High Throughput Hardware Architecture for Accurate Semi-Global Matching

Yan Li, Chen Yang, Wei Zhong, Zhiwei Li, Song Chen*
School of Information Science and Technology
University of Science and Technology of China (USTC)

January 19, 2017

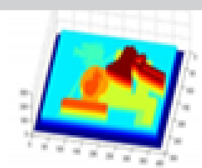
Outline

- **Introduction**
- SGM Algorithm for Hardware Implementation
- Proposed Hardware Architecture
- Experimental Results
- Conclusion

Stereo Vision Systems

- Binocular vs. Monocular
  - Depth information by Triangulation
  - Vision distances
- Applications
  - Automobile, Robots, aerospace, etc.

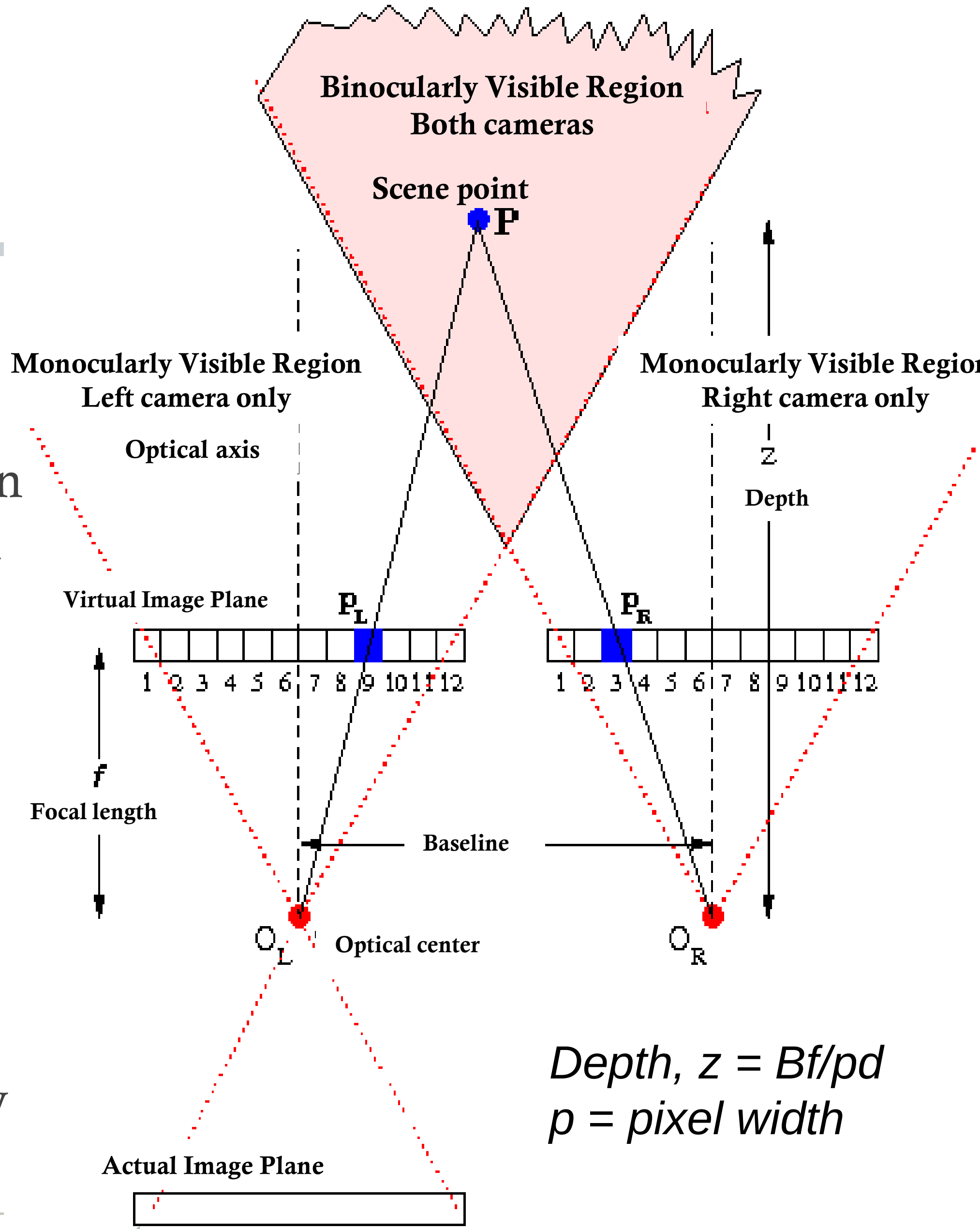
Calibration (offline) → **Rectification** → **Stereo Matching** → **Triangulation** → **Image Recognition and Analysis**



Stereo Matching - basics

- Two cameras: Left and Right
- Optical centers: O_L and O_R
- Virtual image plane: projection of actual image plane through optical center
- Baseline, B: the separation between the optical centers
- Scene Point: P
- Disparity, d = $p_L - p_R = 6$

How to find the pixel p_L and its corresponding pixel p_R accurately and fast?



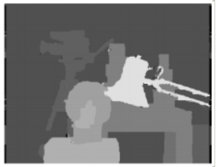
Depth, z = Bf/pd
p = pixel width

Stereo Matching – Challenges & Algorithms

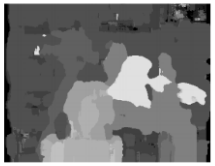
- Challenges: Accuracy, Real time, and Low power
- Global matching algorithms:
  - High accuracy + Low throughput
    - DP, BP, Graph cuts


DP

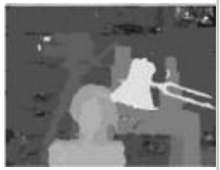

BP


Graph cuts

- Local matching algorithms:
  - Lower accuracy+ High throughput
    - SAD: foreground fattening
    - ADSW[Yoon, IEEE TPAMI 2006]: exponential operation, hardware unfriendly
    - GIF[Ttofis, DATE 2014]: poor disparity accuracies in the low-texture and occluded regions


SAD

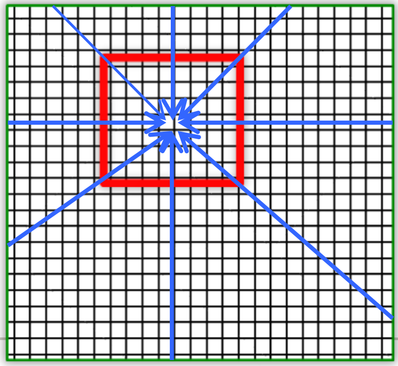

ADSW


GIF

Stereo Matching - Solution

- Semi-global Matching [Hirschmuller, IEEE TPAMI 2008]
  - High Accuracy:
    - Aggregate matching costs along paths(8/16)
    - Make trade-off between accuracy and hardware resources

- Customized Hardware Design
  - Real time processing & Low power consumption
  - Parallel processing
  - Full Pipeline
  - Hardware accelerators (FPGA, ASIC)



