

計算機システム

第2回 CA0302:マルチメディア命令とVLIW

<http://archlab.naist.jp/Lectures/ARCH/ca0302/ca0302j.pdf>

Copyright © 2021 奈良先端大 中島康彦

II-1. ステレオ画像処理

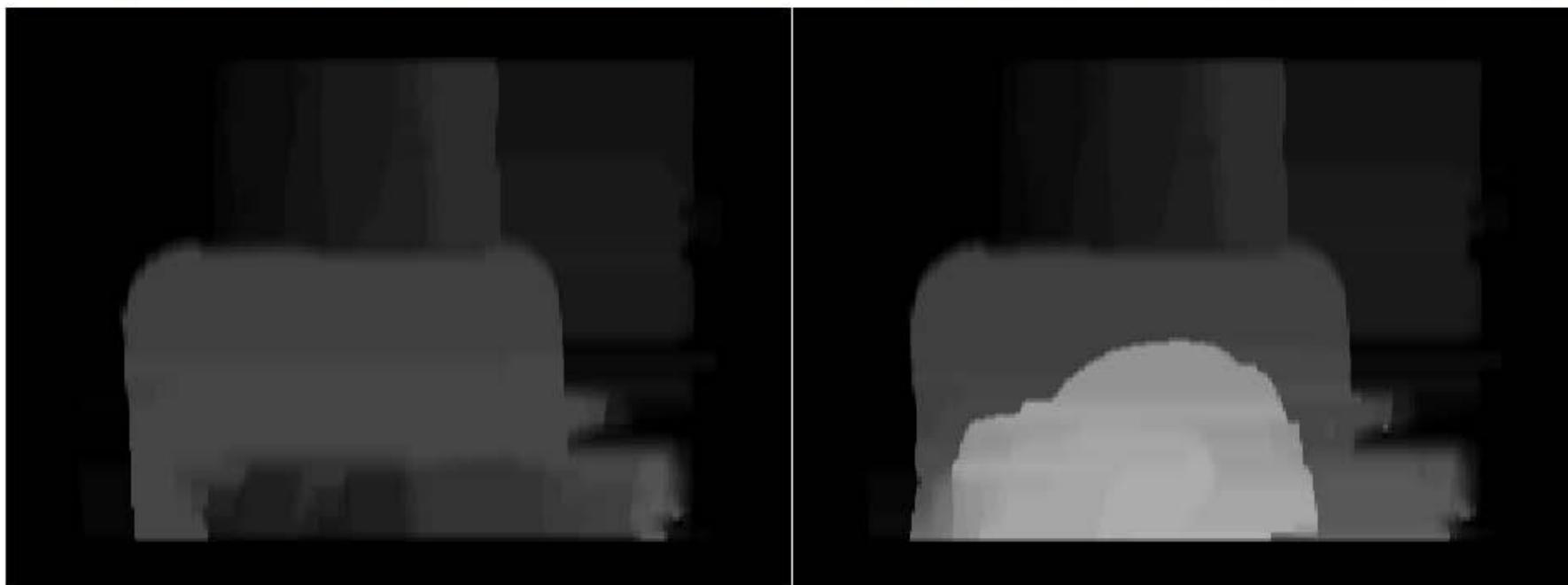
左目画像と右目画像から，奥行き画像を生成



II-4. 普段はない物体の検知

近距離に出現した物体の検出

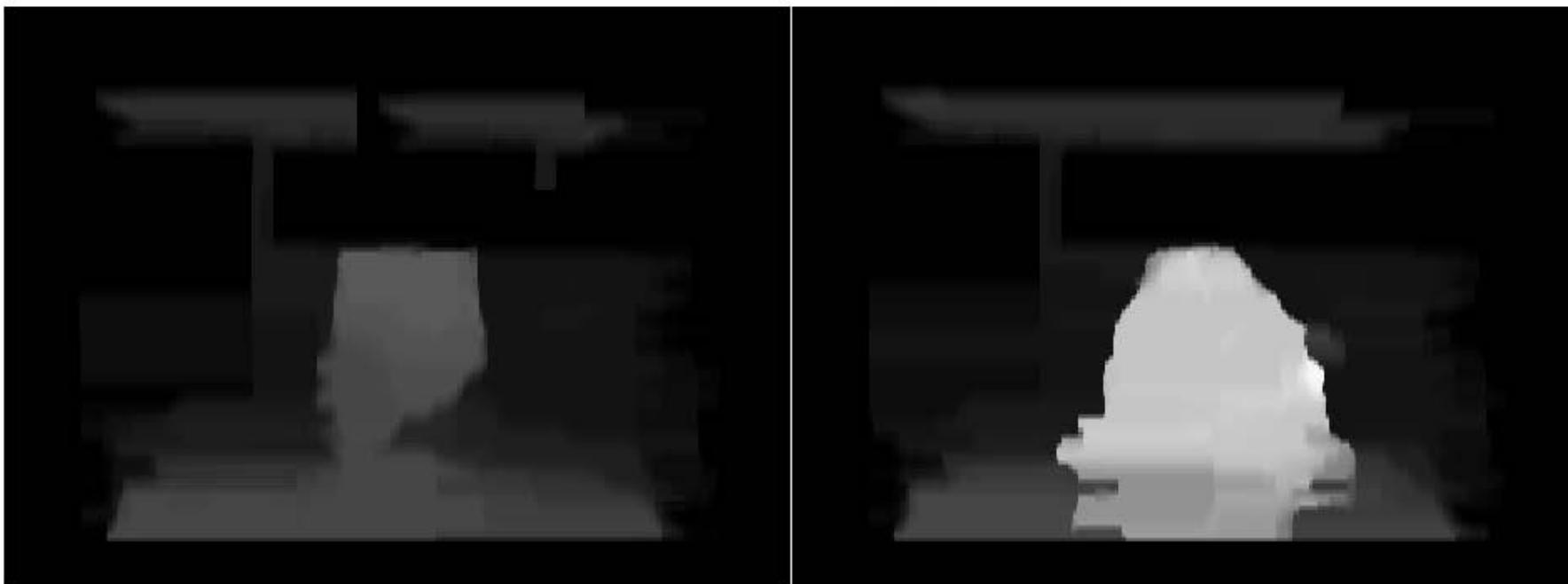
