ES ELECTRONIC SYSTEMS

TU/e

European Commission

oCPS

Platform-aware Model-driven Optimization of Cyber-Physical Systems

Approximate Computing for Energy Efficient Computing

Sayandip De

Eindhoven University of Technology, The Netherlands

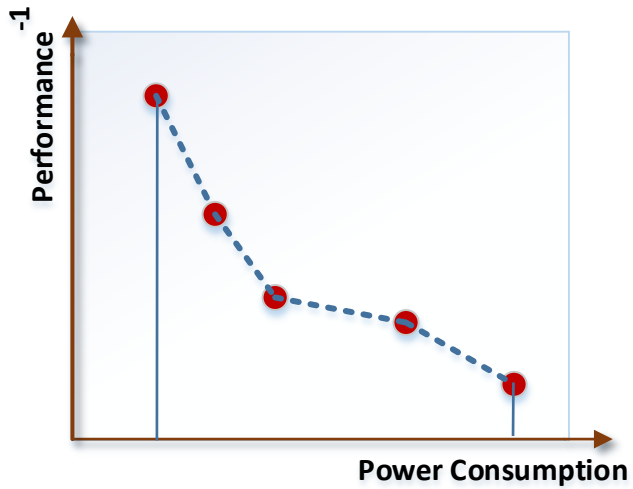
Supervisors: Henk Corporaal, Jos Huisken

This project has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under grant agreement no 674875.

oCPS

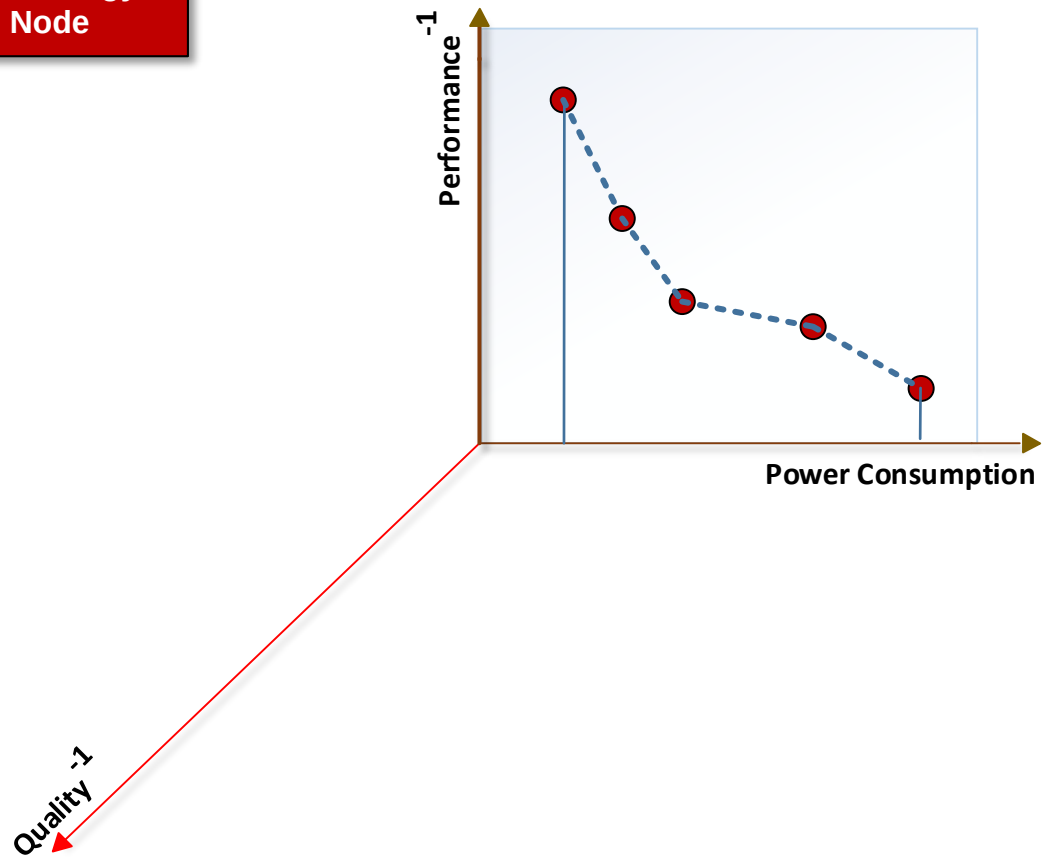
What is “approximate computing”?

Particular Technology Node



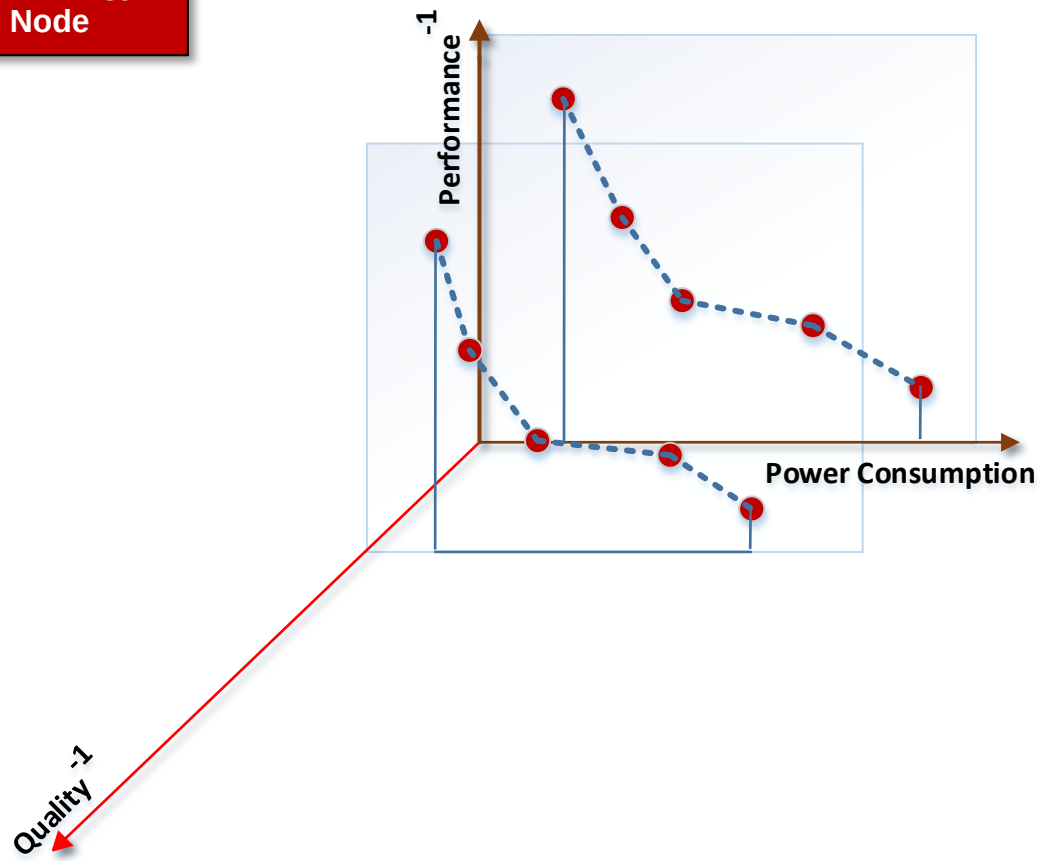
What is “approximate computing”?

Particular Technology Node



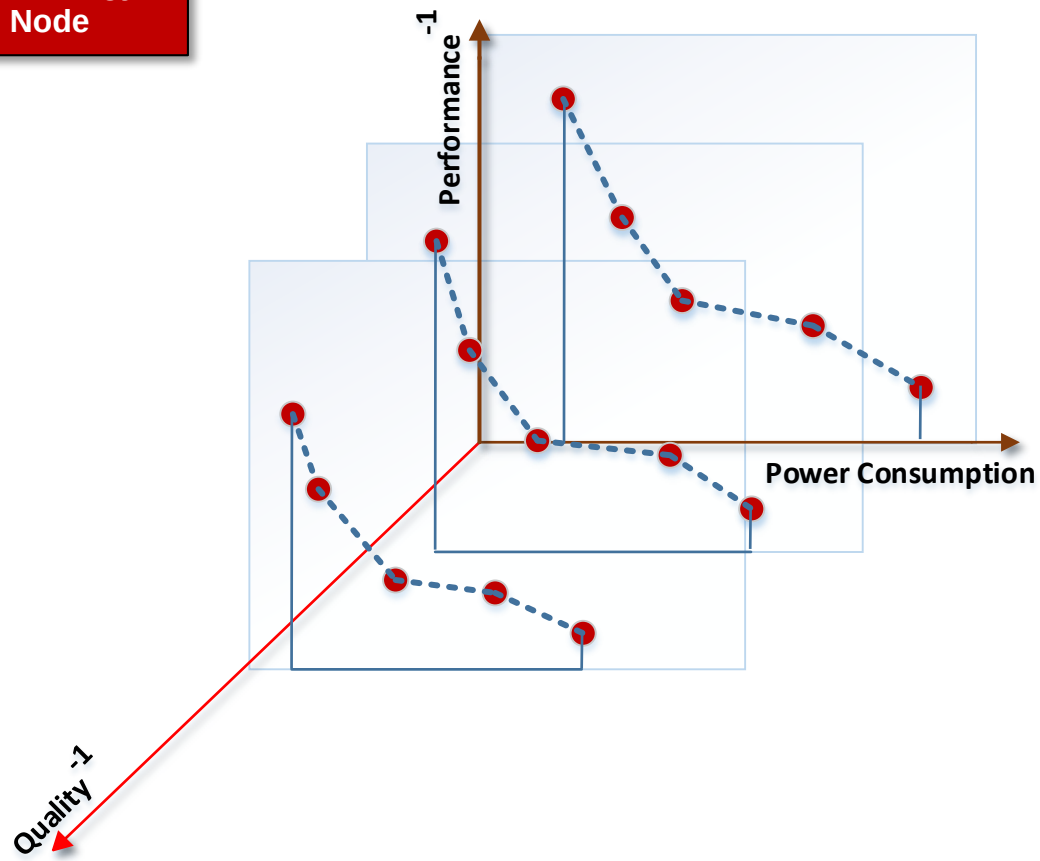
What is “approximate computing”?

Particular Technology Node



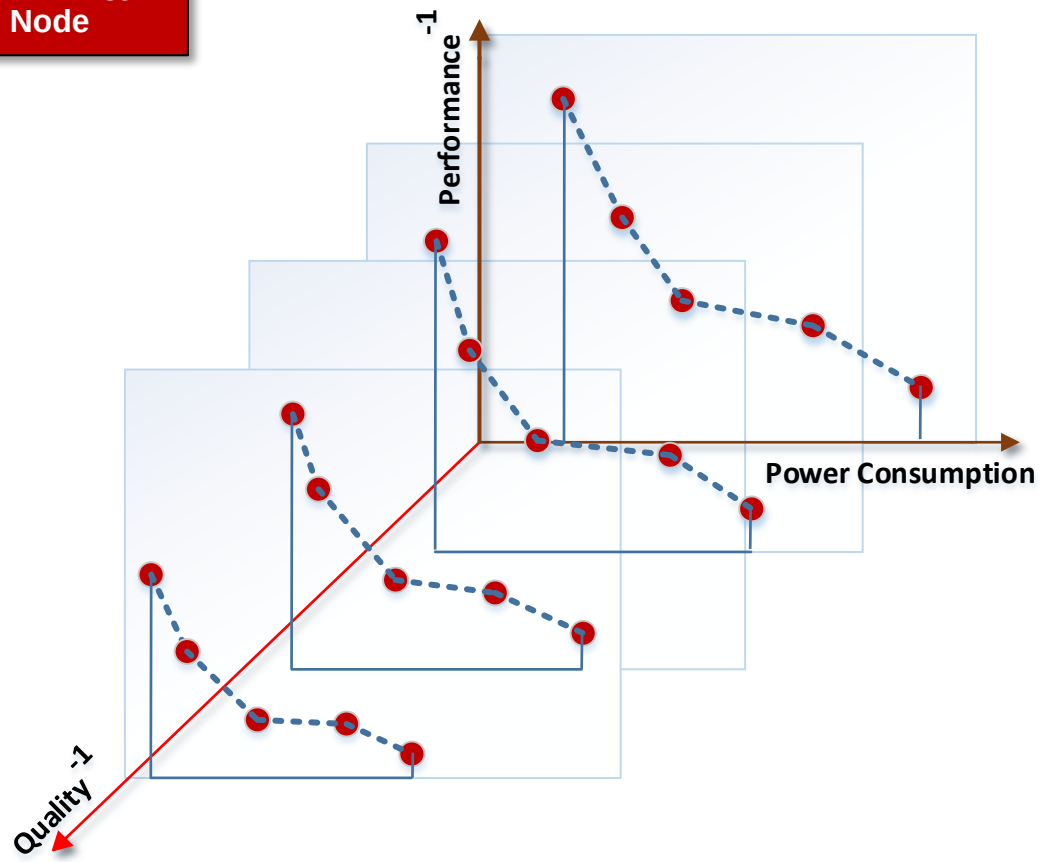
What is “approximate computing”?

Particular Technology Node

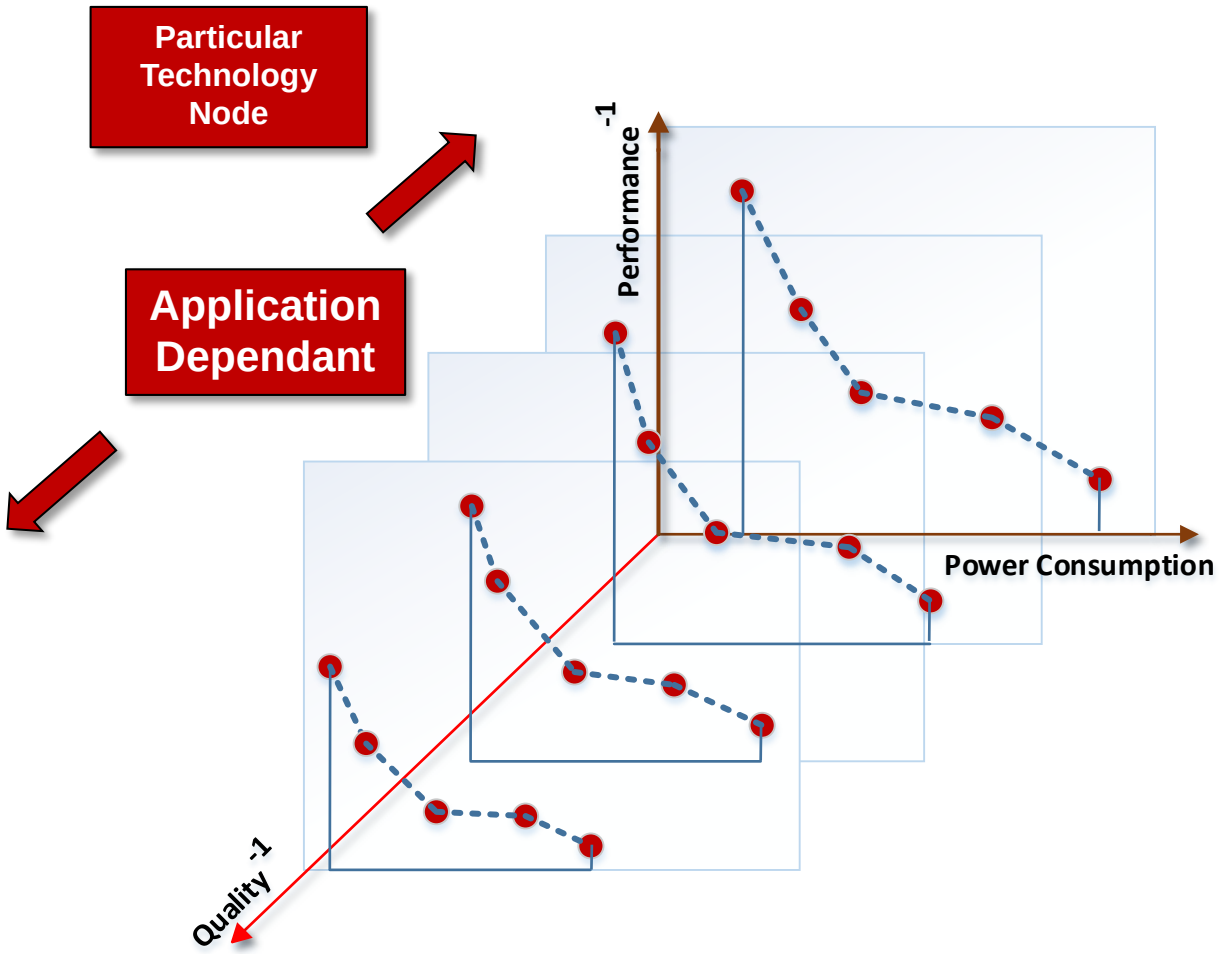


What is “approximate computing”?

