ApproxEye: enabling approximate computation reuse for micro-robotic computer vision

Xin He, Guihai Yan, Faqiang Sun, Yinhe Han and Xiaowei LI

Speaker: **Yinhe Han**

State Key Laboratory of Computer Architecture,

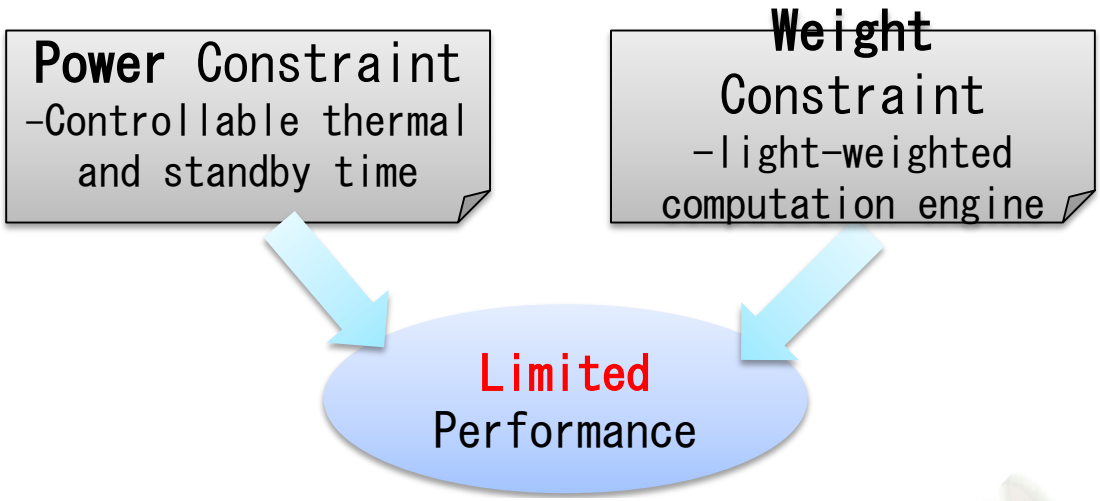
Institute of Computing Technology , Chinese Academy of Sciences

ERA
ENERGY-EFFICIENT RELIABLE ARCHITECTURE

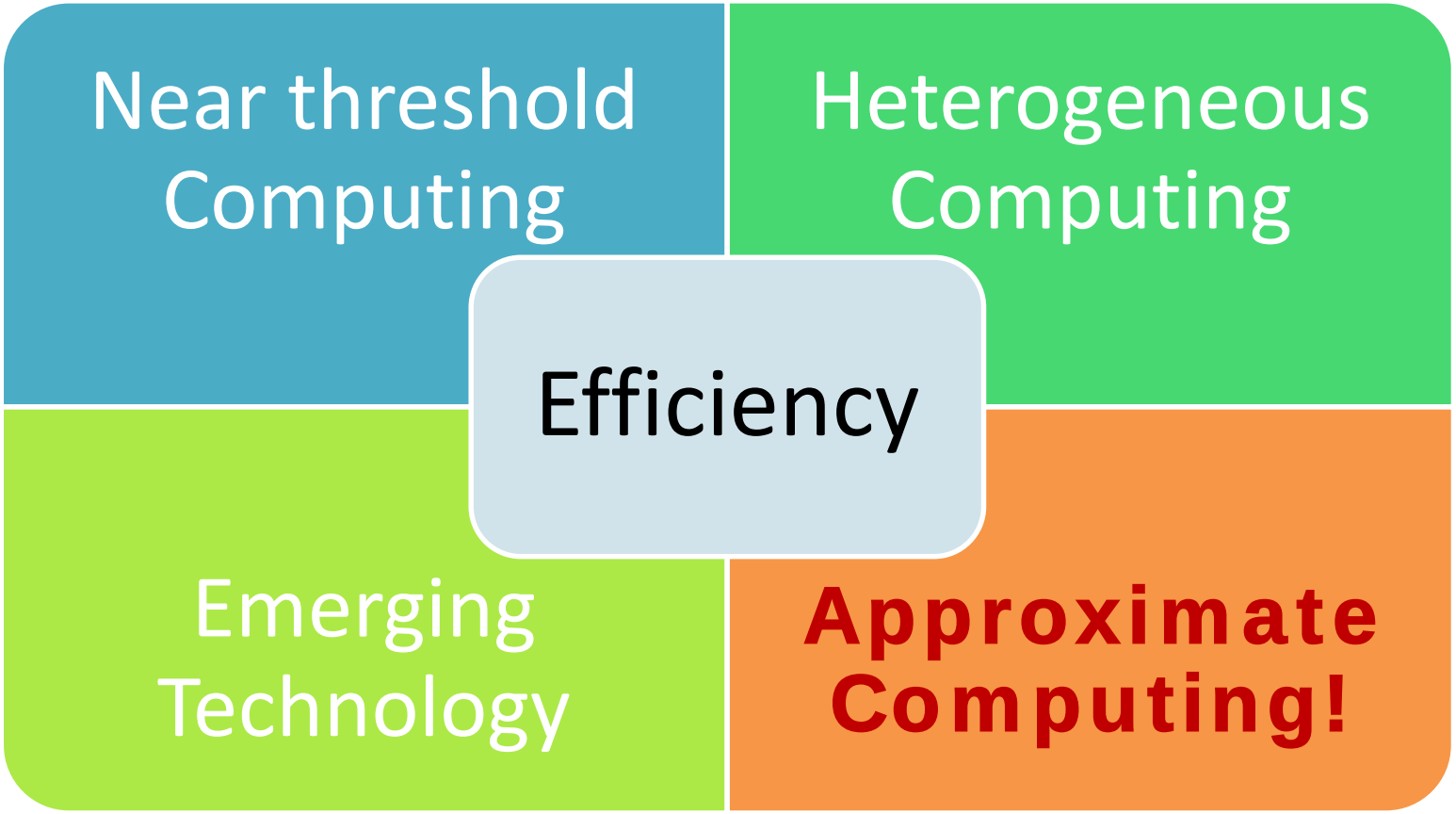
中国科学院计算技术研究所
Institute of Computing Technology,Chinese Academy of Sciences

