

第三章 側邊動態偵測電路分析與系統架構

3.1 概念

本論文所提之側邊動態偵測電路之原理是根據影像關聯法為基礎所設計的，其概念結構方塊圖如圖 3.1 中所顯示。在圖 3.1 中的 PH 即是感光電路，在本論文中的感光電路其輸出是電壓值，為二值訊號，當感光到有物體時則輸出高電位值或以邏輯 1 表示，反之，則輸出低電位值或以邏輯 0 表示。在 REG 與 D 皆代表暫存器，其兩者間的功用分別是：REG 是用來儲存該像素單元(pixel cell)此刻時間所感光且經由感光電路輸出的感光訊號，而 D 的功用則是用來儲存該像素單元前一刻時間的感光訊號，儲存在 D 此一暫存器的原因是為了產生延遲時間，在本論文中的側邊動態偵測電路是以暫存器來實現影像關聯法中的延遲機制。圖 3.1 中的 C 代表關聯運算子(correlator)，此處是以邏輯閘中的 AND 閘來實現關聯運算子。SR 代表著移位暫存器(shift register)。

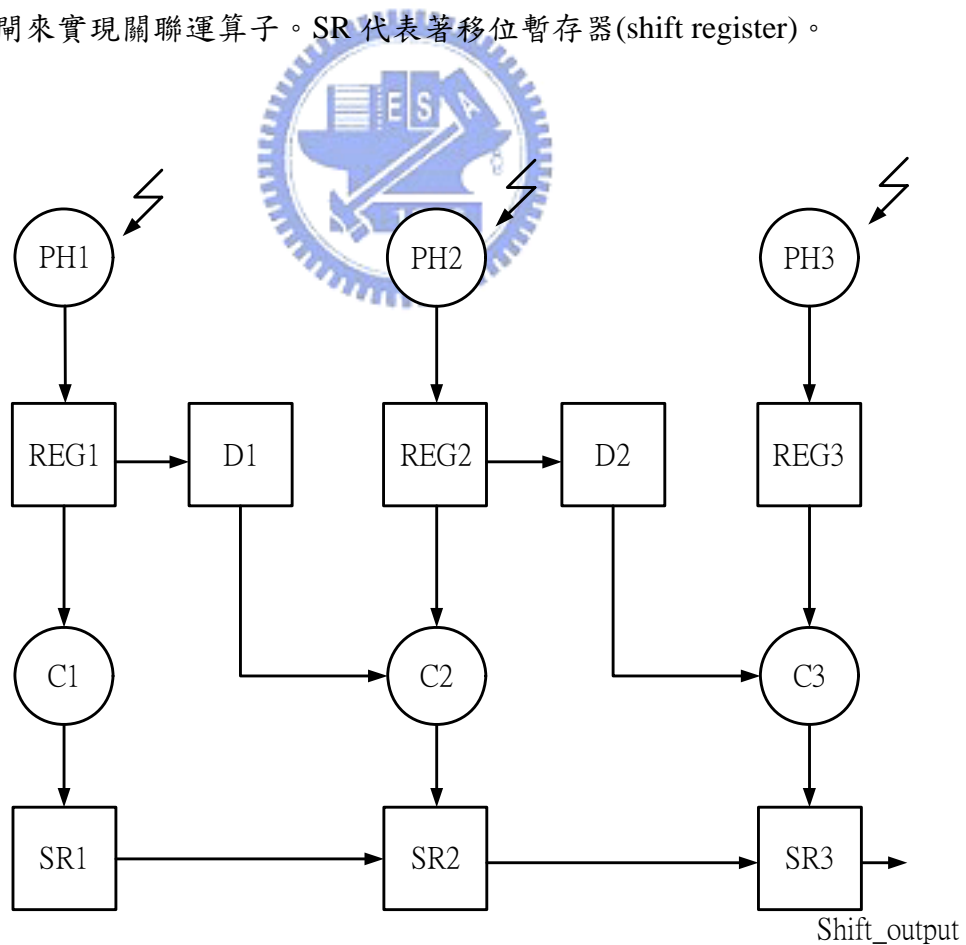


圖 3.1 概念結構方塊圖

圖 3.2 與圖 3.3 中顯示概念結構方塊圖其運作方式，圖 3.2 與圖 3.3 分別為兩個不同時間下的方塊圖，在圖 3.2 是在 T1 時間時的方塊圖，在圖 3.3 則是在 T2 時間時的方塊圖。在 T1 時間的時候，有一移動物體經過 PH2，則 PH2 感光輸出邏輯 1 並傳送至 REG2 中，如圖 3.2 所示，由於 D1、D2 與 D3 皆為邏輯 0，因此經過關聯運算子作關聯運算後所得到的輸出皆為邏輯 0。此時移位暫存器進入移位的模式進行移位，將之前所存在移位暫存器的關聯運算輸出一一移位輸出，所以我們得到 000，故可知並沒有偵測到動態物體的移動。當時間由 T1 至 T2 時，該移動物體由原本在 PH2 的位置移動至 PH3，PH3 則輸出邏輯 1 並傳送至 REG3 中，如圖 3.3 所示。由於 T1 時間時存於 REG2 中的邏輯 1 經過時間延遲於 T2 時間傳送到了 D2，因此在 C3 關聯運算子 D2 與 REG3 作關聯運算得到邏輯 1 的輸出，並傳至移位暫存器 SR3，接著移位暫存器進入移位的模式進行移位，移位後所得到的輸出為 001，故我們可以得知此電路偵測到一移動物體在 T1 至 T2 的時間內向著其關聯運算的方向(+X)移動。

由於圖 3.2 與圖 3.3 中其關聯運算的方向是向右或以+X 方向來表示，其所能偵測到移動物體的運動方向就只有右邊(+X)的方向。因此若將關聯運算的方向由原本的一個拓展成六個關聯運算的方向，即向右、向左、向右上、向右下、向左上與向左下，如此一來，其所能偵測物體的運動方向就比較多。而其中向右上、向右下、向左上與向左下等方向的資訊對於側邊運動偵測很有幫助，可藉由此四個方向的關聯運算來偵測出物體運動呈現著接近或者遠離的狀態。在本論文所設計之側邊動態偵測電路中其關聯運算的方向共有六個，分別是向左(L)、向右(R)、向左上(LU)、向右上(RU)、向左下(LD)與向右下(RD)此六個方向，其示意圖如圖 3.4 所示。而圖 3.4 中的六個關聯值可以以數學式表示之，如下所示：

$$L = \sum_{R=1}^m \sum_{C=2}^n V1_{(R,C)} \cdot V2_{(R,C-1)} \quad (3.1)$$

$$R = \sum_{R=1}^m \sum_{C=1}^{n-1} V1_{(R,C)} \cdot V2_{(R,C+1)} \quad (3.2)$$

$$LU = \sum_{R=2}^m \sum_{C=2}^n V1_{(R,C)} \cdot V2_{(R-1,C-1)} \quad (3.3)$$

$$RU = \sum_{R=2}^m \sum_{C=1}^{n-1} V1_{(R,C)} \cdot V2_{(R-1,C+1)} \quad (3.4)$$

$$LD = \sum_{R=1}^{m-1} \sum_{C=2}^n V1_{(R,C)} \cdot V2_{(R+1,C-1)} \quad (3.5)$$

$$RD = \sum_{R=1}^{m-1} \sum_{C=1}^{n-1} V1_{(R,C)} \cdot V2_{(R+1,C+1)} \quad (3.6)$$

式中的 m 與 n 分別是影像陣列的尺寸 $m \times n$ ，而 R 與 C 則分別代表著列數與行數， $V1_{(R,C)}$ 代表著前一刻之影像陣列中第 R 列第 C 行像素的感光值，是以邏輯 0 與邏輯 1 表示，而 $V2_{(R,C)}$ 則代表著下一刻之影像陣列中第 R 列第 C 行像素的感光值。

3.2 關聯運算之計算方法

圖 3.5 至圖 3.6 中顯示出移動中的物體往上述所提的六個方向運動，圖中 $T1$ 是指時間，而 $T2$ 則是指下一刻的時間。當移動的物體（圖中之黑方格）於 $T1$ 至 $T2$ 的時間內往右移動，則 R 這個輸出經過關聯運算後就會輸出邏輯 1，若物體於 $T1$ 至 $T2$ 的時間內往左下方移動，則 LD 這個輸出經過關聯運算後則會輸出邏輯 1，同理依此類推，如此一來便可以得知該移動中物體的移動方向。