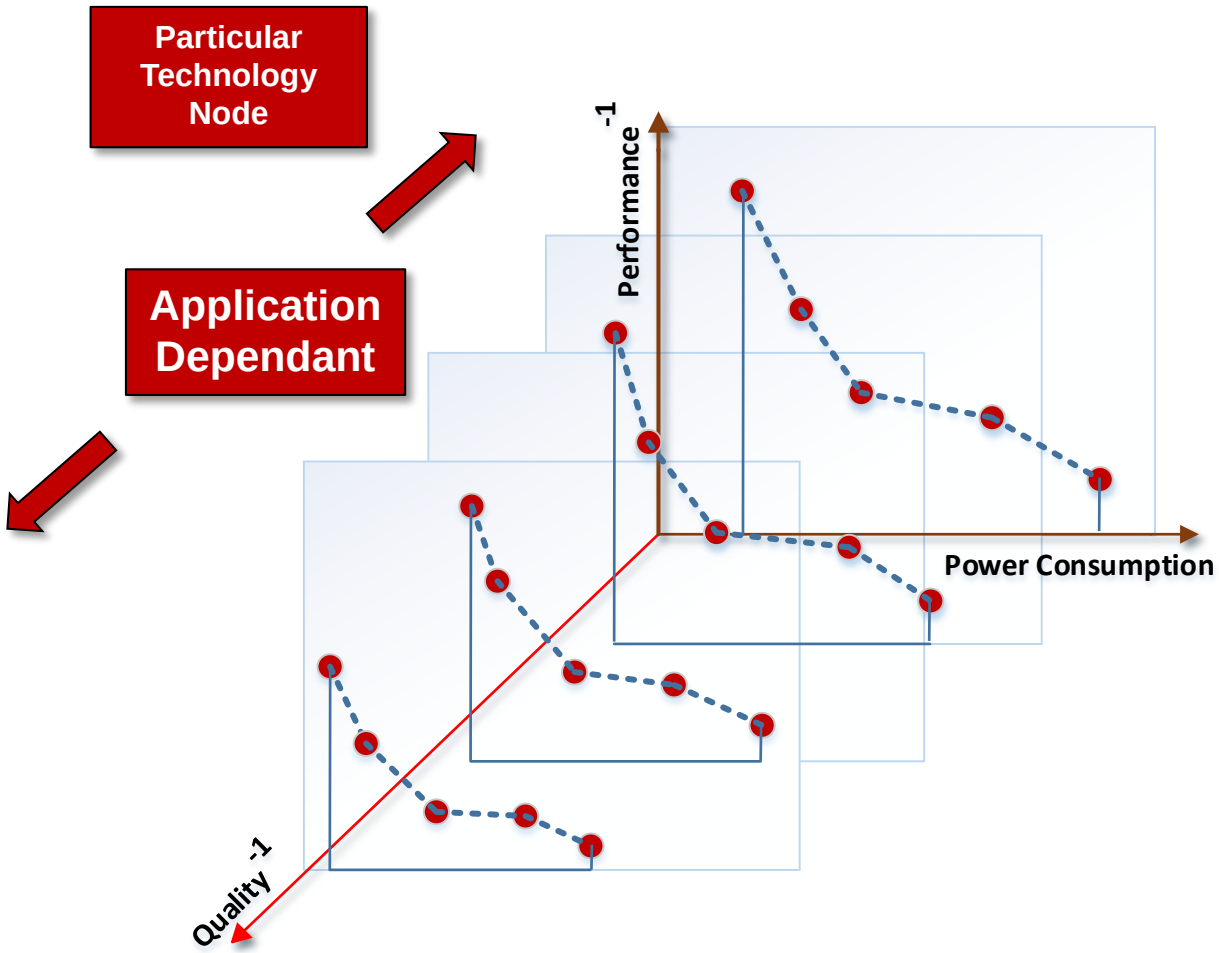
Particular Technology Node

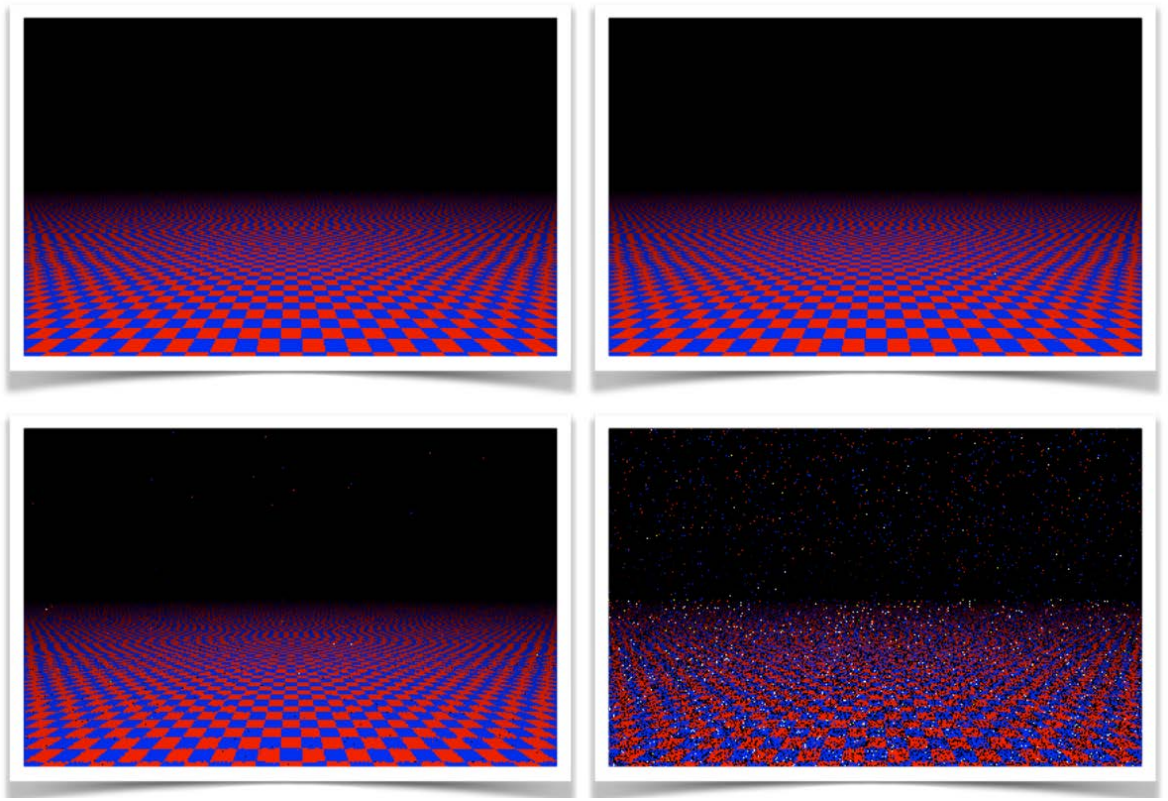


What is “approximate computing”?



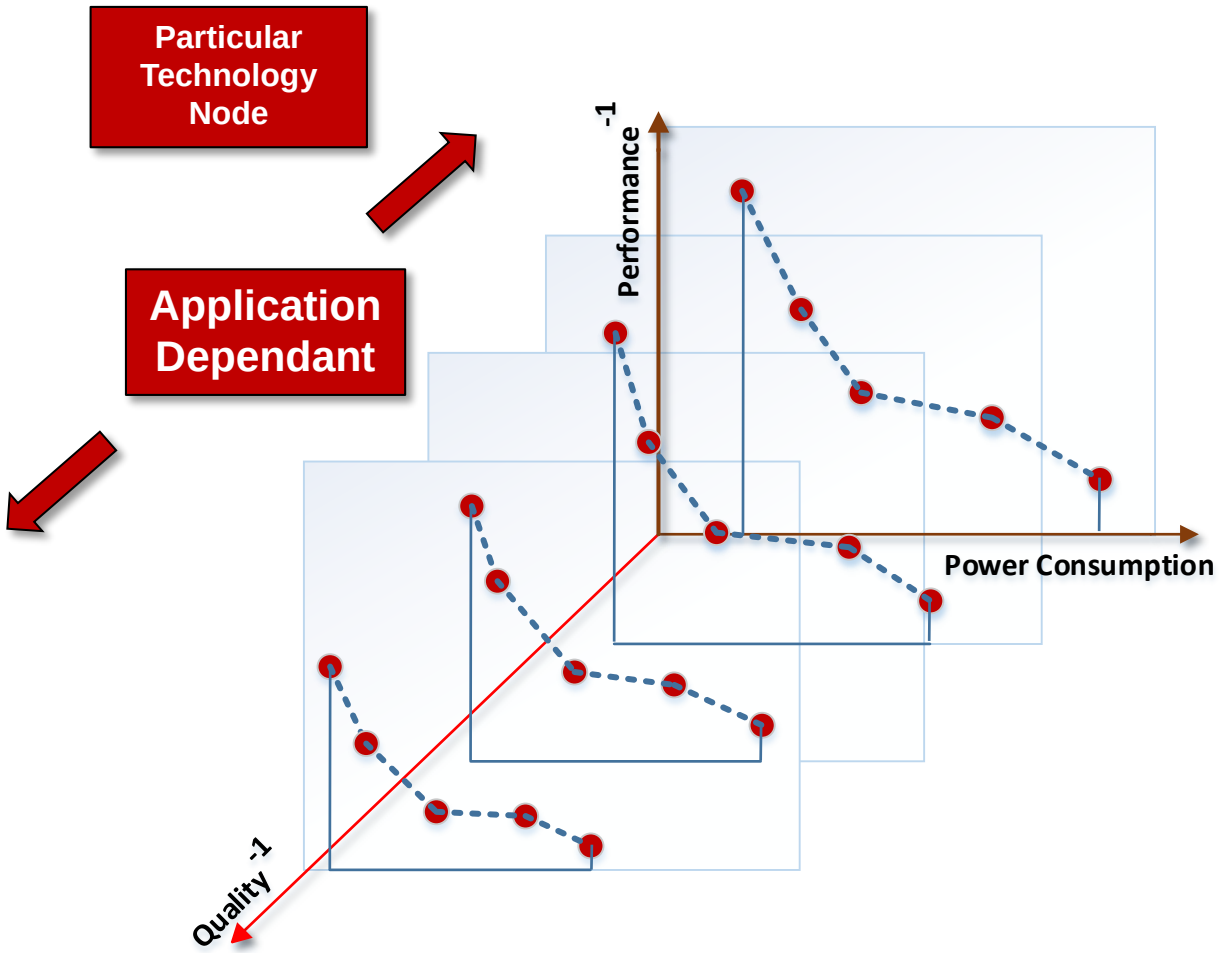
What is “approximate computing”?

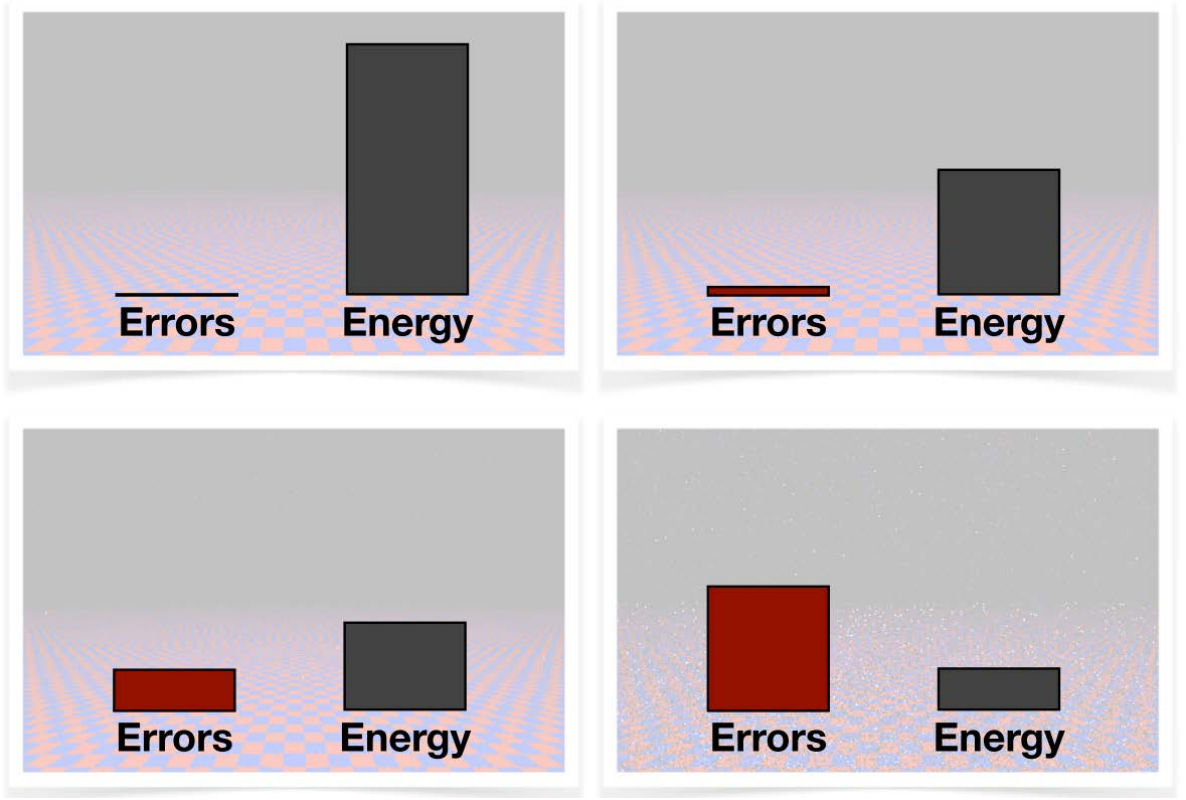




[Image Source: http://arch2neu.saclay.inria.fr/NIAC/luis_ceze.pdf]

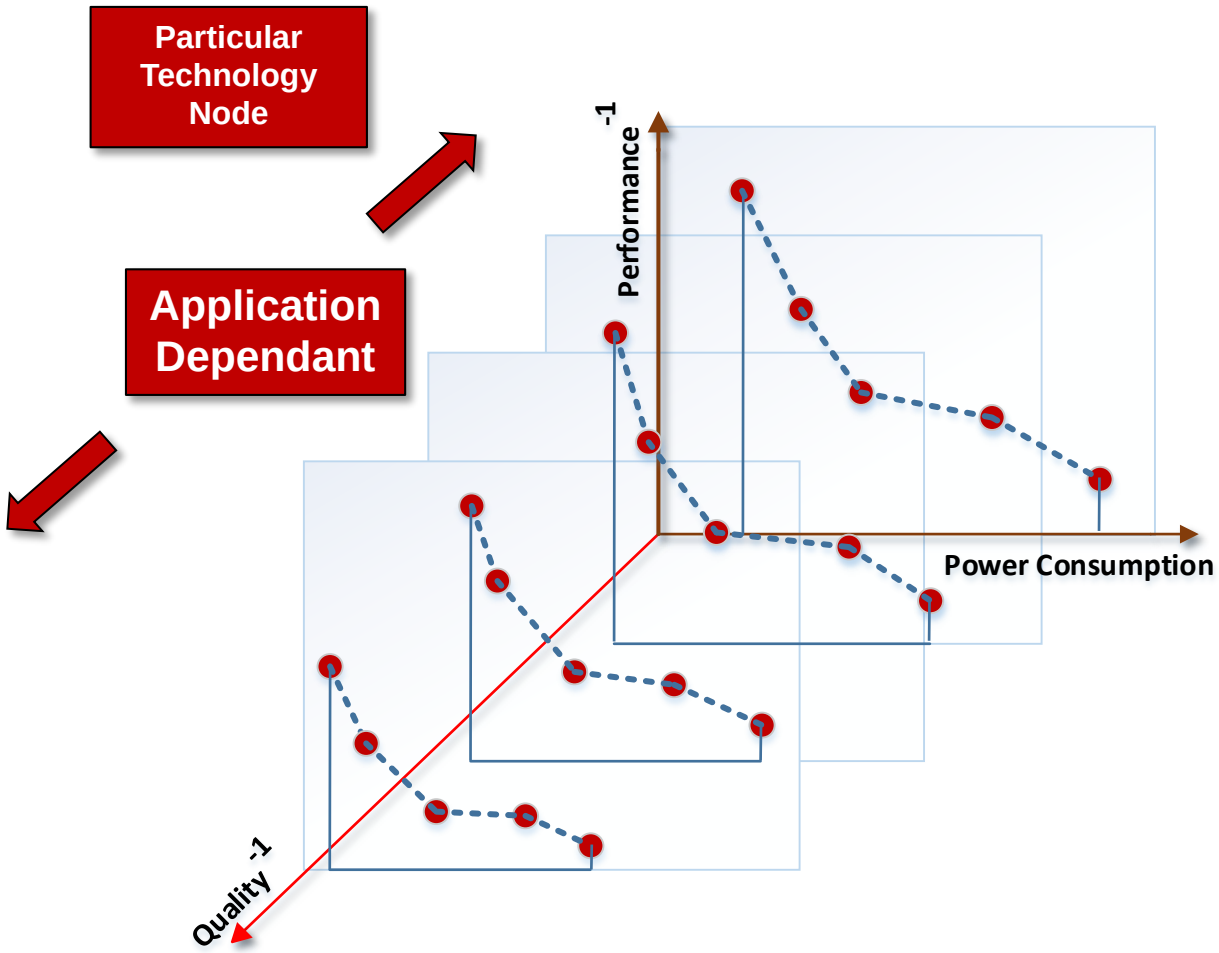
What is “approximate computing”?