Inefficiency of nowadays micro-sized system

- Existing micro-sized computing system (e.g. micro-robotics) fails to meet targeted performance requirement
  - Due to:

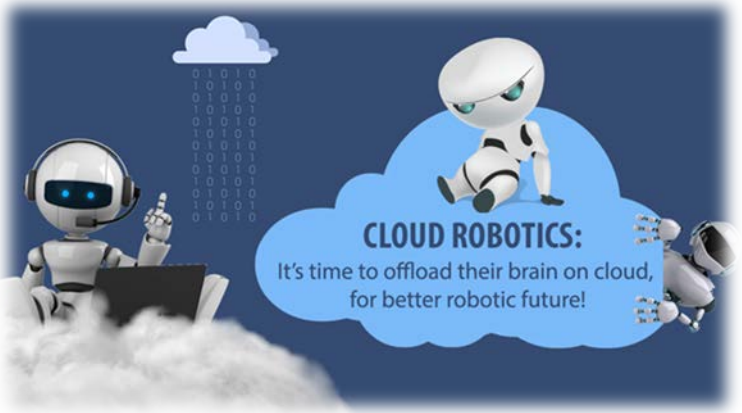


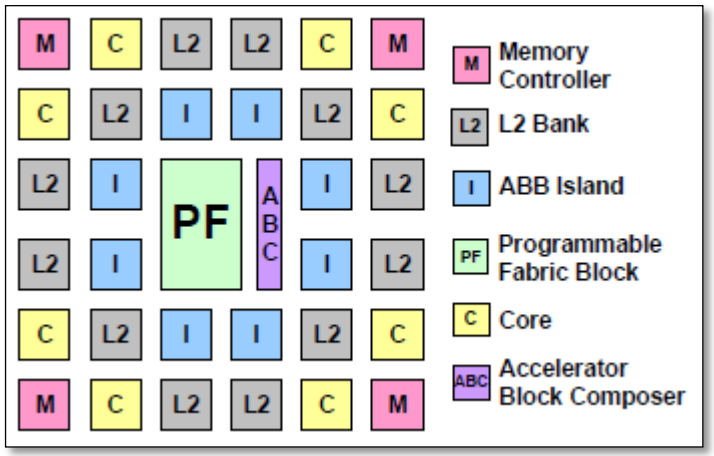
Energy efficient computing

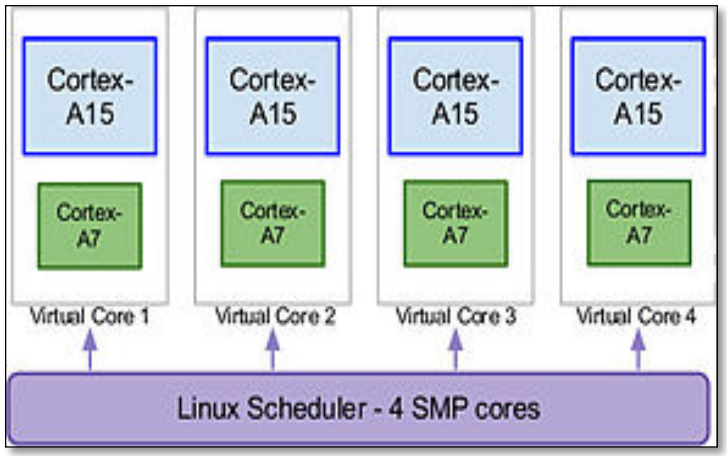


Prior ways to boost system performance

- To perform complex tasks like rescuing, tracking abnormal object tracking and so on
  - Computation Offloading
    - Responsiveness and bandwidth
  - Customized accelerator
    - Portability issue
  - Energy efficient scheduling
    - Limited achievable speedup







Approximate Computing(Domain specific approach)