Hardware SGM – Previous works

Work	Path num.	Row paralleliz-ation	Weight path cost	External memory	Cost
Gehrig, IEEE ICVS 2009	8	No	No	Yes	ZSAD
Banz, IEEE SAMOS 2010	4	Yes	No	-	Rank Transform
Roszkowski, IEEE DDECS 2014	4	No	Yes	-	Census
Wang, IEEE TCSVT 2015	4	Yes	No	-	AD-census+ Cost Aggregation
Proposed	5	Yes	Yes	No	TAD

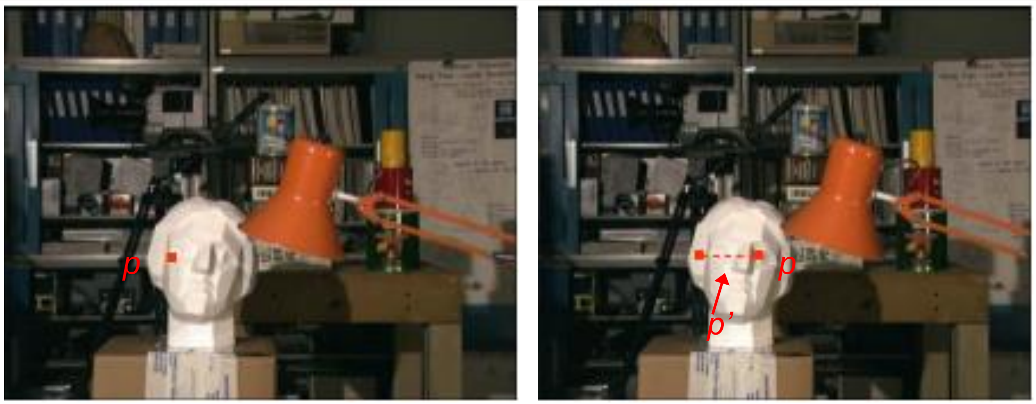
Contributions

- Hardware Architecture
  - Full pipeline
  - Two-row, path and disparity parallelization
  - High throughput: 197fps, 1280x960, 64 disparity levels
- Cost aggregation
  - Five aggregation paths
  - Adaptive weighted path cost aggregation
  - Laplace filter is used to enhance the edge of image
- Disparity refinement
  - Spike removal (SPF)

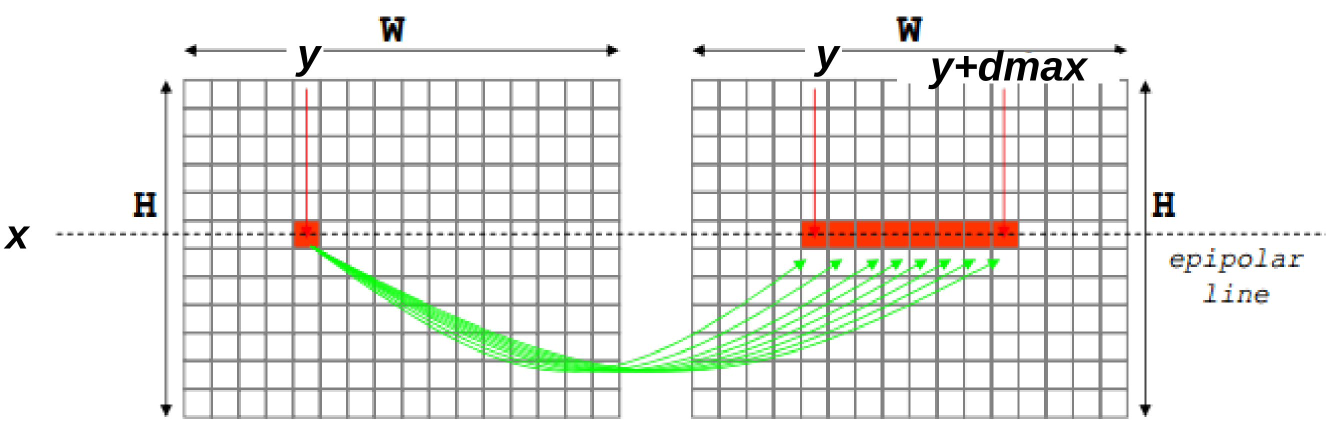
Outline

- Introduction
- **SGM Algorithm for Hardware Implementation**
  - **Pre-processing**
  - **Initial matching cost volume construction**
  - **Semi-global cost aggregation**
  - **Disparity computation**
  - **Post-processing**
- Proposed Hardware Architecture
- Experimental Results
- Conclusion

Stereo Matching

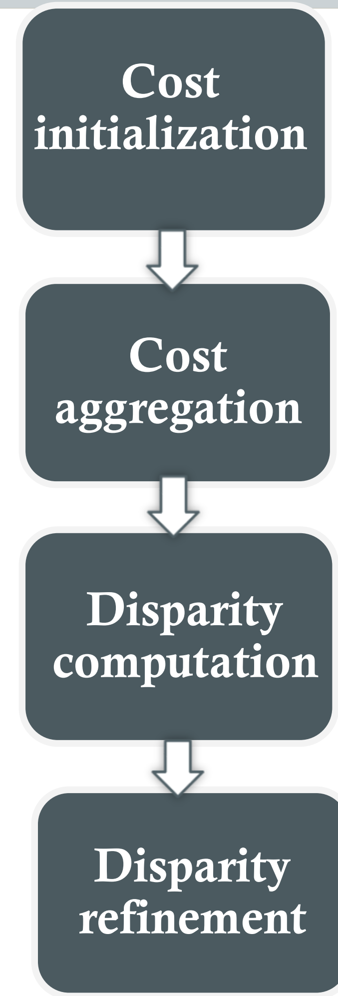


Stereo image pairs



Reference pixel

Stereo correspondence



Pre-processing & Initial matching cost volume construction

- Pre-processing
  - Gaussian filter removes the noise
  - Laplace filter enhances the edge

Guassian

1	2	1
2	4	2
1	2	1

Laplace

-1	-1	-1
-1	9	-1
-1	-1	-1

Gx

-1	0	1
-2	0	2
-1	0	1

Gy

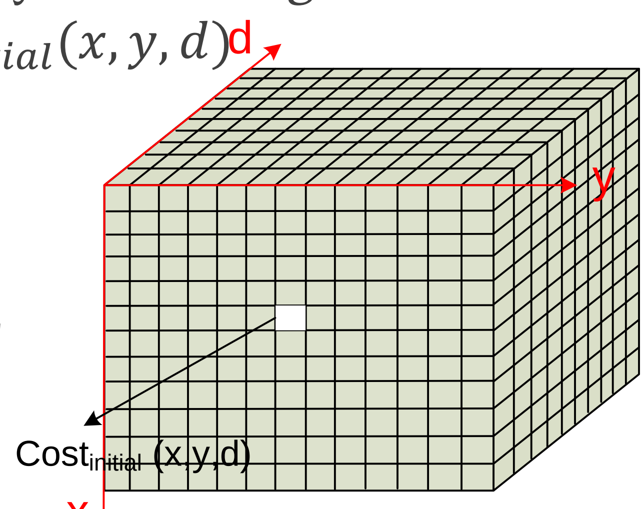
-1	-2	-1
0	0	0
1	2	1

- Initial matching cost computation
  - The truncated absolute difference of intensity and Sobel gradient are used to compute the pixel-wise cost $Cost_{initial}(x, y, d)$

$$M(x, y, d) = |I_l(x, y) - I_r(x, y - d)|,$$

$$G(x, y, d) = \left(\left|\nabla_x\left(I_l(x, y)\right) - \nabla_x\left(I_r(x, y - d)\right)\right| + \left|\nabla_y\left(I_l(x, y)\right) - \nabla_y\left(I_r(x, y - d)\right)\right|\right)/2$$

$$Cost_{initial}(x, y, d) = \min\left(M(x, y, d), T_{gray}\right) + \min\left(G(x, y, d), T_{grad}\right)$$