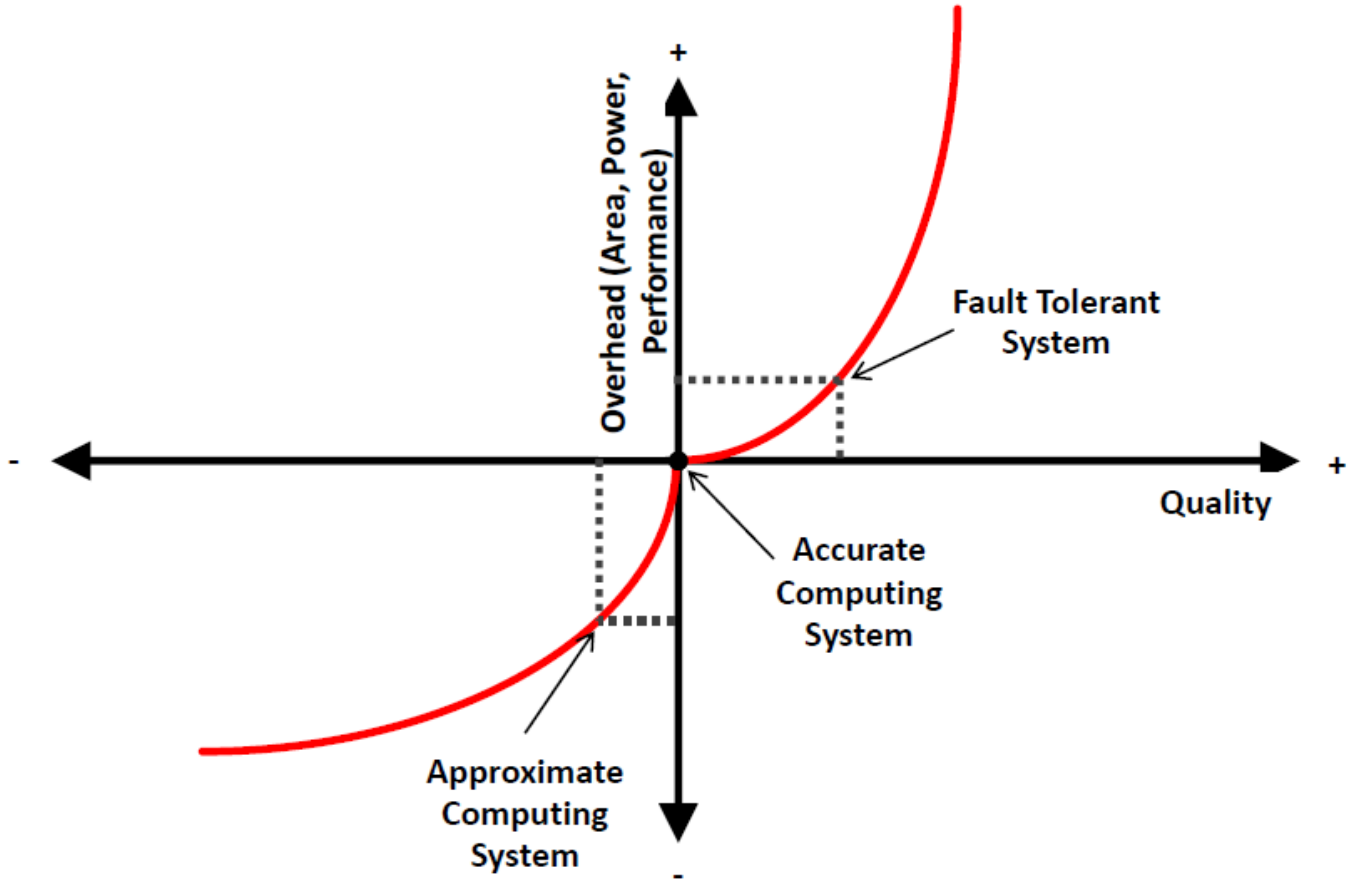
[Image Source: http://arch2neu.saclay.inria.fr/NIAC/luis_ceze.pdf]

What is “approximate computing”?



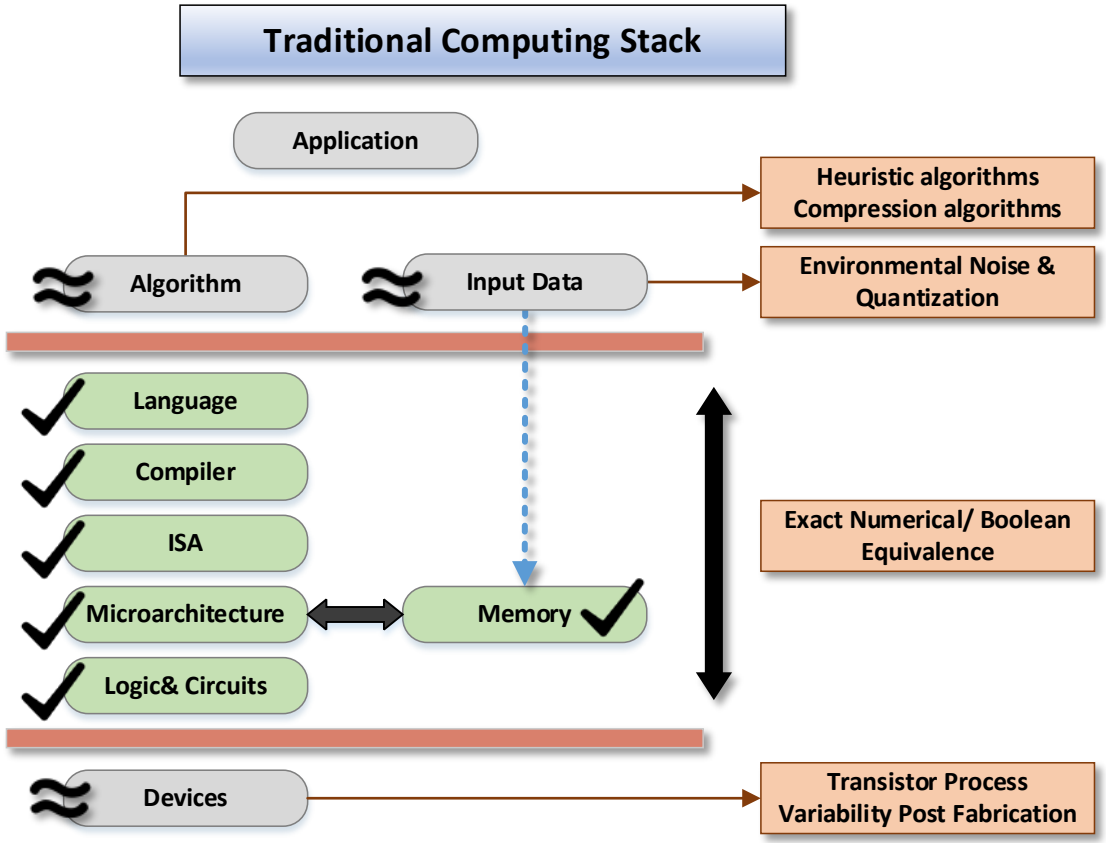
[Image Source: http://arch2neu.saclay.inria.fr/NIAC/luis_ceze.pdf]

Opposite View to Traditional Computing

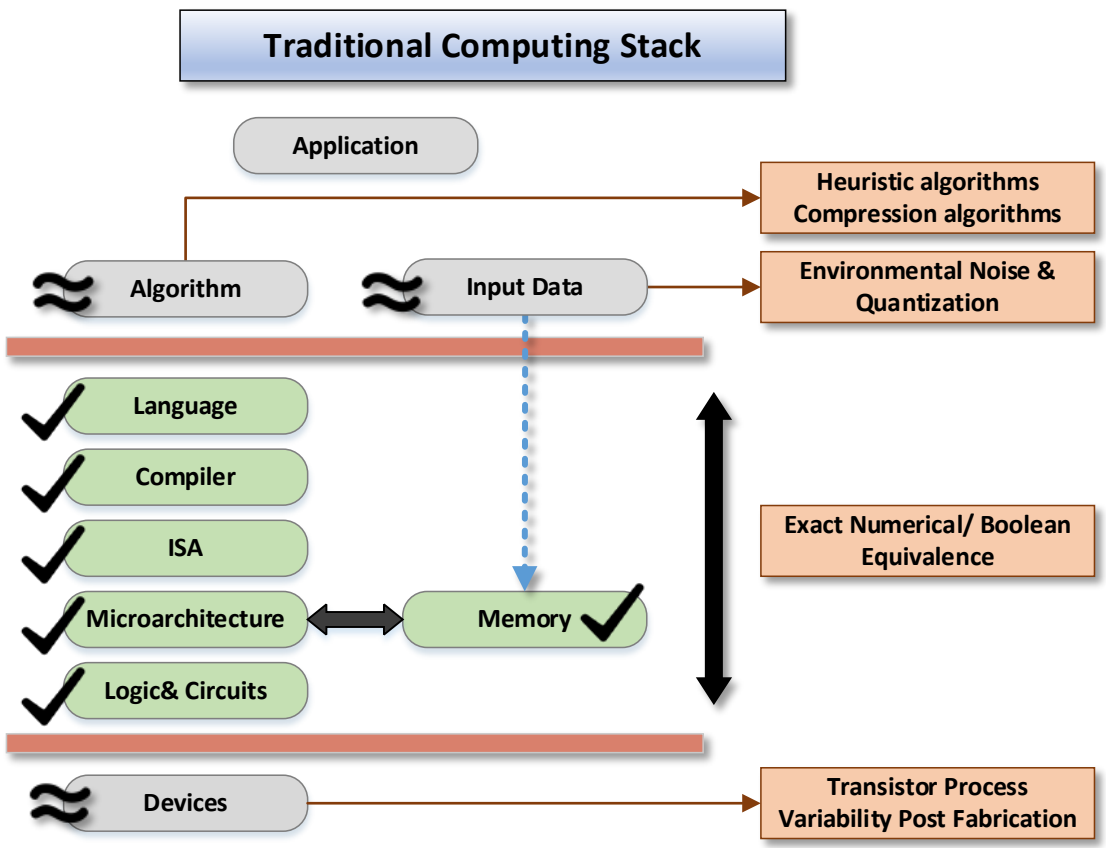


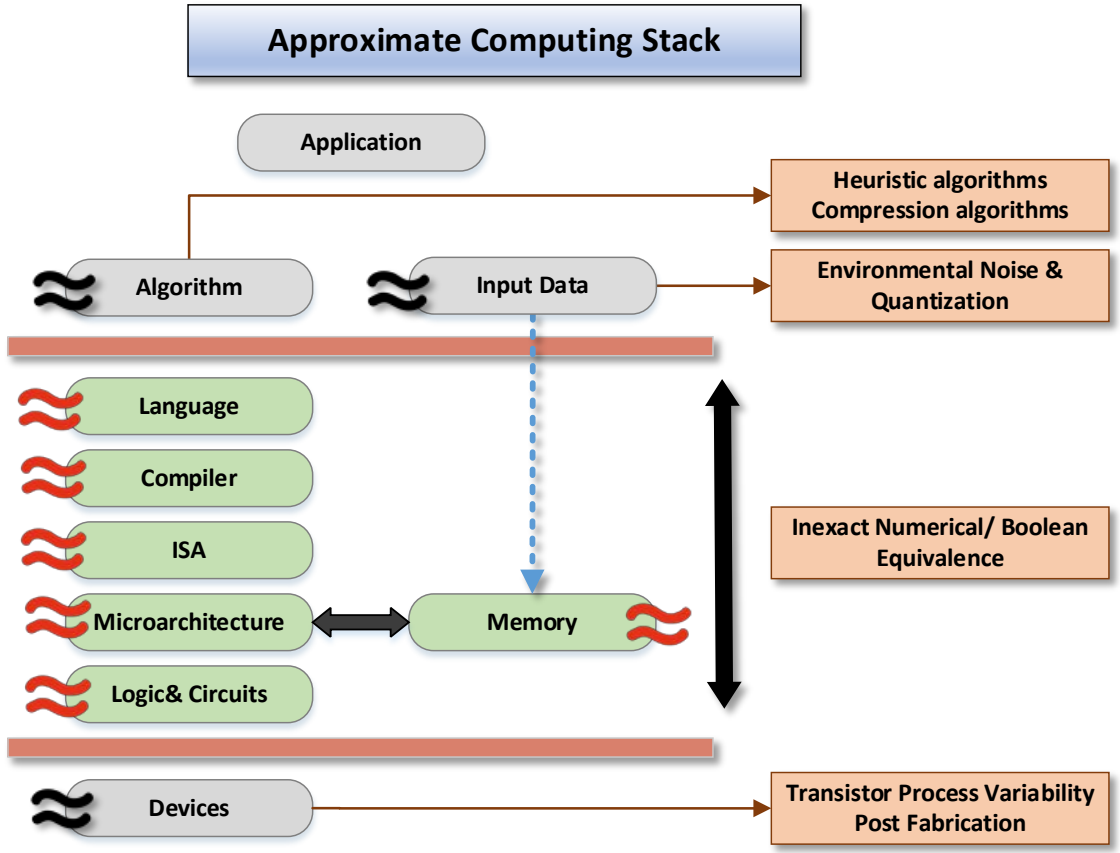
[11] [Source: A Comprehensive Analysis of Approximate Computing Techniques: From Component- to Application-Level [DATE 2019]

Approximations across the system stack

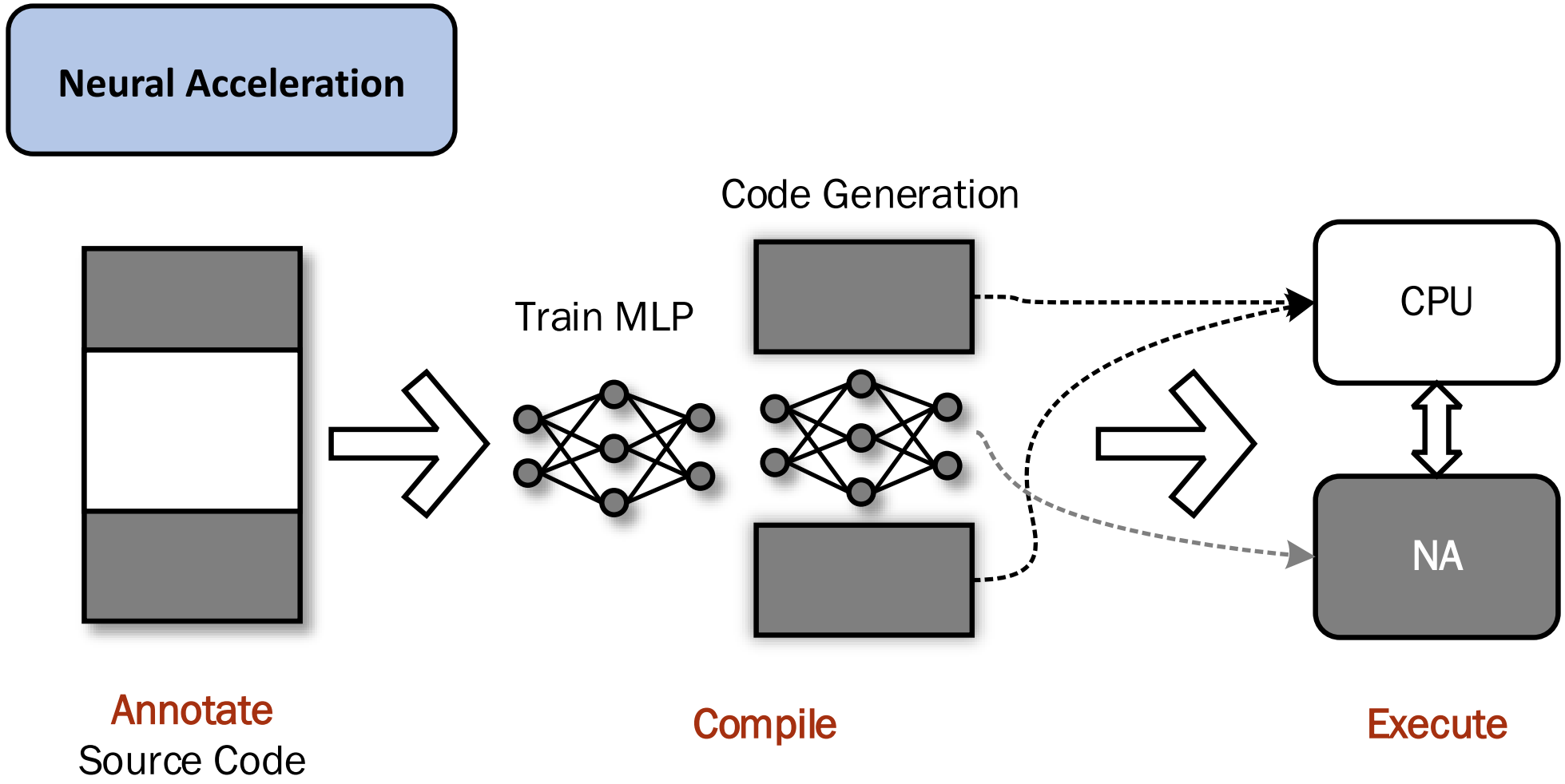


Approximations across the system stack





Some Examples



Neural Acceleration for General-Purpose Approximate Programs, Sampson et al.[MICRO 2012]

Some Examples

Loop Perforation

Some Examples

Loop Perforation

```
for (i = 0; i < n; i += 1) {
        body
 }
```

Some Examples

Loop Perforation

```
for (i = 0; i < n; i += 1) {
    body
}
```

```
for (i = 0; i < n; i += 2) {
    body
}
```

Some Examples

Loop Perforation

```
for (i = 0; i < n; i += 1) {
    body
}
```

```
for (i = 0; i < n; i += 2) {
    body
}
```

```
for (i = 0; i < n; i += 4) {
    body
}
```

Some Examples

Loop Perforation

```
for (i = 0; i < n; i += 1) {
    body
}
```

```
for (i = 0; i < n; i += 2) {
    body
}
```