▶ 白領域の出現



II-5. 接近する物体の検知

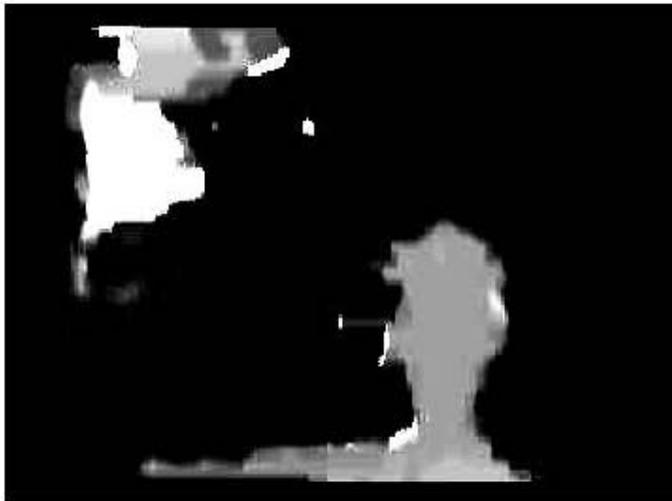
物体の接近

- ▶ 白領域の明度増大



II-6. 画像認識の前処理

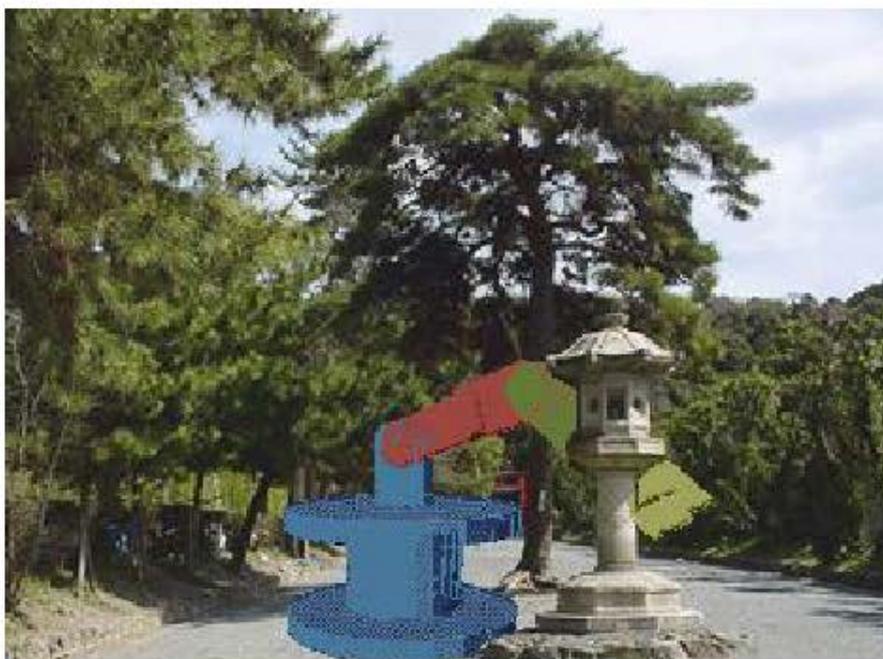
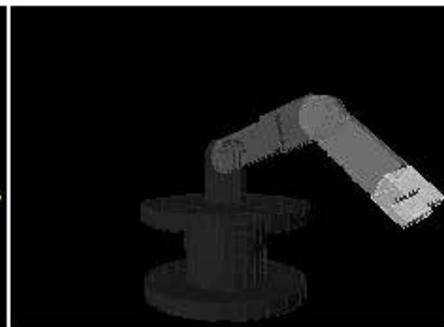
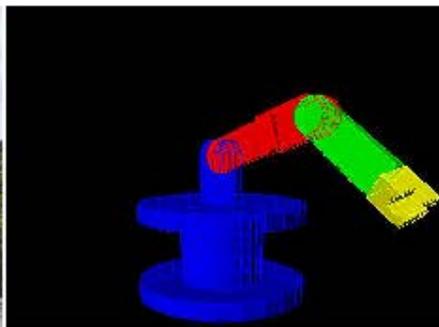
物体の切り出しと認識