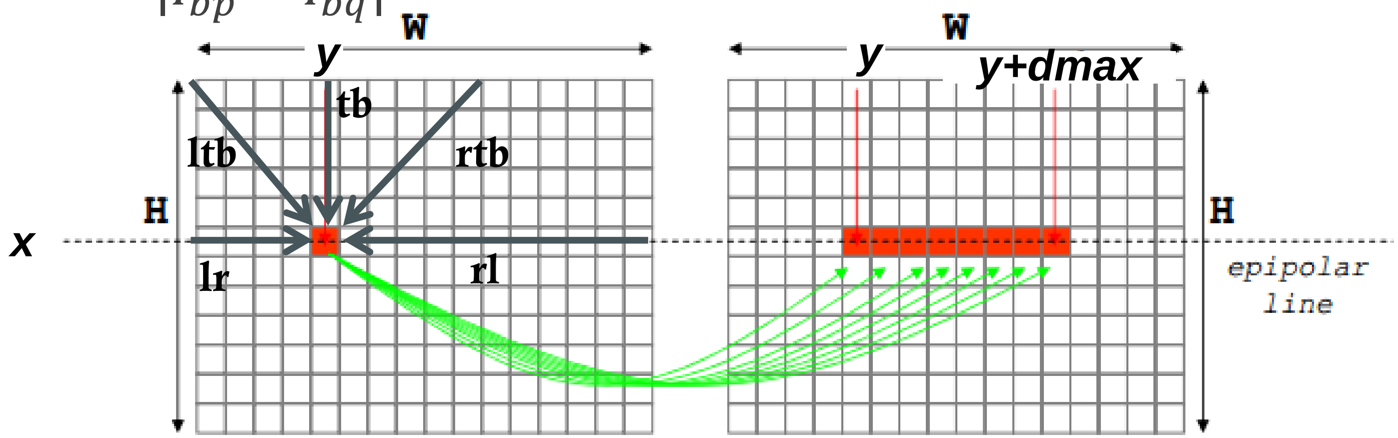
Initial Matching Cost Volume

Semi-global cost aggregation

- The $L_r(p,d)$ along a path r of the pixel p at disparity d is computed as follows[Hirschmuller, IEEE TPAMI 2008] :

$$L_r(p,d) = Cost_{initial}(p,d) + \min\begin{pmatrix} L_r(p-r,d), \\ L_r(p-r,d\pm 1)+P_1, \\ \min_k L_r(p-r,k)+P_2 \end{pmatrix} - \min_k L_r(p-r,k)$$

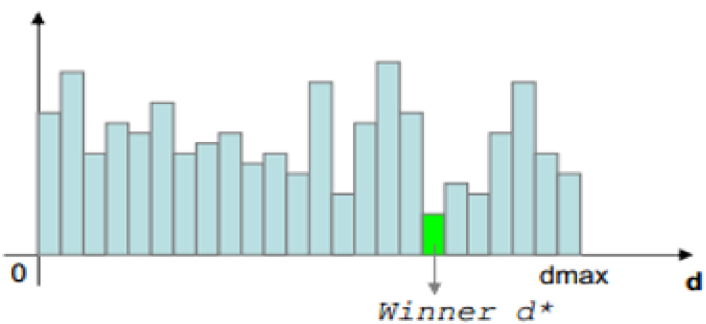
$$P_2 = \frac{P_2'}{|I_{bp}-I_{bq}|}, \mathrm{r} = \mathrm{ltb}(1,1), \mathrm{tb}(1,0), \mathrm{rtb}(1,-1), \mathrm{lr}(0,1), \mathrm{rl}(0,-1)$$

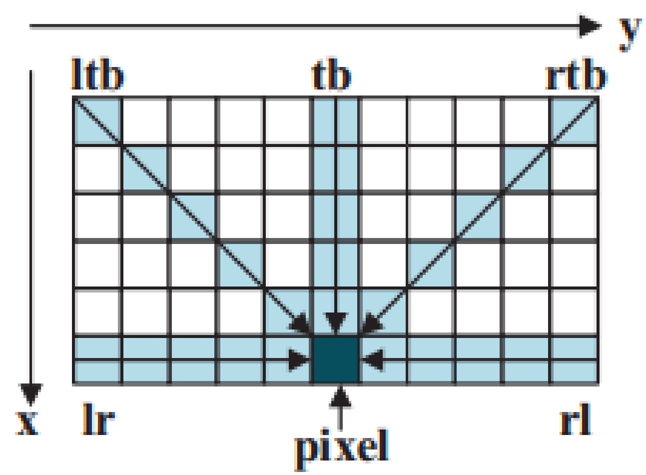


Stereo image pairs

Semi-global cost aggregation & Disparity computation

- The adaptive weight of path costs: 6×11 windows
  - $W_r(p) = \sum_{i=0,1,2,\ldots,R} W_{ir}(p - i \cdot r), i = D_{chebyshev}(p, p - i \cdot r)$,
  - $W_{ir}(p - i \cdot r) = \begin{cases} (R+1) - |i| & |I(p) - I(p - i \cdot r)| < Th \\ 0 & else \end{cases}$,
  - $S(p, d) = \sum_r W_r(p) \cdot L_r(p, d)$.





- Disparity computation
  - The disparity of each pixel is computed by WTA approach.

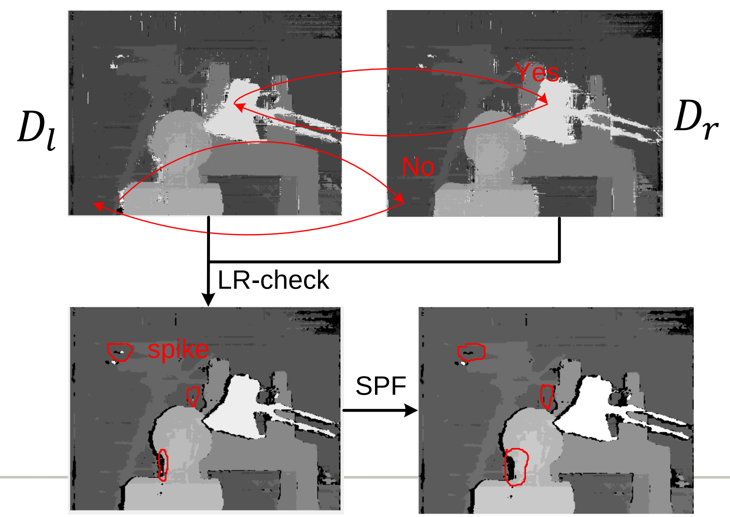
$$D_p = argmin_{0 \leq d \leq d_{max}-1} S(p, d)$$