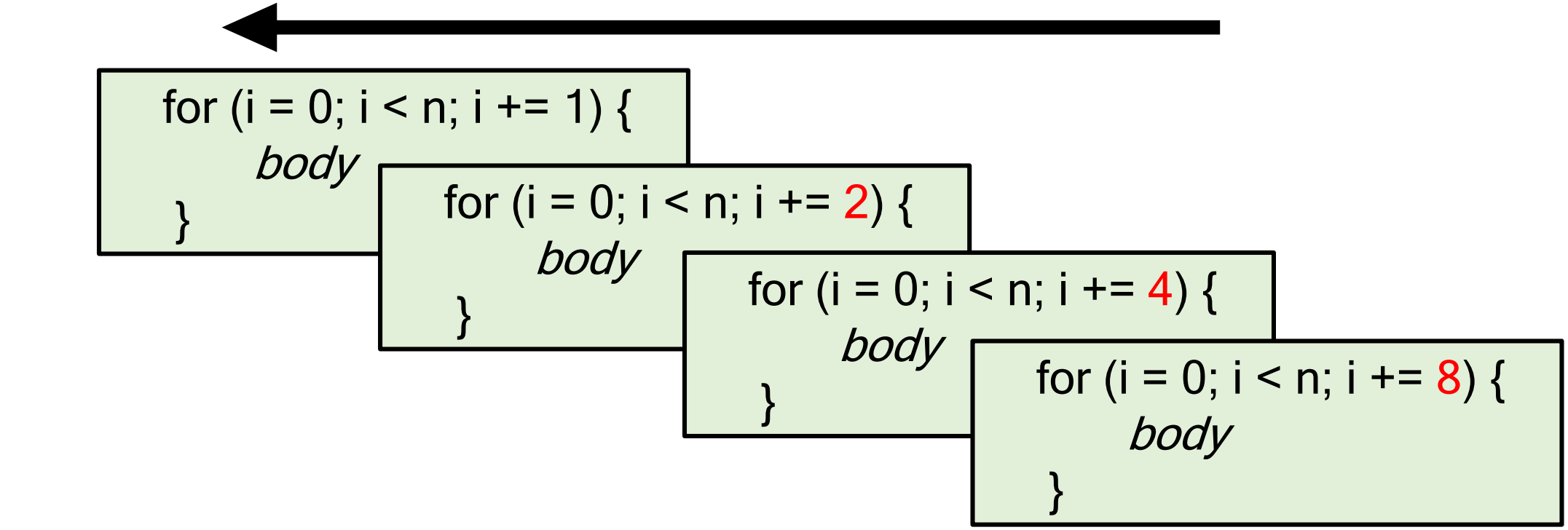
```
for (i = 0; i < n; i += 4) {
    body
}
```

```
for (i = 0; i < n; i += 8) {
    body
}
```

Some Examples

Loop Perforation

Quality



Some Examples

Loop Perforation

Quality

```
for (i = 0; i < n; i += 1) {
    body
}
```

```
for (i = 0; i < n; i += 2) {
    body
}
```

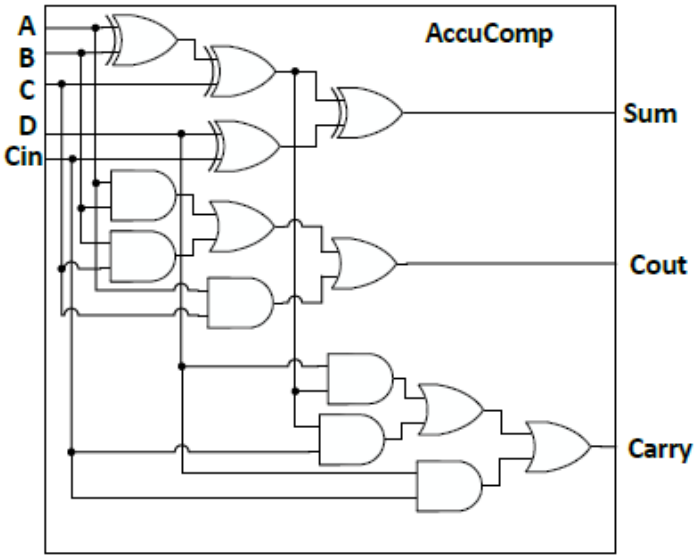
```
for (i = 0; i < n; i += 4) {
    body
}
```

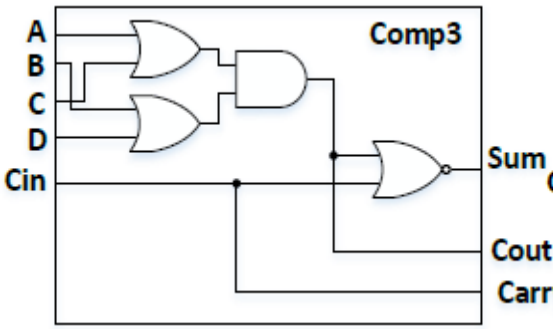
```
for (i = 0; i < n; i += 8) {
    body
}
```

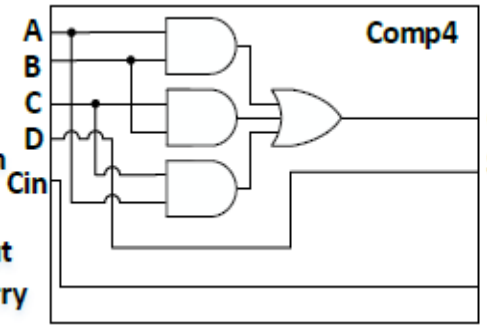
Save Execution Time

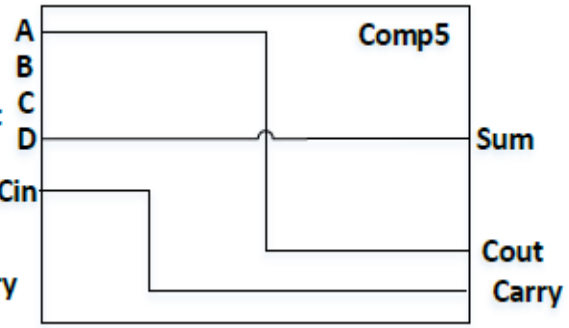
Some Examples

Imprecise Hardware





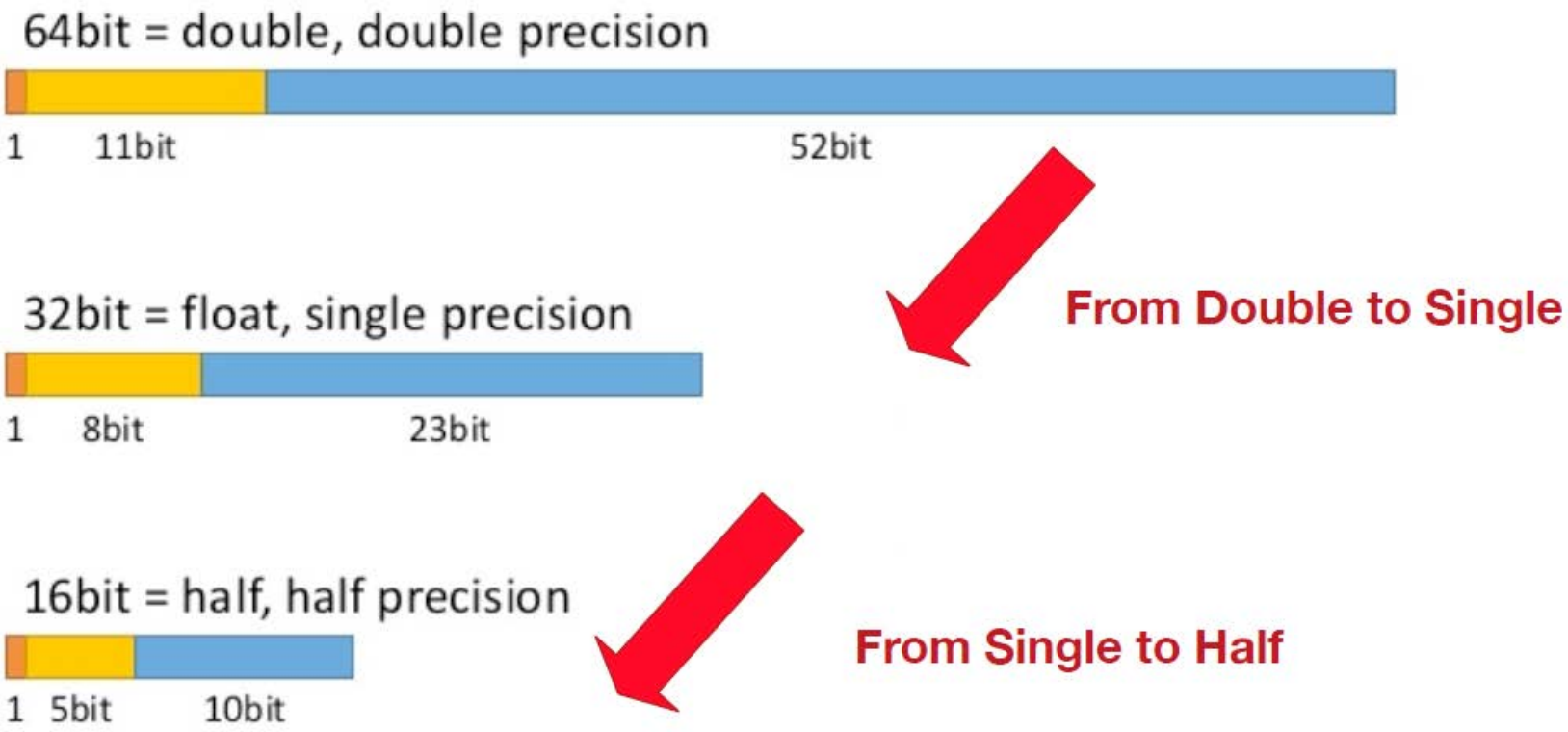




Designing Energy Efficient Approximate Multipliers for Neural Acceleration, S. De et al. [DSD 2018]

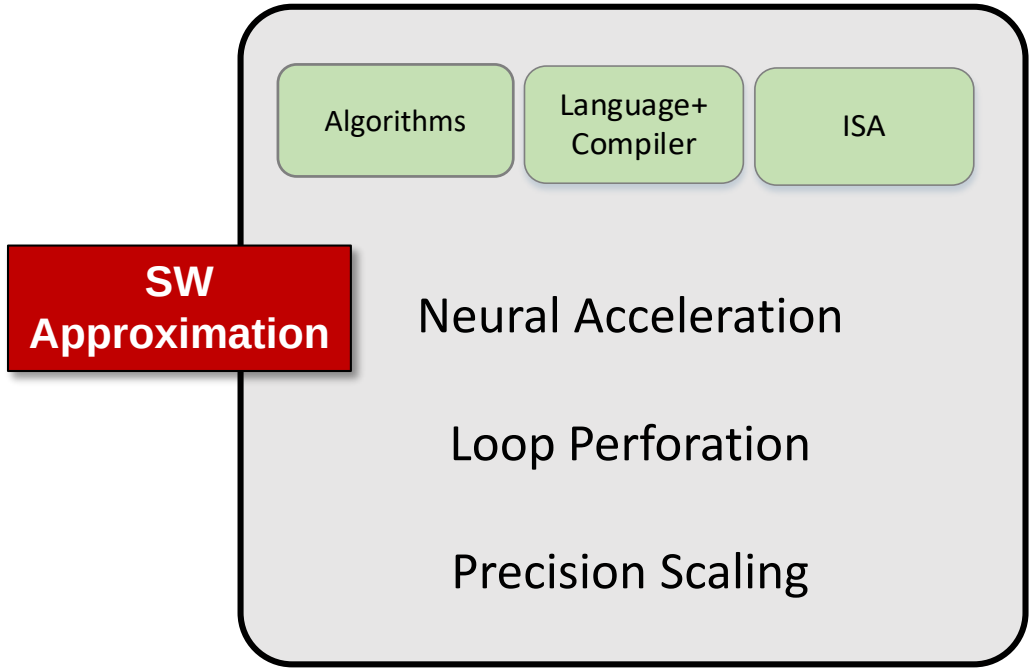
Some Examples

Precision Scaling

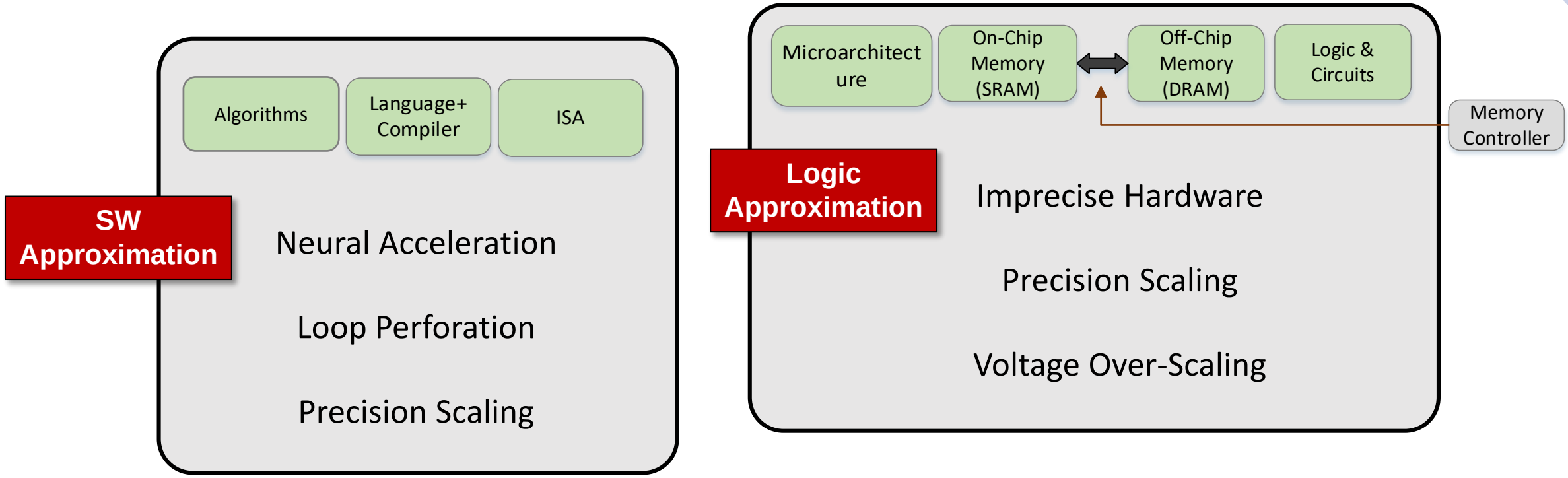


[23] [Source: A Comprehensive Analysis of Approximate Computing Techniques: From Component- to Application-Level [DATE 2019]

