SRI-SURF: A Better SURF Powered by Scaled-RAM Interpolator on FPGA

Xijie Jia[1], Kaiyuan Guo[1], Wenqiang Wang[3],
Yu Wang[1,2] and Huazhong Yang[1]
[1]E.E. Dept., TNLIST, Tsinghua University, Beijing, China
[2]yu-wang@mail.tsinghua.edu.cn
[3]Microsoft Research Asia, Beijing, China

Nano-scale Integrated Circuit and System Lab,
Department of Electronic Engineering, Tsinghua University

Outline

- Introduction
- Methods
- Experiments
- Conclusion

Outline

- Introduction
  - Background
  - Related Work
  - SURF Algorithm
  - Contributions
- Methods
- Experiments
- Conclusion

Background – Local Feature Extraction

- Main Goal:
  - Find representative regions of a image
  - Find robust expression for each of them

- What is “robust” feature:
  - Invariant to affine transformations, environment light, etc.

- Algorithms:
  - SIFT (Scale Invariant Feature Transform) [IJCV04]
  - PCA-SIFT (Principle Component Analysis SIFT) [CVPR04]
  - GLOH (Gradient Location-Orientation Histogram) [PAMI05]
  - SURF (Speed-Up Robust Feature) [ECCV06]

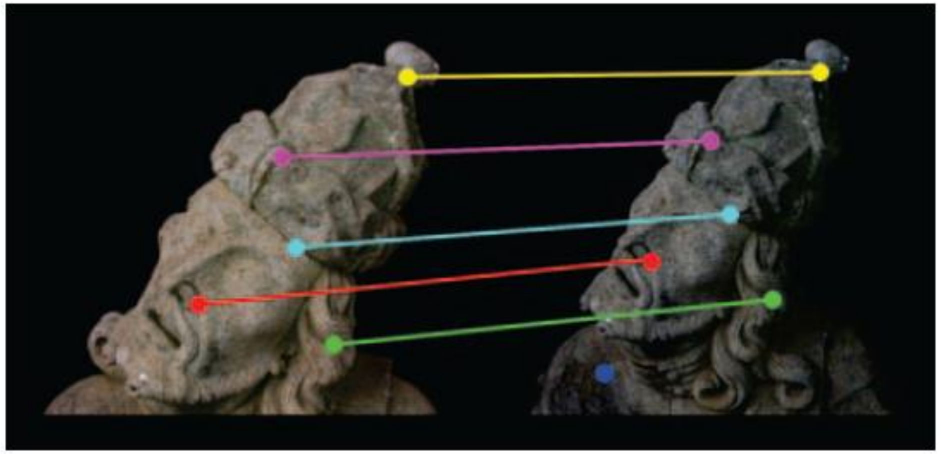
Background - Applications

- Image mosaic[ICISE09]
- Object recognition[SMC09]
- 3D reconstruction[ICIP12]
- Crowd counting[TCEC14]

- Requirements
  - Real-time processing
  - High matching precision at high resolution

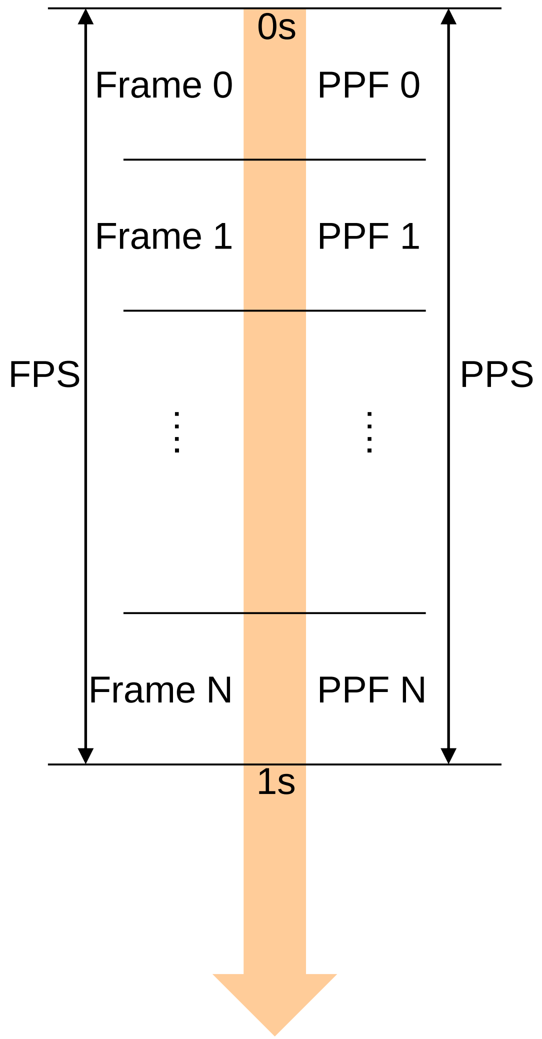
Figure.7 Images a, b

Figure.8 Mosaic Result of image a, b



Background - Performance Evaluation

- Frames Per Second (FPS)
- Feature Points Per Frame (PPF)
  - Related to image resolution and texture complexity
- Feature Points Per Second (PPS)
  - MAX-PPS: represents the calculation capacity of the system
  - ACT-PPS: represents the requirements of the application



Related Work – SURF Acceleration

Serial platform

CPU
OpenSURF[2009]

- Easy to realize
- Low performance

Parallel platform

GPU
clSURF[GPGPU2011]

- Good portability from CPU
- Low energy efficiency

ASIC
SURFEX[CICC2013]

- Best energy efficiency
- Low flexibility

FPGA
SURF[FPT2013]

- Good energy efficiency
- Long develop cycle

Related Work – SURF Acceleration

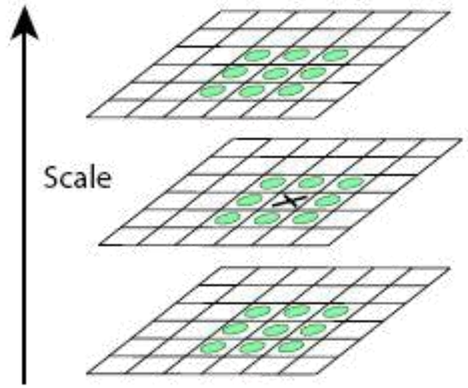
Version	Clock	Resolution	FPS	PPF	PPS	Octave	Chip	Function
[GPGPU11]	1.4GHz	791x704	40	800	32K	NA	GTX480	FD+OG+DG
[ReConFig11]	100MHz	640x480	~2	~49	0.1K	8	Virtex 5+PowerPC	FD+OG+DG
[BEC12]	25MHz	640x480	60	100	6.0K	6	3x Virtex 4	FD+OG+DG
[TENCON13]	200MHz	300x300	42	250	10.5K	4	Zynq 7	FD+OG+DG
[FPT13]	156MHz	640x480	356	100	35K	6	Virtex 6	FD+OG+DG
[ReConfig14]	25MHz	640x480	131	1614	211K	6	Zynq 7	FD+OG
[CICC13]	200MHz	1920x1080	57	5000	285K	12	ASIC	FD+OG+DG

- Early work on GPU: high performance by powerful chip
- Works on FPGA: performance was still insufficient
  - Simplification -> precision problem
  - Low computation capacity
  - High resource occupation
- Work on ASIC: high performance by specific device

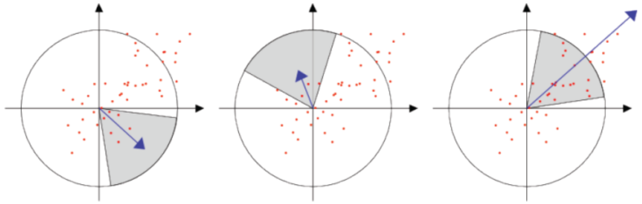
FD： Feature Detection
OG： Orientation Generation
DG： Descriptor Generation