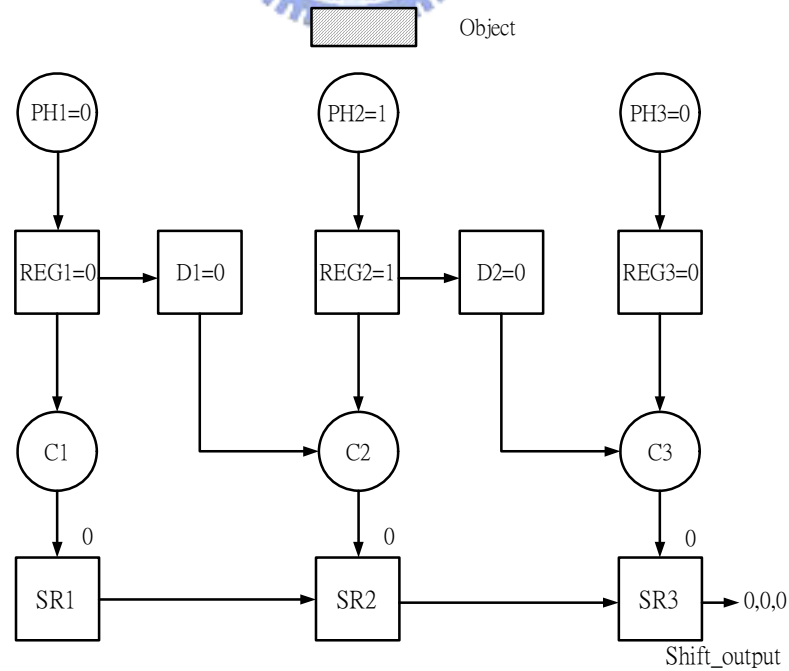


圖 3.2 $T1$ 時間的方塊圖

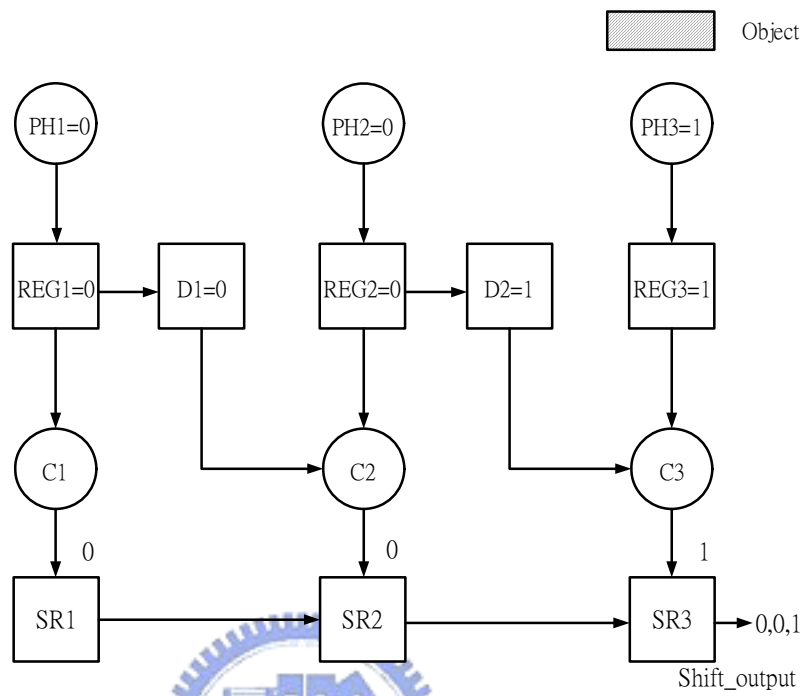


圖 3.3 T2 時間的方塊圖

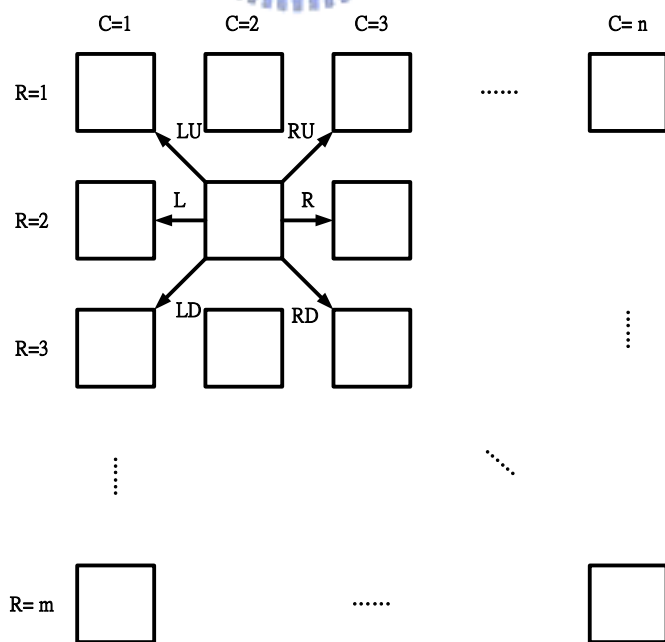


圖 3.4 關聯方向示意圖

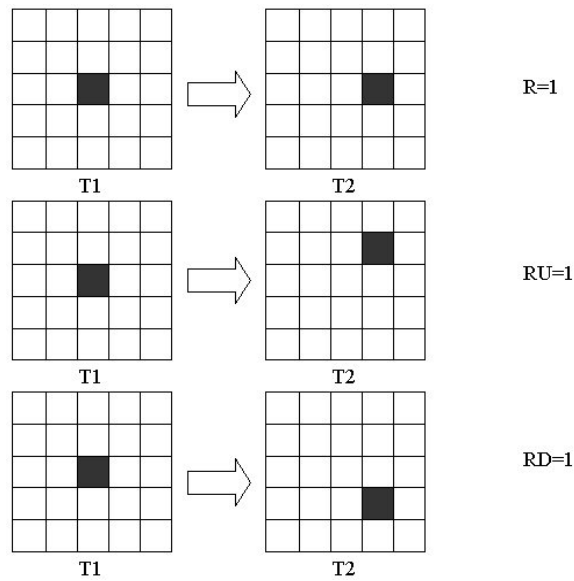


圖 3.5 影像關聯運算方式

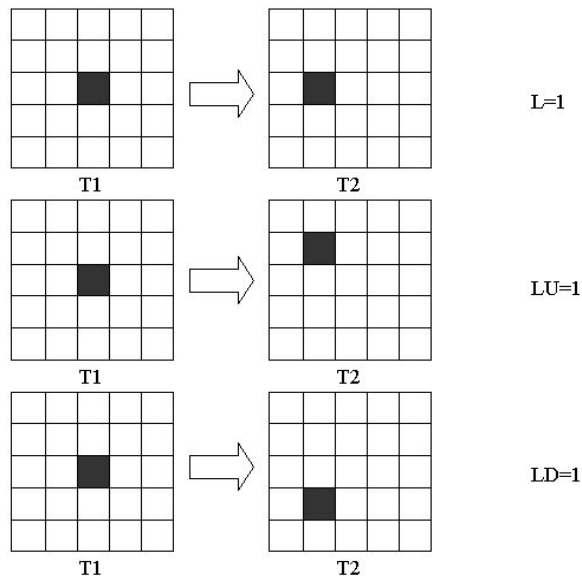


圖 3.6 影像關聯運算方式

圖 3.5 至圖 3.6 中所顯示的移動物體皆是佔一個像素單元的面積，而圖 3.7 所顯示的移動物體則是佔了四個像素單元。圖中移動物體往右邊移動了一個像素單元的距離，當物體從時間 T1 至 T2 向右移動時，於圖 3.7 T1 時刻圖中座標(2,2)的像素(黑方塊代表邏輯 1)與圖 3.7 T2 時刻圖中座標(2,3)的像素(黑方塊代表邏輯 1)作關聯運算(AND 運算)，得到了邏輯 1 的結果。同理圖 3.7 T1 時刻圖中座標(2,3)、(3,2)與(3,3)的像素分別與圖 3.7 T2 時刻圖中座標(2,4)、(3,3)與(3,4)的像素作關聯運算共得到了三個邏輯 1 的結果，因此在 R 方向的關聯運算結果為四個邏輯 1，記數為四。在 RU 方向的關聯運算共得到了兩