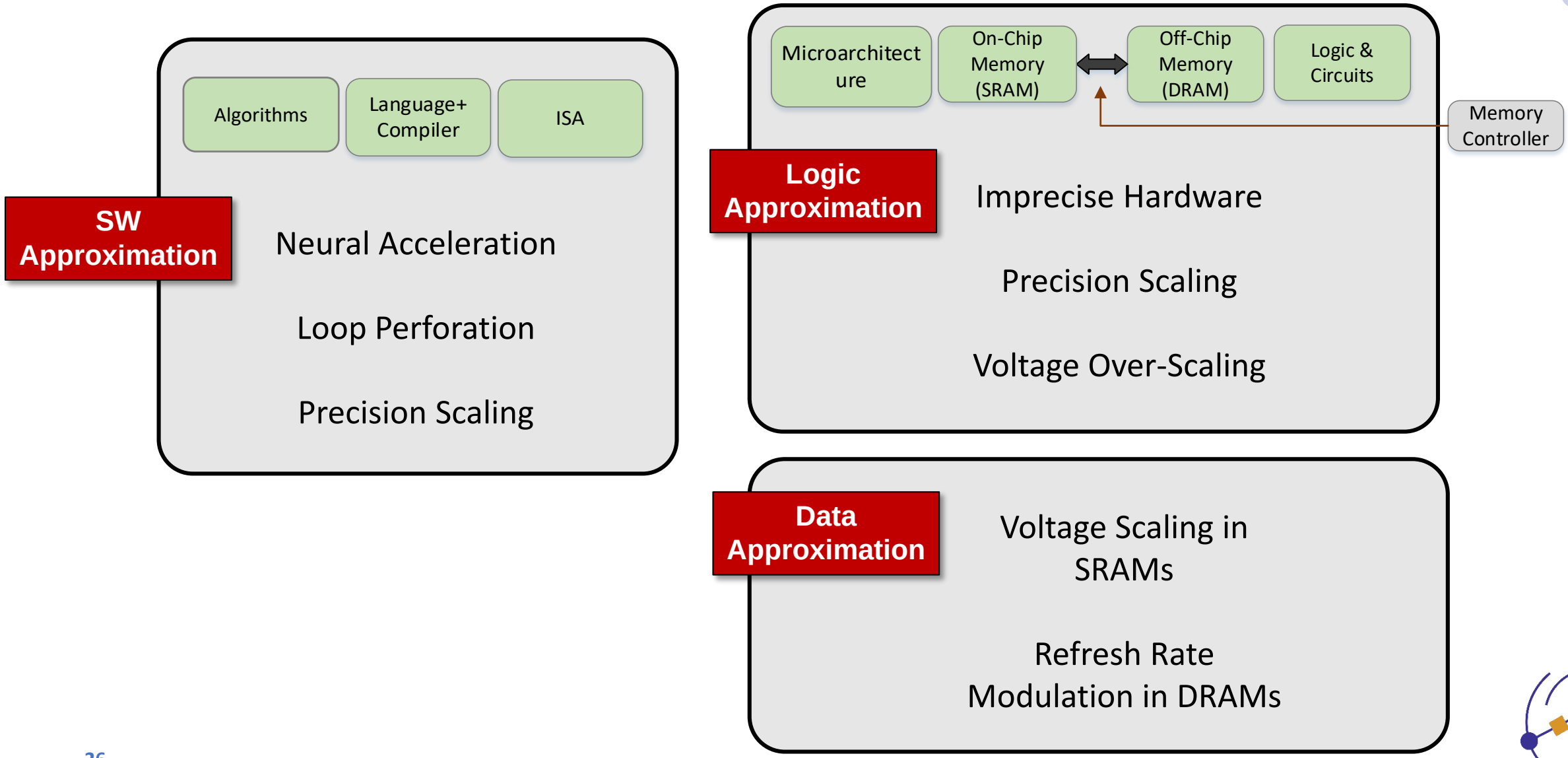
Approximations across the system stack



Approximations across the system stack

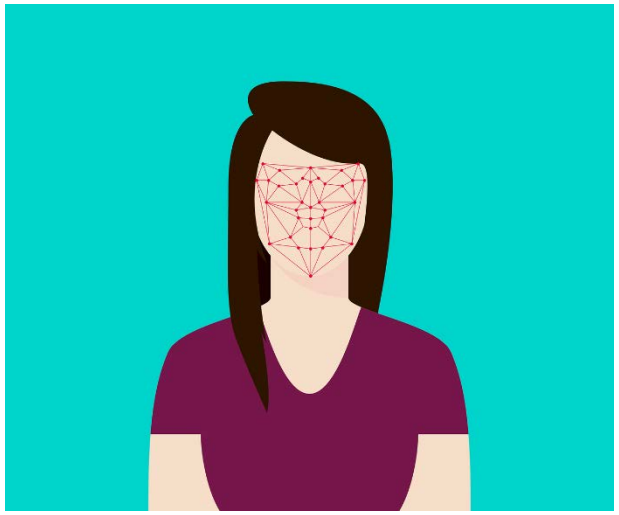


Approximations across the system stack

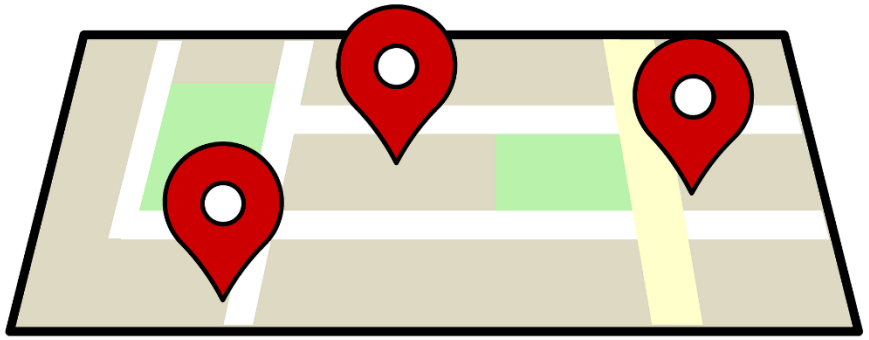


Error Resilient Applications

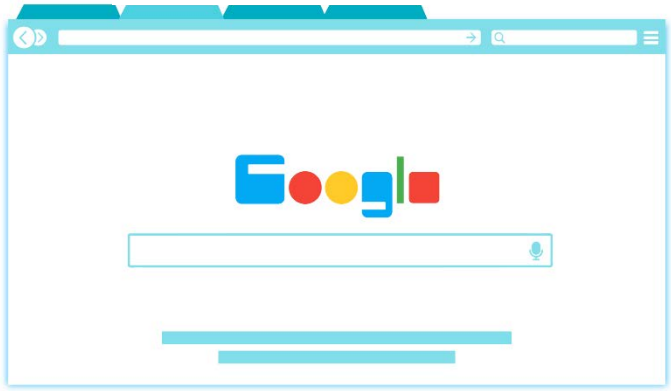
Image Processing & Compression



Biometric Security



Navigation



Web Browsing

No single accurate result!!!

Image Processing benefits from Approximate Computing!

But they are always part of bigger systems.

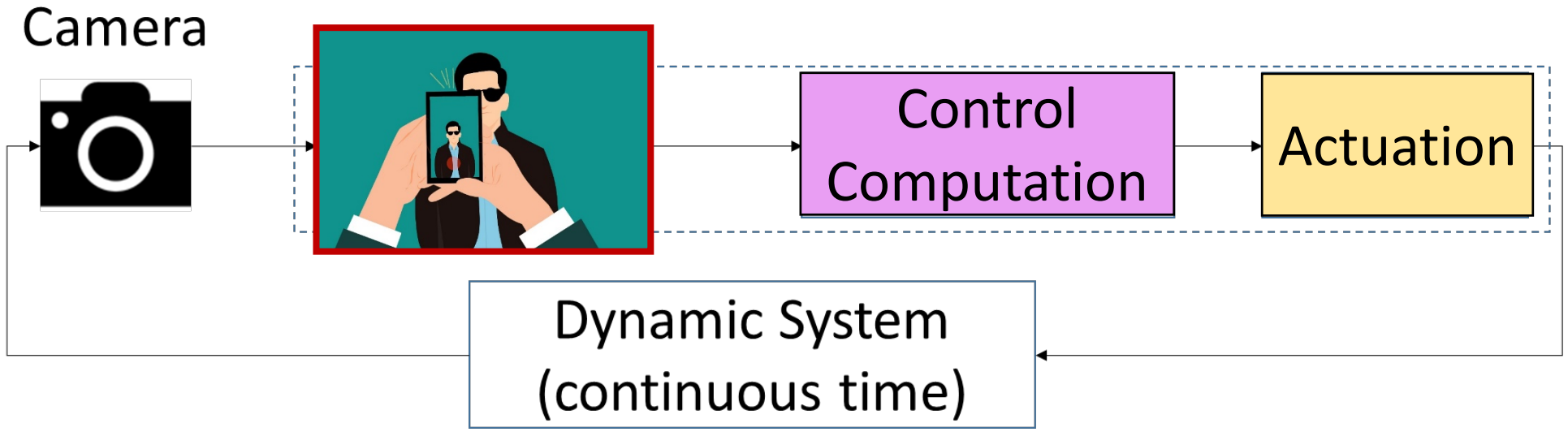
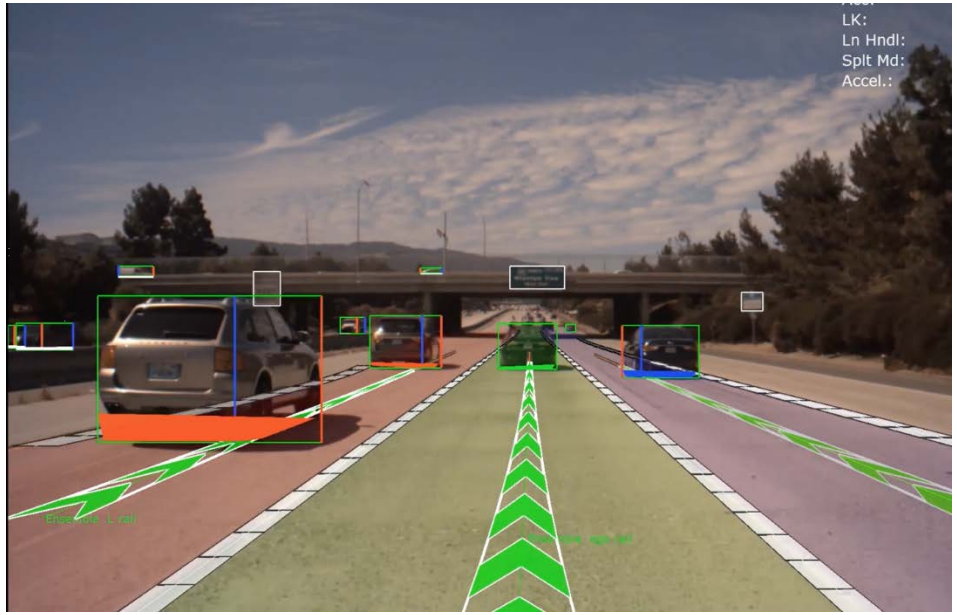
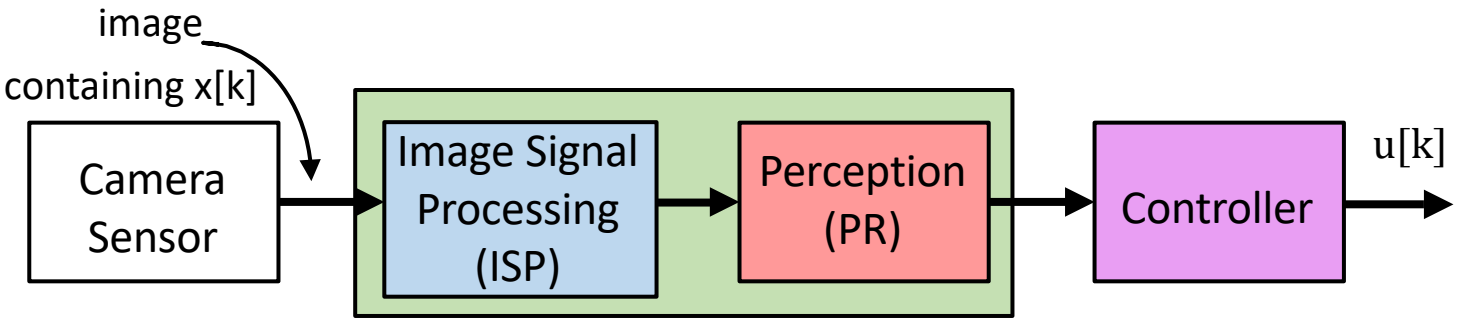


Image-approximation in-the-loop

Lane keeping assist system (LKAS)

Lane keeping assist system (LKAS)

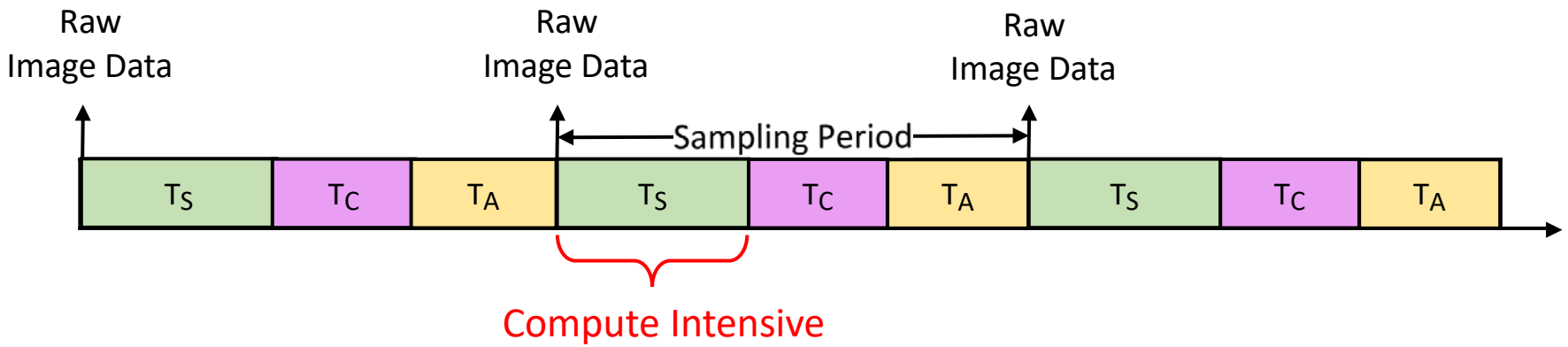


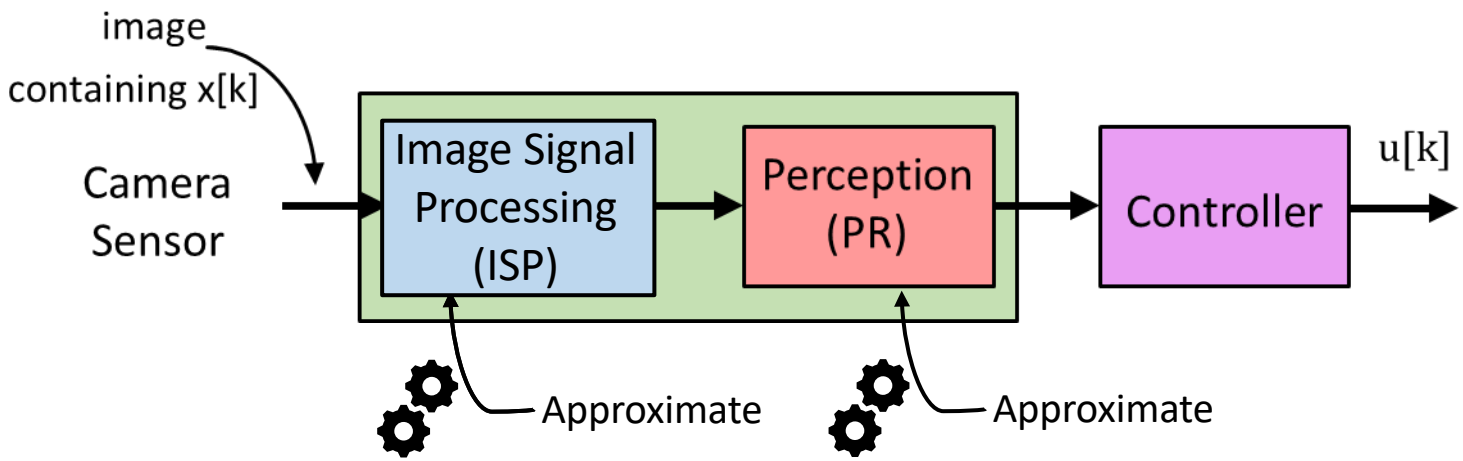


What is the impact of image approximation on the bigger closed-loop system?

How to analyse this impact?

What should we approximate?

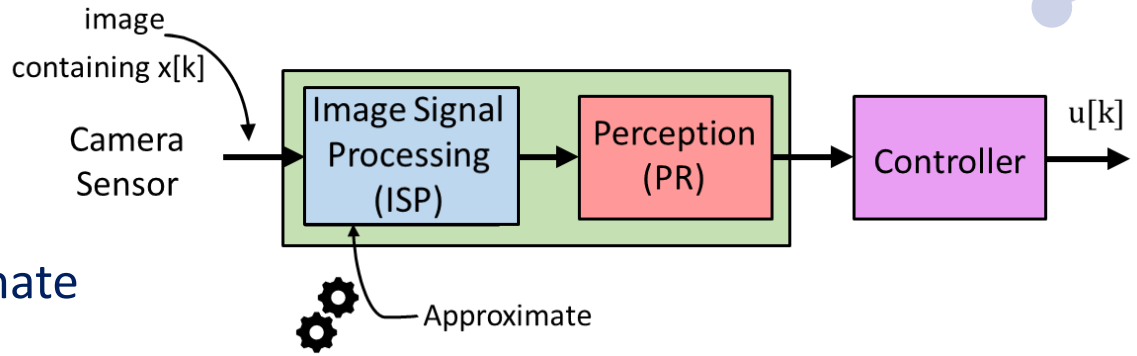


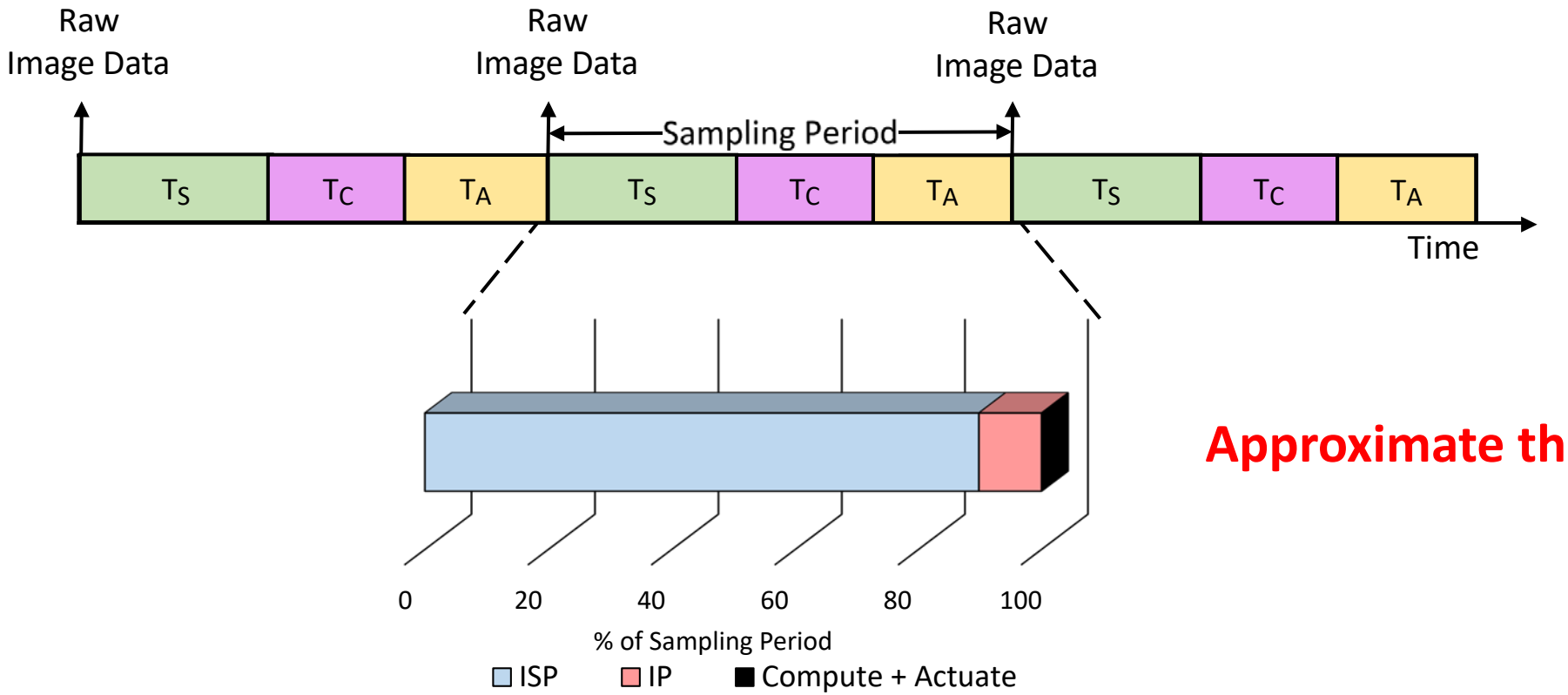


What should we approximate?