- Leveraging intrinsic application resilience to improve efficiency

E. g. DCT Image Compression

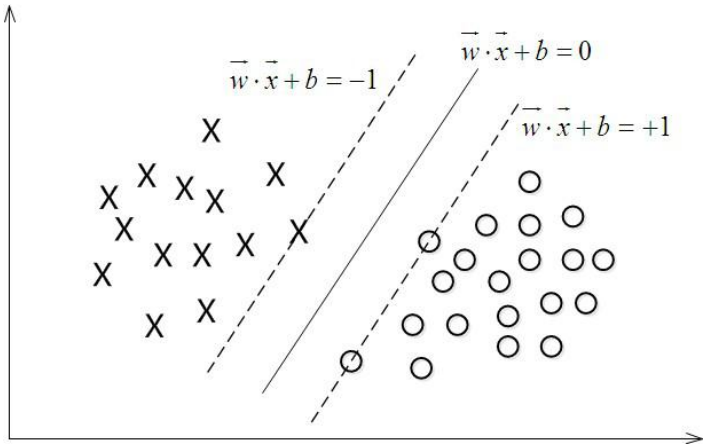
Original method ⇨ Approximate method

By using inexact multiplier

Outcome: **2.82%** reduction on PSNR

Performance: **31.44%** Improvement

Machine Learning



By reducing excess iterations

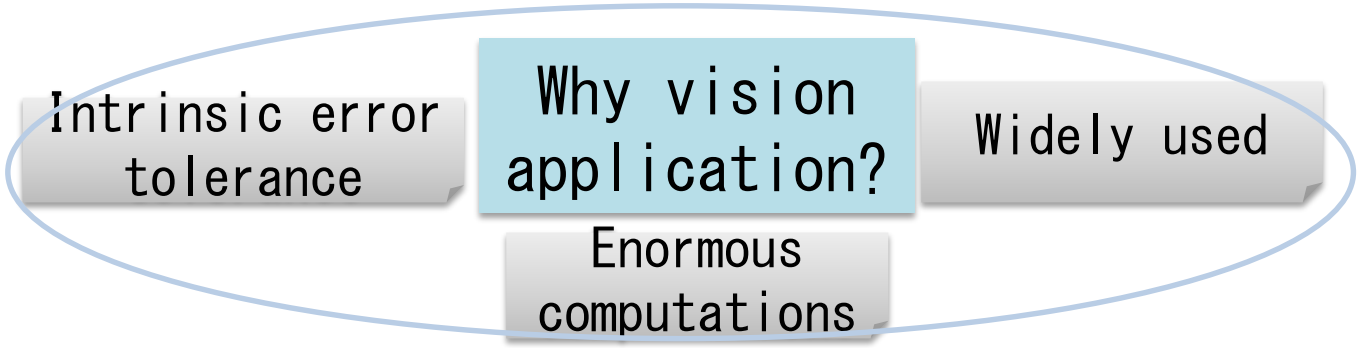
Outcome: **1.3%** accuracy reduction

Performance: **4.97X** Improvement

Approximate Computing opens another road towards computing efficiency

Targeting and observation

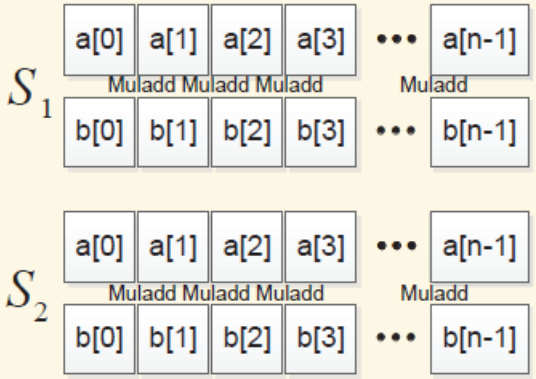
- Focusing on accelerating the computer vision algorithm through approximate computing:



- We select optical flow to conduct approximate computing

Kernel of Optical flow:

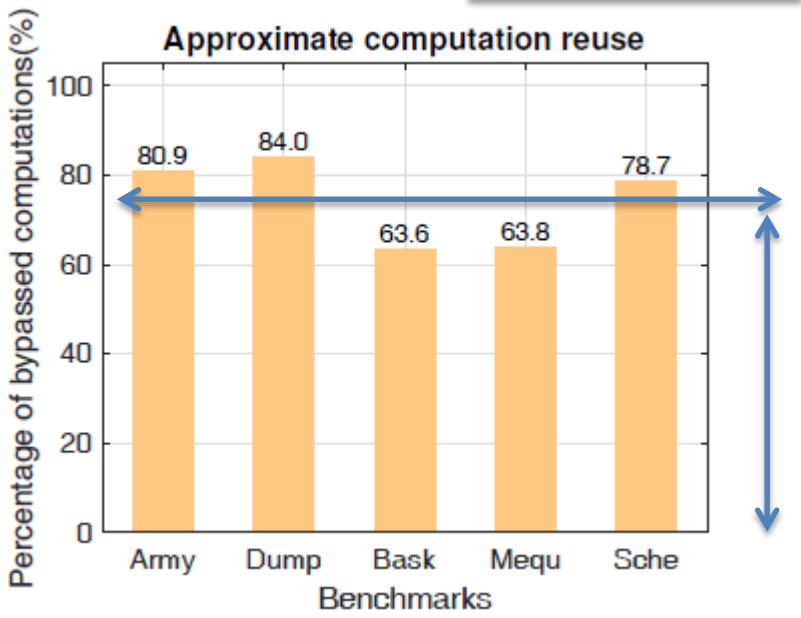
```
for (i......)
  for (j......){
    load(a,b);
    begin:
    for(k=0;k<n;k++)
      result+=a[k]*b[k];
    end:
    'result' is used...
  }
```



Optical flow as target killer application

- We select optical flow to conduct approximate computing
  - Optical flow is widely used for visual surveillance, motion estimation, object tracking.
  - Its computation kernel is a typical kind of operations seen in computer vision

Observation



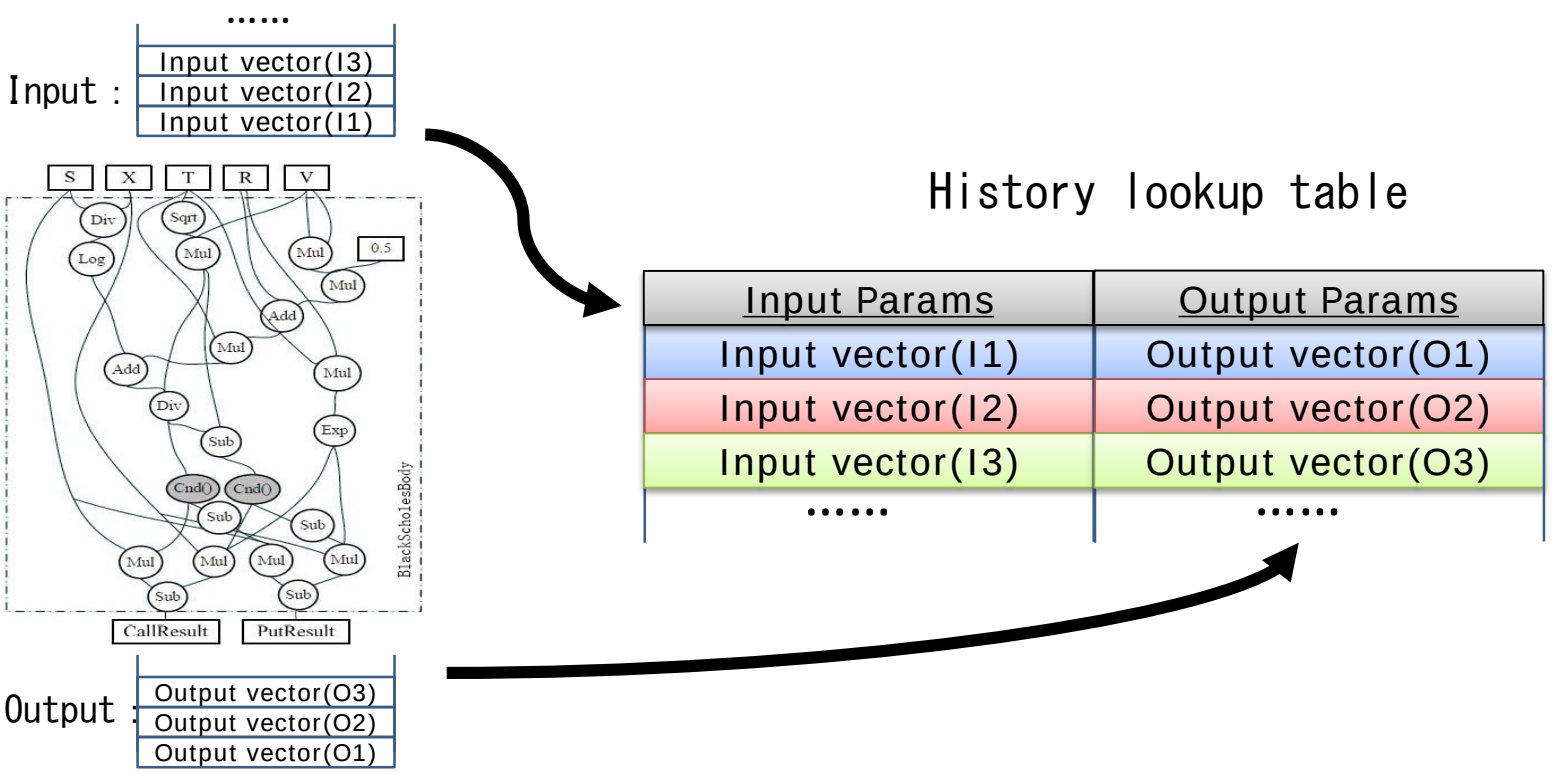
- 74. 20%的相似计算量
  [10%的输出误差]
- Could we leverage this observations to accelerate computing?