個邏輯 1 的結果，分別是圖 3.7 T1 時刻圖中座標(3,2)與(3,3)此二像素依序與圖 3.7 T2 時刻圖中座標(2,3)與(2,4)像素作關聯運算的結果。在 RD 方向的關聯運算得到了兩個邏輯 1 的結果，分別是圖 3.7 T1 時刻圖中座標(2,2)與(2,3)此二像素依序與圖 3.7 T2 時刻圖中座標(3,3)與(3,4)像素作關聯運算的結果。在 L 方向的關聯運算沒有得到任何邏輯 1 的結果，因為圖 3.7 T1 時刻圖中座標(2,2)、(2,3)、(3,2)與(3,3)像素依序與圖 3.7 T2 時刻圖中座標(2,1)、(2,2)、(3,1)與(3,2)像素彼此間的關聯運算皆不為邏輯 1。在 LU 方向的關聯運算沒有得到任何邏輯 1 的結果，因為圖 3.7 T1 時刻圖中座標(2,2)、(2,3)、(3,2)與(3,3)像素依序與圖 3.7 T2 時刻圖中座標(1,1)、(1,2)、(2,1)與(2,2)像素彼此間的關聯運算皆不為邏輯 1。而在在 LD 方向的關聯運算沒有得到任何邏輯 1 的結果，因為圖 3.6 T1 時刻圖中座標(2,2)、(2,3)、(3,2)與(3,3)像素依序與圖 3.7 T2 時刻圖中座標(3,1)、(3,2)、(4,1)與(4,2)像素彼此間的關聯運算皆不為邏輯 1。

經過以上的關聯運算後，得到了六個輸出值於圖中所示，由於 R 的輸出值為四，大於其他五個輸出，所以可以得知該移動物體整體的運動方向是朝右邊所移動，與實際上所運動的方向相符且一致。同理，圖 3.8 與圖 3.9 中的移動物體的整體運動方向都可藉由經過關聯運算後的輸出而得知。

3.3 動態偵測分析

本論文所設計之側邊運動偵測其原理可應用於側邊防撞，以下將會把汽車當作側邊的移

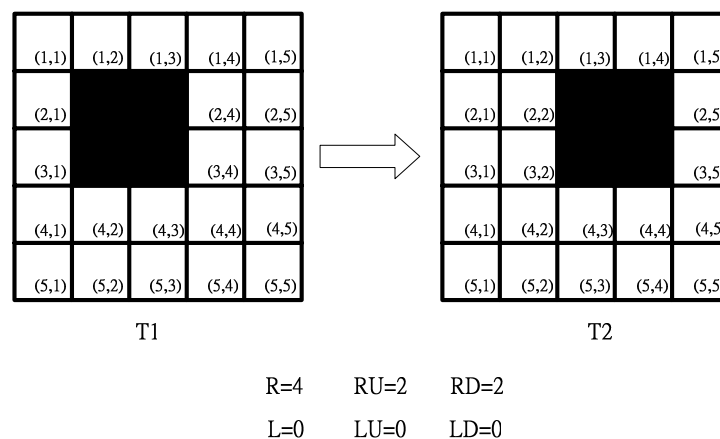


圖 3.7 向右方移動之影像關聯運算

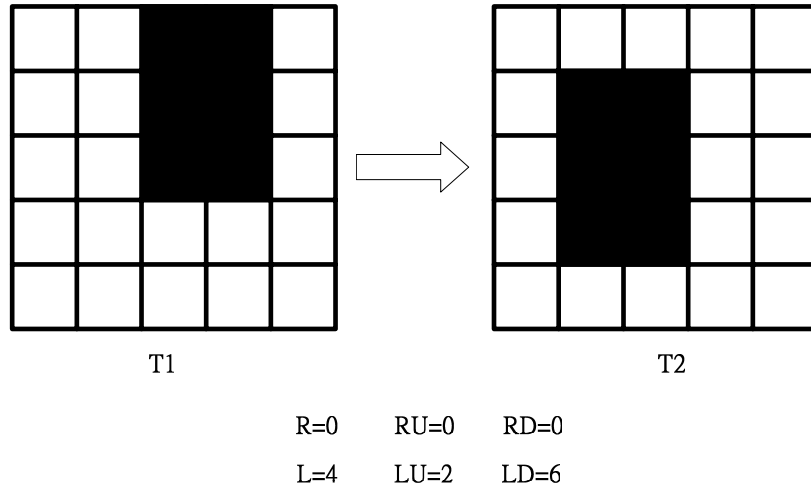


圖 3.8 向左下方移動之影像關聯運算

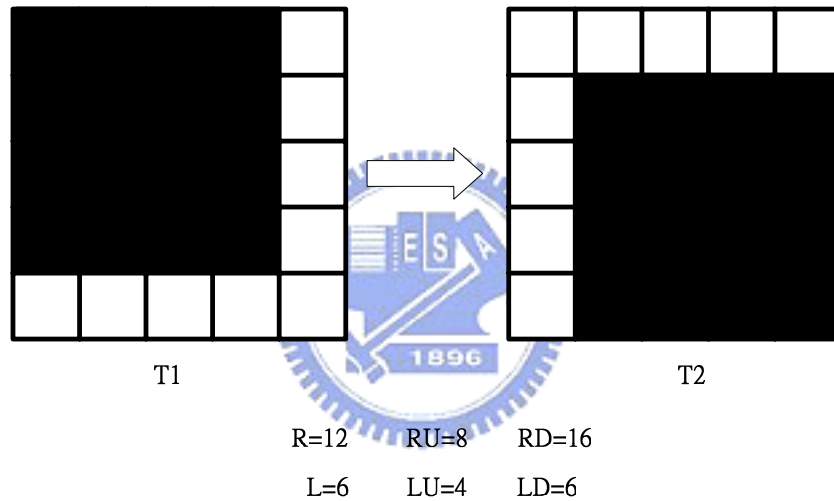


圖 3.9 向右下方移動之影像關聯運算

動物體以進行分析。一般而言，當車子在行進間時，當右邊鄰車的行車速度相對地快於左車時，會有以下三種狀況發生，分別是右邊鄰車平行超越左車、右邊鄰車越來越靠近左車以及越來越遠離左車，以上所述的三種狀況分別如圖 3.10(a)、(b)、(c)所示。在圖 3.10(a)中，左車亦即左邊的長方形方塊則是我們的座車，並在左車的右邊上架設一個 CMOS 影像感測器，用來偵測右邊的路況；而長方形方塊前方的箭頭代表的是行車速度，箭頭越大即速度越快；反之，箭頭越小則速度越慢。同理，若右邊鄰車的行車速度慢於左車時，其所發生的狀況如圖 3.11(a)、(b)、(c)所示。圖 3.10 與圖 3.11 中的六個狀況則是側邊動態物體運動的所有可能狀況，即是行車間側邊鄰車可能行駛的所有狀況。

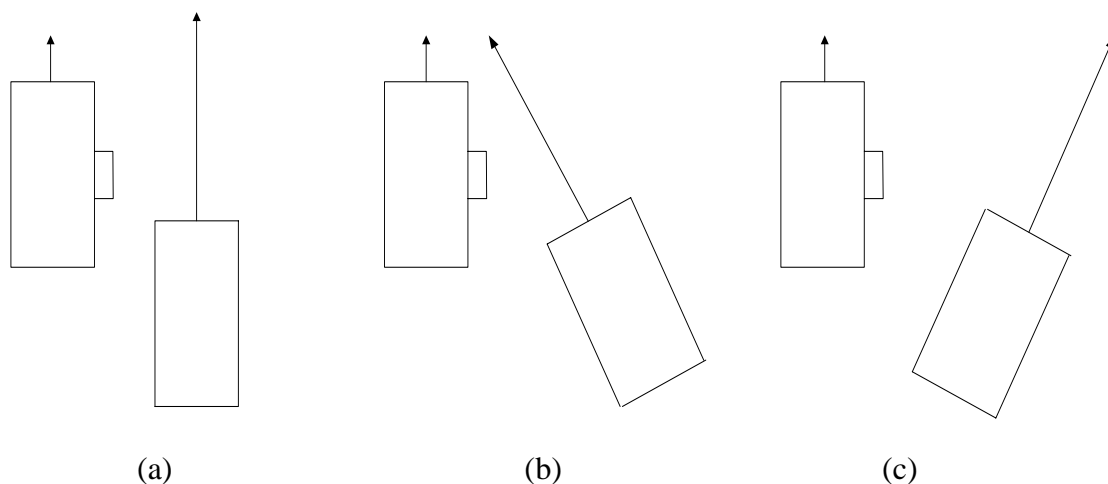


圖 3.10 車輛行進間可能狀況 (右車車速 > 左車車速)

