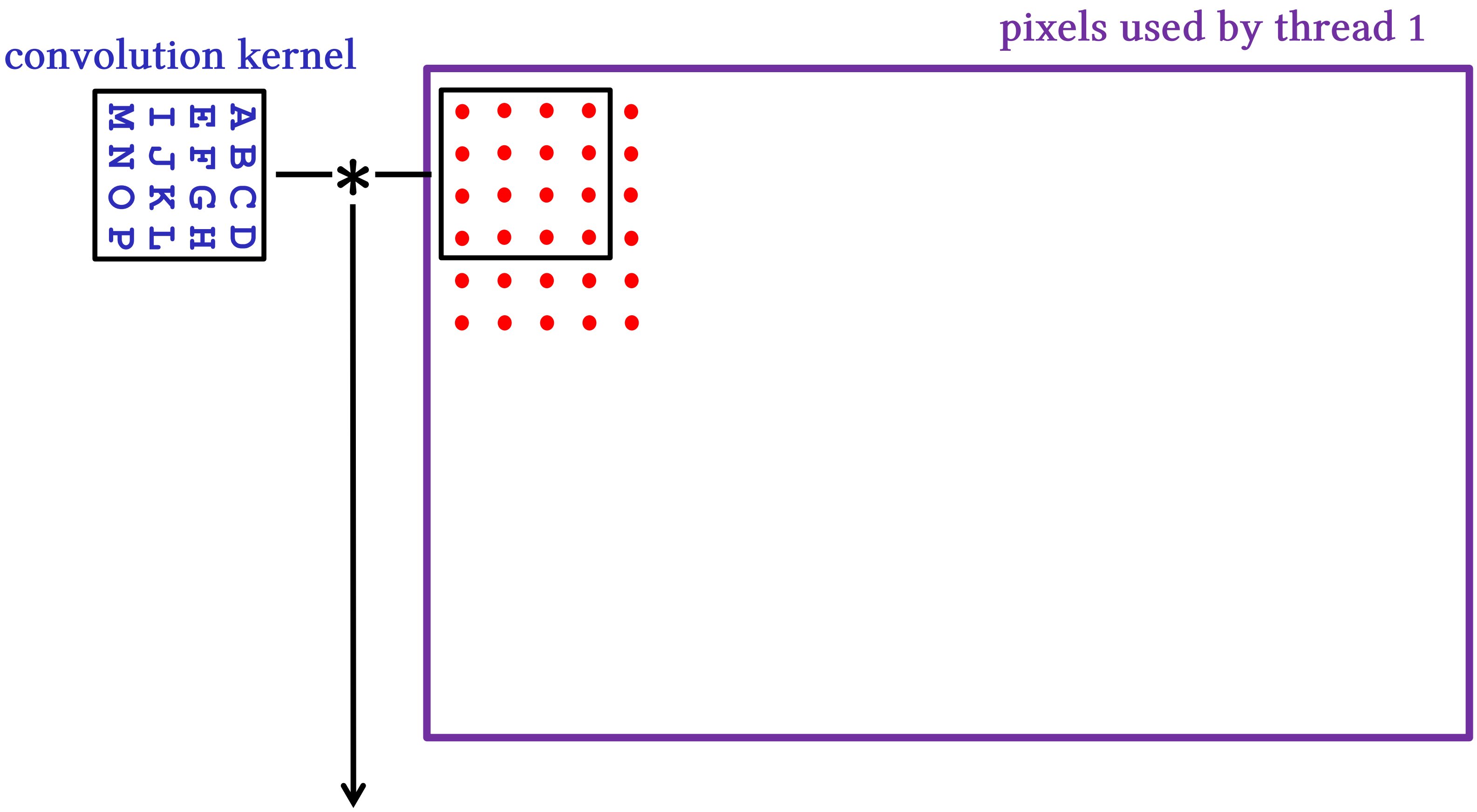
- Quantize
- Truncate
- Increment histogram bin