Post-processing

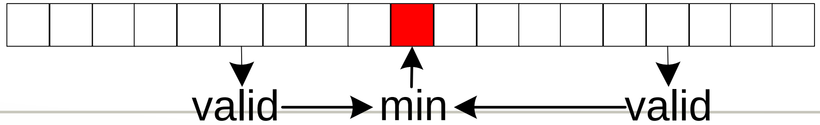
- Detect the mismatching and occluded pixels by L-R check.

$$D_p \begin{cases} D_l & |D_l - D_r| \leq 1 \\ 0 & else \end{cases}$$

- Remove spikes by SPF
  - $W_p(x+i, y+j) = \begin{cases} 1 & if \frac{D_p(x,y)\times(256-spf\_r)}{256} \leq D_p(x+i, y+j) \leq \frac{D_p(x,y)\times(256+spf\_r)}{256} \\ 0 & else \end{cases}$
  - $N_p(x,y) = \sum_{-r\leq i\leq r, -r\leq j\leq r} W_p(x+i, y+j)$
  - $D_p(x,y) = \begin{cases} D_p(x,y) & if N_p(x,y) \geq Th \\ 0 & else \end{cases}$



- Fill the disparity of invalid pixel

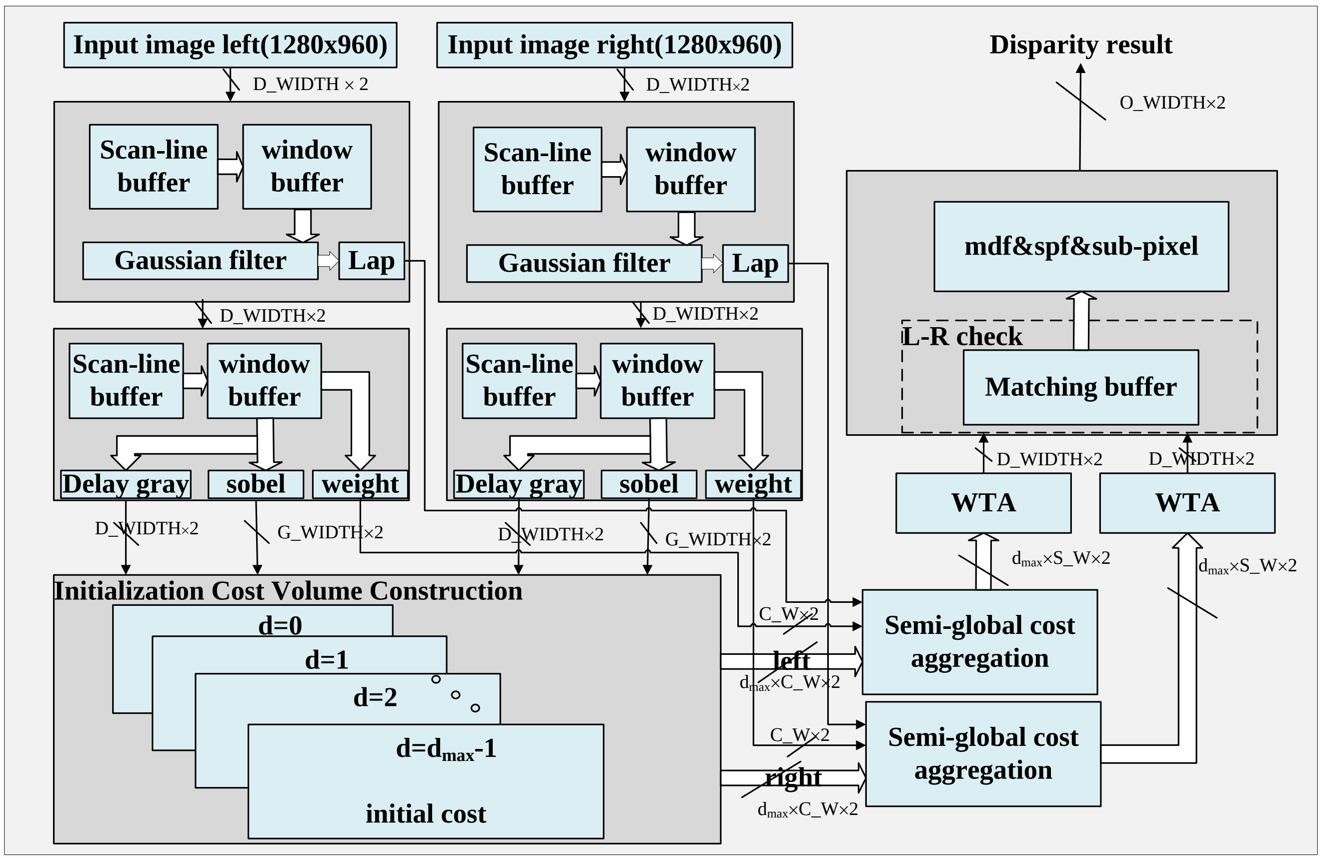


Outline

- Introduction
- SGM Algorithm for Hardware Implementation
- **Proposed Hardware Architecture**
  - **Overview**
  - **Pre-processing module**
  - **Initial matching cost volume construction**
  - **Semi-global cost aggregation architecture**
  - **Post-processing**
- Experimental Results
- Conclusion