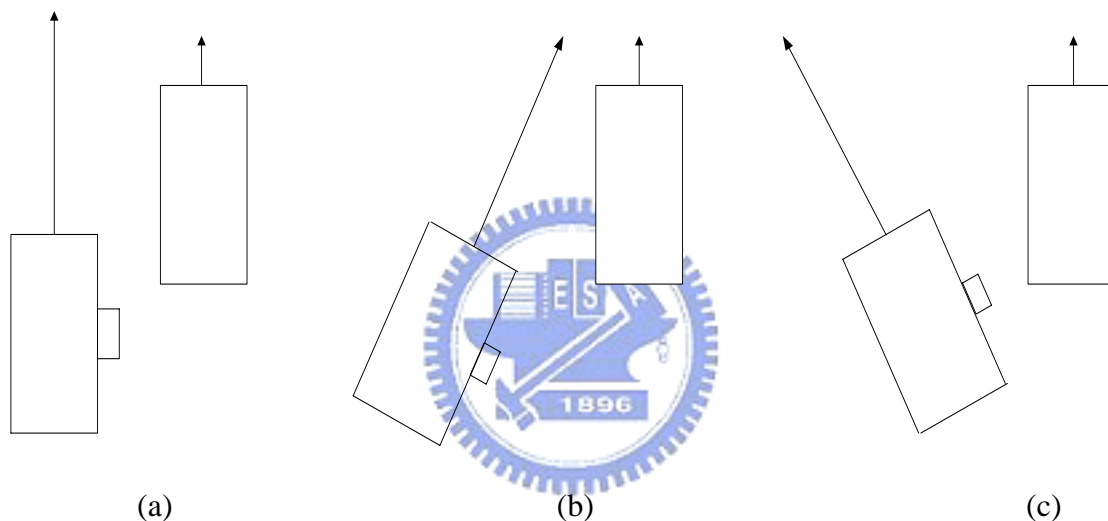


圖 3.11 車輛行進間可能狀況 (右車車速 < 左車車速)

以下將用黑白二階的圖片顯示當圖 3.10 中的狀況發生時，其 CMOS 影像感測器所擷取到相對應的影像變化。圖 3.12、圖 3.13 以及圖 3.14 分別顯示出圖 3.10 中(a)、(b)及(c)三個狀況的影像變化。而圖 3.15 圖 3.16 圖 3.17 則分別表示圖 3.11(a)、(b)及(c)三個狀況的影像變化。其 CMOS 感測器的觀察角度是如同左車中的人向窗外看出般，而圖中黑色長方形則代表右邊鄰車車體，亦即有物體移動經過 CMOS 影像感測器感光電路。

由圖 3.12 至圖 3.17 我們可以發現，移動中的物體有一定的運動方向。例如在圖 3.12 中物體整體的移動方向均是向左移動，此即代表左邊的鄰車是超越我們前去而且並沒有碰撞的危險。圖 3.8 中物體整體的移動方向均是向左下方移動，若是套用在行車時的狀況，則表示鄰車越來越向我們靠近，有碰撞發生的危險。圖 3.9 中物體整體的移動方向

均是向左上方移動，此即表示著右邊鄰車正在越來越遠離我們，此現象並無碰撞的可能性。所以由此可知，我們只要掌握其物體整體的移動方向，便可以藉由此資訊判別出鄰車與我們的關係。當鄰車對於我們有碰撞危險的可能時，系統則發出警告，提醒駕駛人避免碰撞的發生，以達行車的安全。