II-7. 実写映像とCGの合成

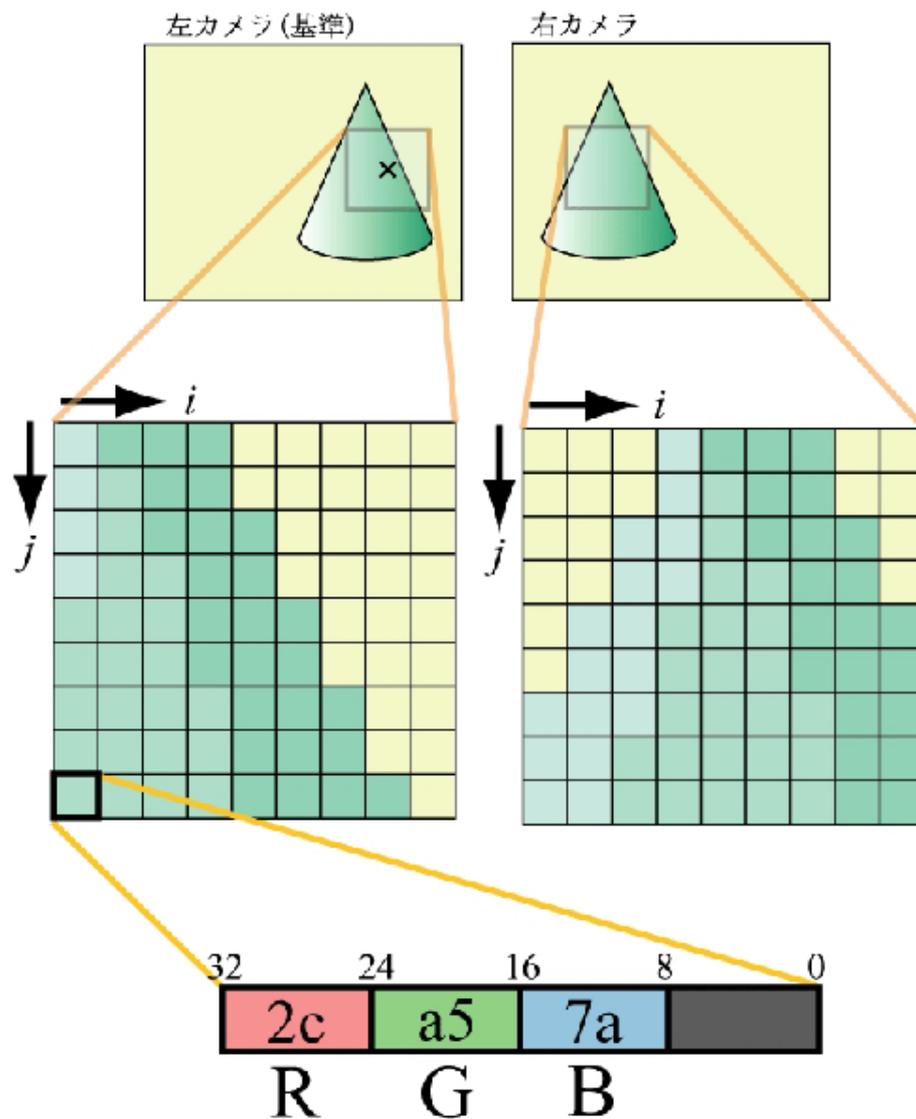
奥行き情報に基づく合成 (Z-keying)

- ▶ 原理的には実写どうしても可能



III-4. 視差を求めるのは意外に難しい (つづき)

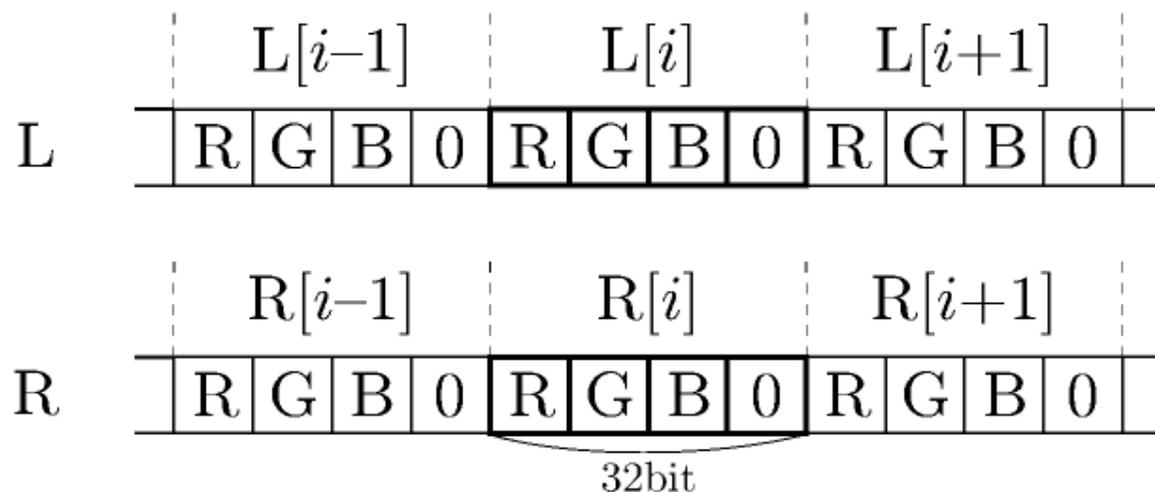
左右の画像から、互いに最も類似する一部分を見つけ出す



III-5. 画素差の計算

SAD (Sum of Absolute Difference)…画素差絶対値の総和

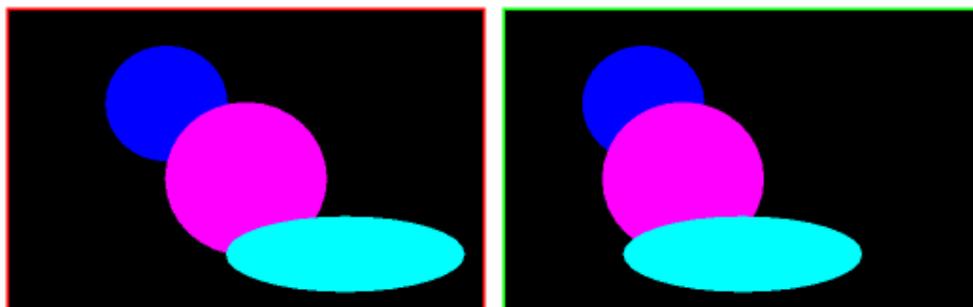
- ▶ 全体解像度320×240に対し、21×21ピクセル程度のウィンドウ
- ▶ 各ピクセルはRGB各8ビットで4バイト境界に配置