Overview – Hardware Architecture

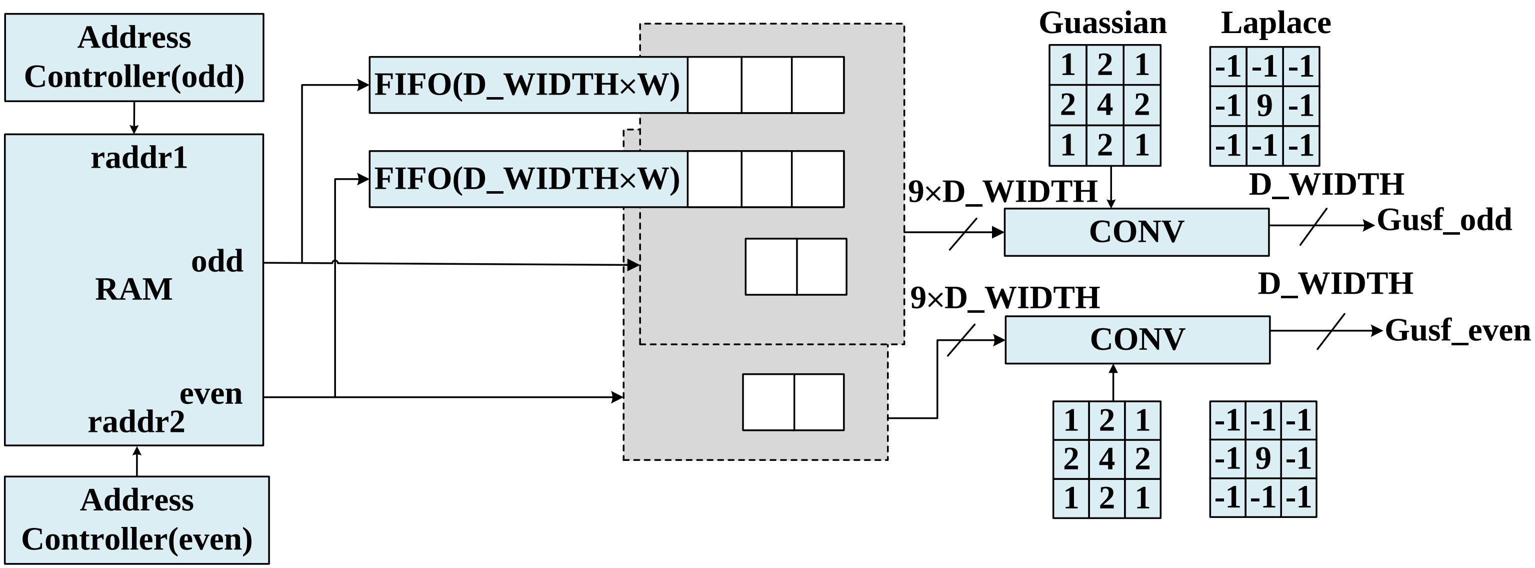
- Hardware Architecture



- Fully pipeline
- Disparity, path, and row parallelization
- Throughput: 960p/197fps
- F_{max}:156MHz
- Disparity range: 64/16
- Flexible Resolution

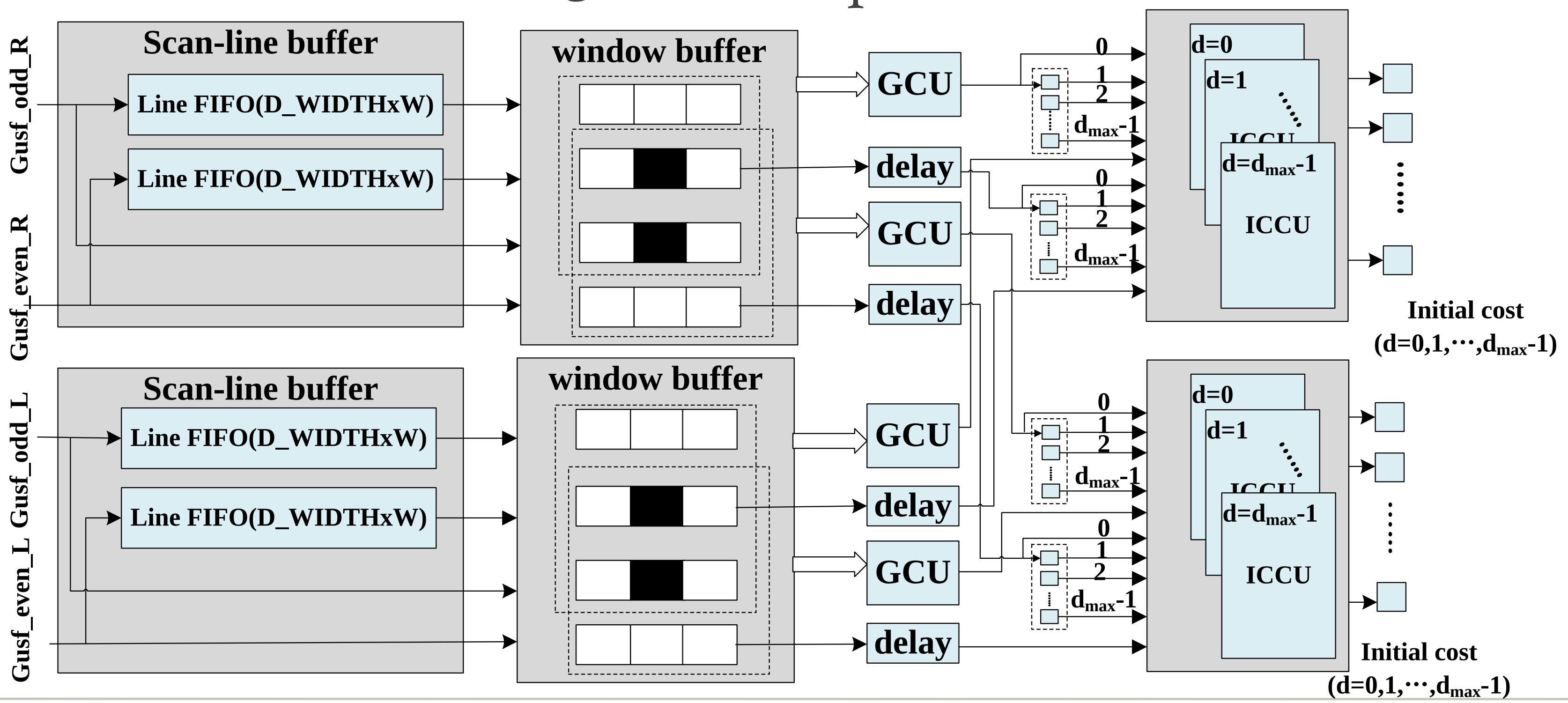
Pre-processing module – Gaussian and Laplace filter

- Two line FIFOs capture pixels from four adjacent rows.
- CONV performs convolution of 3x3 windows consisting of a set of registers with the filter kernels



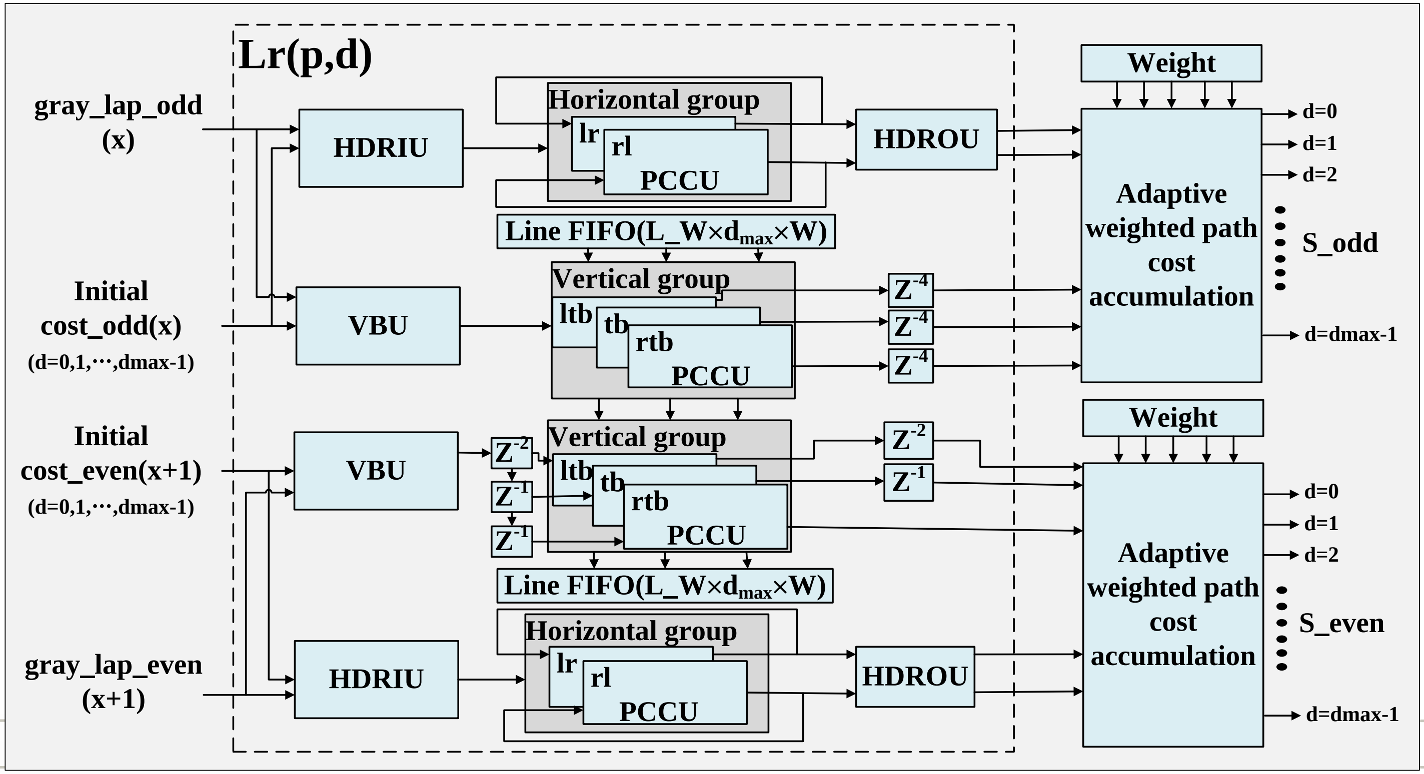
Initial matching cost volume construction