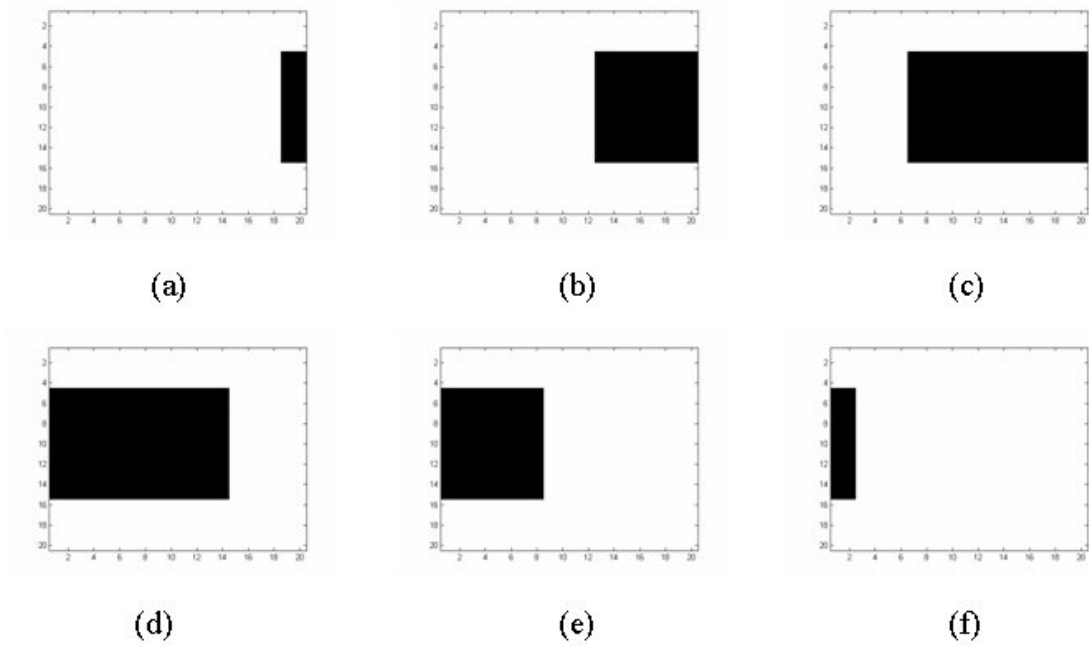


圖 3.12 向左方移動之模擬示意圖

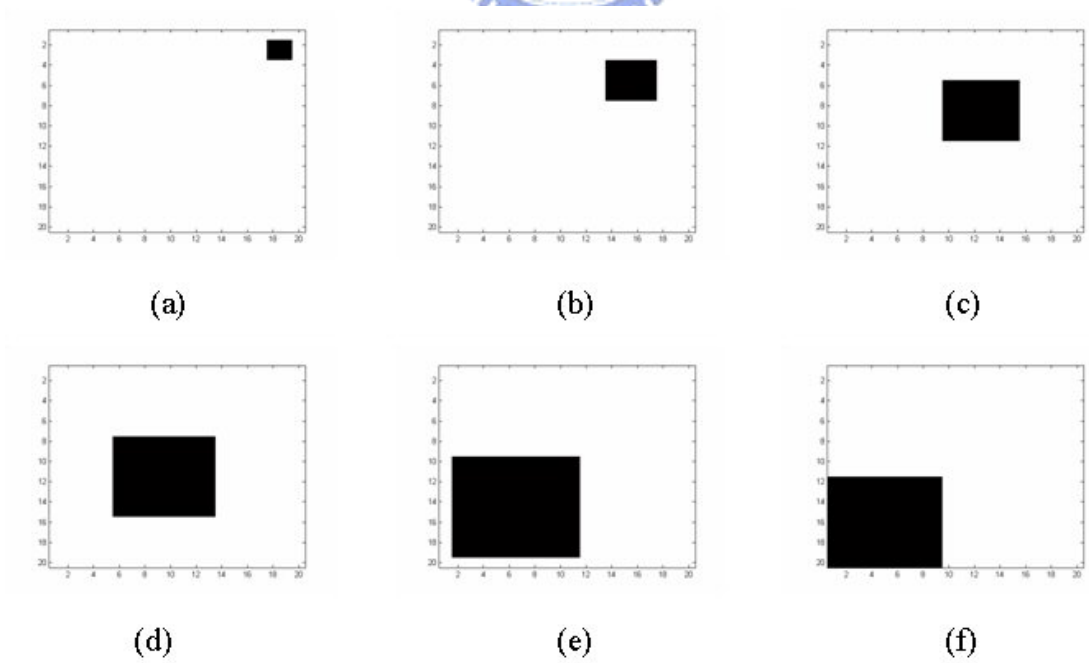


圖 3.13 向左下方移動之模擬示意圖

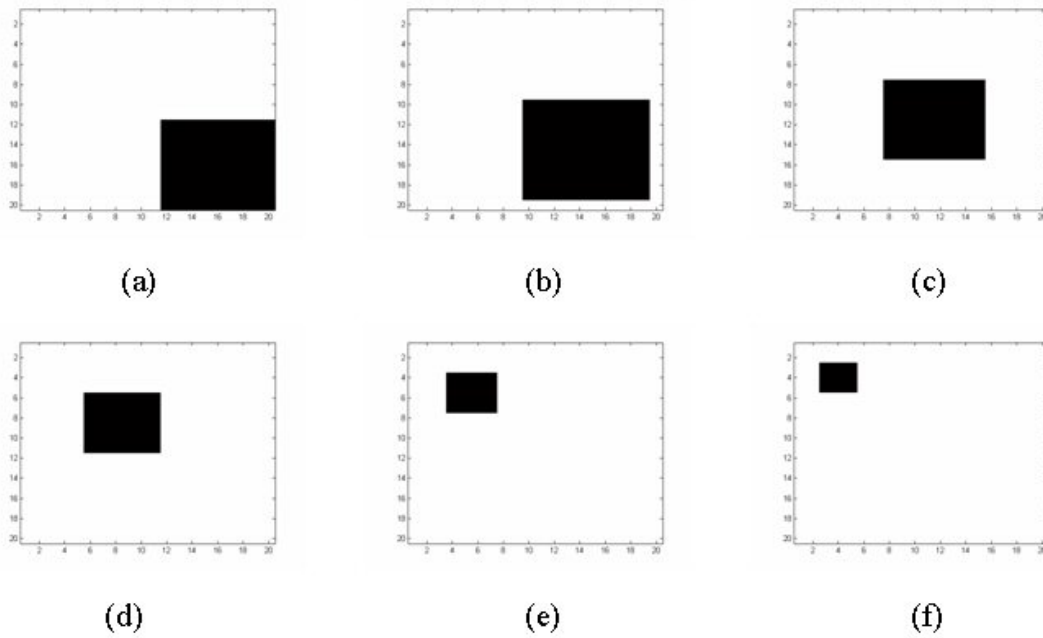


圖 3.14 向左上方移動之模擬示意圖

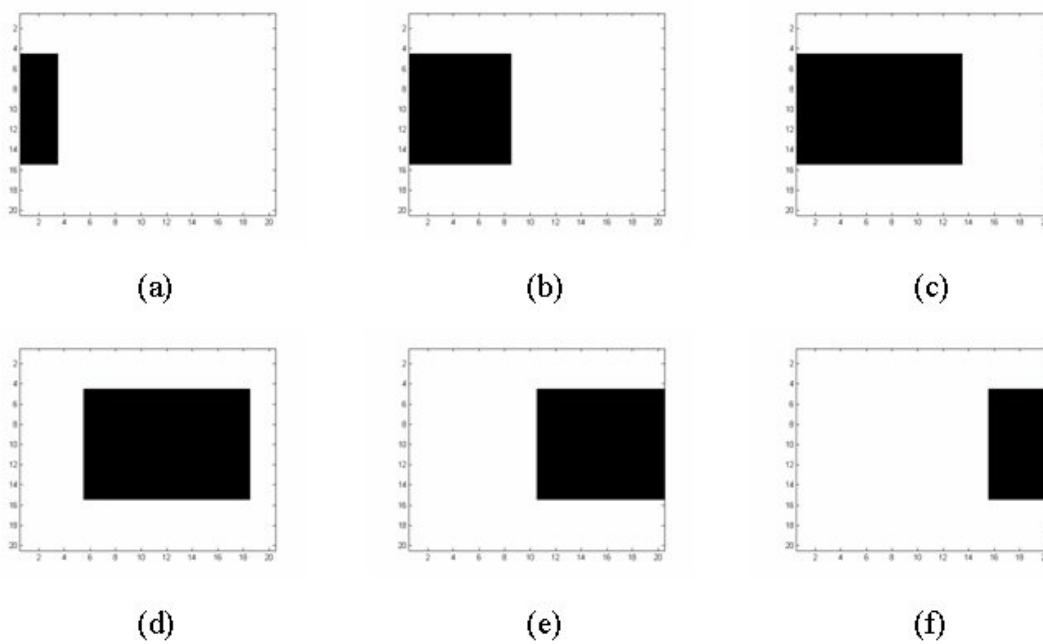


圖 3.15 向右方移動之模擬示意圖

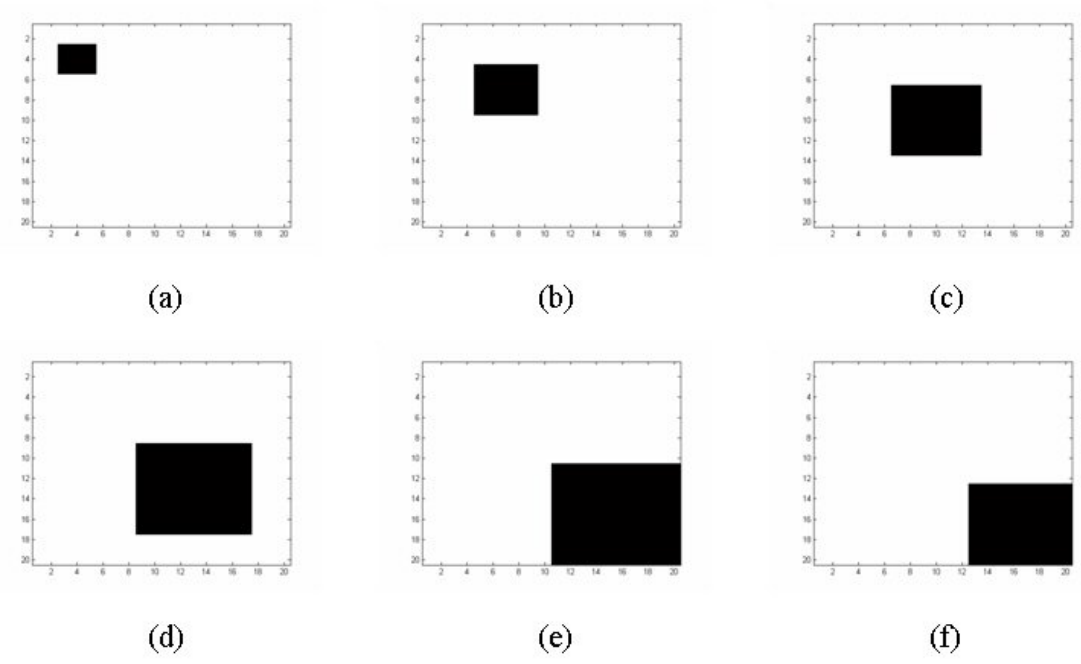


圖 3.16 向右下方移動之模擬示意圖

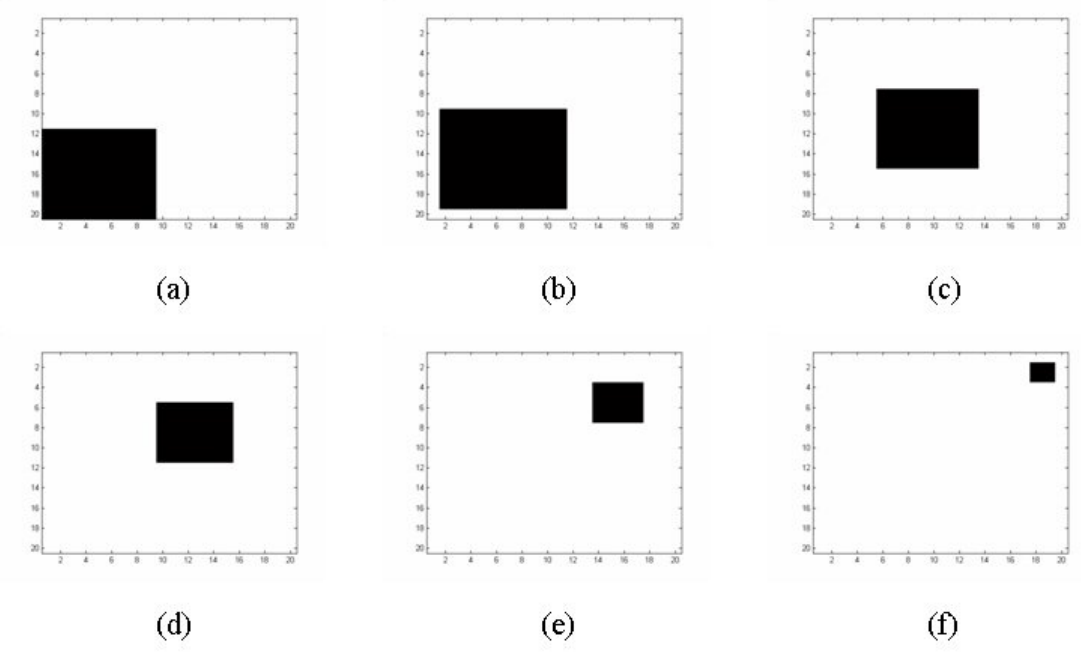


圖 3.17 向右上方移動之模擬示意圖

接著下來，將圖 3.12 至圖 3.17 共六個情況，利用 Matlab 軟體進行其功能模擬，此模擬是利用 20 X 20 個像素陣列，模擬以上所述之六個情況，並即時在陣列與陣列間作運算，其運算方式如圖 3.5 至圖 3.6 所示。表 3.1 至表 3.6 分別為圖 3.12 至圖 3.17 相對應之模擬輸出結果。