- To reuse the prior computations:
  - Exact reuse scheme (100% accurate)
    - Traditional computation reuse
      - Fail to reuse similar computations
  - Approximate reuse scheme (allow accuracy degradation)
    - RACB scheme
      - Fail to fully extract reuse opportunities
    - Proposed ApproxEye scheme
      - Significance aware computing
      - Adaptive reuse granularity selection

For traditional reuse scheme

- Traditional computation reuse scheme
  - Leveraging computation locality to reuse computations
    - But the tight requirement is used for 100% precision

Phase 1: **"Memorize" computation history**

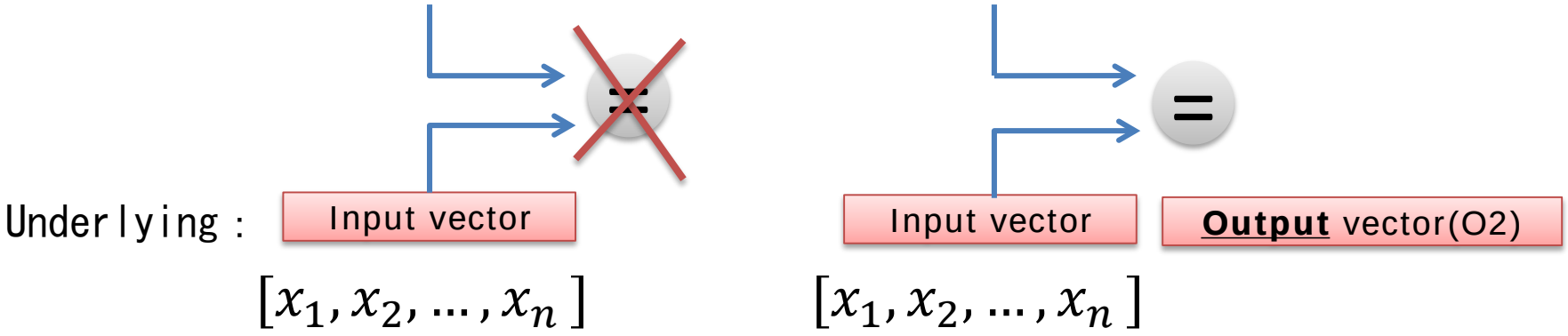


For traditional reuse scheme

- Traditional computation reuse scheme
  - Leveraging computation locality to reuse computations
    - Require all inputs of a reused item is **exactly equal** to underlying computation

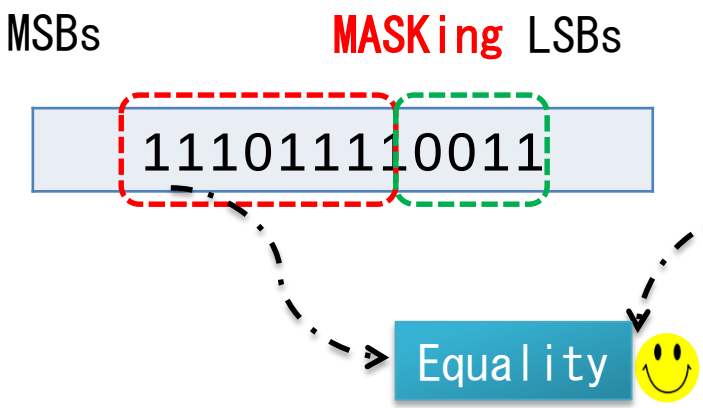
Phase 2: **"Reusing" prior computations**

Input Params	Output Params
Input vector(I1)	Output vector(O1)
Input vector(I2)	Output vector(O2)
Input vector(I3)	Output vector(O3)
.....	.....