One thread



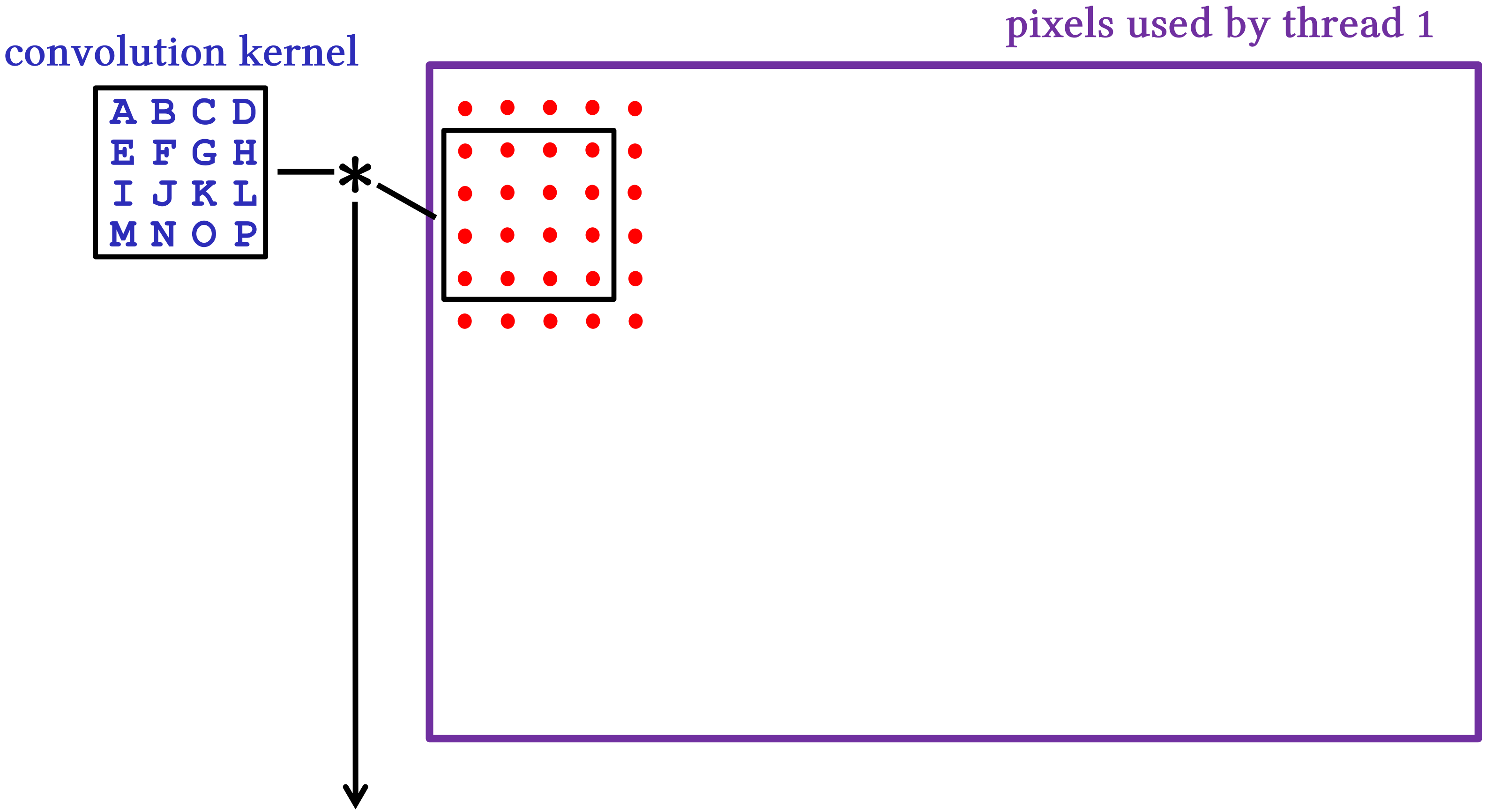
- Quantize
- Truncate
- Increment histogram bin