- Disparity parallel: 64
- Scanline buffer: FIFO
- Window buffer: 3x3 register
- GCU: Sobel gradient computation
- ICCU: 64x2, initial matching cost computation



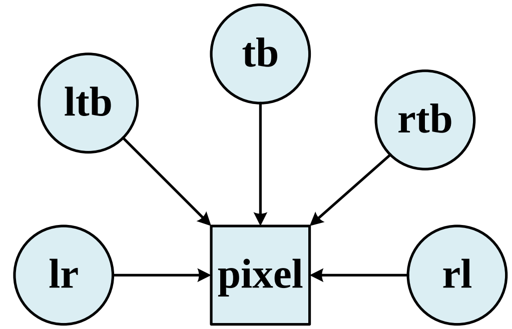
Overview - Semi-global cost aggregation architecture

- Horizontal group: lr and rl.
- Vertical group: ltb, tb and rtb. The path cost computation depends on the path costs in the upper row.



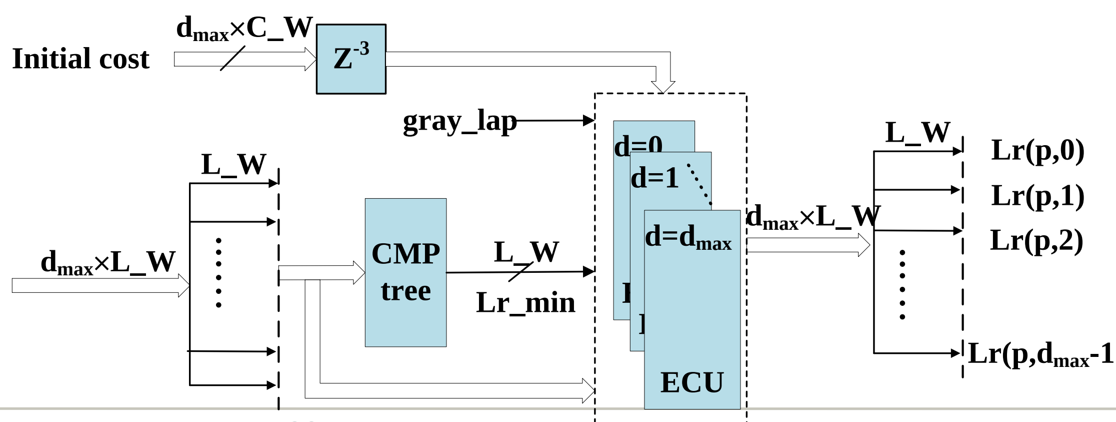
Semi-global cost aggregation architecture – Path cost computation

- PCCU(path cost computation)
  - Compare tree
  - Three-stage pipeline
  - fixed point divider (P2): four-stage pipeline



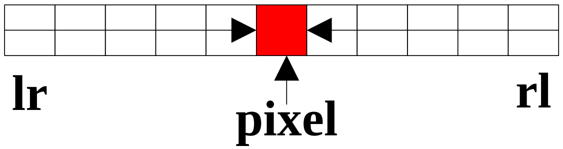
$$L_r(p,d) = Cost_{initial}(p,d) + \min\begin{pmatrix} L_r(p-r,d), \\ L_r(p-r,d\pm 1)+P_1, \\ \min_k L_r(p-r,k)+P_2 \end{pmatrix} - \min_k L_r(p-r,k)$$

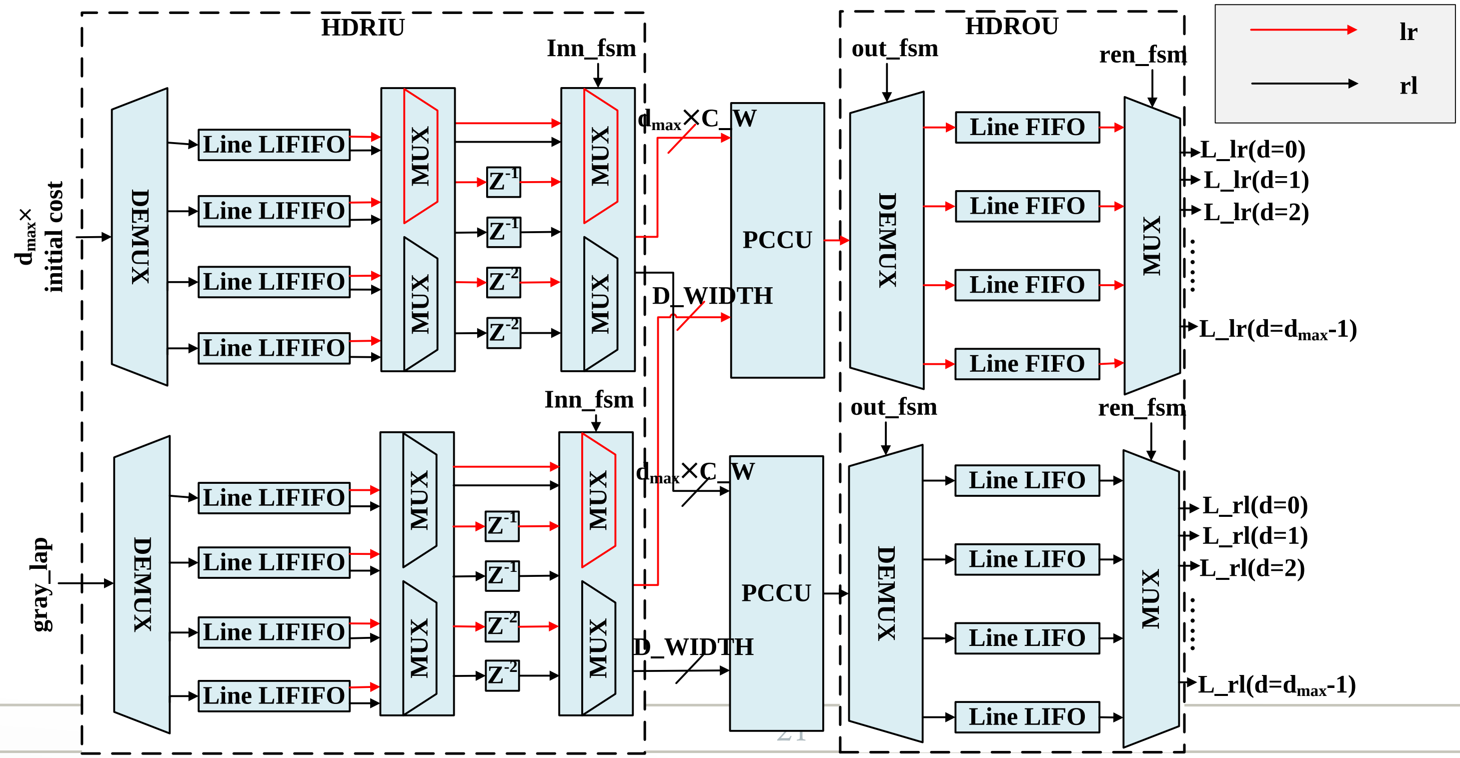
$$P_2 = \frac{P_2'}{|I_{bp} - I_{bq}|}$$