Introduction to SURF - Algorithm

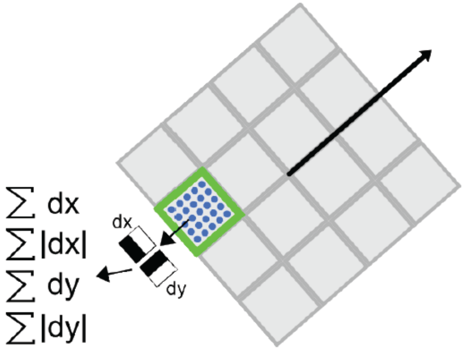
- Feature Detection
  - Calculate integral image ——base data
  - Calculate $\det(\mathcal{H}_{aprrox})_{\mathrm{norm}}$ ——locate in each interval
  - Find local-maximum ——locate among neighbor interval
  - Up-sampling interpolation ——sub-pixel correction



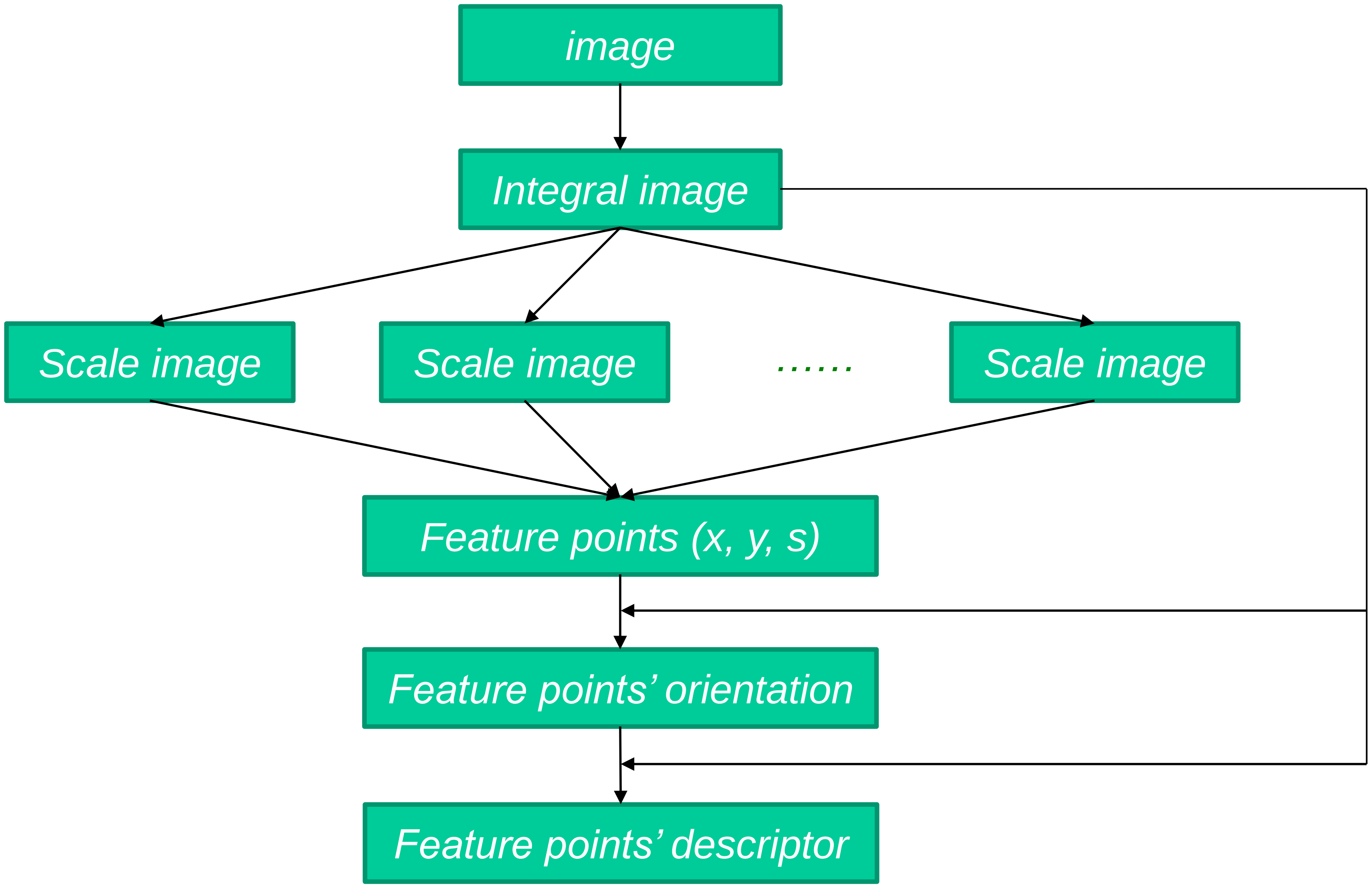
- Orientation Generation
  - Calculate Haar wavelet ——base data
  - Add-up Slide-Window ——locate orientation



- Descriptor Generation
  - Calculate Haar wavelet ——base data
  - Sum-up Sub-Neighbor-Region ——generate 4x4x4 descriptors



Introduction to SURF - Algorithm



Introduction to SURF - Complexity

Op.	Determinant	Find localMax UpSamp-Intp	Orientation	Descriptor	Total
	Resolution	Candidate Point	Feature Point	Feature Point	
	640x480	520	520	500	
Read RAM	9,059,904		453,440	2,304,000	11,817,344
Plus	7,361,172	6,480	1,152,320	4,864,000	13,383,972
Minus	3,963,708	4,860	340,080	1,728,000	6,036,648
Multiply	566,244		165,360	1,296,000	2,027,604
Square	283,122		37,440		320,562
Divide	283,122				283,122
Compare		14,040	18,720		32,760
Equation Set		540			540
Rotate			56,680	576,000	632,680
ATAN			520		520

High computation complexity

Bottleneck of serial processing
Good parallelism

SOLVED [FPT13]

Points are computed serially,
Bottleneck is single point processing

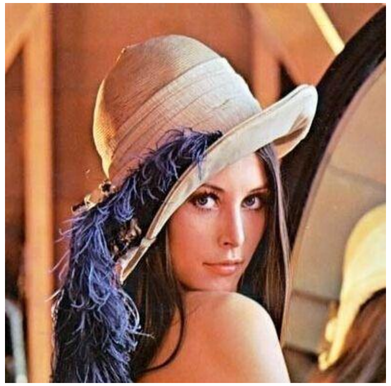
UNSOVED

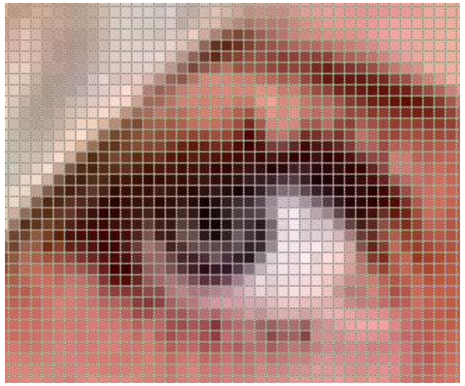
Introduction to SURF - approximation

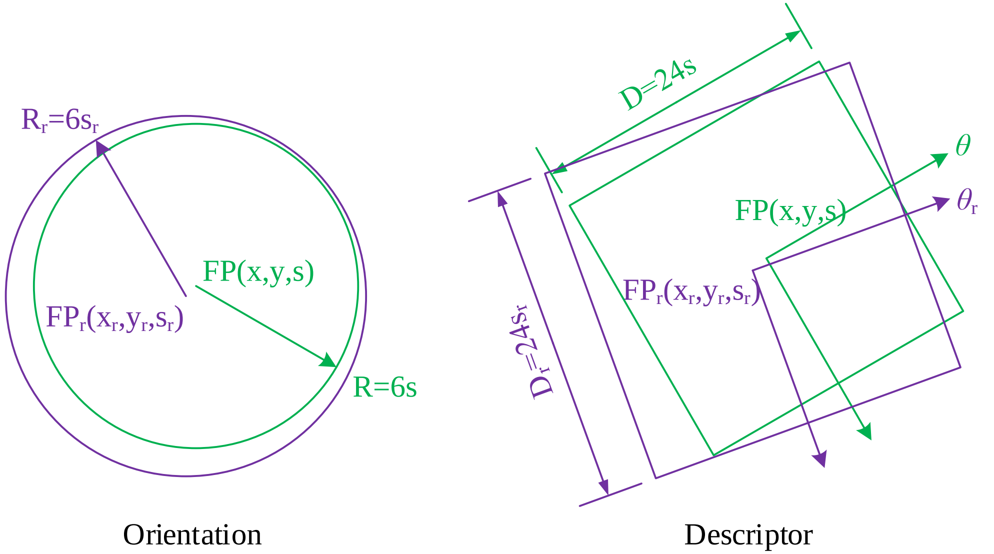
- Feature points are from different scales