Execution time analysis: profiling

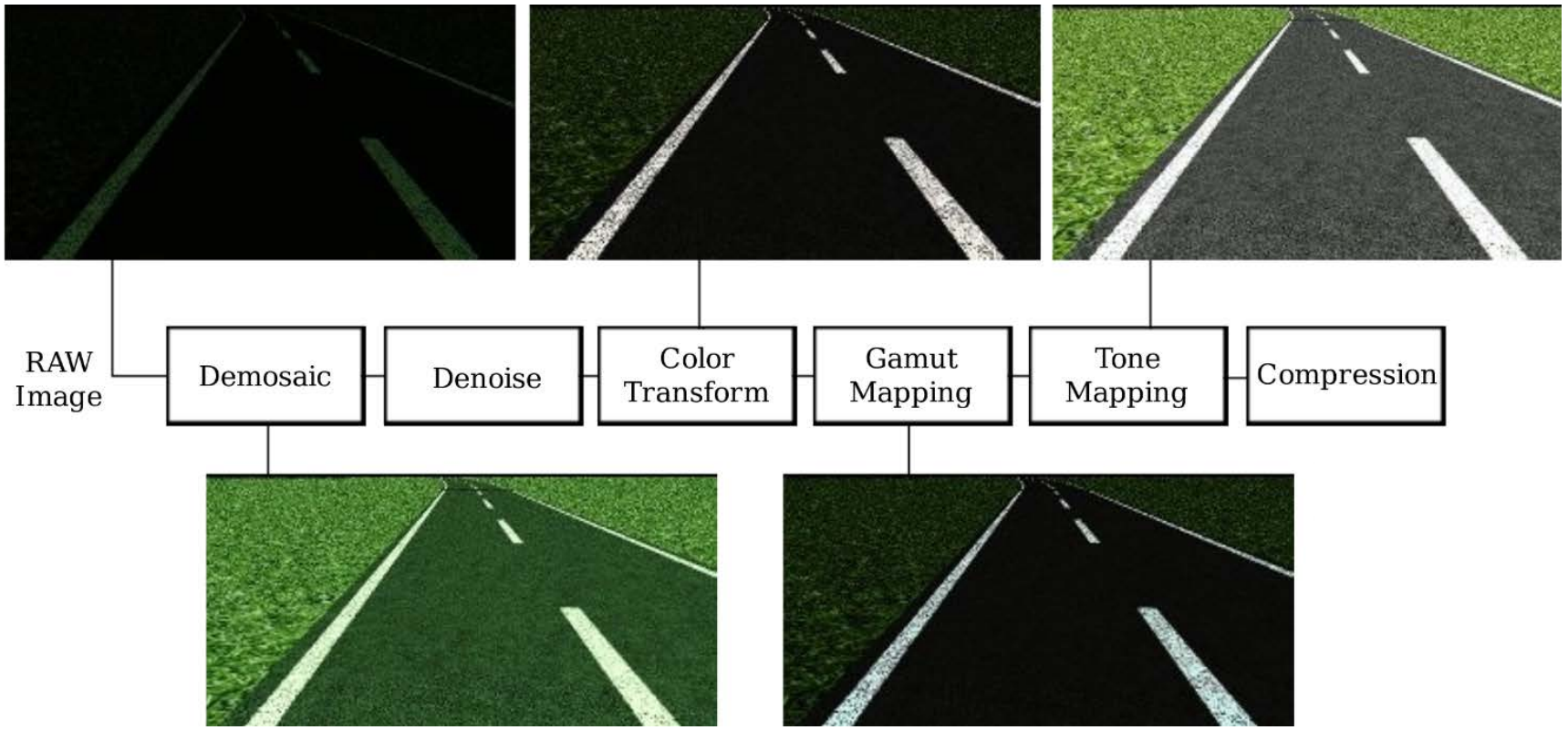
- Intel i7 processor @2.6GHz
- ISP stage is the most compute-intensive → approximate





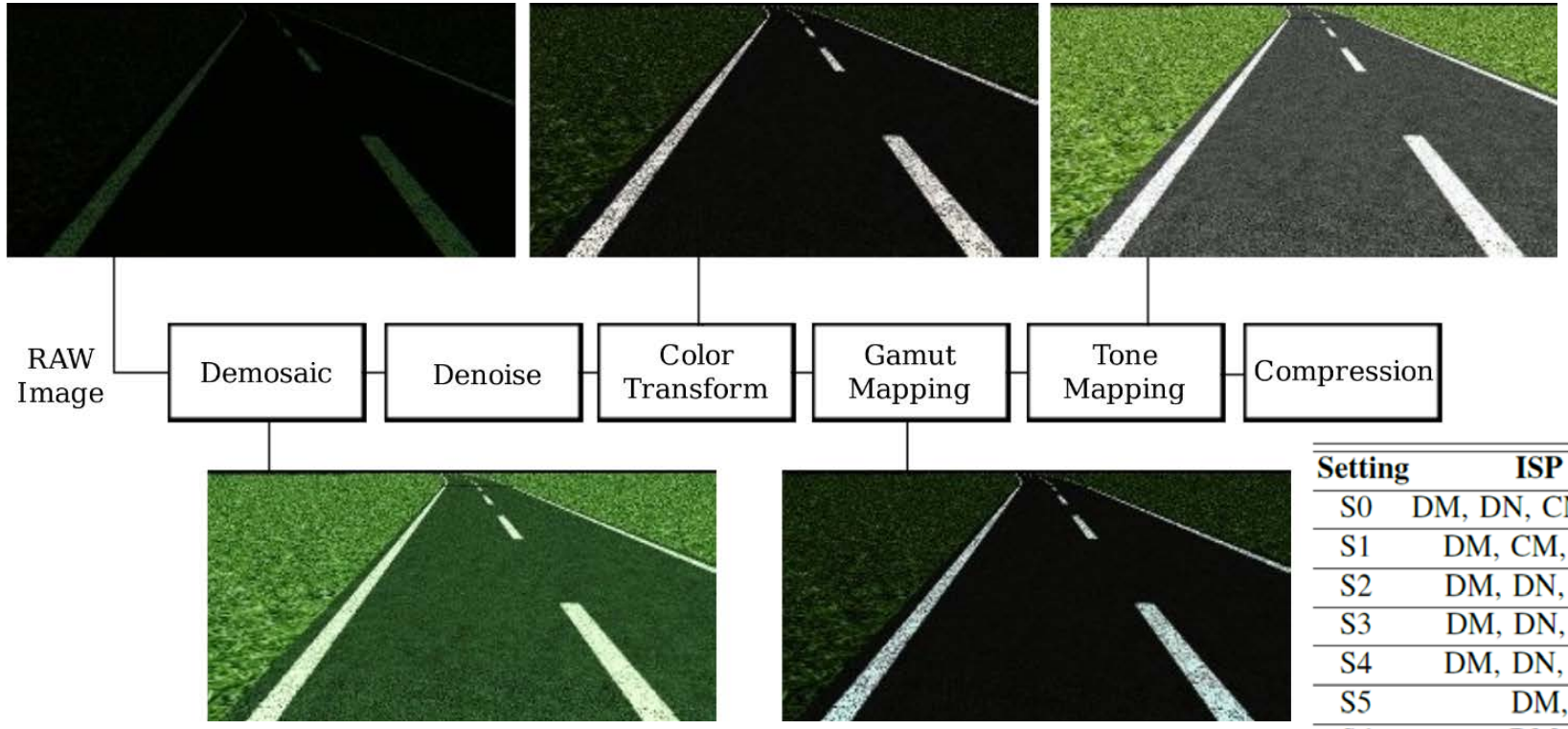
Approximate the ISP!!

Image Signal Processor [ISP]



Traditional pipelines are optimized for vision.
Do we really need a vision optimized pipeline for control?

ISP Approximation: tuning knobs

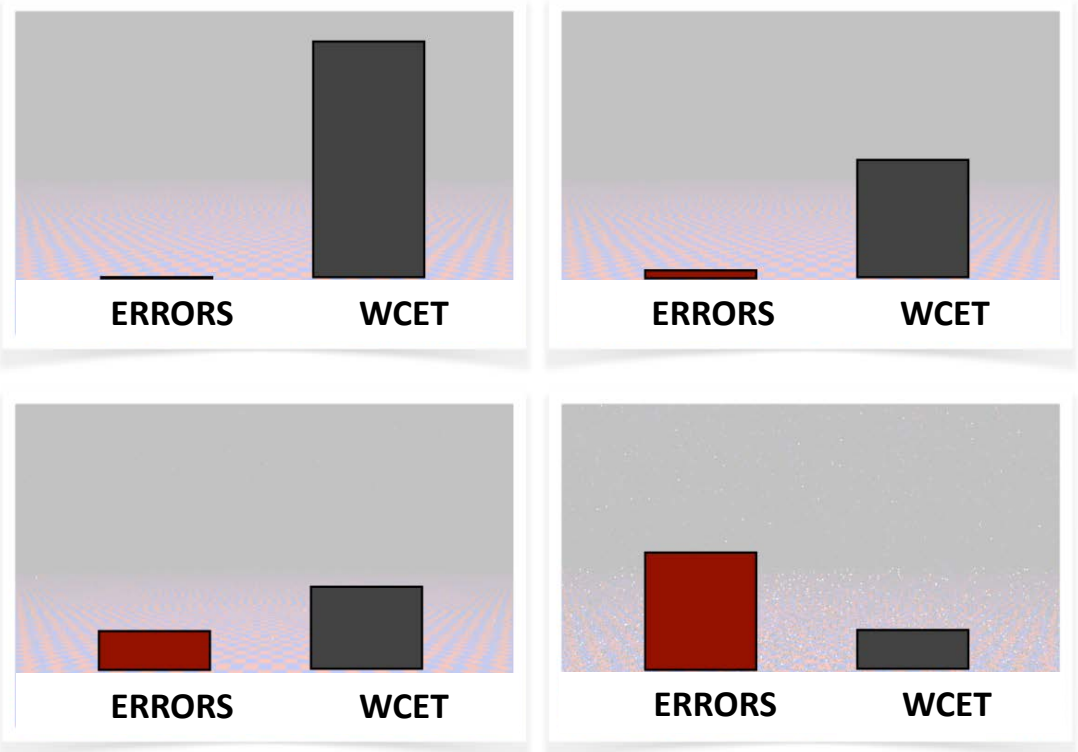


Setting	ISP Stages	Description
S0	DM, DN, CM, GM, TM, C	Accurate (all stages included)
S1	DM, CM, GM, TM, C	Skip Denoising
S2	DM, DN, GM, TM, C	Skip Color Mapping
S3	DM, DN, CM, TM, C	Skip Gamut Mapping
S4	DM, DN, CM, GM, C	Skip Tone Mapping
S5	DM, DN, C	Keep only Denoising
S6	DM, CM, C	Keep only Color Mapping
S7	DM, GM, C	Keep only Gamut Mapping
S8	DM, TM, C	Keep only Tone Mapping

Reconfiguring the Imaging Pipeline for Computer Vision, Buckler et al. [ICCV 2017]

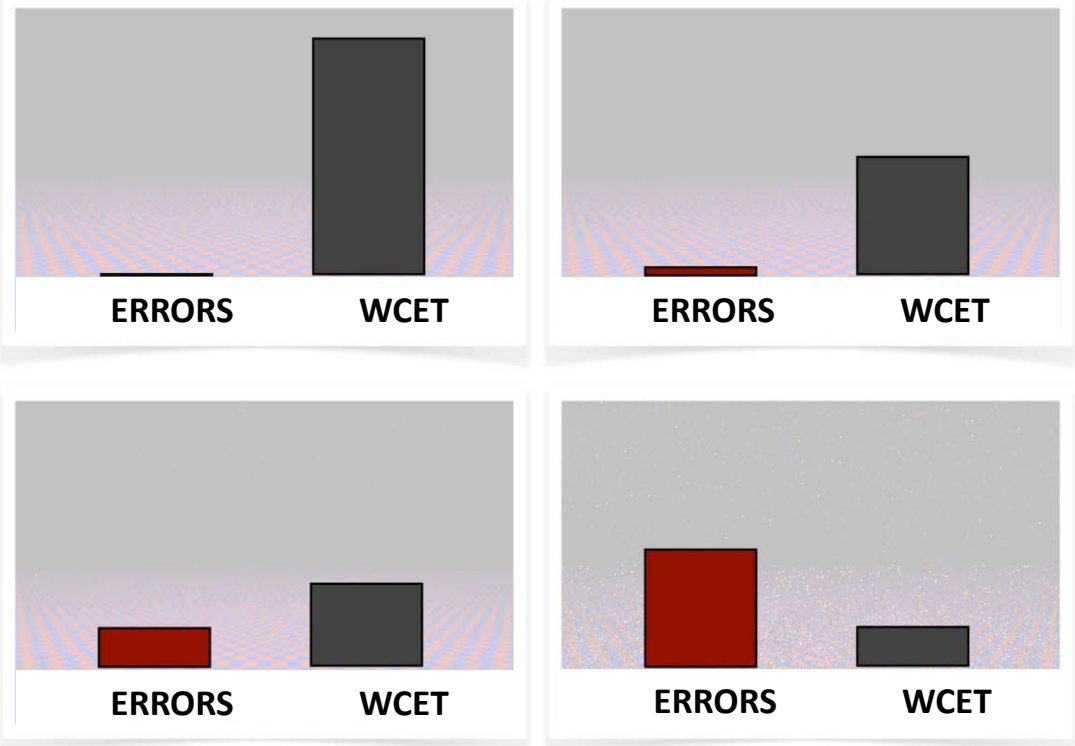
What is the impact of approximation on IBC?

Reduced Execution Time



What is the impact of approximation on IBC?

Reduced Execution Time



Loss in Image Quality

What is the impact of approximation on IBC?

Reduced Execution Time

→ shorten both sampling period h and delay τ

→ better control performance

Loss in Image Quality

→ inaccurate computation of state y_L

→ errors might be significant

What is the impact of approximation on IBC?

Reduced Execution Time —due to→ **Degree of Approximation**

Their interplay determines if we gain or lose

Loss in Image Quality —due to→ **Quality-of-Control (QoC)**

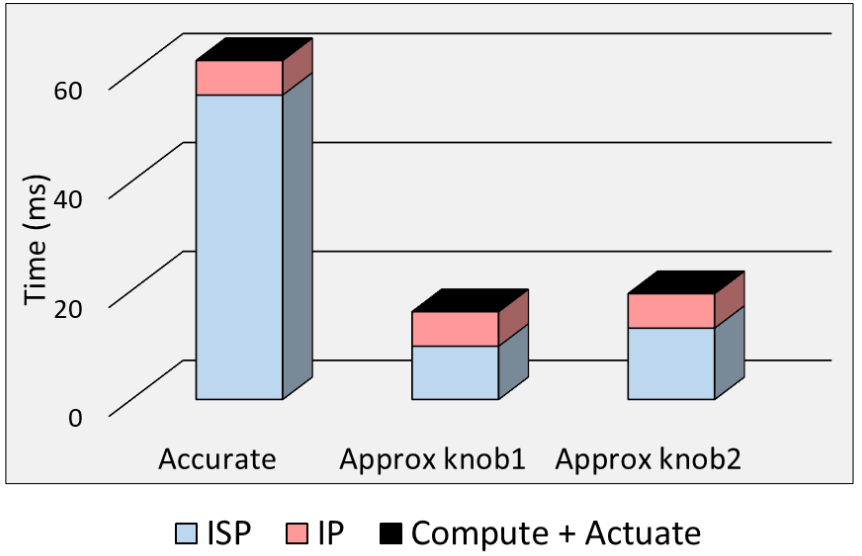
How can we gain on IBC system performance?