One thread



- Quantize
- Truncate
- Increment histogram bin

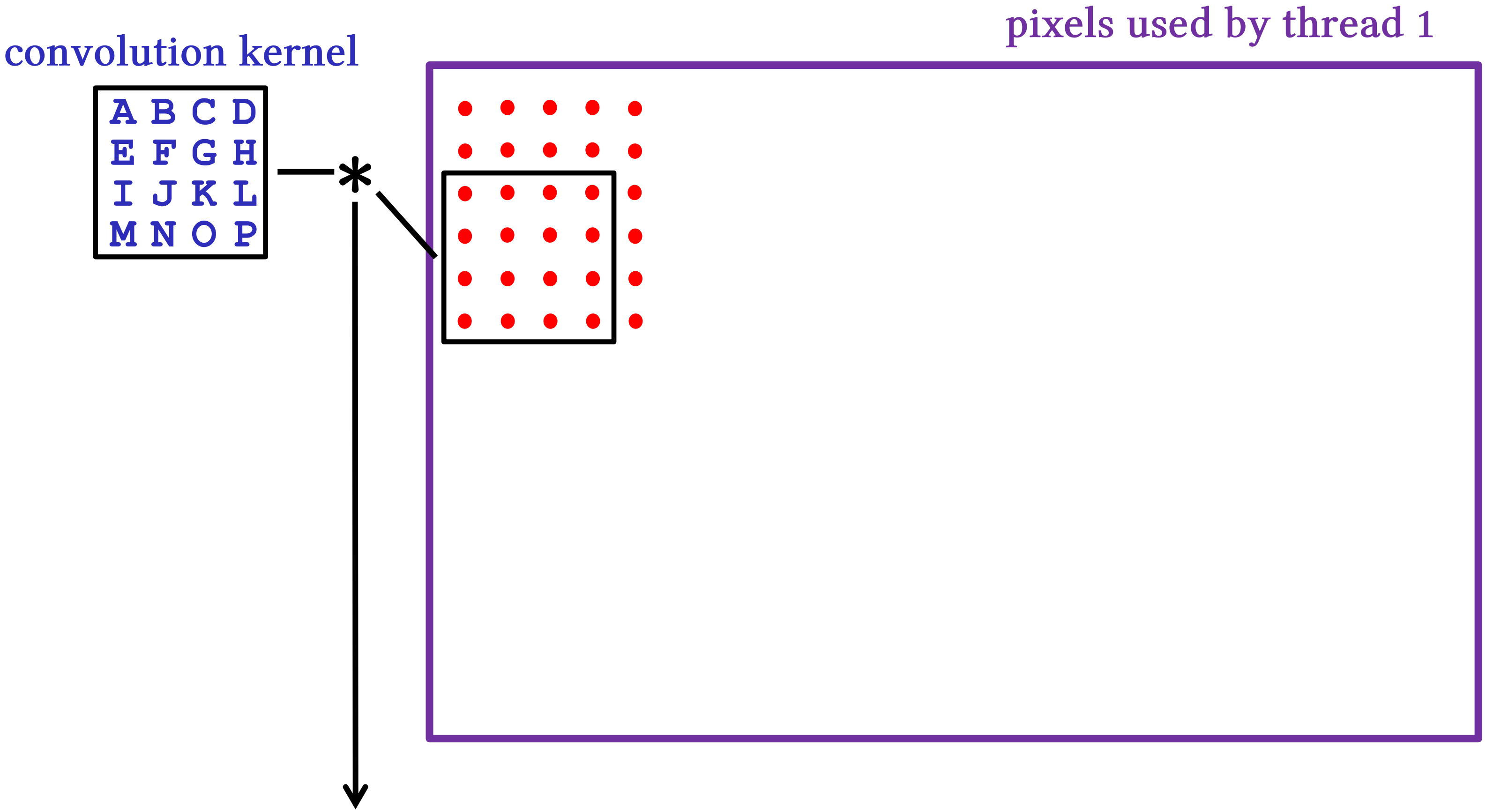
One thread



A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

- Quantize
- Truncate
- Increment histogram bin