$$DIFF_i = |L[i]_R - R[i]_R| + |L[i]_G - R[i]_G| + |L[i]_B - R[i]_B|$$

IV-5. 2台のカメラによる「寄り目」法（つづき）

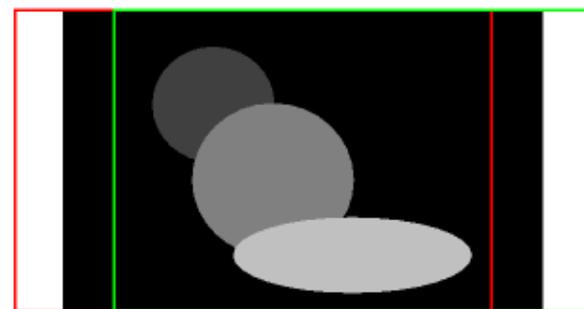
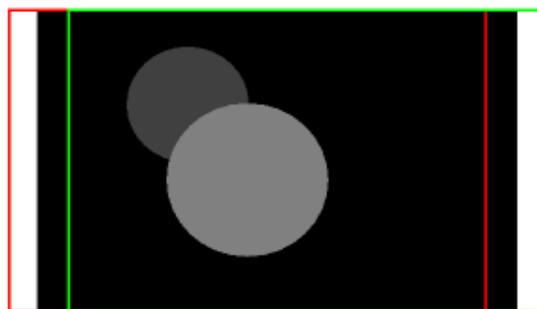
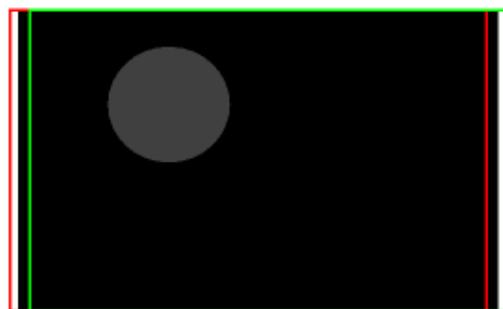
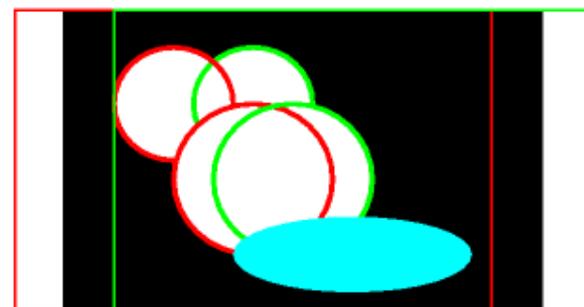
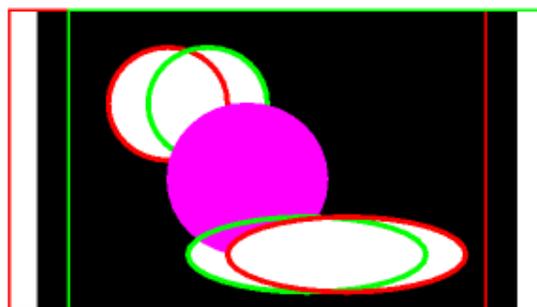
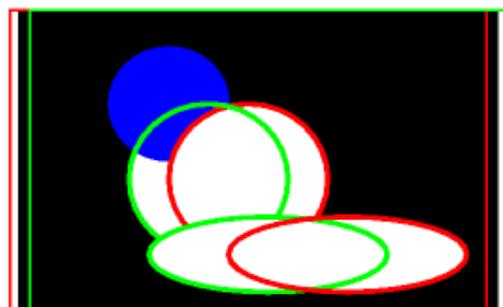
遠方から手前に向かって物体を探索



遠くに焦点

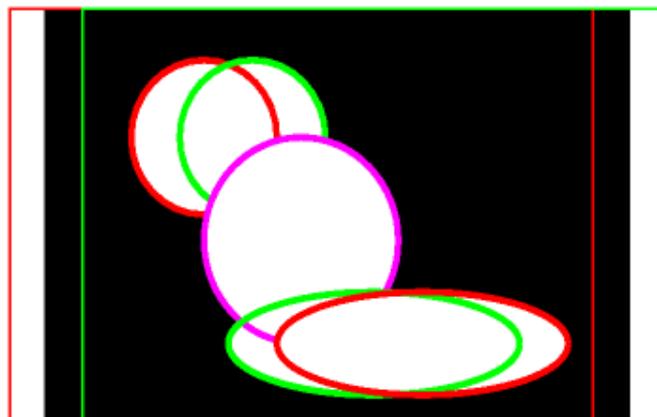
中間に焦点

近くに焦点



IV-5. 2台のカメラによる「寄り目」法 (つづき)