Semi-global cost aggregation architecture – Horizontal path cost computation

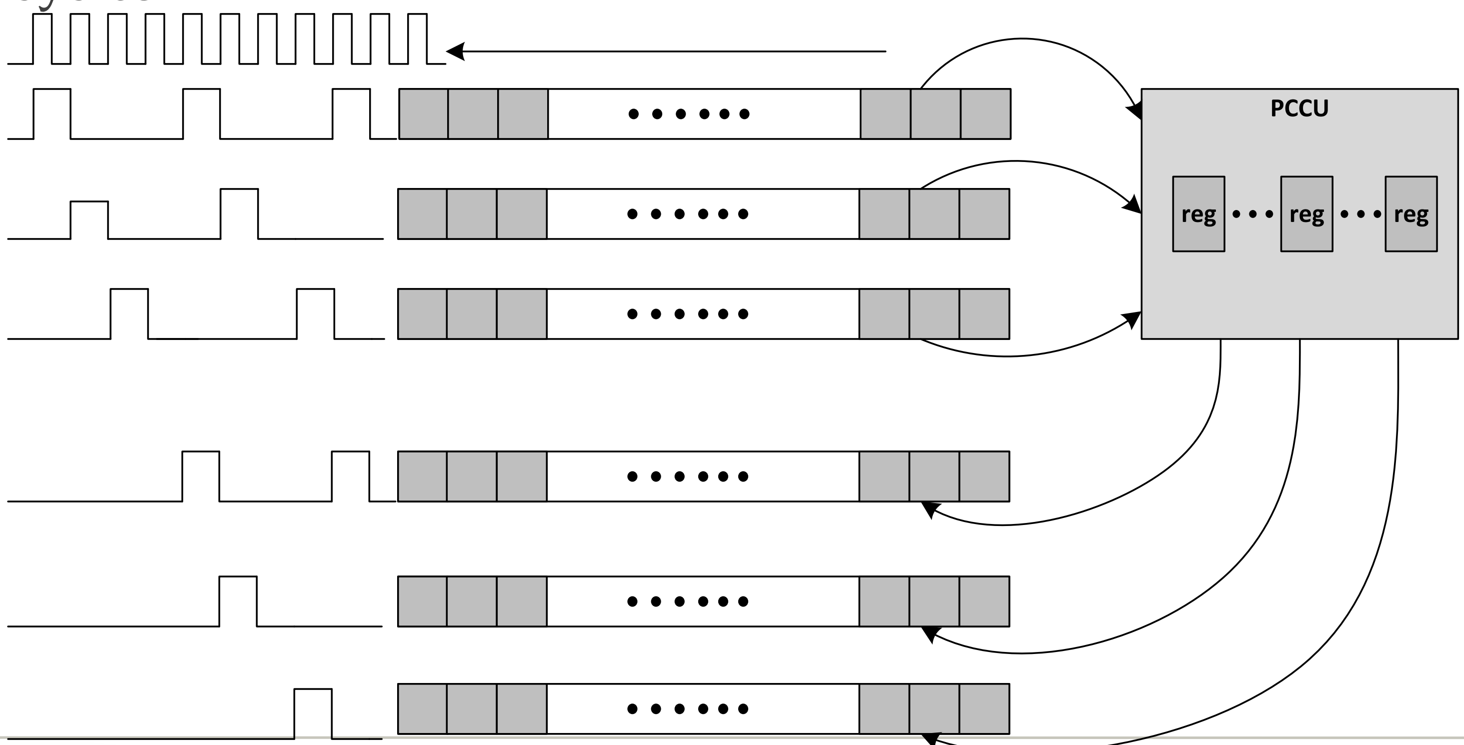
- Horizontal Architecture
  - ping-pong buffer
  - LIFIFO(dual-port RAM), LIFO





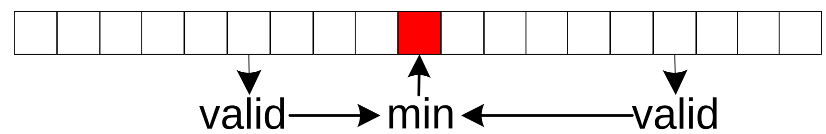
Semi-global cost aggregation architecture – Horizontal path cost computation

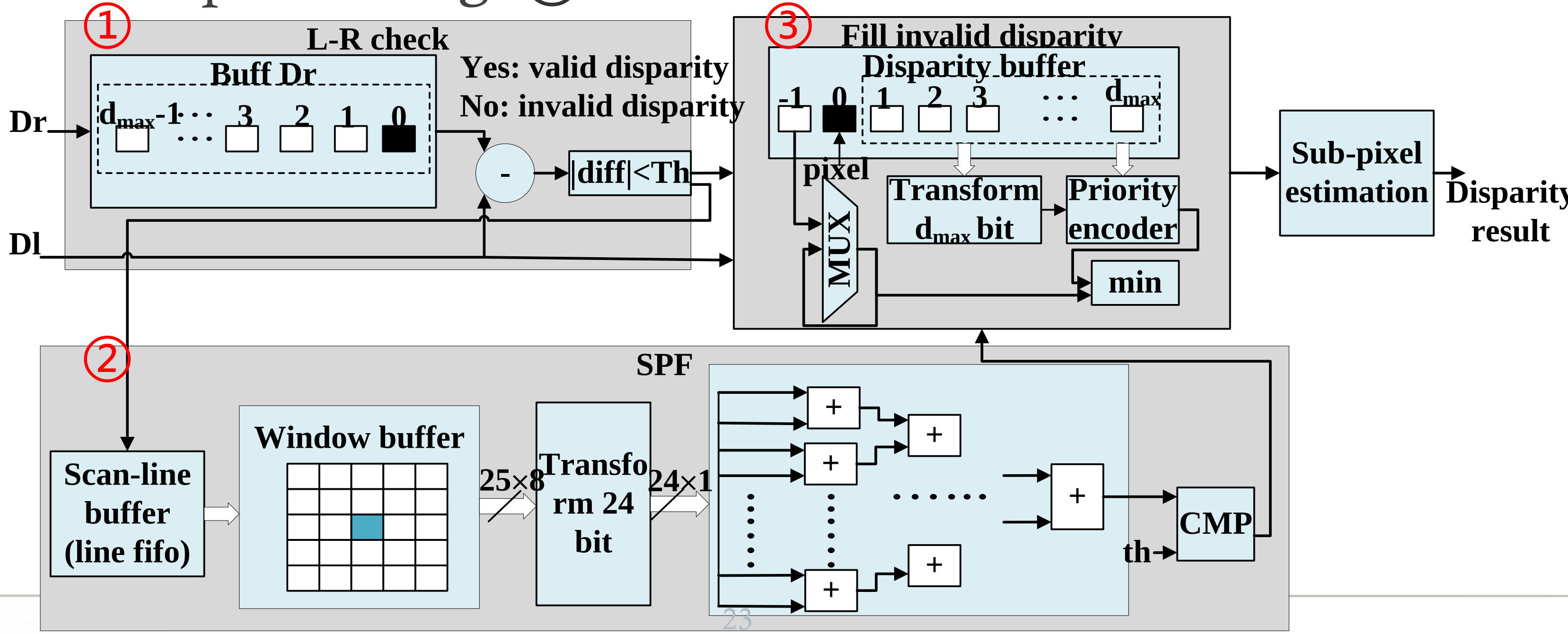
- Horizontal Architecture
  - Compute three pixels from three different rows in three clock cycles