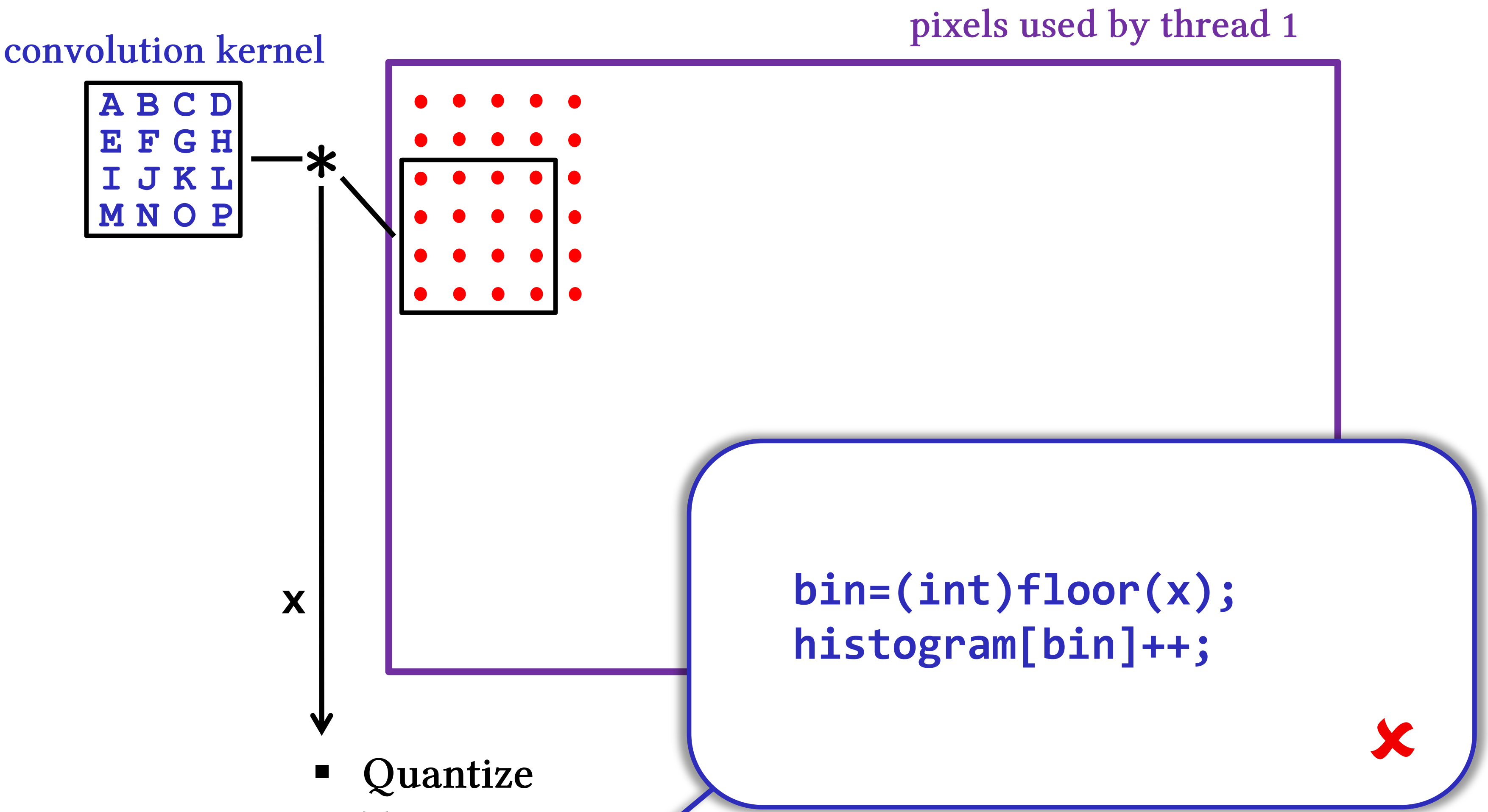
One thread



A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

- Quantize
- Truncate
- Increment histogram bin

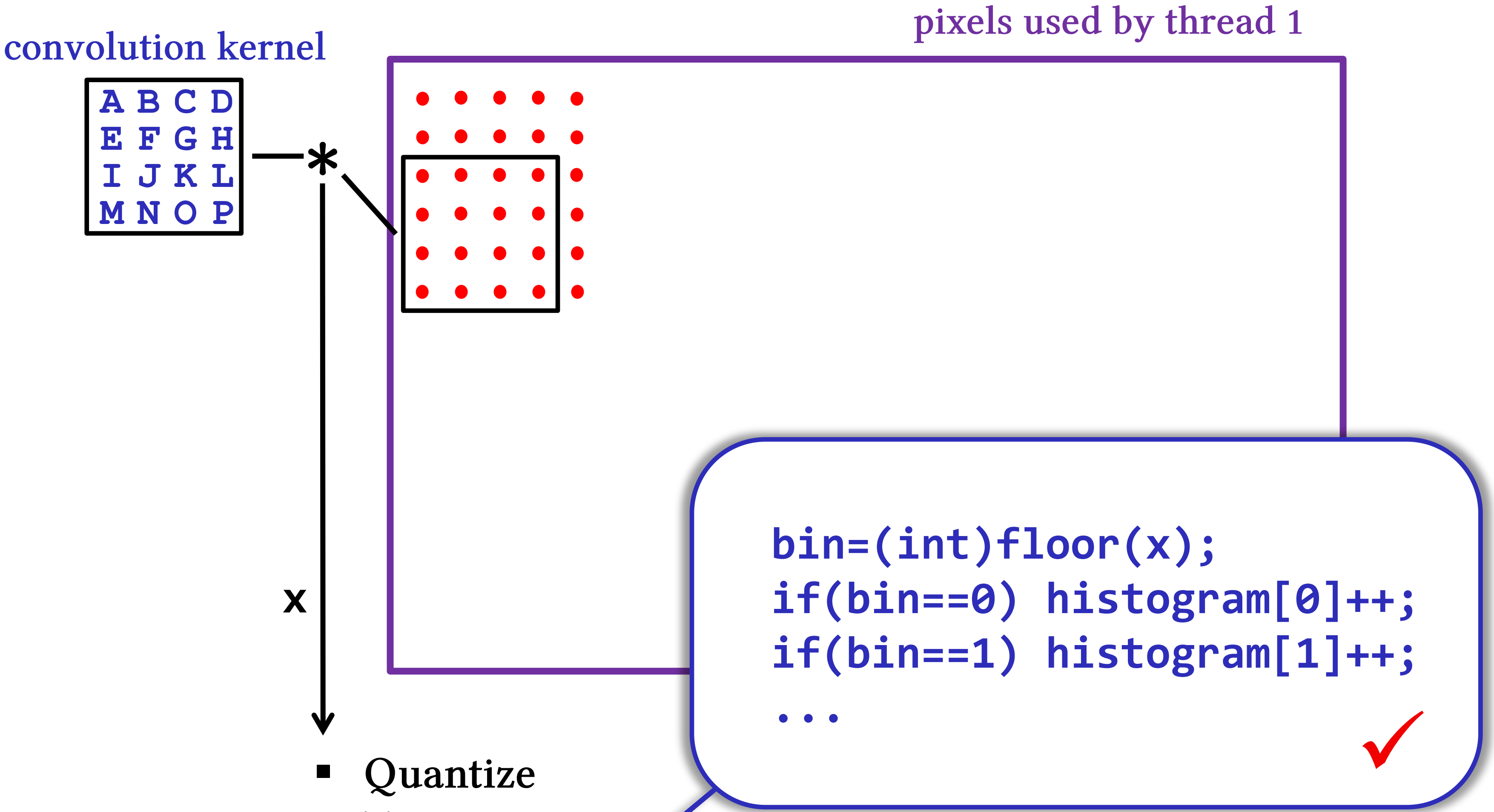
One thread



```
bin=(int)floor(x);
histogram[bin]++;
```