- Non-integer coordinate feature points

- How to use integral image?
- In OpenSURF, all the integral image data are from integer coordinates
- How about interpolation



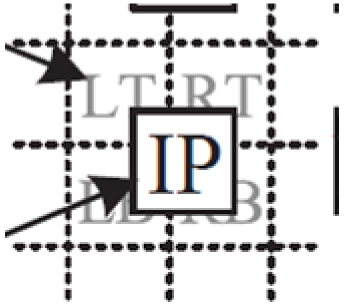


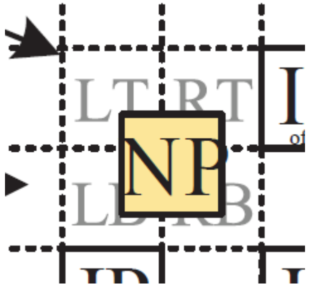


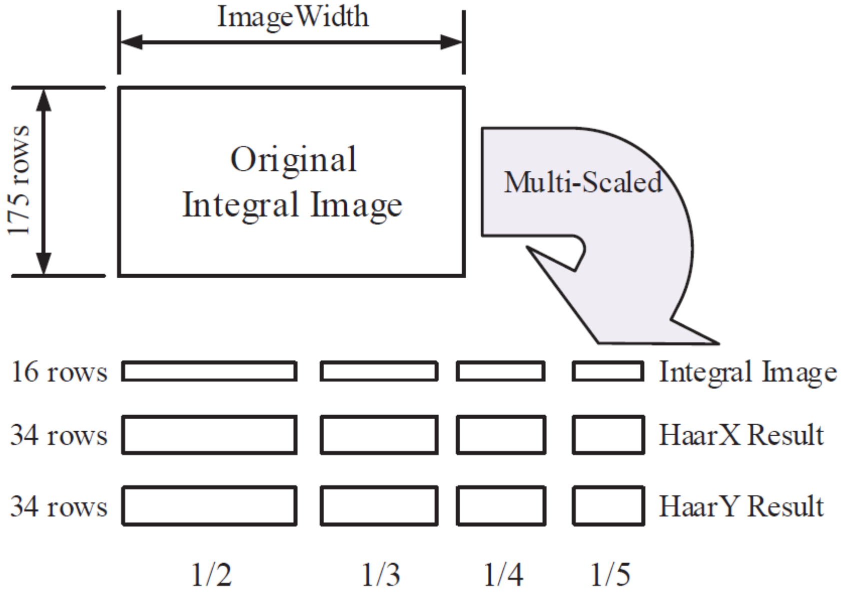
The index deviation caused by rounding error
FP: original feature point
FP_r: rounded-coordinates-and-scale feature point

Contribution

- Interpolation of Integral Image (I^3)
  - For better matching precision
- Compromise of Interpolation of Integral Image (CI^3)
  - Halve the memory access, by decreasing a bit accuracy
  - For higher processing speed
- Multi-Scaled RAM (MSR)
  - For lower storage occupation







Outline

- Introduction
- Methods
  - Interpolation of Integral Image (I^3)
  - Compromise of Interpolation of Integral Image (CI^3)
  - Multi-Scaled RAM (MSR)
  - Implementation
- Experiments
- Conclusion

Interpolation of Integral Image

Quantization Error of Image System

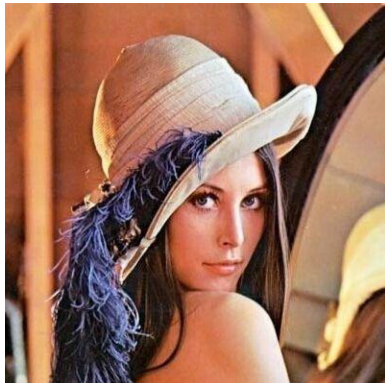
Continuous image-> Acquisition -> Pixels

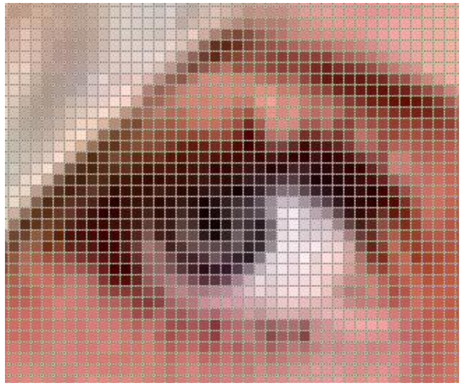
Decimal coordinates-> Truncation -> Integer

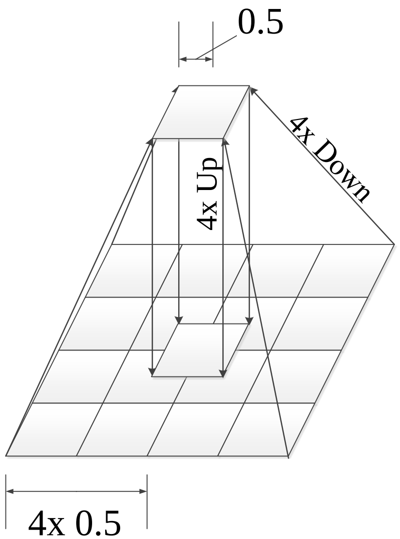
Loss of image detail

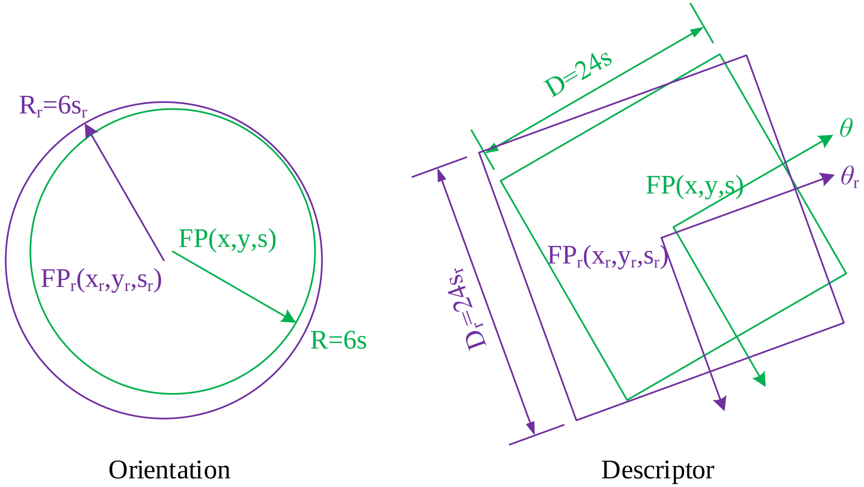
Index deviation

Cumulative error is enlarged step by step





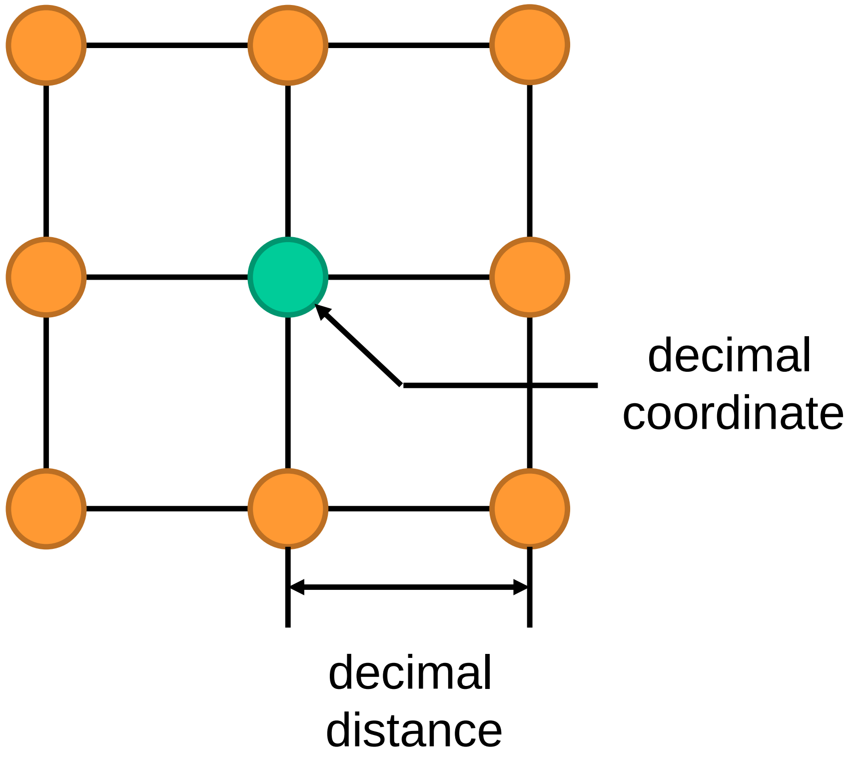




The index deviation caused by rounding error
FP: original feature point
FP_r: rounded-coordinates-and-scale feature point