輪郭部分の画素差最小をもって、物体と認識



L

画素
輪郭



R

画素
輪郭

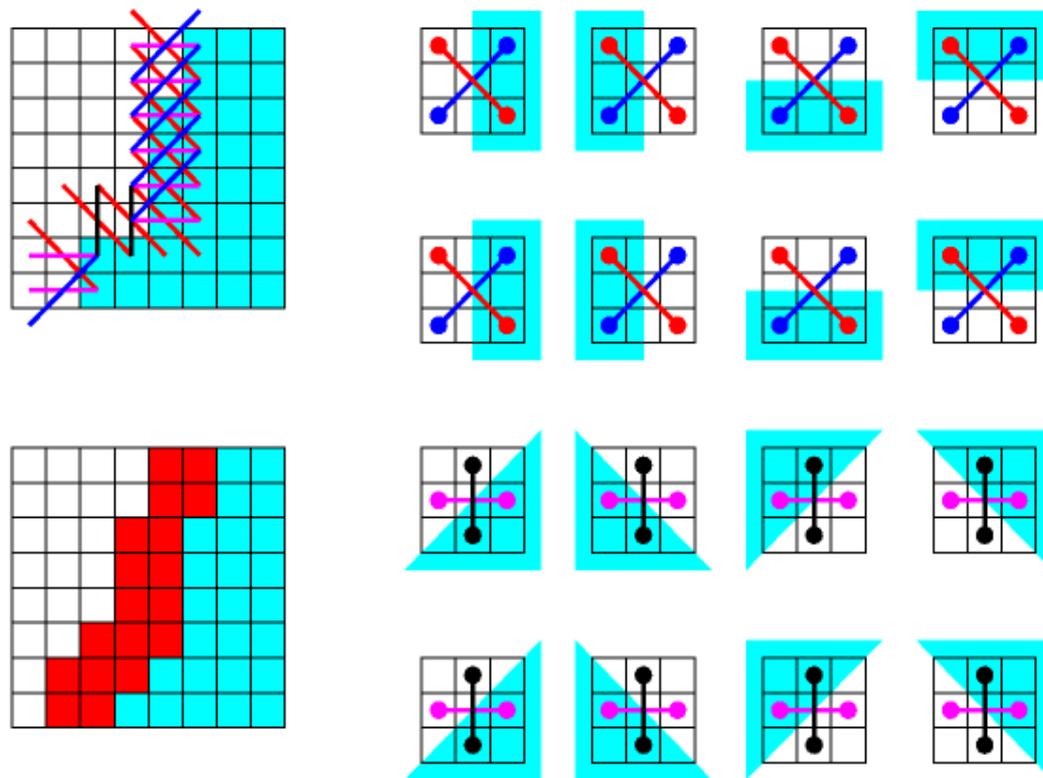


両方に輪郭があり、画素差が最小の時、一致と見なす

IV-5. 2台のカメラによる「寄り目」法 (つづき)

輪郭抽出

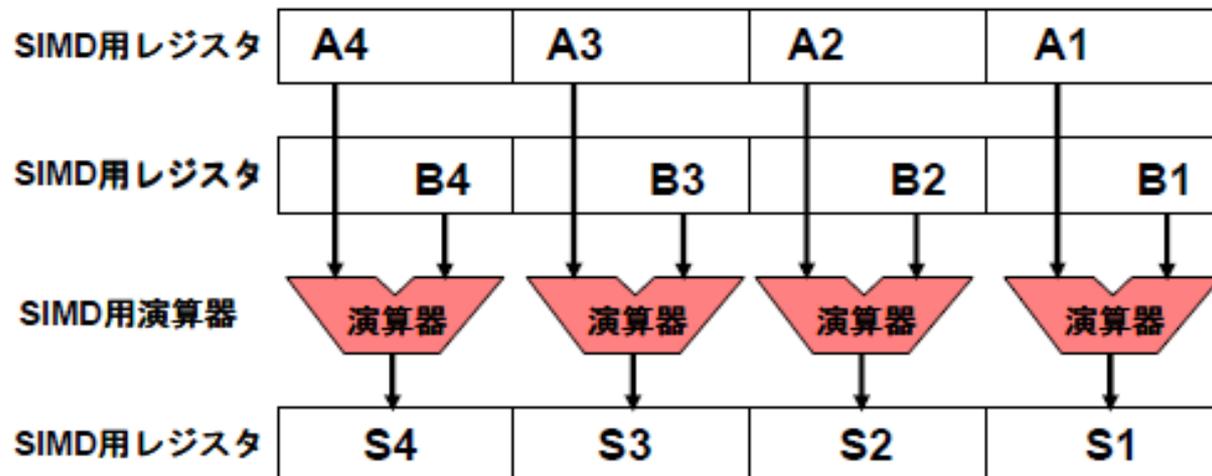
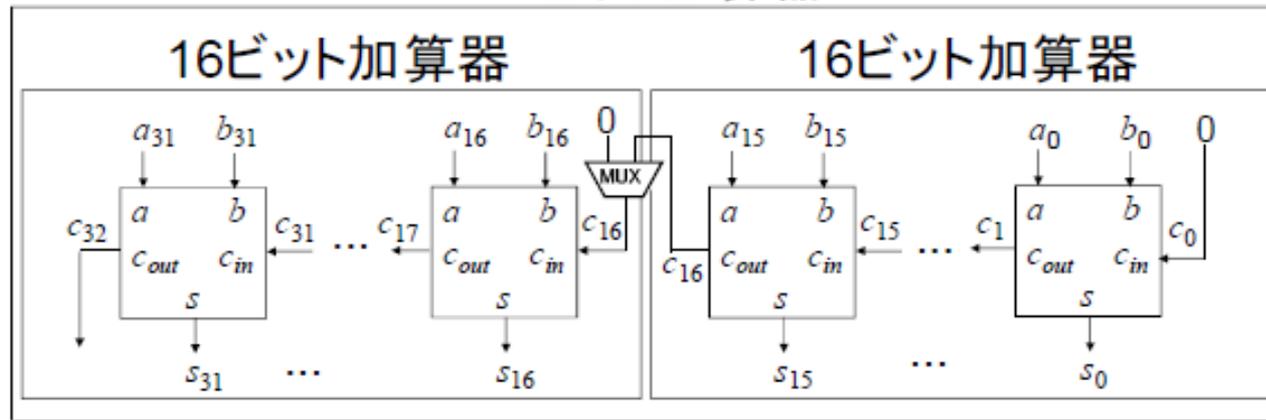
- ▶ 3×3 領域の 4 組の画素差の合計が閾値を超えた場合、輪郭があると見なす