- Quantize
- Truncate
- Increment histogram bin

One thread



```
bin=(int)floor(x);
if(bin==0) histogram[0]++;
if(bin==1) histogram[1]++;
...
```

✓

- Quantize
- Truncate
- Increment histogram bin

Benchmarks

Machine: 16-core 2.0GHz SandyBridge Xeon

Implementation	wallclock extraction time for 1Mpix image	
▪ Reference C++	29588 s	
▪ Reference MATLAB *single-thread*	1554 s	
▪ Reference MATLAB *multi-thread*	1100 s	(2186 s CPU)
▪ Optimized CUDA *using 1×TESLA K20*	2.6 s	potentially <1 s

Accuracy

Steganalysis experiment:

- 10000 BOSSBase v1.01 cover images (256Kpix).
- HUGO embedding, 0.4bpp.
- Measure Ensemble FLD error on disjoint testing sets.

	# projections per residual	dimension	testing error rate	Extraction of 256Kpix image
Reference PSRM	55	12870	12.98%	491 s
GPU-PSRM	55	12870	14.34%	0.59 s
	40	9360	14.75%	0.45 s
	30	7020	14.78%	0.36 s
	20	4680	14.88%	0.27 s
	10	2340	15.71%	0.20 s

Accuracy

Steganalysis experimen

- 10000 BOSSBase v1
- HUGO embedding,
- Measure Ensemble

This single experiment:

- *2732 core hours.*
- *Costs £136 ($223) on Oxford University cluster (internal prices).*
- *Would cost twice as much on EC2.* ☹

	# projec per resi			f ge
Reference PSRM	55	12870	12.98%	491 s
GPU-PSRM	55	12870	14.34%	0.59 s
	40	9360	14.75%	0.45 s
	30	7020	14.78%	0.36 s
	20	4680	14.88%	0.27 s
	10	2340	15.71%	0.20 s

Conclusions

- PSRM features require massive amounts of computation.

  GPU implementation the only possibility for a quick result.

- GPU-PSRM features are slightly modified, optimization-friendly.

  Lose a little in variety, but only 1% additional error.
  400-1000 times faster than current CPU implementations.

- Should consider cost/benefit analysis of new features.

  A practitioner might prefer speed to accuracy.

- Optimize implementation of previous-gen. features? (SRM/JRM)

  Need not necessarily involve a GPU.

Conclusions

- PSRM features require massive amounts of computation.

  GPU implementation the only possibility for a quick result.

- GPU-PSRM features are slightly modified, optimization-friendly.

  Lose a little in variety, but only 1% additional error.
  400-1000 times faster than current CPU implementations.

- Should consid

  A practitione

Source will be available from

`http://www.cs.ox.ac.uk/andrew.ker/gpu-psrm/`

☺

- Optimize implementation of previous-gen. features? (SRM/JRM)

  Need not necessarily involve a GPU.