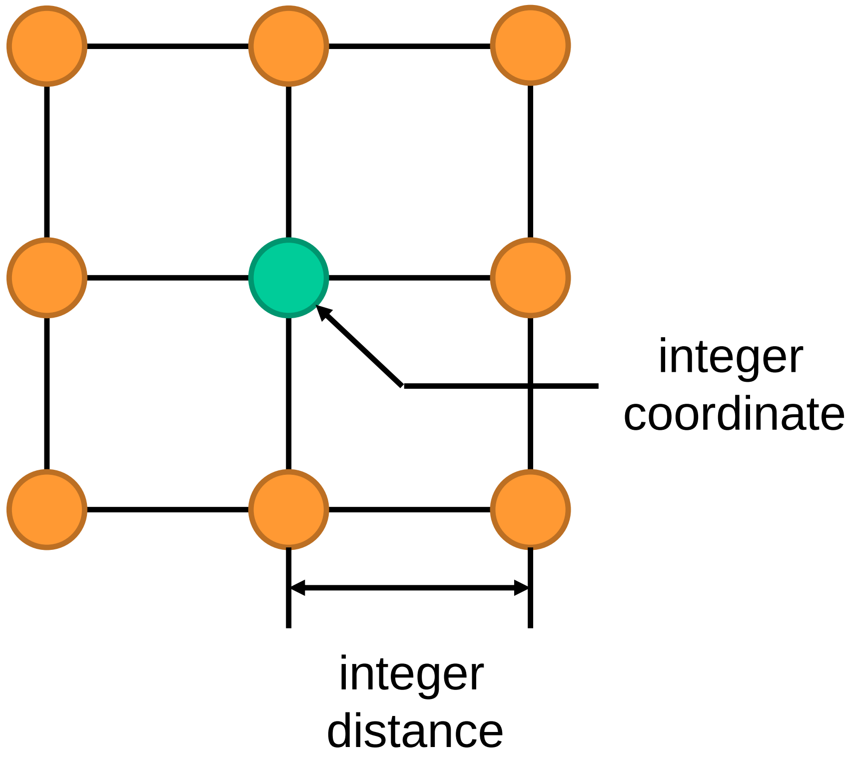
Interpolation of Integral Image

- Haar wavelet - math



Theoretical situation
Approximate by interpolation

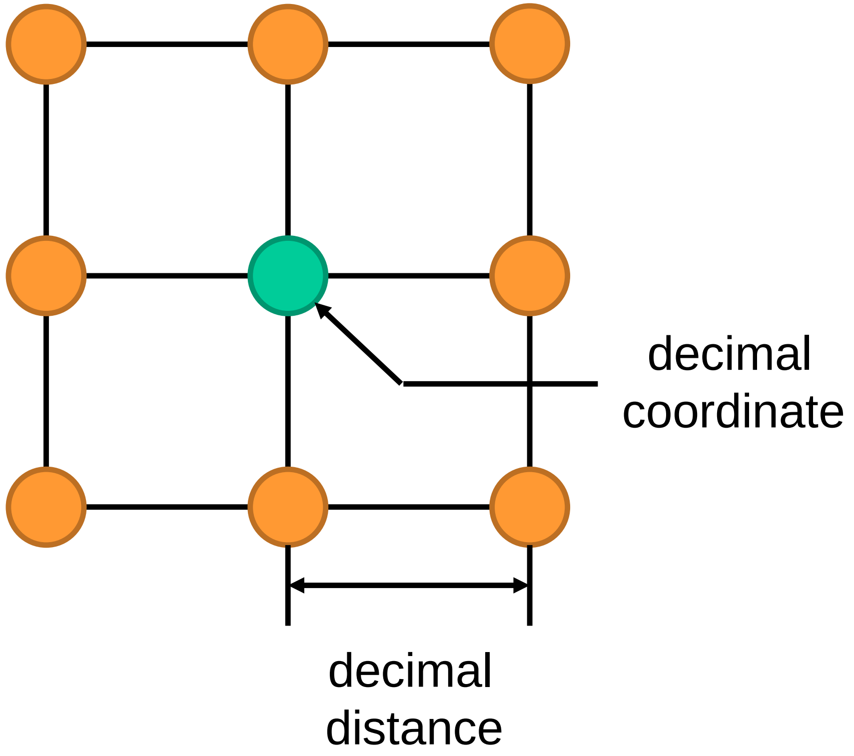
- OpenSURF



Directly read from integral image

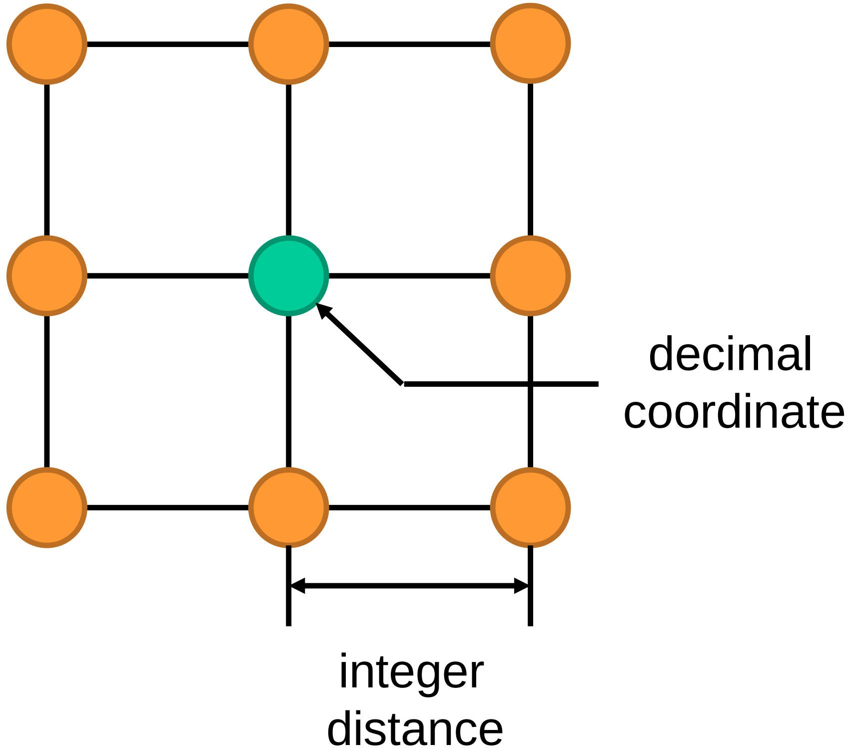
Compromise of Interpolation of Integral Image (CI^3)