Impact on execution times: Profiling

Output of accurate$_{ISP}$

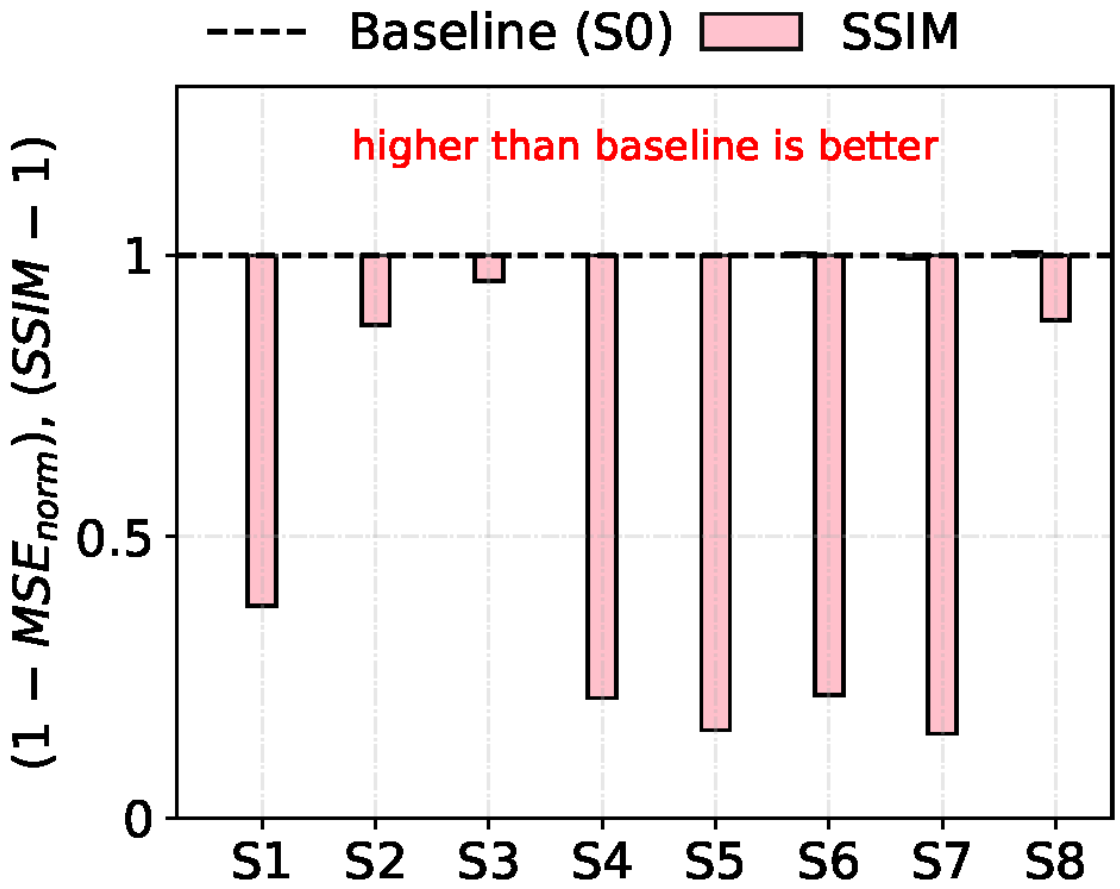
Output of approximate$_{ISP}$ knob2

Output of approximate$_{ISP}$ knob1



Impact on image quality

- Low SSIM
  - More noise



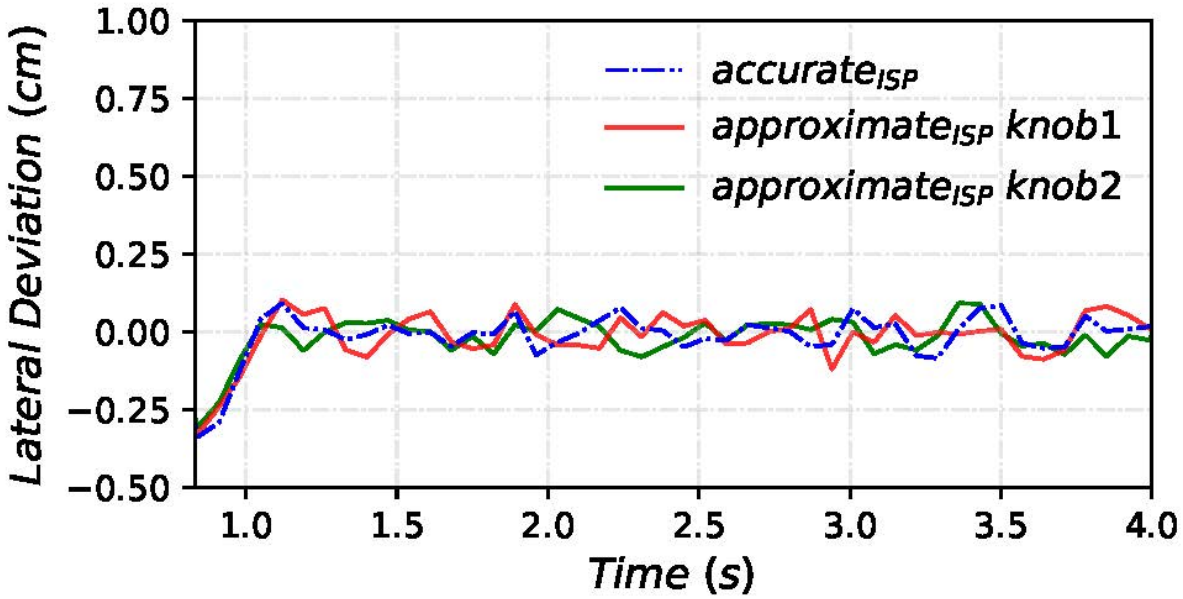
Impact on image quality

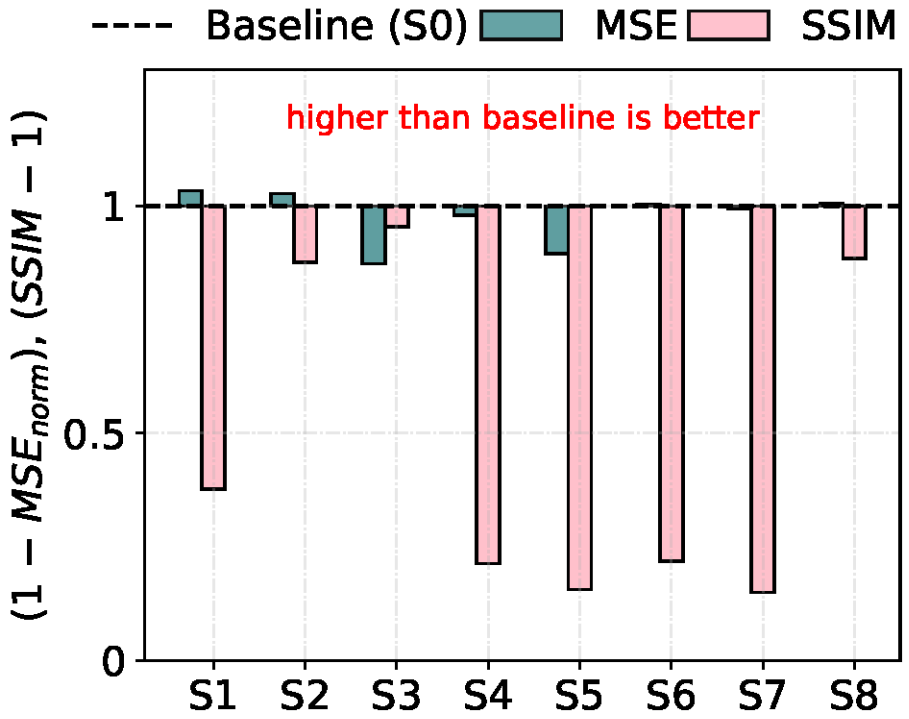
- Features of image may not be detected
  - Algorithm should be resilient to approximation
  - Application-specific testing needed!

Impact of image quality on QoC

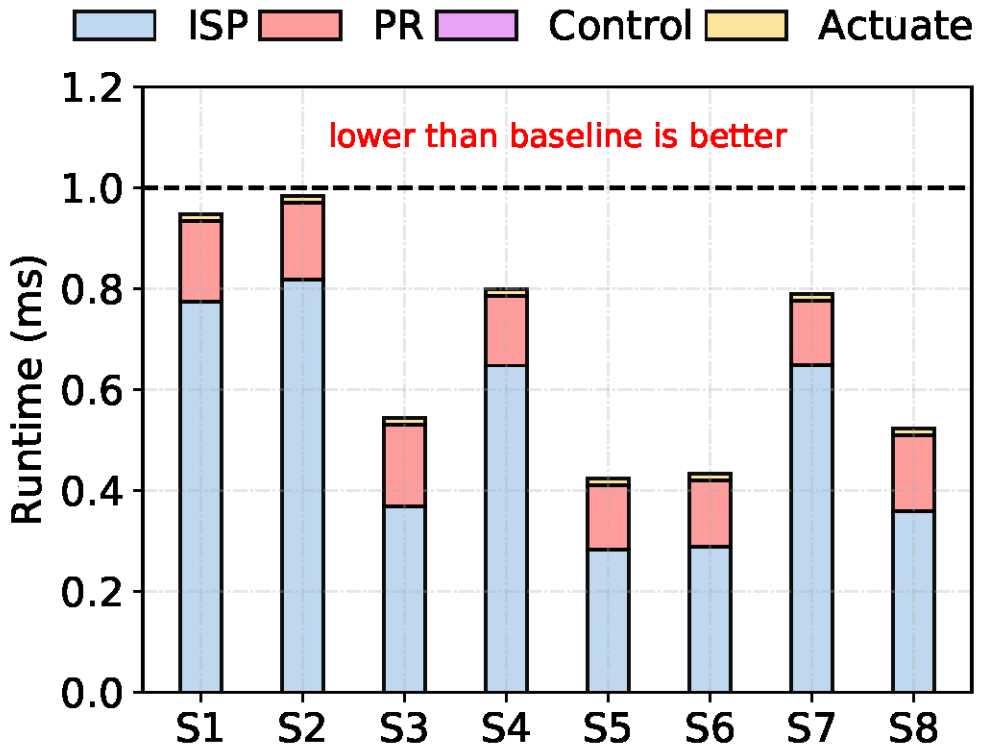
Without considering improved timing

- Performance deteriorates for approximated images (S1 – S8)
  - Still acceptable for control



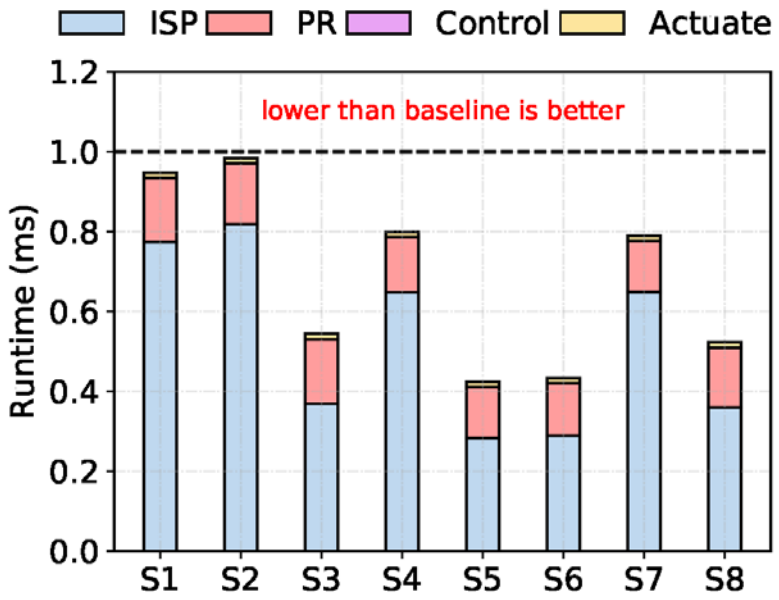


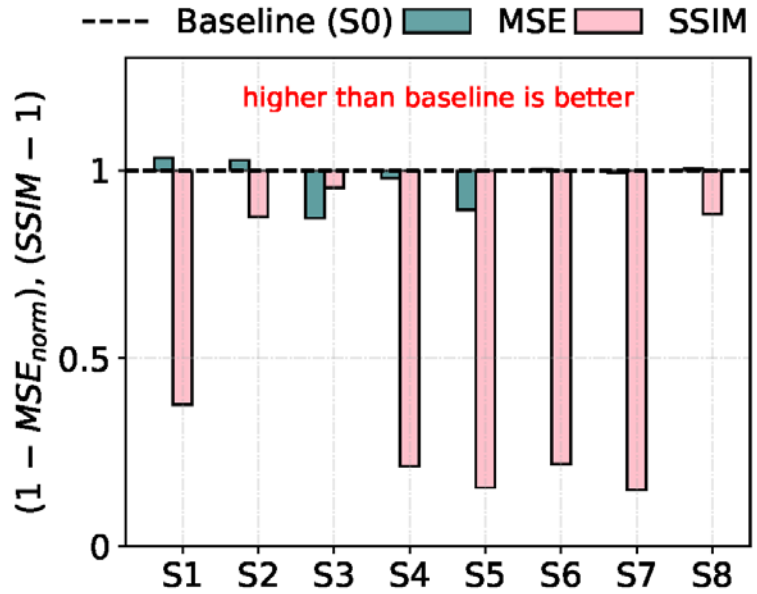
But timing is improved due to approximation!



What is the impact of approximation on IBC?

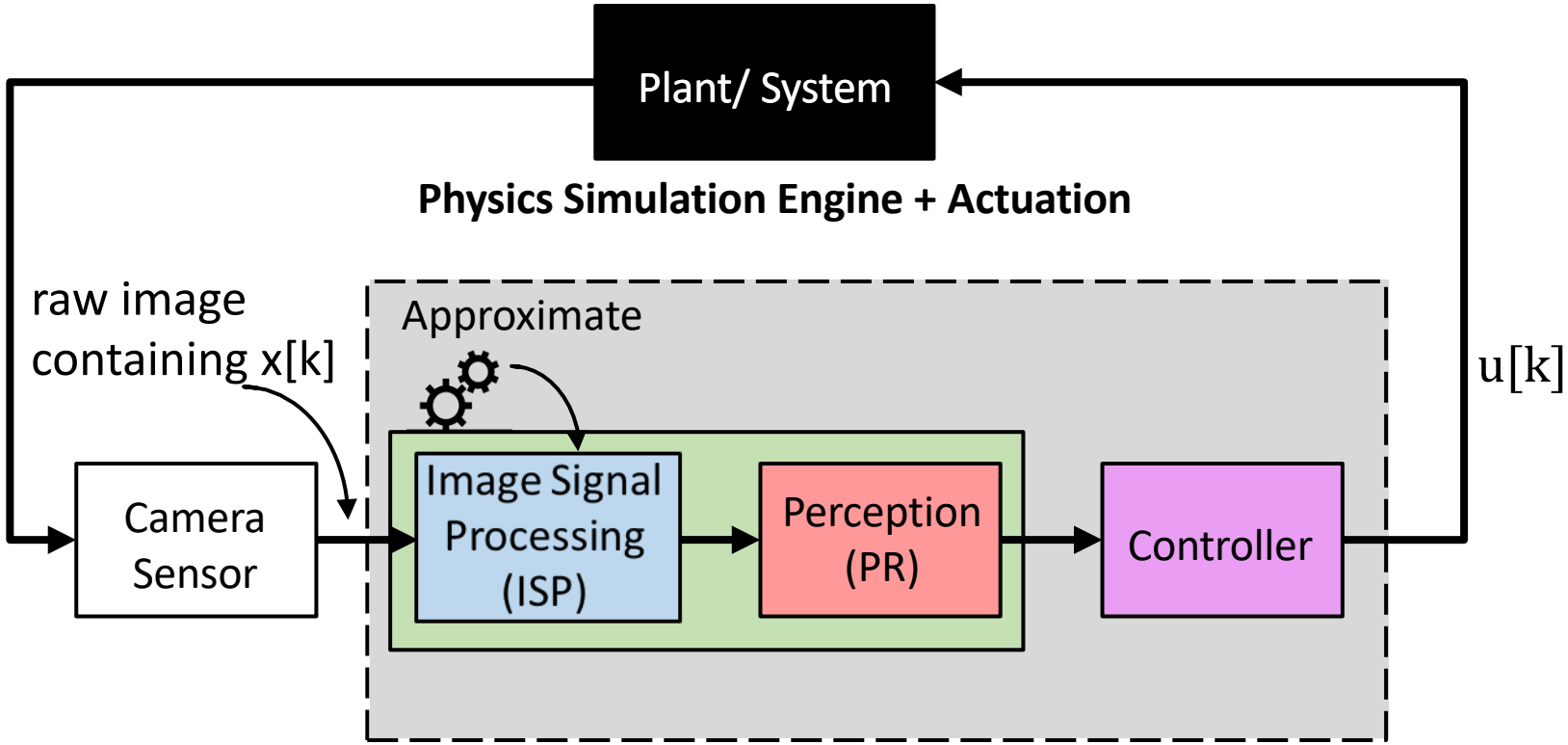
Reduced Execution Time —due to→ **Degree of Approximation**





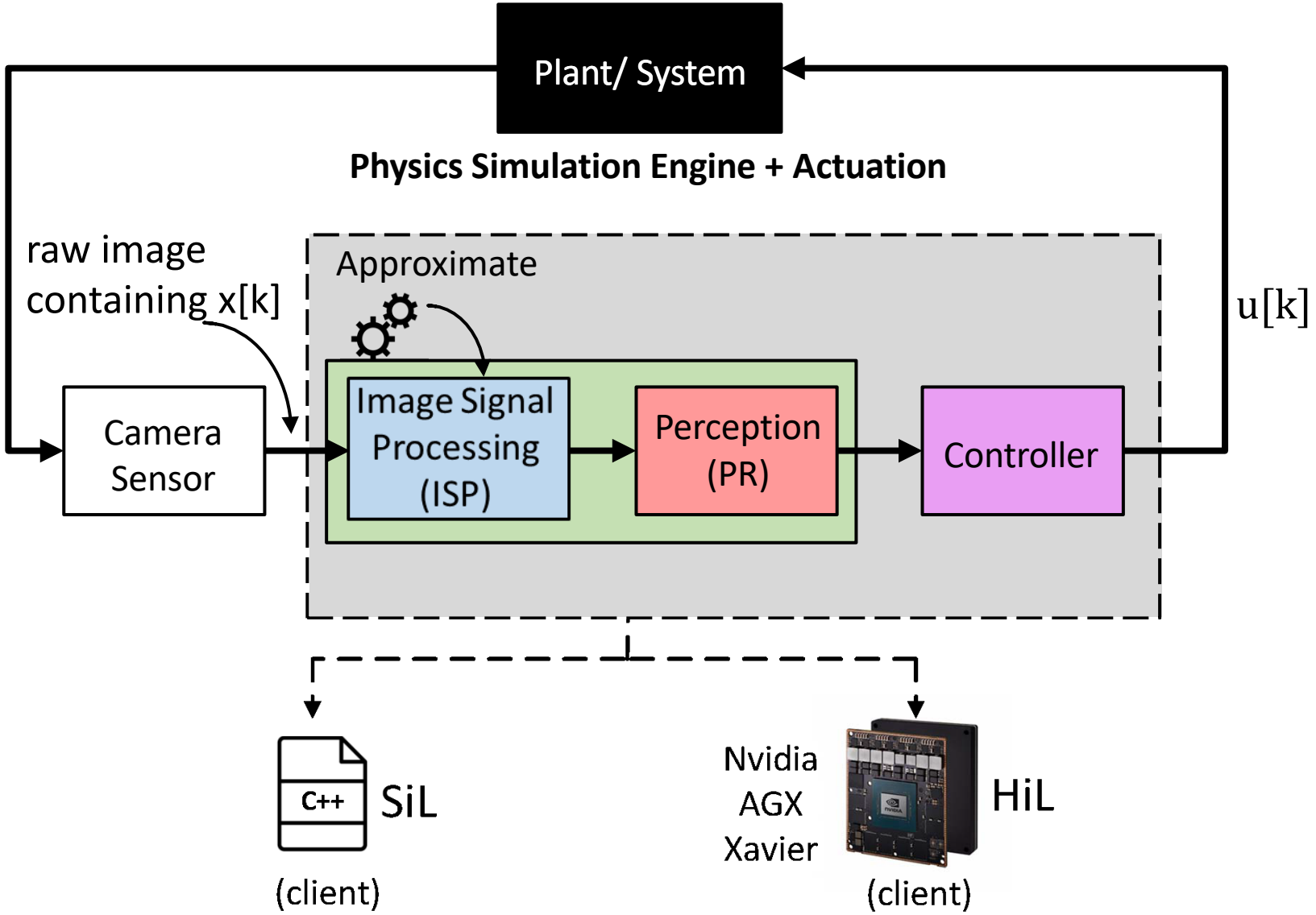
Loss in Image Quality —due to→ **Quality-of-Control (QoC)**

How to check if it is safe to approximate images for safety-critical IBC systems?