表 3.1 圖 3.12 之 Matlab 模擬輸出數據(取其中八組)

Sequence		L	R	LU	LD	RU	RD
1	(a) to (b)	22	11	20	20	10	10
2	(b) to (c)	88	77	80	80	70	70
3	(c) to (d)	99	77	90	90	70	70
4	(d) to (e)	88	77	80	80	70	70
5	(e) to (f)	22	11	20	20	10	10
6		0	0	0	0	0	0

表 3.2 圖 3.13 之 Matlab 模擬輸出數據(取其中八組)

Sequence	L	R	LU	LD	RU	RD
1	2	0	0	4	0	0
2	6	2	3	9	1	3
3	12	6	8	16	4	8
4	20	12	15	25	9	15
5	30	20	24	36	16	24
6	42	30	35	49	25	35
7	56	42	48	64	36	48
8	72	56	63	81	49	63

表 3.3 圖 3.14 之 Matlab 模擬輸出數據(取其中八組)

Sequence	L	R	LU	LD	RU	RD
1	90	72	100	80	80	64
2	72	56	81	63	63	49
3	56	42	64	48	48	36
4	42	30	49	35	35	25
5	30	20	36	24	24	16
6	20	12	25	15	15	9
7	12	6	16	8	8	4
8	6	2	9	3	3	1

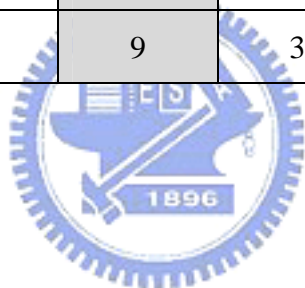


表 3.4 圖 3.15 之 Matlab 模擬輸出數據(取其中八組)

Sequence		L	R	LU	LD	RU	RD
1	(a) to (b)	22	33	20	20	30	30
2	(b) to (c)	77	88	70	70	80	80
3	(c) to (d)	77	99	70	70	90	90
4	(d) to (e)	77	99	70	70	90	90
5	(e) to (f)	44	55	40	40	50	50
6		0	0	0	0	0	0

表 3.5 圖 3.16 之 Matlab 模擬輸出數據(取其中八組)

Sequence	L	R	LU	LD	RU	RD
1	0	2	0	0	0	4
2	2	6	1	3	3	9
3	6	12	4	8	8	16
4	12	20	9	15	15	25
5	20	30	16	24	24	36
6	30	42	25	35	35	49
7	42	56	36	48	48	64
8	56	72	49	63	63	81

表 3.6 圖 3.17 之 Matlab 模擬輸出數據(取其中八組)

Sequence	L	R	LU	LD	RU	RD
1	72	90	80	64	100	80
2	56	72	63	49	81	63
3	42	56	48	36	64	48
4	30	42	35	25	49	35
5	20	30	24	16	36	24
6	12	20	15	9	25	15
7	6	12	8	4	16	8
8	2	6	3	1	9	3

表 3.1 與表 3.4 中之序列 1 分別為圖 3.12(a)至圖 3.12(b)與圖 3.15(a)至圖 3.15(b)之運算結果，而表 3.1 與表 3.4 之其他序列所對應到的圖分列於表 3.1 與表 3.4 中所示，其他表由於數據太多不似表 3.1 與表 3.4 般只有六組輸出數據，所以我們取其中連續八組數據製表。從表 3.1 可以明顯的看出在序列 1 至序列 6 六個輸出中都是 L 此一個輸出值為最大，L 代表著是左邊的意思，由此現象可以得知模擬的移動物體整體的移動方向是朝畫面上的左邊進行移動，與所進行模擬的圖 3.12 中其物體整體的移動方向是一致相符的。而表 3.2 得到的輸出數據則是顯示出 LD 此一個輸出值為最大，LD 代表著是左下方的意思，此即代表此移動物體是往畫面上的左下方進行移動。與其所進行模擬的圖 3.13 中其物體整體的移動方向是一致相符的。由於此 CMOS 影像感測器設置是架設於汽車側邊，必定使用於道路上，所以當發現其輸出顯示出側邊物體是朝左下方移動時，即可明白此側邊物體越來越接近 CMOS 影像感測器，亦即明白有發生碰撞的危險性。同理，藉由表 3.3 至表 3.6 的輸出數據就可以了解此移動中的物體其行進的方向。

由於在圖 3.13 中所進行模擬的移動方向是往畫面的左下方移動，也就是左下 45 度角的方向。但是在真實世界中，物體移動的方向不會恰為 45 度角，所以接下來我們再模擬兩個情況，分別是物體往「左下角偏左方」移動及「左下角偏下方」移動。其模擬示意圖分別如圖 3.18 及圖 3.19 中所示，在圖 3.19 中可以明顯的看出其與圖 3.13 比較起來，整個移動物體的移動方向除了是往左下方移動外還更偏左方移動，而圖 3.19 中的移動物體則是往左下偏下方移動。

在經過 Matlab 模擬後，圖 3.18 及圖 3.19 此二狀態的六個方向有關聯性的像素個數值(輸出值)分別列表於表 3.7 與表 3.8。由表 3.2、表 3.7 及表 3.8 中可以看出無論當移動物體所移動的方向是為「左下方」、「左下偏左方」或是「左下偏下方」，都可以藉由其模擬的輸出看出，亦即 LD 為最大。同理，其他三個斜角的移動方向也是可以由具關聯性的像素個數得知。因此可以得到一結論如下：在利用關聯法進行動態物體的偵測時，其相鄰像素間具六個方向關聯性之累加總和數最大者，即是所偵測的移動物體之主要移動的方向。例如，有一物體朝畫面上的左方進行移動時，則相鄰像素間具左邊方向關聯性累加之總和(L)必為最大。因此我們可以藉由觀察六個方向中具關聯性的像素個數多寡