Limitation of prior approximate reuse scheme

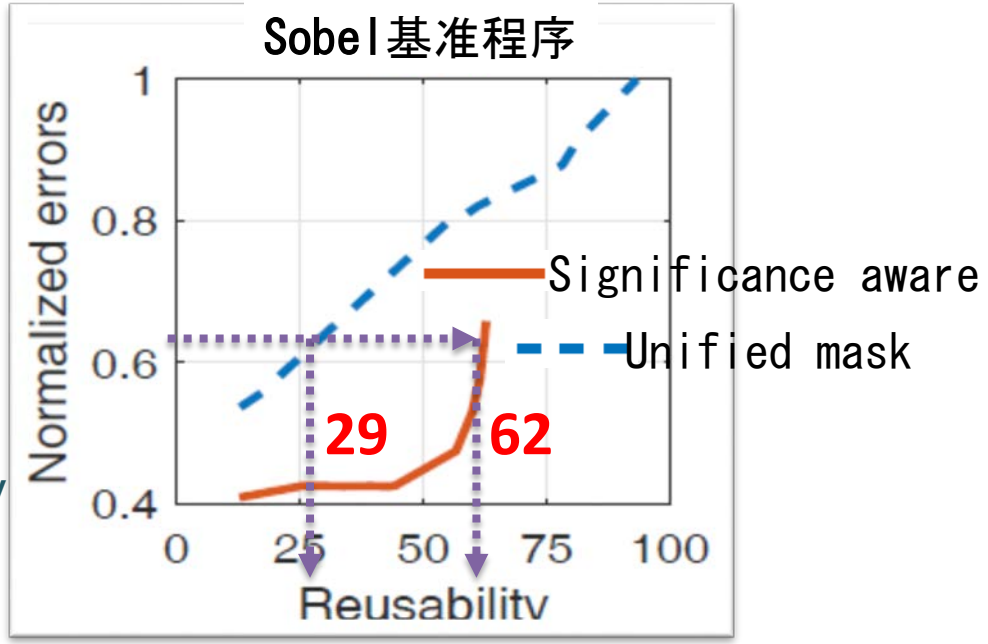
- Prior approx reuse（e.g. RACB scheme [islped2005]）
  - Reuse requirement：the equality of Most Significant Bits



Input vector	Output
110111110110	10
110101110110	10
……	……
000110111110	50

Cons：

- It's non-trivial to determine the length of LSBs mask
- A unified mask for all different inputs limits the effectiveness
- Limited application scope
- Predetermined reused granularity



Proposed ApproxEye scheme

- To fully extract the potential of computation reuse, ApproxEye proposes both
  - "Simple but effective" significance aware quantification
  - Adaptive reuse granularity selection

- Workflow of ApproxEye
  1. Fully extract computation with high reuse possibilities
  2. Based on statistical analysis, the optimal reuse granularity is specified
  3. Calculate input significance and set masks adaptively

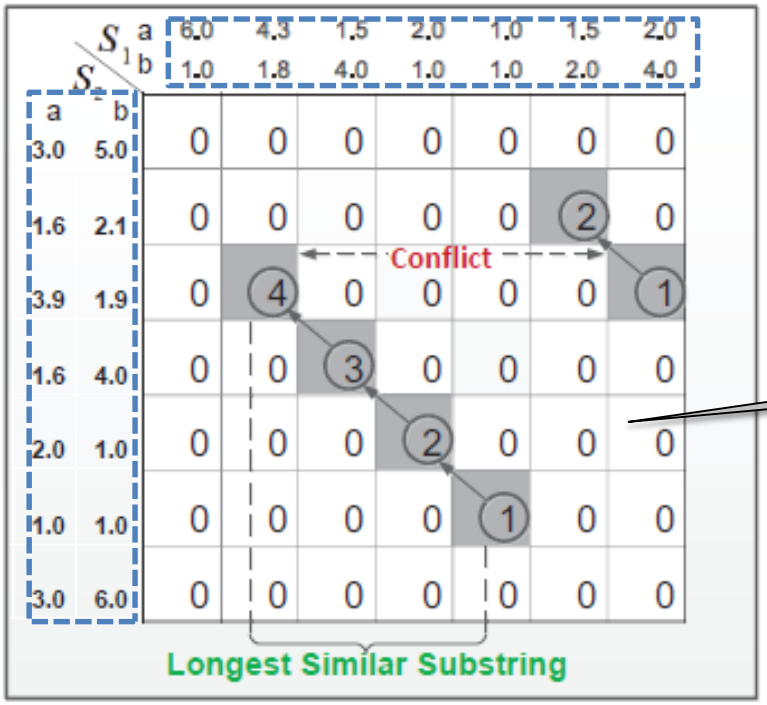
Proposed ApproxEye scheme

- Fully extract computation with high reuse possibilities
  - Prohibit missing any reuse opportunities from history

Given: different optical flow kernel "multiply add" sequences, e.g. S1 and S1

Determine: similar substring inside both S1 and S2

S1:



S1:

Questions：extracting **all** the longest similar substring

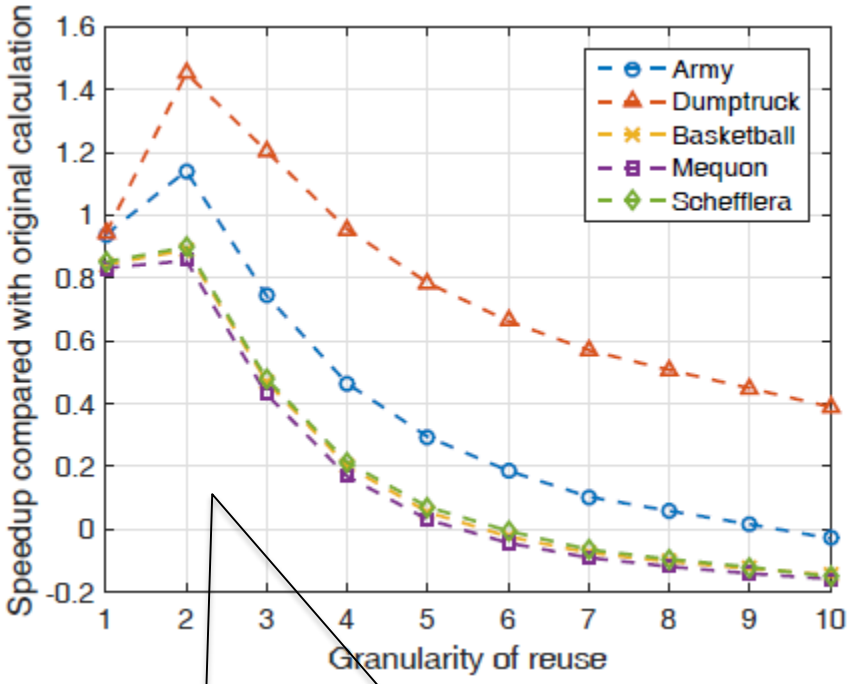
Solution：Enhanced version of Dynamic Programing