- Haar wavelet - math



Need 32 number from integral image
Different interpolation parameter

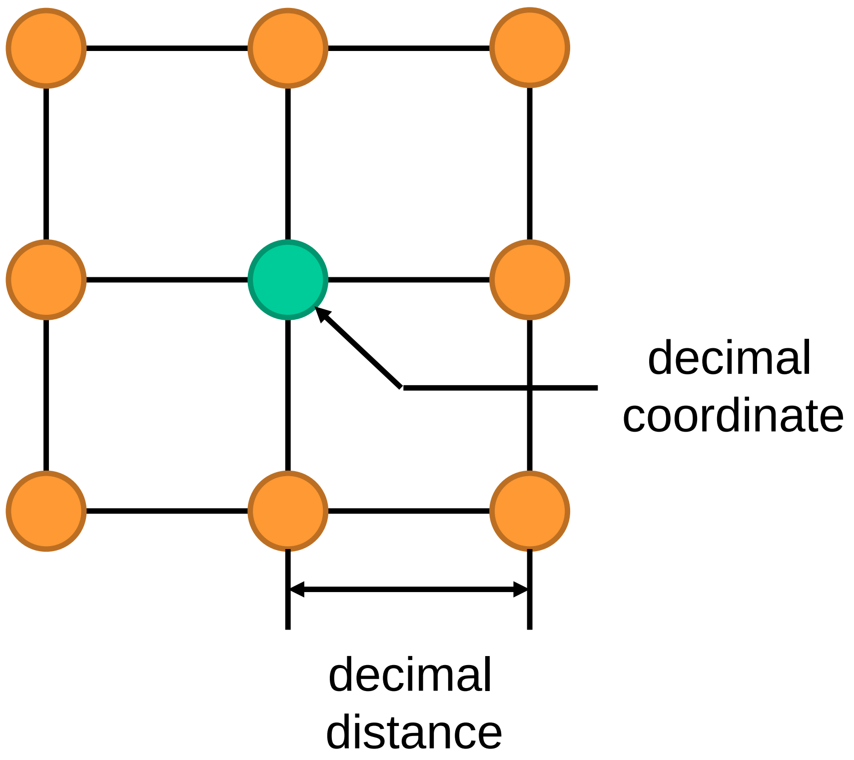
- A trade-off version



Need 32 number from integral image
Same interpolation parameter

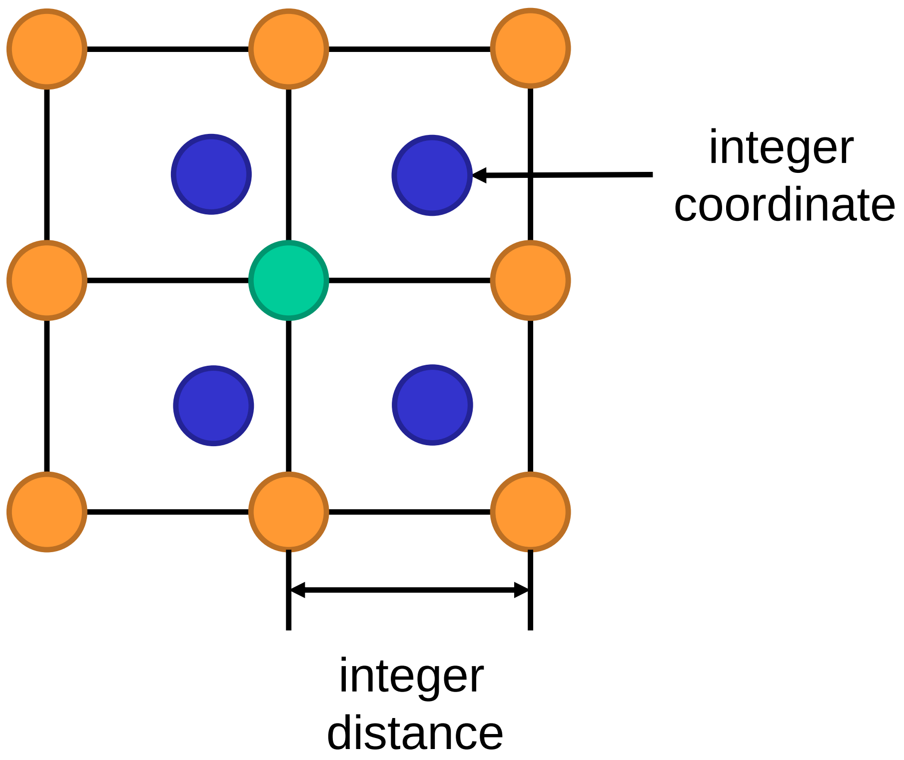
Compromise of Interpolation of Integral Image (CI^3)

- Haar wavelet - math



Need 32 number from integral image
Hard to fetch in parallel

- Proposed



Pre-compute the Haar wavelets on integer coordinates
Need 4 pre-computed number

Compromise of Interpolation of Integral Image (CI^3)

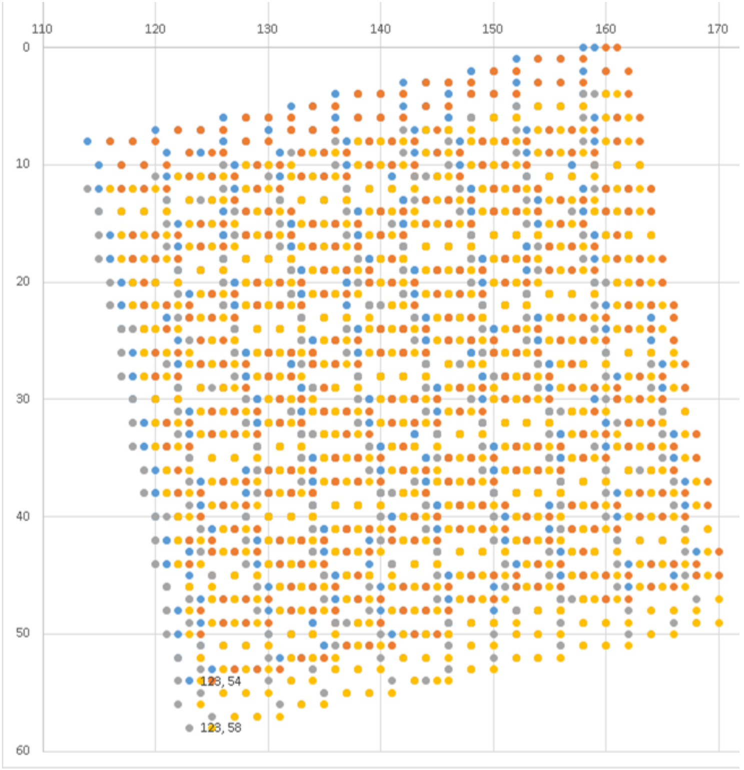
- Advantage:
  - Use interpolation to improve accuracy
  - Remains the data access pattern predictable
- Weakness:
  - RAM occupation is doubled for pre-computed Harr wavelets.
  - Not exactly as the mathematical solution

Version	Point Type	Coord.Type	Index Level	Coords. Deviation
Trad.	All	Rounded Integer	Pixel	Large
Proposed	FP	Fixed Decimal	Sub-Pixel	Small
	NP	Fixed Decimal	Sub-Pixel	Small
	IP	As Trad.	As Trad.	As Trad.

RAM Occupation Problem

Comparison of FP Distribution and Buffer Utilization

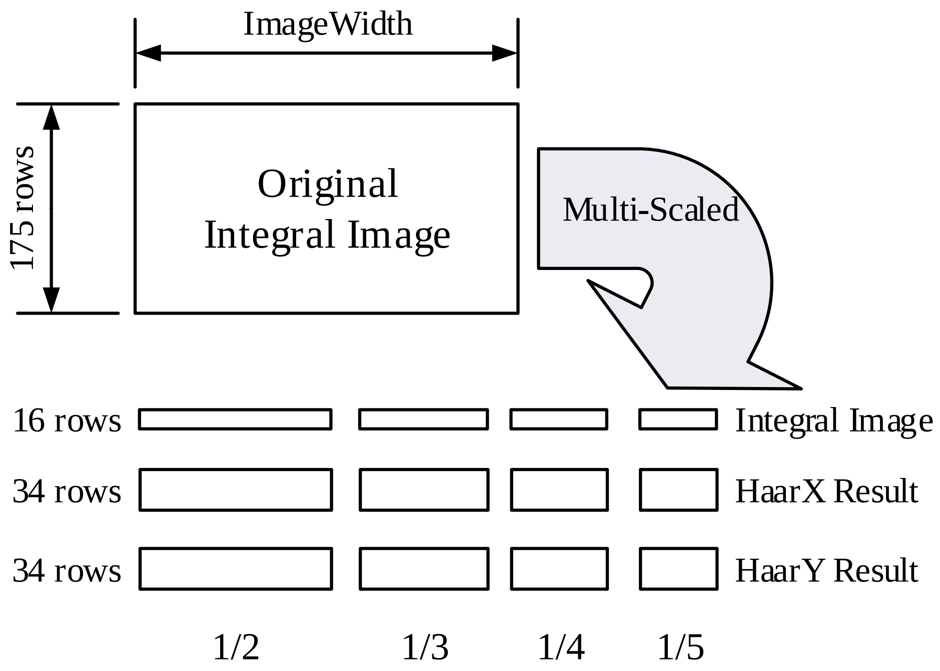
s_0	Distribution of Extracted FPs	Rows Needed	Row-Width			
			320	640	1280	1920
2	54%	71	20.28%	10.14%	5.07%	3.38%
3	29%	105	13.71%	6.86%	3.43%	2.29%
4	11%	140	10.29%	5.14%	2.57%	1.71%
5	5%	175	8.23%	4.11%	2.06%	1.37%