便可得知此移動中的物體行進方向為何，以及該移動物體是否有接近及遠離的現象。

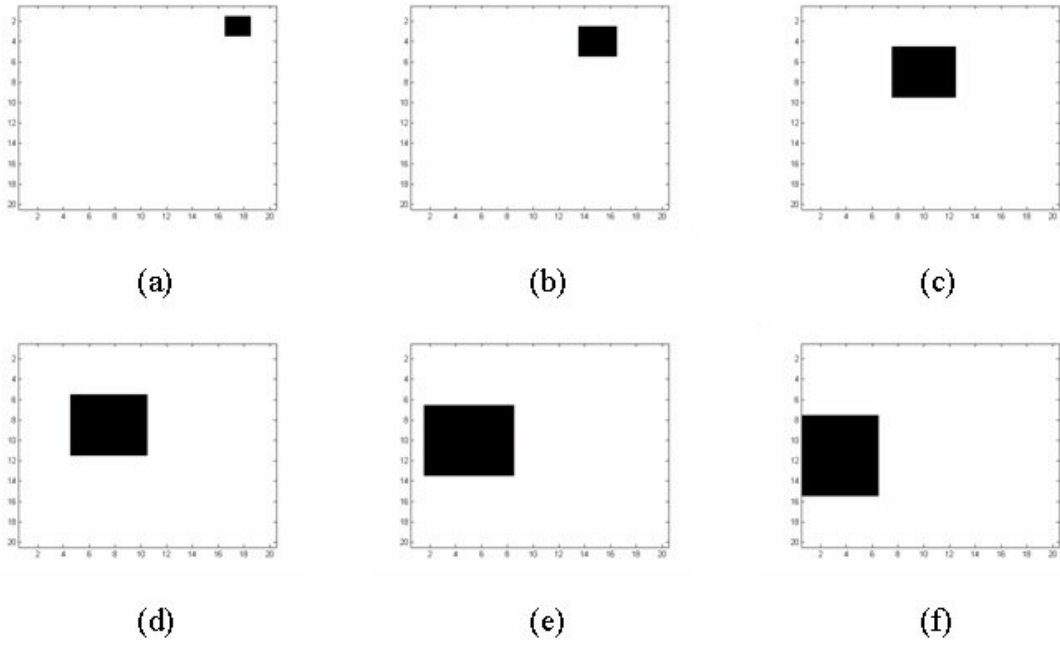


圖 3.18 向左下偏左方移動之模擬示意圖

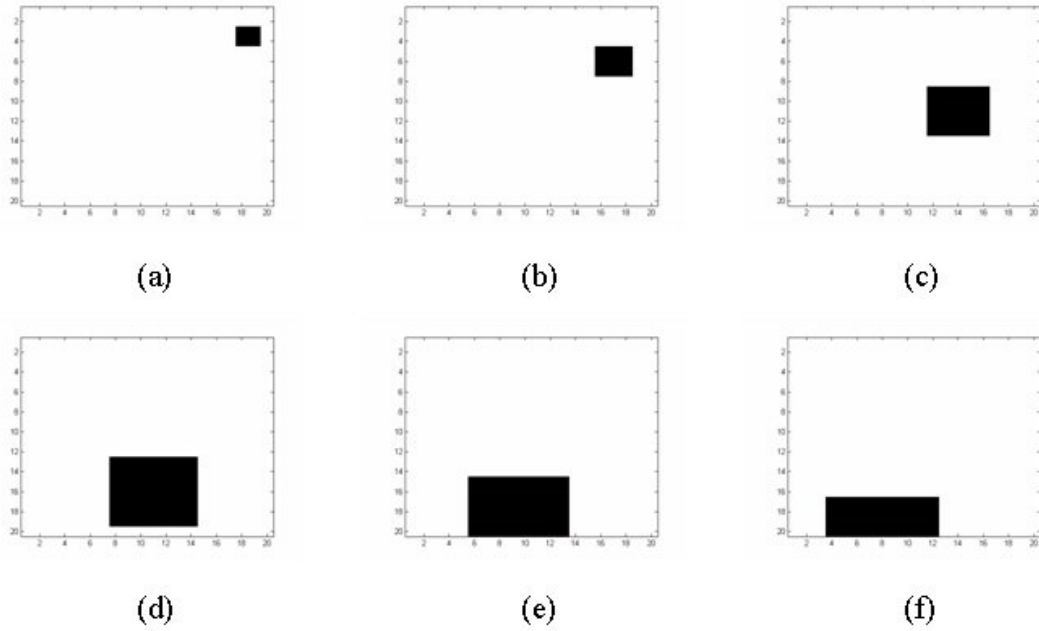


圖 3.19 向左下偏下方移動之模擬示意圖

表 3.7 圖 3.18 之 Matlab 模擬輸出數據(取其中八組)

Sequence	L	R	LU	LD	RU	RD
1	1	0	0	2	0	0
2	4	0	2	6	0	0
3	9	3	6	12	2	4
4	16	8	12	20	6	10
5	25	15	20	30	12	18
6	36	24	30	42	20	28
7	28	21	24	32	18	24
8	16	8	14	18	7	9

表 3.8 圖 3.19 之 Matlab 模擬輸出數據(取其中八組)

Sequence	L	R	LU	LD	RU	RD
1	0	0	0	2	0	0
2	3	1	0	6	0	2
3	8	4	4	12	2	6
4	15	9	10	20	6	12
5	24	16	18	30	12	20
6	35	25	28	42	20	30
7	30	24	24	32	18	24
8	15	14	9	18	7	14

3.4 系統架構

本論文所設計的側邊動態偵測系統架構其系統架構圖如圖 3.20 所示。於圖 3.20 中，CMOS 側邊動態偵測電路是由 20 X 20 的像素單元所建構的陣列像素，在工作時需要外部提供偏壓及脈波波形去控制內部的時序，此晶片的輸出共有六個輸出，其內部的陣列的安排佈局如圖 3.21 中所示。在圖 3.21 中的正方形方塊列是一個像素單元 (pixel cell)，一共是 400 個像素。圖 3.22 所顯示的是其電路內部控制之時序圖，CLK1 與 CLK2 是外加之訊號，在 CLK1 為高電位(3.3V)時感光電路進行感光，而 CLK1 由高電位轉態至低電位(0V)時，則開始作關聯運算，當 LOAD/SHIFT 為高電位時，靠著移位暫存器將關聯運算的輸出移位至輸出端。

圖 3.23 中所顯示的是單一個像素單元 (single pixel cell) 中的內部架構圖。每一個像素單元包含了一個感光電路 (photocircuit)，感光電路中的感光元件是採用 N+/PWELL 型態的感光二極體 (photodiode)。在感光後，感光電路中的感光二極體會因為感測到光的強弱而產生相對應的光電流。當感測到的光強度越強時，則感光二極體所產生的光電流則越大，反之則越小。此光電流透過感光電路後轉換成電壓值，該電壓值由於感光電路設計的緣由 (將於本論文第四章說明) 會與光電流成反比例之關係，意謂著當感光二極體所感到的光強度越強則感光電路的輸出電壓將會越小，反之則越大。並經由電壓比較器在適當的 V_b 偏壓下把感光電路所輸出的電壓分為二值(3.3V 或 0V)並存放於圖 3.23 的 Register1 中。除此之外還利用了另一個 Register2 配合著時序控制而達到圖 3.1 中 D1 運算元時間延遲的效果，接著透過 CLK1 由高電位轉態至低電位時，將此 Register1 的輸出分別送往周圍的六個鄰近像素單元的 Correlators 之輸入端。例如圖 3.23 中 Correlator1 的輸入端 L 則是由該像素其左邊的鄰近像素單元的 Register1 之輸出所輸入，同理 Correlator2 的輸入端 R 則是由其右邊的鄰近像素單元的 Register1 之輸出所輸入。當訊號傳達至 Correlator 後便要作影像關聯運算，其周圍作影像關聯運算相互間的連結狀態如圖 3.24 中所示。在圖 3.24 中所示共有 20 X 20 的陣列像素單元，圖 3.24 中的箭頭則表示著該像素中的 Register1 輸出至周圍鄰近的六個像素單元而欲作影像關聯

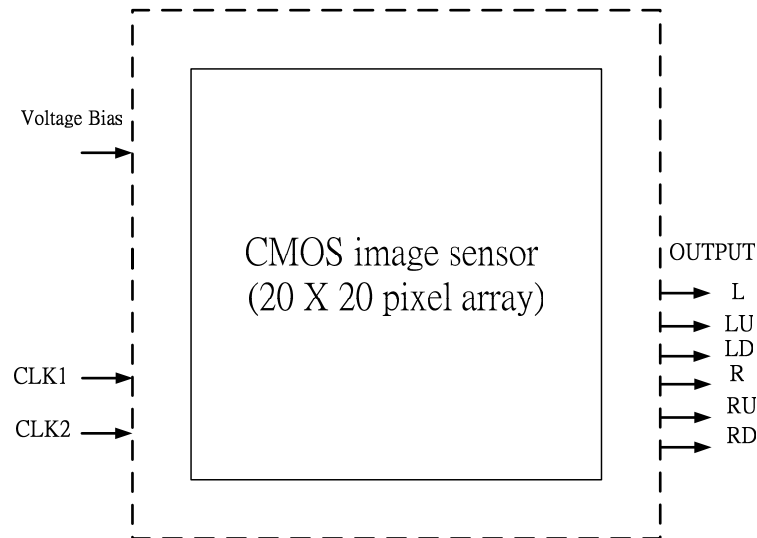


圖 3.20 系統晶片架構

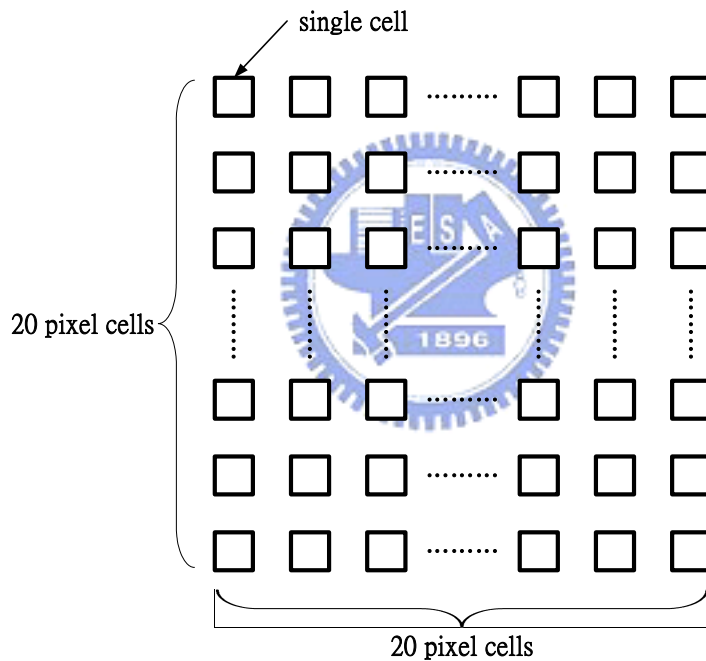


圖 3.21 Pixel array 之排列

運算的輸出方向。由於我們是作六個方向的影像關聯運算，故有六個 Correlator，此六個 Correlator 分別將該像素單元此該所感光到的電壓值與前一刻中鄰近的六個像素單元所感光到的電壓值相互作影像關聯運算。最後將經由 Correlator 作影像關聯運算後的輸出送至圖 3.23 中的 Shift Register，並配合其 LOAD/SHIFT 的時序脈波，在 SHIFT 狀態的時候以圖 3.25 中的連結路徑將其結果一一輸出。