- Mark the relative position of similar items
- Traverse route map to obtain the substrings

Proposed ApproxEye scheme

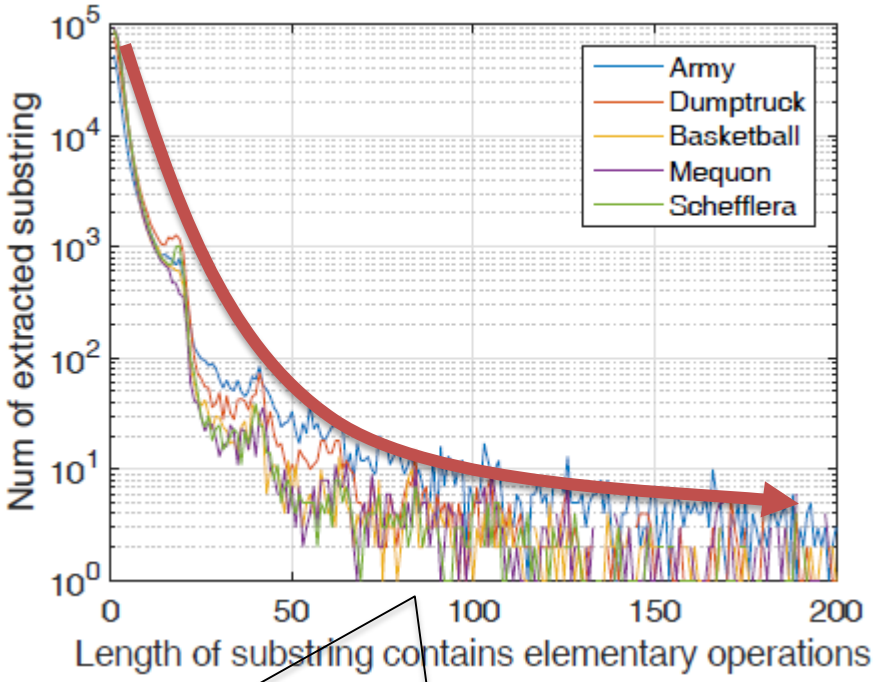
- Once the similar computation is obtained, statistically analysis is conducted to
  - determine reuse granularity that gives highest speedup