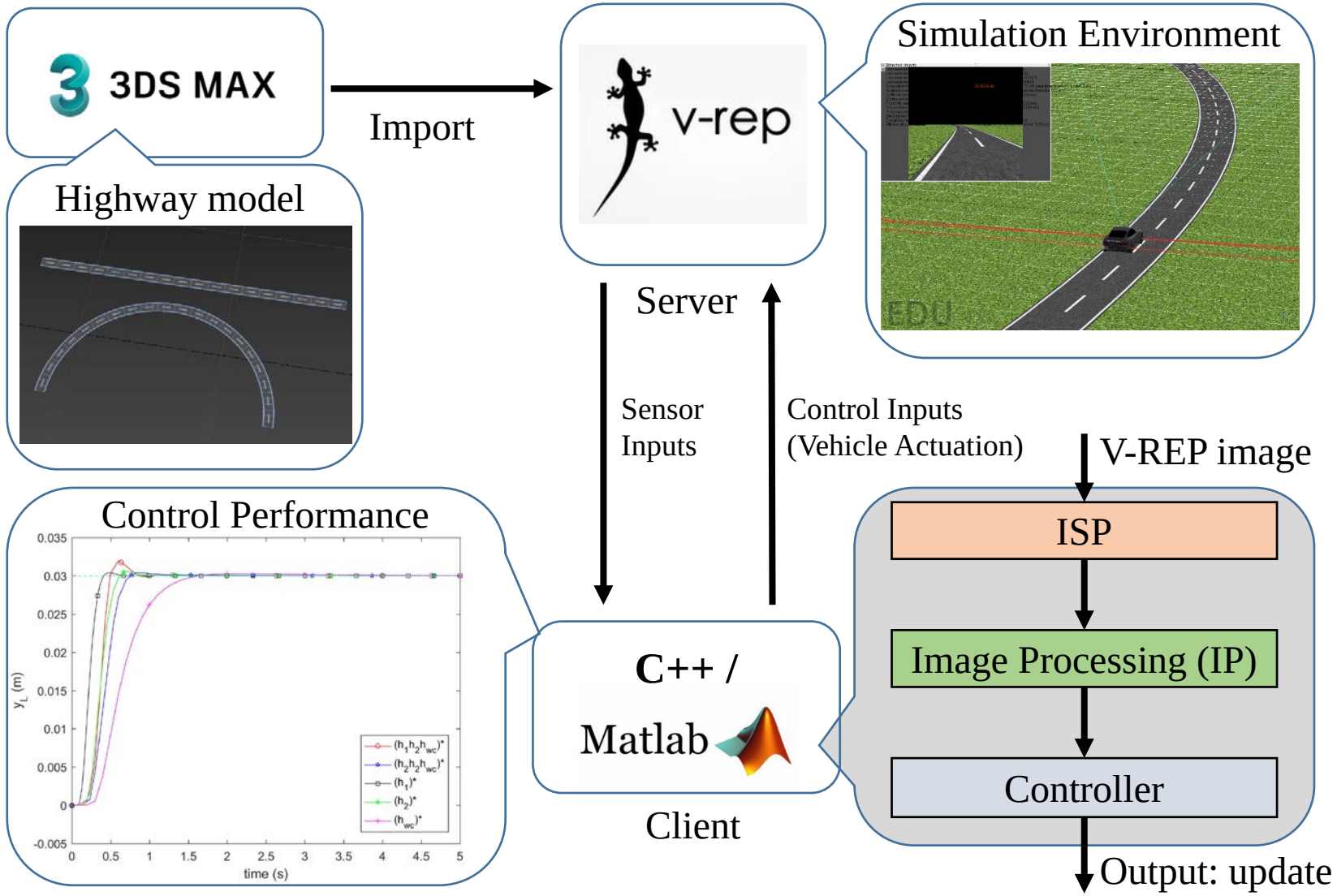
An analysis framework is needed!!!

IMACS framework: software-in-the-loop & hardware-in-the-loop simulation



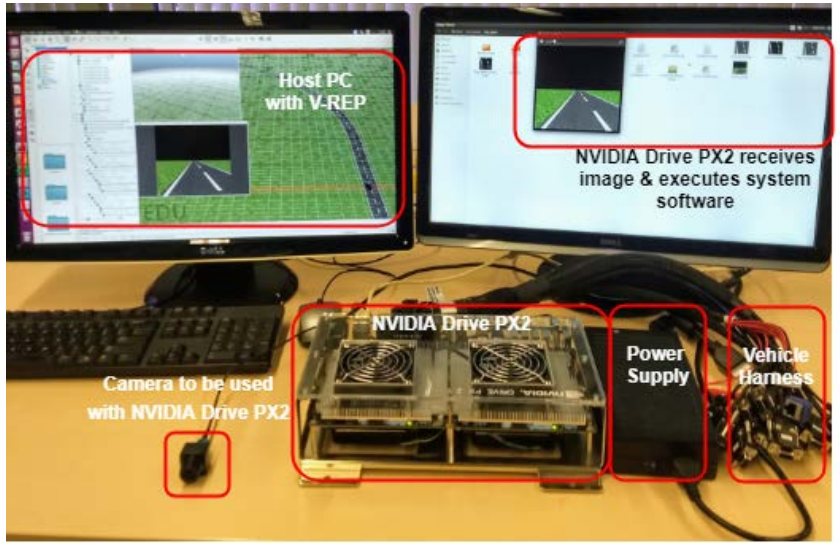
S. Mohamed *et al.*, "IMACS: A Framework for Performance Evaluation of Image Approximation in a Closed-loop System," *MECO*, 2019.

IMACS: SiL simulator



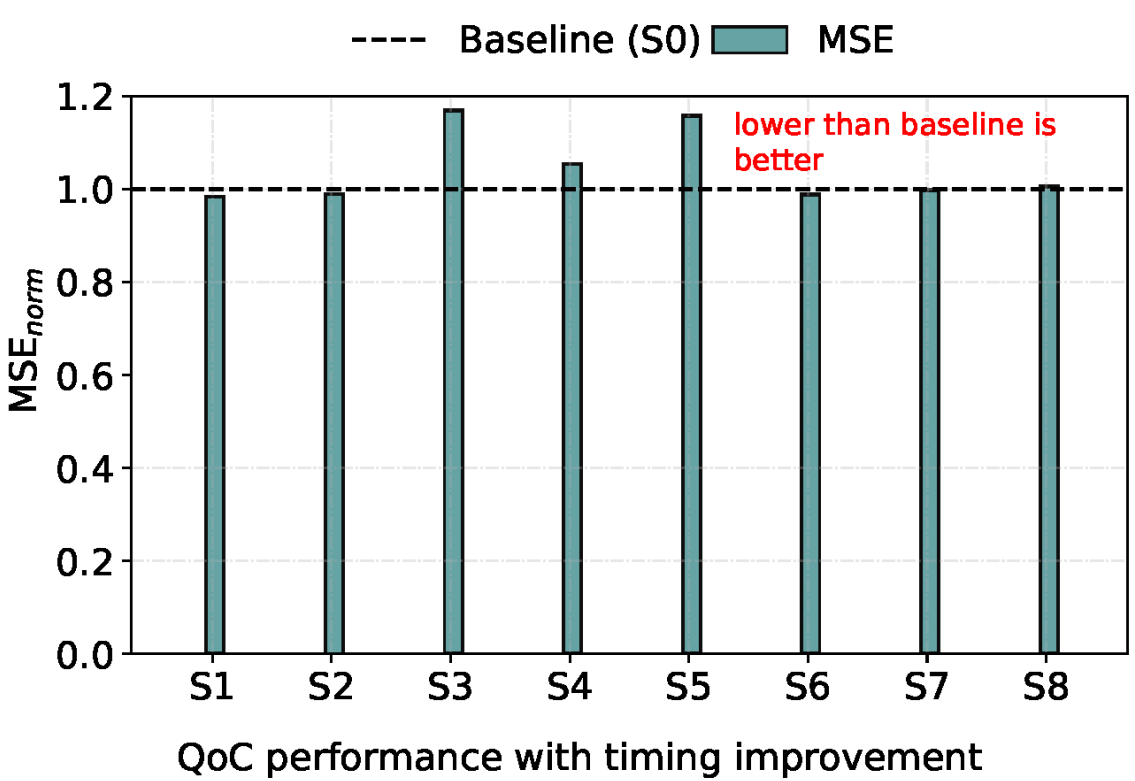
http://www.es.ele.tue.nl/cps/automotive/#imacs

IMACS: HiL simulator

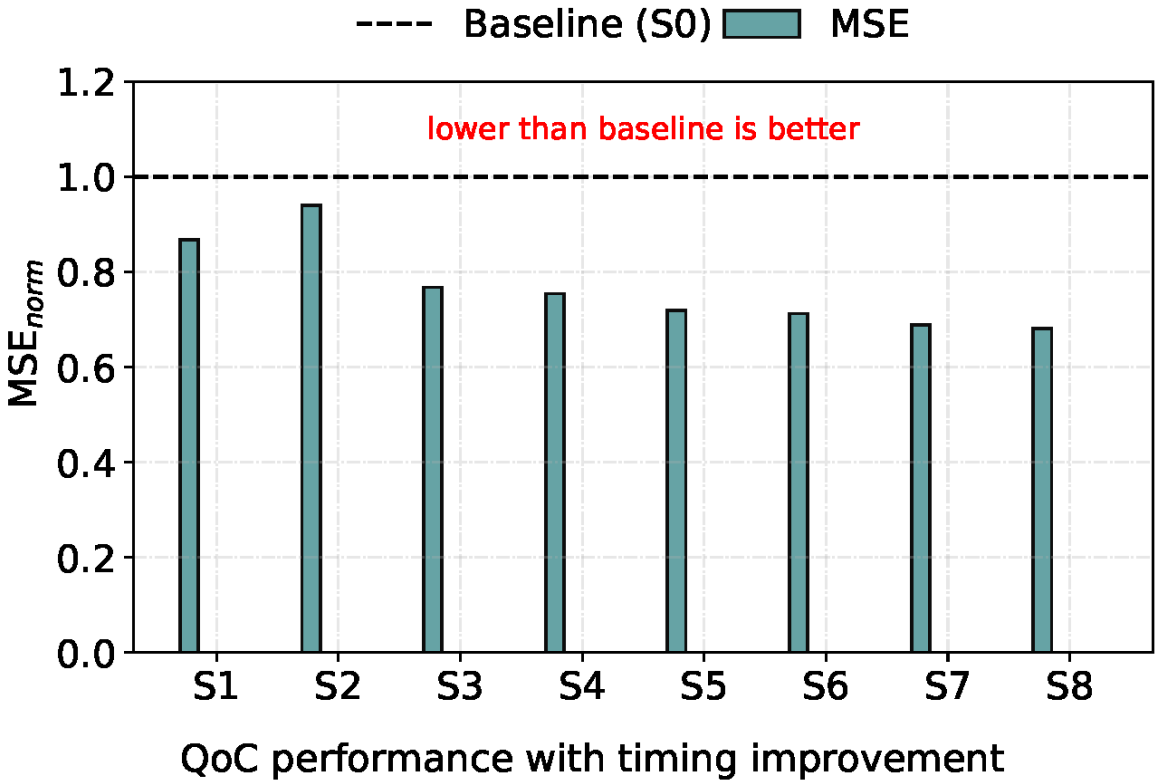


http://www.es.ele.tue.nl/cps/automotive/#imacs

Results: Degree of approximation vs QoC

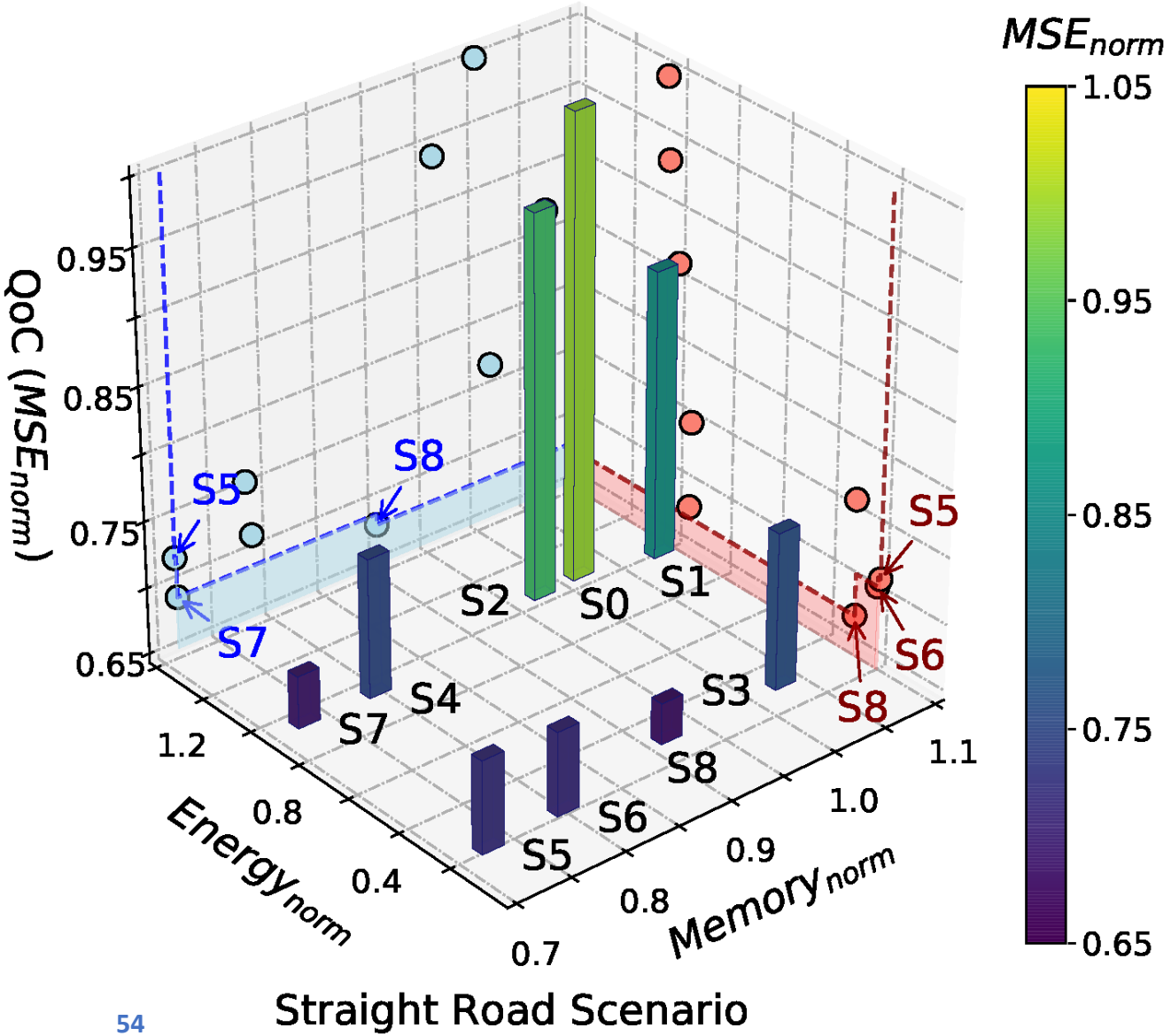


QoC performance with timing improvement



QoC performance with timing improvement

Results: Trade offs w.r.t Degree of Approximation



Setting	ISP Stages	Description
S0	DM, DN, CM, GM, TM, C	Accurate (all stages included)
S1	DM, CM, GM, TM, C	Skip Denoising
S2	DM, DN, GM, TM, C	Skip Color Mapping
S3	DM, DN, CM, TM, C	Skip Gamut Mapping
S4	DM, DN, CM, GM, C	Skip Tone Mapping
S5	DM, DN, C	Keep only Denoising
S6	DM, CM, C	Keep only Color Mapping
S7	DM, GM, C	Keep only Gamut Mapping
S8	DM, TM, C	Keep only Tone Mapping

Conclusions

- Image-based control suffers from long processing delay;
- Image-approximation is one promising approach to deal with long delay and save compute energy;
- There are several knobs that decides the performance of the overall IBC system;
- Extensive design space exploration is required to design sweet-spots;

sayandip.de@tue.nl

oCPS