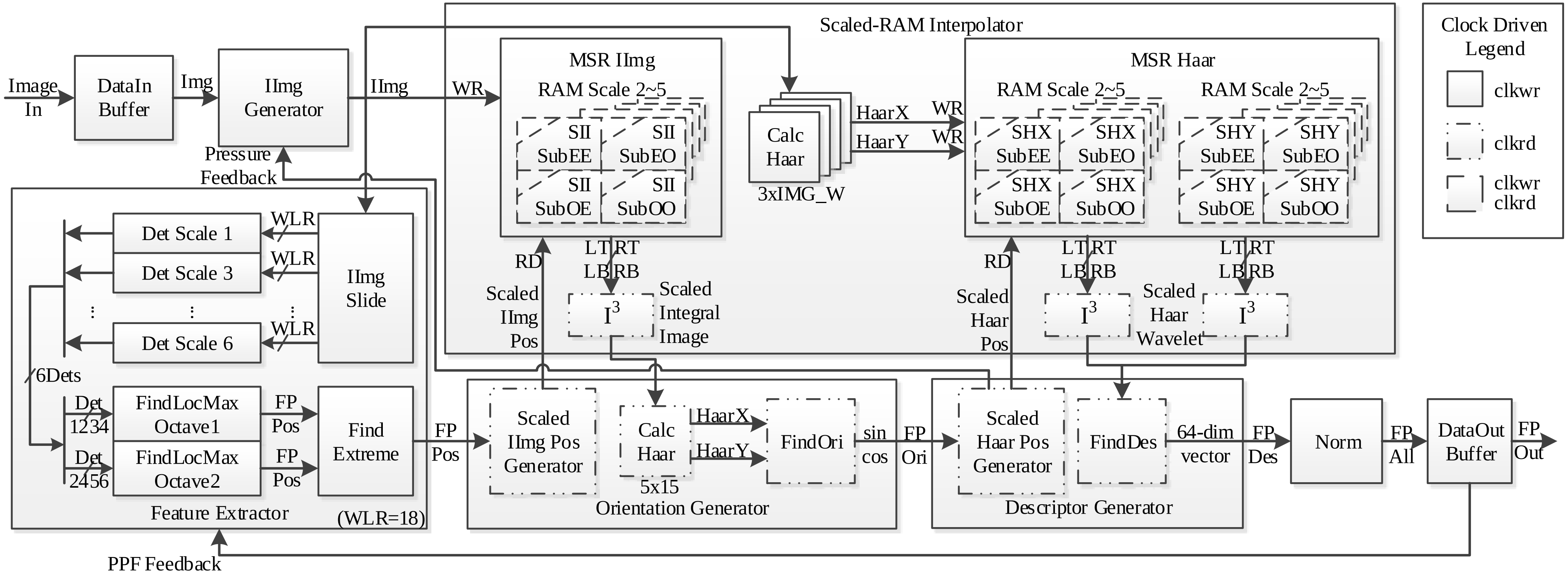
- A large number of rows are required:
  $$span_{\mathrm{IP,max}} = \sqrt{2}(23s_0 + 1) + 2s_0$$
- Only a few of the data are used:
  24x24x8=4608

Multi-Scaled RAM (MSR)

- Scaled Integral Image -> Multi-Scaled RAM
- Haar results of NP are processed on the corresponding scaled RAM
- Normalized scale -> uniform RAM access pattern
- Adjust utilization:
  - 39%, 26%, 19.5%, 15.5%
- Reject redundant data -> save RAM
  - $(16 + 34 \times 2) \times \left(\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5}\right) = 108$
  - RAM saved: $1 - 108/175 = 38\%$



Hardware Implementation



- 16-bit fractional part
- Dual clock domain: I/O and calculation
- Two closed-loop feedbacks:
  - Stall the process of reading image if needed
  - Reduce the number of feature points in a frame

Outline

- Introduction
- Methods
- Experiments
  - Precision evaluation: Better than OpenSURF, $ARMSE_{SW,HW} < 3x10^{-6}$
  - Performance evaluation: MAX-PPS=241KPPS, avg-ACT-PPS=212KPPS
  - Resource evaluation: 22% logic, 43% RAM (@1080P)
- Conclusion

Test Dataset

- Local Feature Evaluation Dataset
  [http://www.robots.ox.ac.uk/~vgg/research/affine/]

- 5 different changes, 8 scenes, 6 images each, around 800x640

Blur

1000x700
6 images

Blur

1000x700
6 images

Viewpoint

800x640
6 images

Viewpoint

1000x700
6 images

Zoom+rotation

765x512
6 images

Zoom+rotation

800x640
6 images

Light

921x614
6 images

JPEG compression

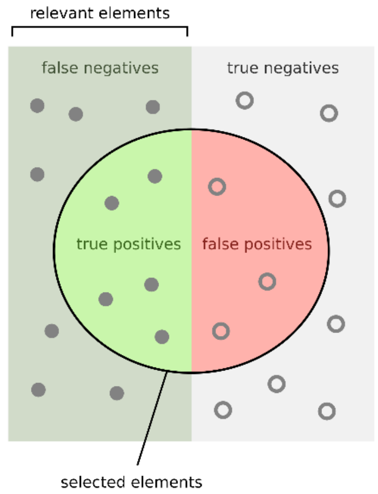
800x640
6 images

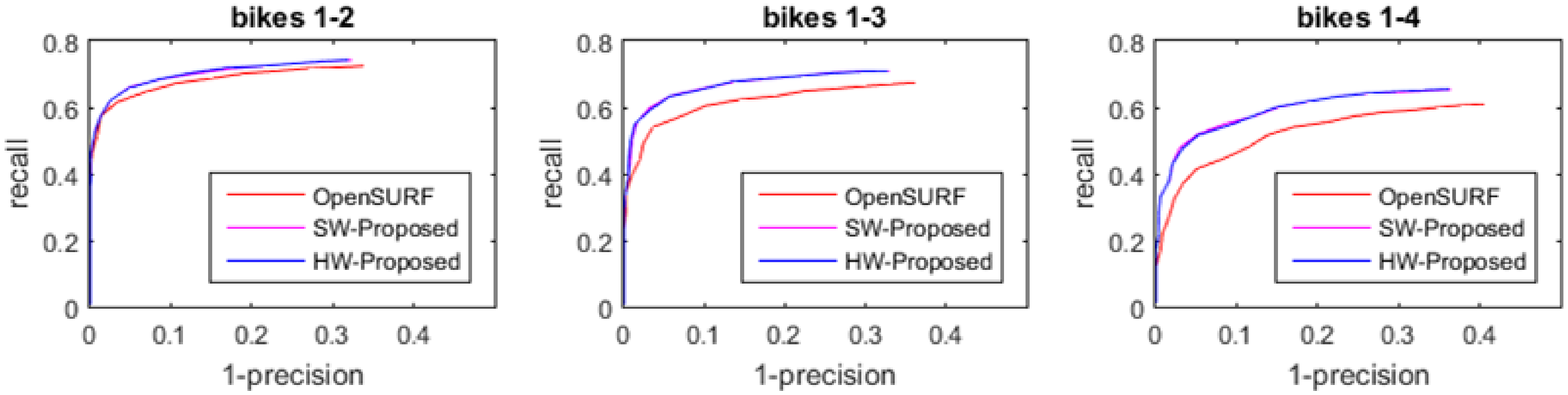
Precision Evaluation

- Curve of *recall*~(1-*precision*)

$$recall = \frac{\#correct\ matches}{\#correspondence}$$

$$1 - precision = \frac{\#\ false\ matches}{\#correct\ matches + \#\ false\ matches}$$