Post-processing: L-R check, SPF, Subpixel interpolation and Median filter

- Occlusion and mismatching detection: ①
- Spike removal: ②
- Invalid pixel filling: ③



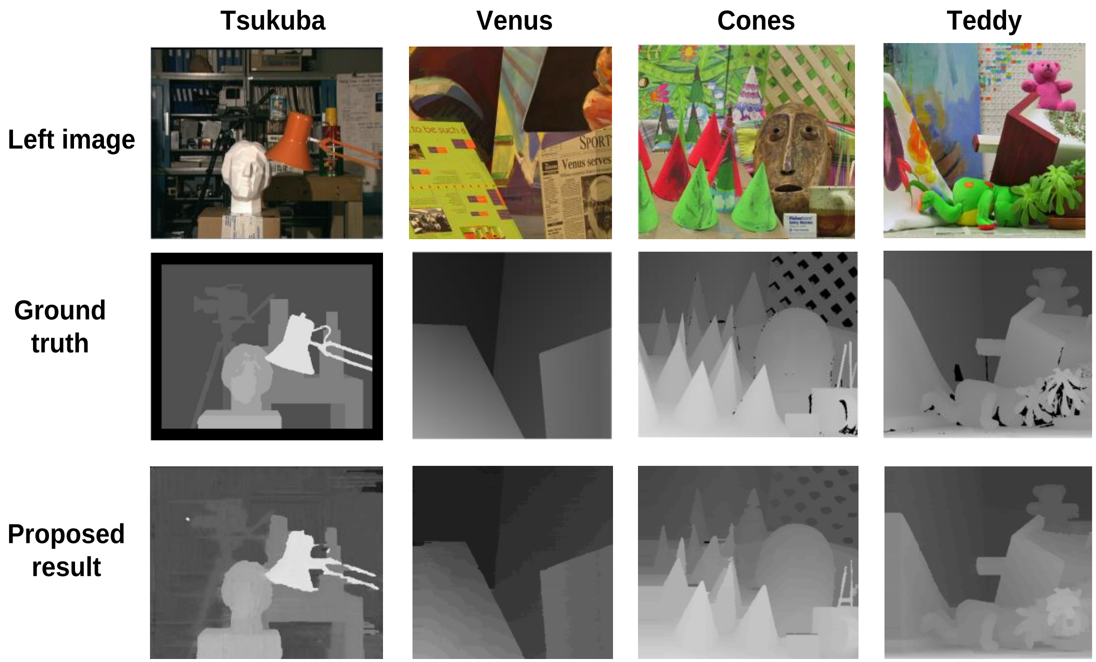


Outline

- Introduction
- SGM Algorithm for Hardware Implementation
- Proposed Hardware Architecture
- **Experimental Results**
  - **Quality of results**
  - **Hardware performance and consumption**
- Conclusion

Experimental results – Quality

- Test bench: Middlebury
- Test images: Tsukuba, Venus, Teddy and Cones
- Average error rate: 6.03%
  - Lower accuracy in Venus test image with large plain area



Experimental results – Quality

- The influence of the path number and adaptive weighted path costs are given in the following table.
- Using one more path and adaptive weighted path cost aggregation improve the accuracy by 3.69%

Variants	Average error			
	nonocc	all	disc	Overall
5path+weight	2.58	6.19	9.32	6.03
5path	3.19	6.80	13.32	7.77
4path+weight	3.37	7.03	12.31	7.57
4path	4.38	8.01	16.78	9.72

Experimental results – Hardware performance and consumption

- Quartus II, Altera Stratix V

Work	nonocc				Average error rate				Image size	D[1]	Speed (fps[2])	MDE/s[3] (10^6)	platform
	Tsukuba	Venus	Teddy	Cones	nonocc	all	disc	overall					
Wang[11]	2.39	0.38	6.08	2.12	2.74	6	8.10	5.61	1600×1200	128	42.61	10472	FPGA
Proposed	2.79	0.68	4.18	2.67	2.58	6.19	9.32	6.03	1280×960	64	197	15492	FPGA
Ttofis[6]	4.04	1.55	7.52	2.77	3.97	6.80	9.30	6.69	1280×720	64	60	3538	FPGA
Hirschmuller[8]	3.26	1.00	6.02	3.06	3.34	6.87	12.3	7.50	-	-	-	-	CPU
Banz[9]	4.1	2.7	11.4	8.4	6.7	-	-	-	640×480	128	103	4050	FPGA
Gehrig [13]	5.86	3.85	13.28	9.54	8.13	-	-	-	340×200	64	27	118	FPGA
Shan[14]	-	-	-	-	-	17.3	-	-	1280×1024	256	46	15437	FPGA

[1] D: Disparity [2] fps: frame per second [3] MDE/s: Million Disparity Estimation per second (M × N × D × fps)

- Throughput: 1280x960/197fps at 64 disparity levels, MDE=15492M, $f_{max} = 156MHz$
- Hardware resource utilization: logic utilization 96,084/234,720 (41%), total registers 83,156/469,440 (18%), block memory bits 20,329,470/52,428,800 (39%)
- Power: 16.603W

Outline

- Introduction
- SGM Algorithm for Hardware Implementation
- Proposed Hardware Architecture
- Experimental Results
- **Conclusion**

Conclusion

- This work proposed a high-throughput hardware architecture for accurate semi-global matching using two-row parallelization and disparity parallelization.
- Five path costs are adaptively weighted to improve the disparity accuracy by 3.69%.
- The implementation on FPGA shows a throughput of 1280×960/197fps at 64 disparity levels in 156MHz.
- The results show high quality.

Thank you very much!

Any questions?