Speedup under different granularities



Two multiply-add operations gives highest speedup

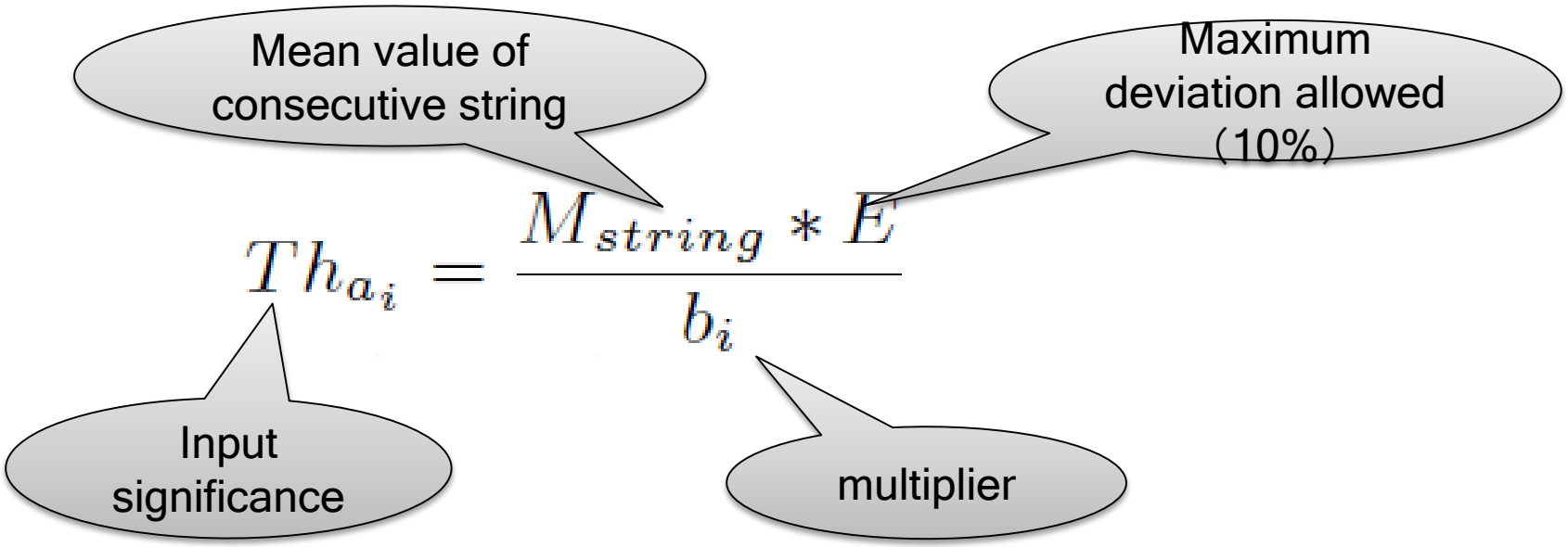
Amount of substring extracted



Amount of substring extracted decreases exponentially

Proposed ApproxEye scheme

- Calculate input significance and set masks adaptively

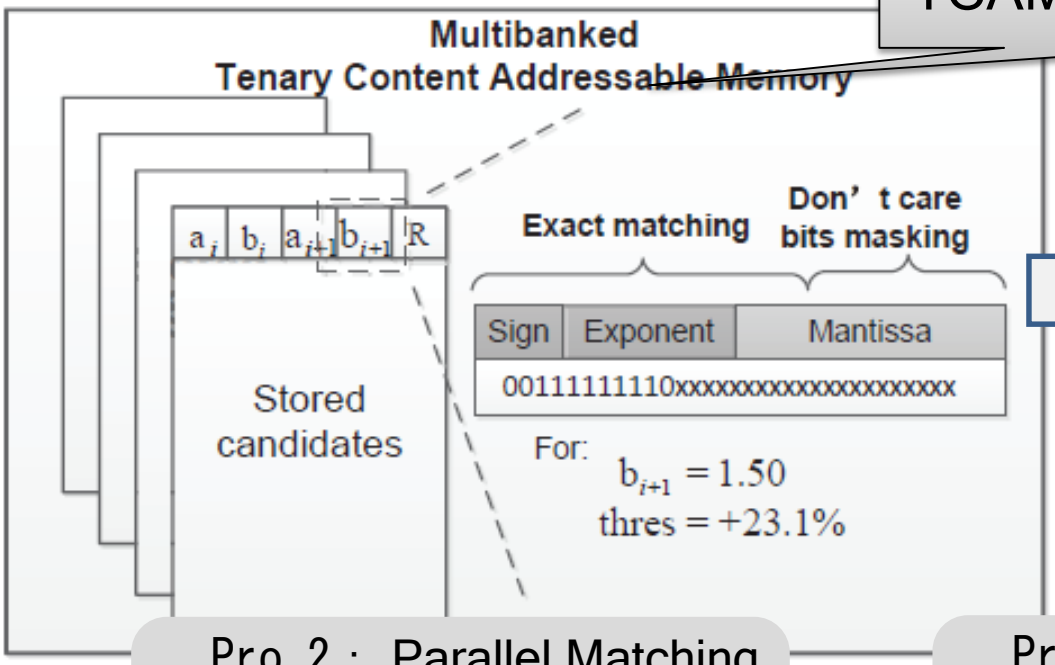


This simple but faithful significance estimation benefits from:
1)This simple form can improve the effectiveness of runtime estimation
2)The significance can be easily used for adaptive LSB masking

创新成果二：可变粒度的近似计算方法

- Parallel search tailored for approximate computing

TCAM for Parallel Matching



Pro 1：Significance aware

- Adaptively set don't care masks based obtained significance

Pro 2： Parallel Matching

- Improve the latency for large amount of data

Pro 3：Consolidation

- Consolidate similarity calculation, checking, and searching

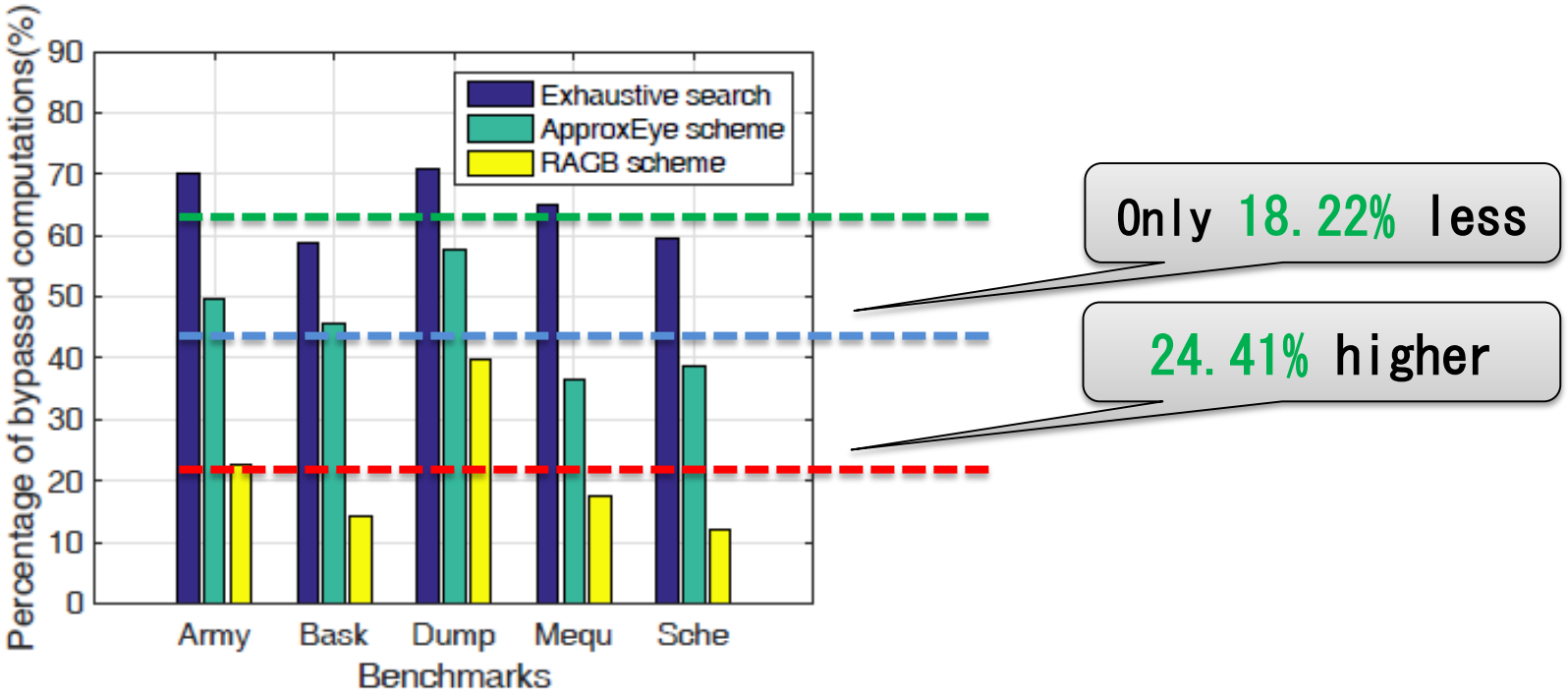
Reducing the time consumption

Experiment

- Experimental setup
  - Target applications
    - Algorithm: State of the art Lucas Kanade pyramid iterative optical flow algorithm
    - Dataset: open source data from Microsoft redmond
  - Performance and power simulation
    - Latency of multiply-add operations is obtained using Cadence tool flow in TMSC 45nm PTM model
    - The latency and power of TCAM is simulated from TCAM model based on Cacti