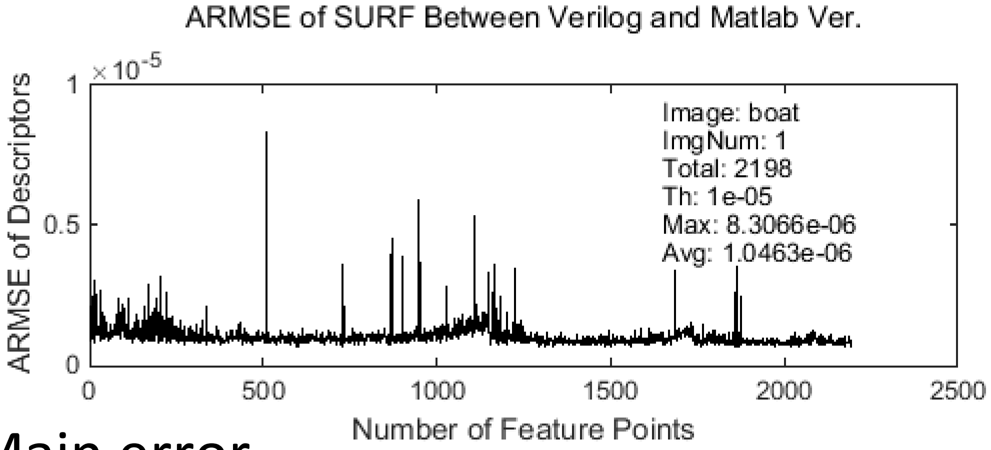


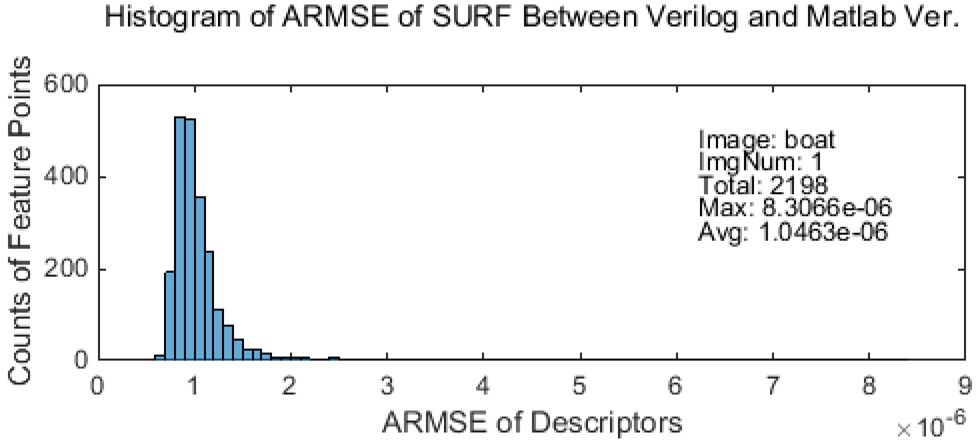


  - Better matching precision than OpenSURF (Matlab version)
  - The loss of detail brought by MSR is compensated by restoring accuracy of scale *s* by I^3
  - Hardware verification results match software (Matlab) results well

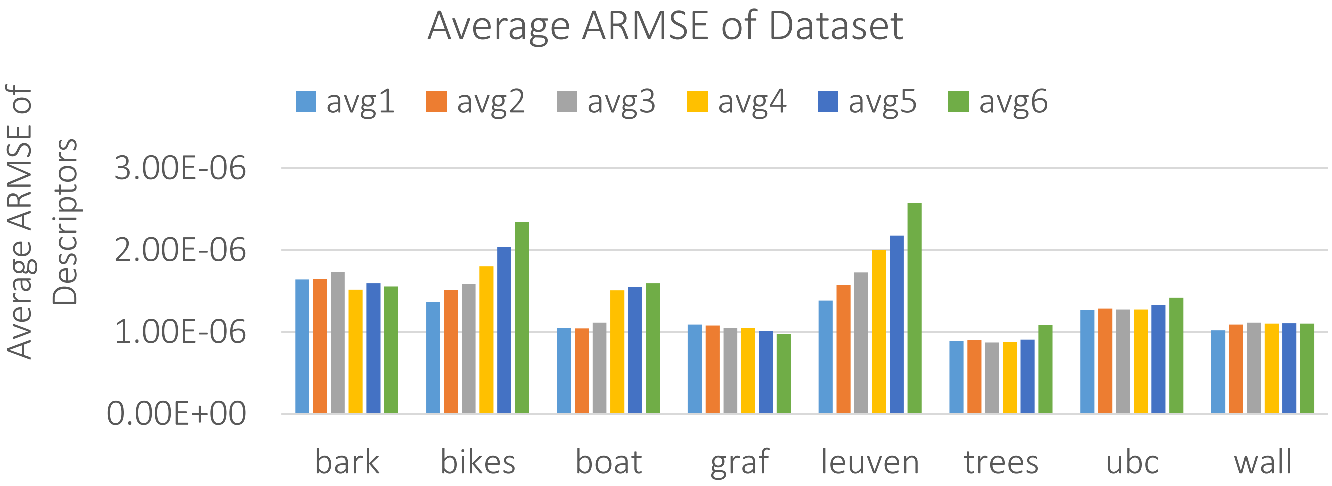
Descriptor Error between SW. and HW.

- Approx-Root-Mean-Square Error: $\mathrm{ARMSE} = \sqrt{\frac{1}{64}\sum_{i=0}^{63}\left(v_{i,\mathrm{SW}} - v_{i,\mathrm{HW}}\right)^2}$



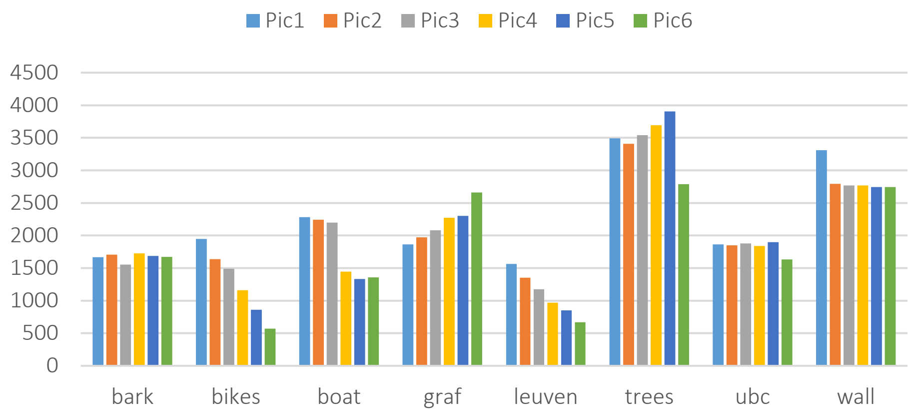


- Main error
  - CORDIC:
    Iterative approximation
- Average ARMSE
  - Less than 3×10^{-6}
  - ±1 bit error for 16-bit descriptor