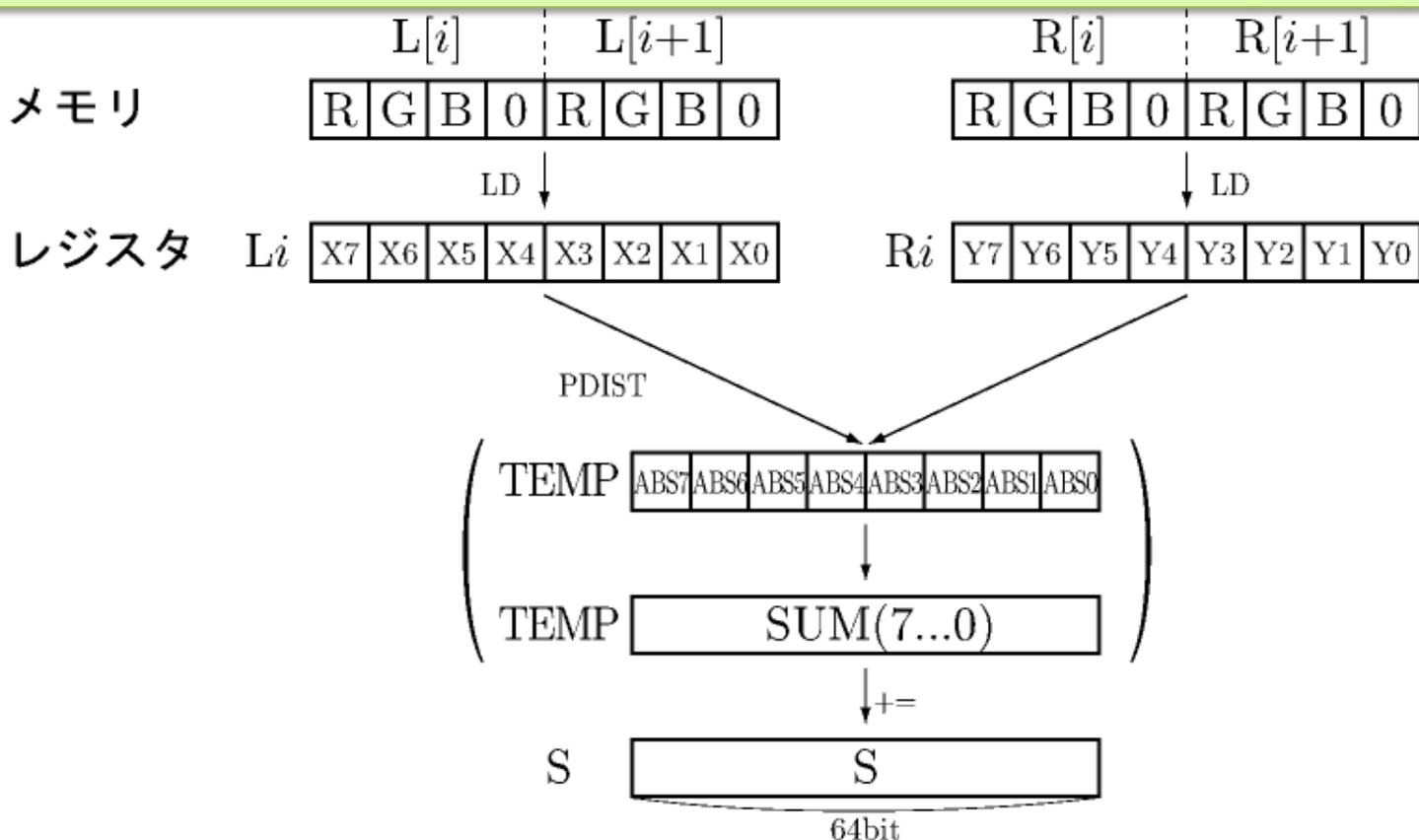
XI. マルチメディア命令による高速化

レジスタを分割使用するマルチメディア命令

32ビット加算器



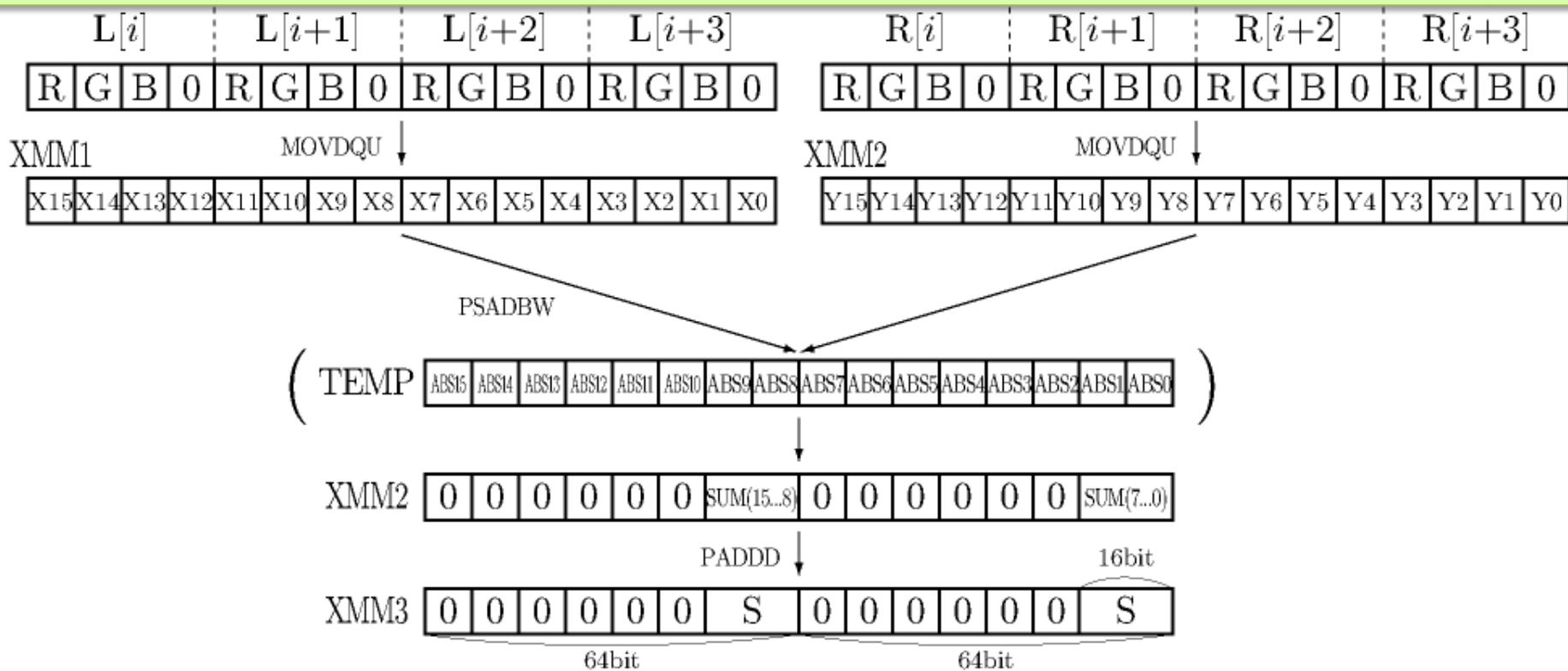
SPARC: Visual Instruction Set (VIS)