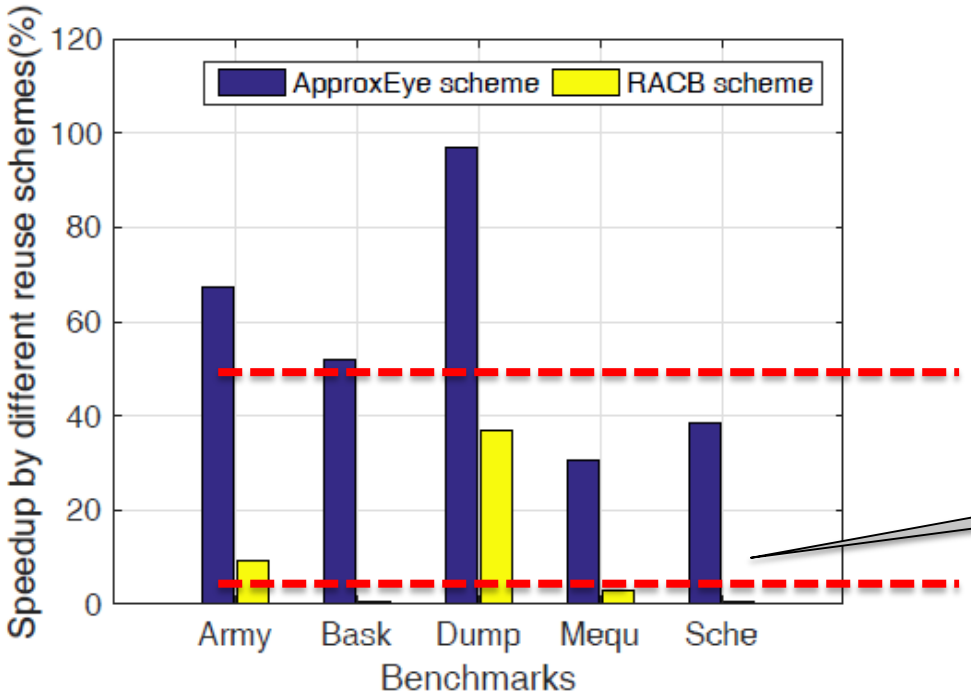
Experimental result

- The amount computation which can be reused by ApproxEye
  - Compared with oracle solution and RACB scheme



Experimental result

- Speedup achieved by ApproxEye



47.13% speedup on average

Prior RACB approximate computing scheme：

- Unfaithful significance quantification

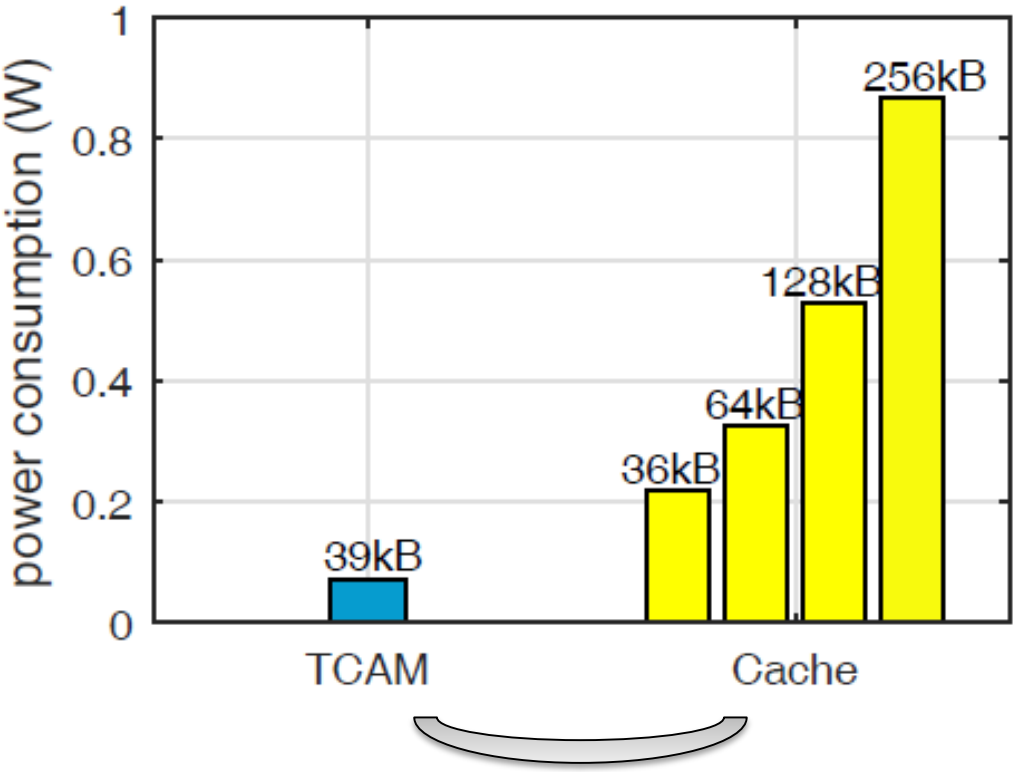
using unified deviation thresholds for all different inputs

- Select the reuse granularity arbitrarily,

miss the opportunities for partial reuse

Experimental result

- TCAM power consumption VS conventional cache



Comparable or less power than Cache

Conclusion

- Approximate computing can exploit the potential of computation reuse in computer vision applications
- To fully extract the potential of computation reuse, one should take a deep look into data distribution and make comprehensive decisions
- Moreover, the engine of approximate computing should also be fine tuned for efficiency

Thanks for listening!

Questions?

Xin He

State Key Laboratory of Computer Architecture,
Institute of Computing Technology, Chinese Academy of Sciences
(ICT, CAS)

ERA
ENERGY-EFFICIENT RELIABLE ARCHITECTURE

中国科学院计算技术研究所
Institute of Computing Technology,Chinese Academy of Sciences