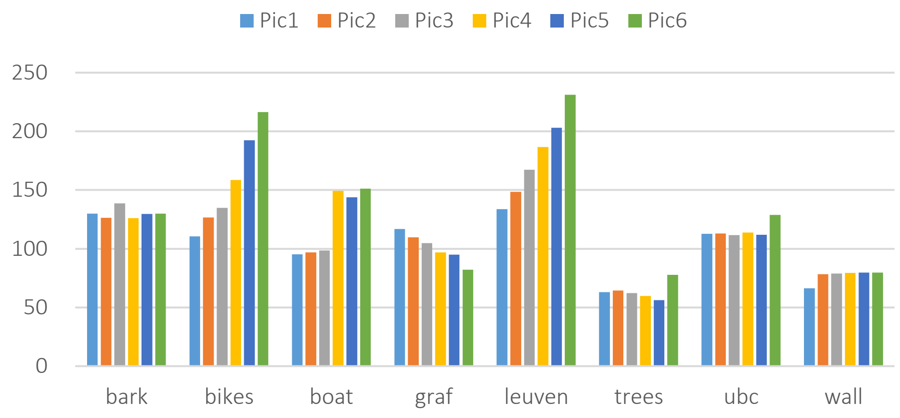


Performance Evaluation

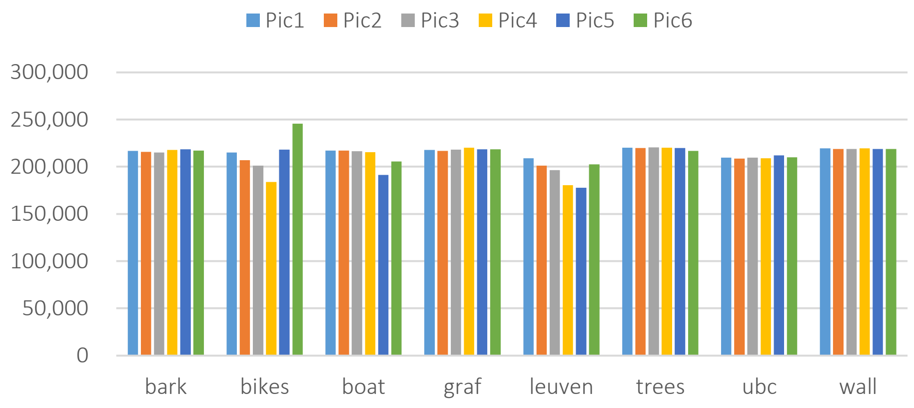
PPF: Relative to resolution and texture
Average is 2K



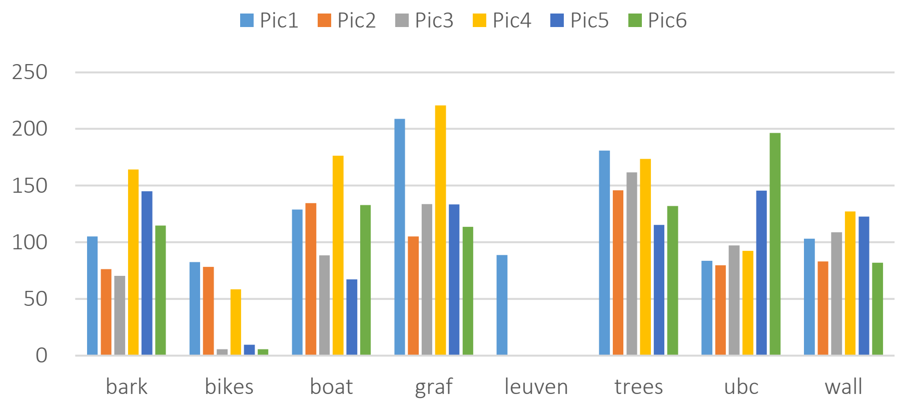
FPS: Negative relative to PPF
Average is 118FPS



PPS: Keep stable. Average is 212K
88.2% of MAX-PPS



Latency(ns): Depend on the amount of FP in the bottom area. Average is 113ns, 1.2% of image



Resource Evaluation

- Altera Stratix III EP3SL340H1152C3

Modules	Registers	18bit DSPs	VGA BRAM bits	1080P BRAM bits
Provided	*270,400*	*576*	*16,662,528*	
IIG+SRI	4.5K/1.7%	21/3.7%	1.7M/10.0%	5.1M/30.7%
FE	25.0K/9.3%	24/4.2%	639K/3.8%	2.0M/12.3%
OG	13.0K/4.8%	12/2.0%	49K/0.3%	50K/0.3%
DG	13.1K/4.9%	32/5.6%	15K/0.09%	16K/0.09%
Norm	5.0K/1.9%	0/0.0%	9K/0.06%	9K/0.06%
Total	60.7K/22.4%	89/15.5%	2.4M/14.3%	7.2M/43.4%

  – Logic resource utilization is below 23%, not sensitive to resolution
  – RAM size is in proportional to row-width, 43% @1920
  – Our system is compact, suit for coexisting with other modules at high resolution

Comparison with Previous Work

Version	Clock	Resolution	FPS	PPF	PPS	Octave	Chip	Function
[GPGPU11]	1.4GHz	791x704	40	800	32K	NA	GTX480	FD+OG+DG
[ReConFig11]	100MHz	640x480	~2	~49	0.1K	8	Virtex 5+PowerPC	FD+OG+DG
[FCCM10]	200MHz	640x480	56	NA	NA	NA	Virtex 5	FD+OG
[BEC12]	25MHz	640x480	60	100	6.0K	6	3x Virtex 4	FD+OG+DG
[TENCON13]	200MHz	300x300	42	250	10.5K	4	Zynq 7	FD+OG+DG
[FPT13]	156MHz	640x480	356	100	35K	6	Virtex 6	FD+OG+DG
[ReConfig14]	25MHz	640x480	131	1614	211K	6	Zynq 7	FD+OG
[CICC13]	200MHz	1920x1080	57	5000	285K	12	ASIC	FD+OG+DG
Proposed	**150MHz**	**640x480**	**488**	**480**	**241K**	**6**	**Stratix III**	**FD+OG+DG**
		1920x1080	**72**	**3250**				

- Points per second
  - MAX-PPS=241KPPS， avg-ACT-PPS=212KPPS
  - Best in FPGA solutions, comparable with the ASIC solution
- Frame rate
  - Best MAX-FPS, avg-ACT-FPS=118FPS

Outline

- Introduction
- Improvement
- Experiments
- Conclusion

Conclusion

- Scaled-RAM Interpolator SURF
  - Interpolation of Integral Image (I^3): Better matching precision
    - better than OpenSURF, $ARMSE_{SW,HW}<3x10^{-6}$
  - Compromise of Interpolation of Integral Image (CI^3): Higher processing speed
    - MAX-PPS=241KPPS, avg-ACT-PPS=212KPPS
  - Multi-Scaled RAM (MSR): Lower storage occupation
    - 22% logic, 43% RAM (@1080P)

- TODO:
  - Making full use of MSR's parallelism among scales

Thanks!

Q&A

References

- [1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," IJCV, 2004.
- [2] Y. Ke et al., "PCA-SIFT: a more distinctive representation for local image descriptors," CVPR, 2004.
- [3] K. Mikolajczyk et al., "A performance evaluation of local descriptors," PAMI, 2005.
- [4] H. Bay et al., "SURF: Speeded Up Robust Features," ECCV, 2006.
- [5] J. Hong et al., "Image Mosaic Based on SURF Feature Matching," ICISE, 2009.
- [6] M.-L. Wang et al., "Object recognition from omnidirectional visual sensing for mobile robot applications," SMC, 2009.
- [7] M. Segundo et al., "Automating 3D reconstruction pipeline by surf-based alignment," ICIP, 2012.
- [8] H. Zhang et al., "Large crowd count based on improved SURF algorithm," TCEC, 2014.
- [9] Evans C. Notes on the OpenSURF Library. Technical report, University of Bristol, 2009.

References Cont.d

- [10] P. Mistry et al., “Analyzing Program Flow Within a Many-kernel OpenCL Application,” GPGPU, 2011.
- [11] L. Liu et al., “SURFEX: A 57fps 1080P resolution 220mW silicon implementation for simplified speeded-up robust feature with 65nm process,” CICC, 2013.
- [12] X. Fan et al., “Implementation of high performance hardware architecture of OpenSURF algorithm on FPGA,” FPT, 2013.
- [13] M. Schaeferling et al., “Object Recognition on a Chip: A Complete SURF-Based System on a Single FPGA,” ReConFig, 2011.
- [14] T. Sledevic et al., “SURF algorithm implementation on FPGA,” BEC, 2012.
- [15] Y.-S. Do et al., “A new area efficient SURF hardware structure and its application to Object tracking,” TENCON, 2013.
- [16] C. Wilson et al., “A power-efficient real-time architecture for SURF feature extraction,” ReConFig, 2014.
- [17]Mikolajczyk K, et al. Local Feature Evaluation Dataset, http://www.robots.ox.ac.uk/~vgg/research/affine/.