▶ PDIST命令は、絶対値総和を第3オペランドに累積

LD	$L[i], L[i+1]$	→ L_i (8バイトロード)
LD	$R[i], R[i+1]$	→ R_i (8バイトロード)
PDIST	L_i, R_i, S	→ S (8バイトSAD命令)

intel: Streaming SIMD Extensions (SSE2)



▶ PSADBW命令が、16バイトに拡張されている

```

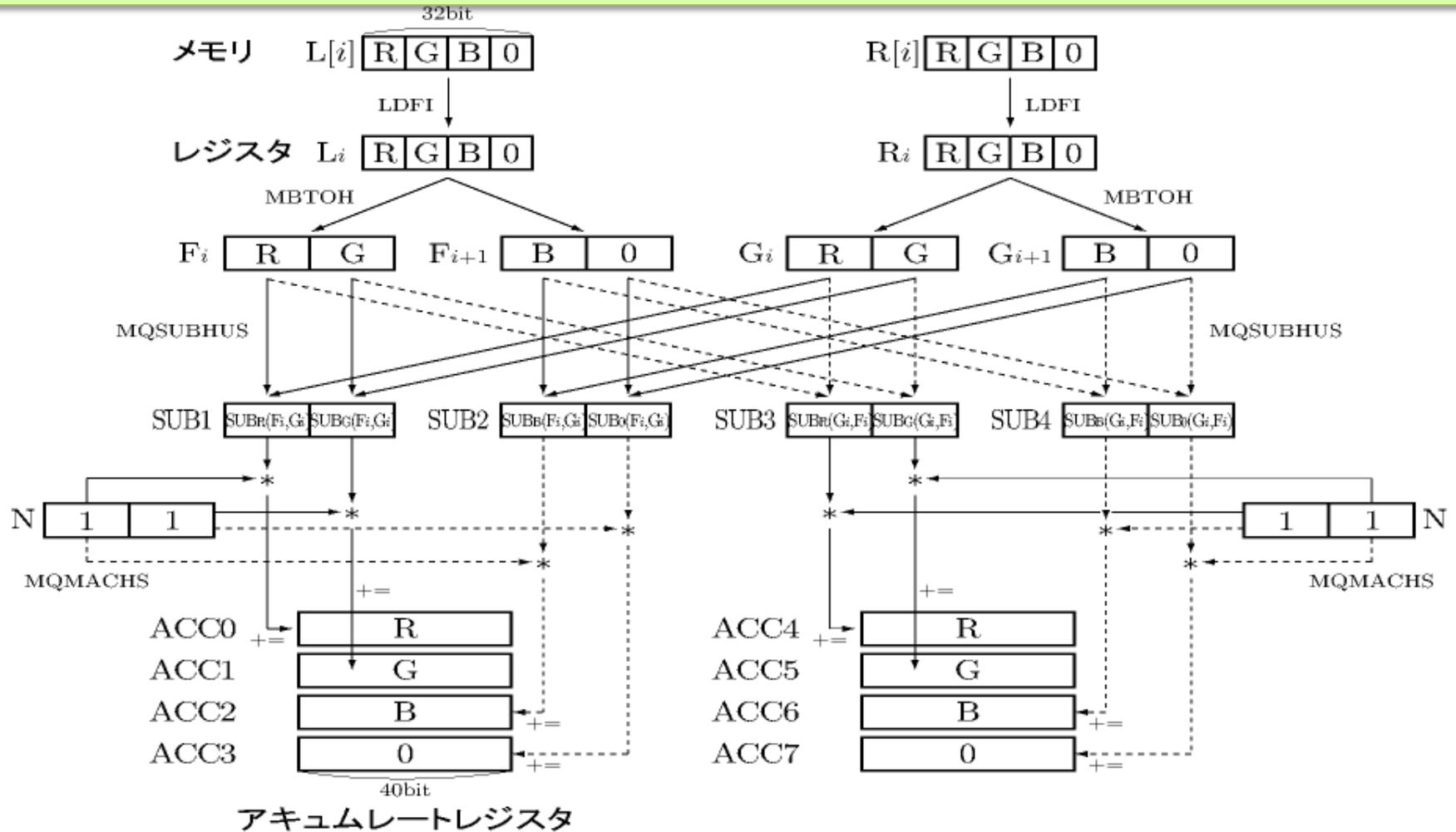
MOVDQU L[i],L[i+1],L[i+2],L[i+3] → xmm1 (16B)
MOVDQU R[i],R[i+1],R[i+2],R[i+3] → xmm2 (16B)
PSADBW xmm1, xmm2                → xmm2
PADD    xmm2, xmm3                → xmm3 (16B)
    
```

intel: Streaming SIMD Extensions (SSE2)

wdif: ウィンドウサイズ=20×20の場合

```
pushl    %ebp
movl    %esp, %ebp
movl    12(%ebp), %ecx /* *lp */
movl    16(%ebp), %edx /* *rp */
pxor    %xmm3, %xmm3
movdqu  0(%ecx), %xmm1 /* line#1 */
movdqu  0(%edx), %xmm2
psadbw  %xmm1, %xmm2
padd    %xmm2, %xmm3
movdqu  16(%ecx), %xmm1
movdqu  16(%edx), %xmm2
psadbw  %xmm1, %xmm2
padd    %xmm2, %xmm3
movdqu  32(%ecx), %xmm1
movdqu  32(%edx), %xmm2
psadbw  %xmm1, %xmm2
padd    %xmm2, %xmm3
movdqu  48(%ecx), %xmm1
movdqu  48(%edx), %xmm2
psadbw  %xmm1, %xmm2
padd    %xmm2, %xmm3
movdqu  64(%ecx), %xmm1
movdqu  64(%edx), %xmm2
psadbw  %xmm1, %xmm2
padd    %xmm2, %xmm3
addl    $1280, %ecx
addl    $1280, %edx
:
movdqu  0(%ecx), %xmm1 /* line#20 */
movdqu  0(%edx), %xmm2
psadbw  %xmm1, %xmm2
padd    %xmm2, %xmm3
movdqu  16(%ecx), %xmm1
movdqu  16(%edx), %xmm2
psadbw  %xmm1, %xmm2
padd    %xmm2, %xmm3
movdqu  32(%ecx), %xmm1
movdqu  32(%edx), %xmm2
psadbw  %xmm1, %xmm2
padd    %xmm2, %xmm3
movdqu  48(%ecx), %xmm1
movdqu  48(%edx), %xmm2
psadbw  %xmm1, %xmm2
padd    %xmm2, %xmm3
movdqu  64(%ecx), %xmm1
movdqu  64(%edx), %xmm2
psadbw  %xmm1, %xmm2
padd    %xmm2, %xmm3
movd    %xmm3, %ecx /*下位取出し*/
psrldq  $8, %xmm3 /*右8Bシフト*/
movd    %xmm3, %eax /*上位取出し*/
addl    %ecx, %eax /*合算*/
leave
ret
```

FR550: 8way Very long instruction word (VLIW)



▶ 各命令はFR500とほぼ同じ

ただし、FR500に対し単純に2倍の命令を実行できるわけではない

FR550: 8way Very long instruction word (VLIW)

ADDL	ADDR			MQSUBHUS S _i +8	MQSUBHUST T _i +8	MQMACHS S _i +1	MQMACHS T _i +1
LDFl _i	LDfR _i			MQADDHUS X	MQADDHUS Z	MQMACHS S _i +2	MQMACHS T _i +2
LDFl _i +1	LDfR _i +1			MQADDHUS Y	MQADDHUS U	MQMACHS S _i +3	MQMACHS T _i +3
LDFl _i +2	LDfR _i +2					MQMACHS S _i +4	MQMACHS T _i +4
LDFl _i +3	LDfR _i +3	MBTOHL _i	MBTOHR _i			MQMACHS X	MQMACHS Z
LDFl _i +4	LDfR _i +4	MBTOHL _i +1	MBTOHR _i +1			MQMACHS Y	MQMACHS U
LDFl _i +5	LDfR _i +5	MBTOHL _i +2	MBTOHR _i +2	MQSUBHUS S _i	MQSUBHUST T _i		
LDFl _i +6	LDfR _i +6	MBTOHL _i +3	MBTOHR _i +3	MQSUBHUS S _i +1	MQSUBHUST T _i +1		
LDFl _i +7	LDfR _i +7	MBTOHL _i +4	MBTOHR _i +4	MQSUBHUS S _i +2	MQSUBHUST T _i +2		
LDFl _i +8	LDfR _i +8	MBTOHL _i +5	MBTOHR _i +5	MQSUBHUS S _i +3	MQSUBHUST T _i +3		
		MBTOHL _i +6	MBTOHR _i +6	MQSUBHUS S _i +4	MQSUBHUST T _i +4		
		MBTOHL _i +7	MBTOHR _i +7	MQSUBHUS S _i +5	MQSUBHUST T _i +5		
		MBTOHL _i +8	MBTOHR _i +8	MQSUBHUS S _i +6	MQSUBHUST T _i +6		
				MQSUBHUS S _i +7	MQSUBHUST T _i +7	MQMACHS S _i	MQMACHS T _i
ADDL	ADDR			MQSUBHUS S _i +8	MQSUBHUST T _i +8	MQMACHS S _i +1	MQMACHS T _i +1
LDFl _i	LDfR _i			MQADDHUS X	MQADDHUS Z	MQMACHS S _i +2	MQMACHS T _i +2
LDFl _i +1	LDfR _i +1			MQADDHUS Y	MQADDHUS U	MQMACHS S _i +3	MQMACHS T _i +3
LDFl _i +2	LDfR _i +2					MQMACHS S _i +4	MQMACHS T _i +4
LDFl _i +3	LDfR _i +3	MBTOHL _i	MBTOHR _i			MQMACHS X	MQMACHS Z
LDFl _i +4	LDfR _i +4	MBTOHL _i +1	MBTOHR _i +1			MQMACHS Y	MQMACHS U
LDFl _i +5	LDfR _i +5	MBTOHL _i +2	MBTOHR _i +2	MQSUBHUS S _i	MQSUBHUST T _i		
LDFl _i +6	LDfR _i +6	MBTOHL _i +3	MBTOHR _i +3	MQSUBHUS S _i +1	MQSUBHUST T _i +1		
LDFl _i +7	LDfR _i +7	MBTOHL _i +4	MBTOHR _i +4	MQSUBHUS S _i +2	MQSUBHUST T _i +2		
LDFl _i +8	LDfR _i +8	MBTOHL _i +5	MBTOHR _i +5	MQSUBHUS S _i +3	MQSUBHUST T _i +3		

MQMACHS: quad 16bit-multiply-and-add
Moreover, 8 instructions can be executed simultaneously.