此時經過 Shift Register 並依圖 3.25 中的連結路徑所傳出來的輸出，可以透過累加器去累加其輸出結果，當這六個輸出經過累加後的結果其值為最大者，則代表此移動物體整體的移動方向是該輸出所表示的移動方向是一致的。

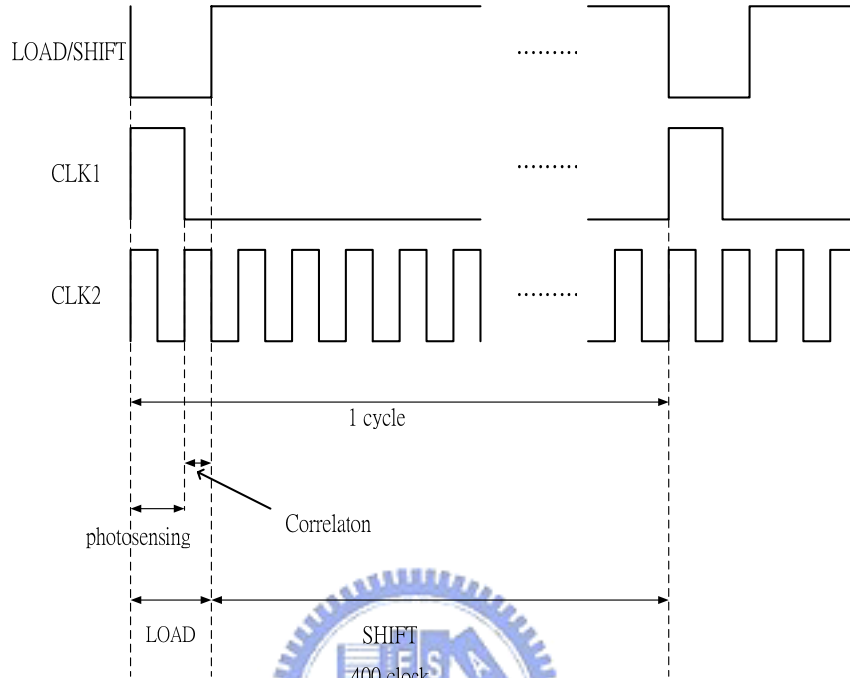


圖 3.22 時序圖

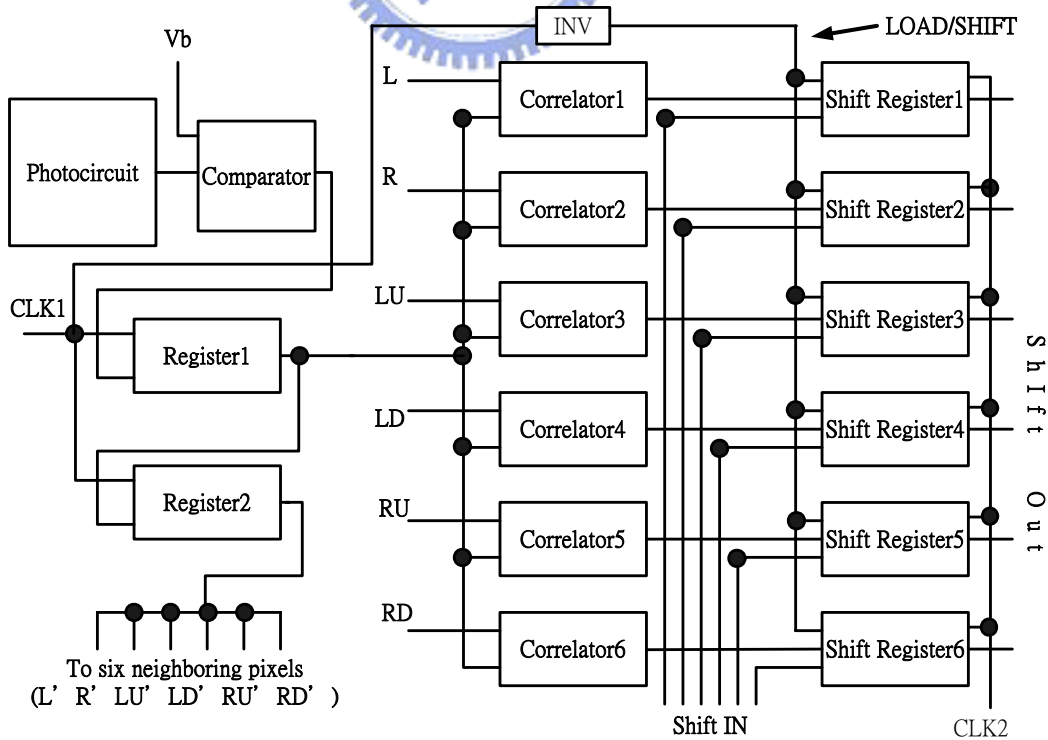


圖 3.23 Single pixel cell 之架構

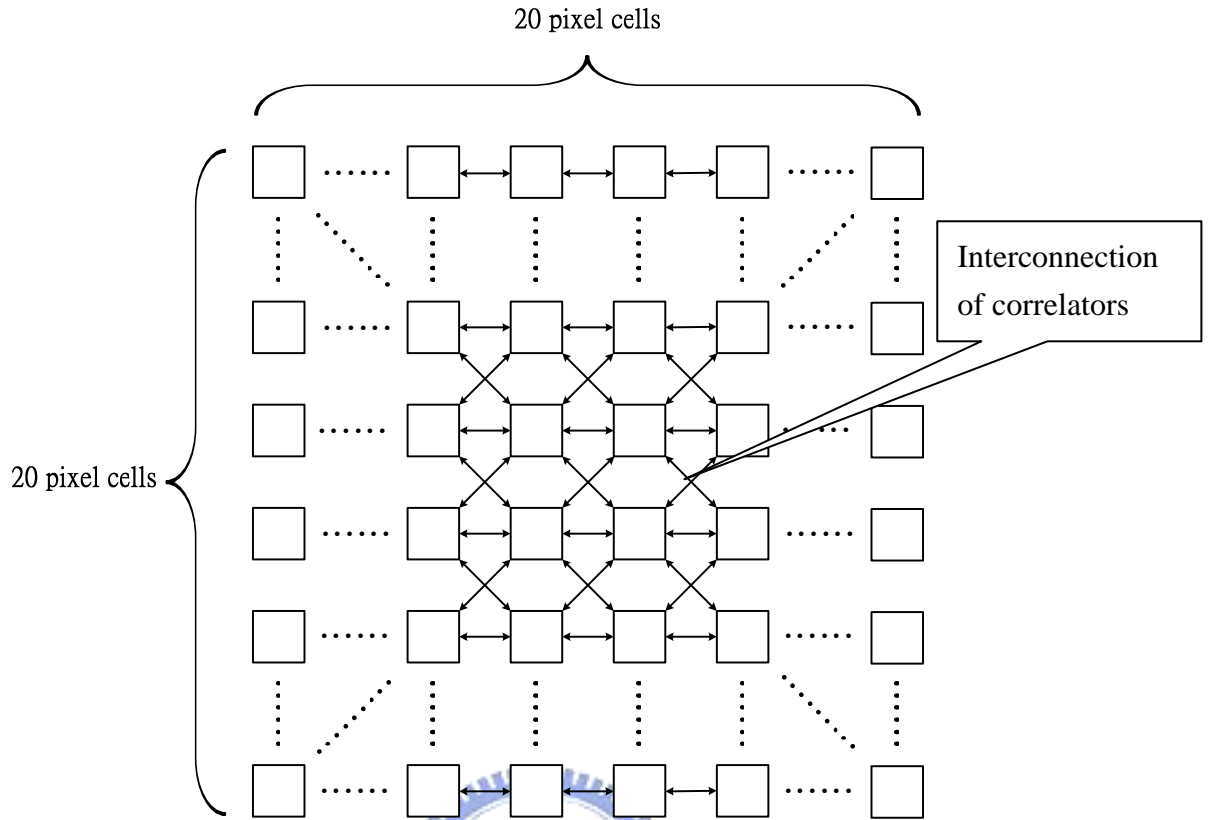


圖 3.24 陣列間關聯運算之連結圖

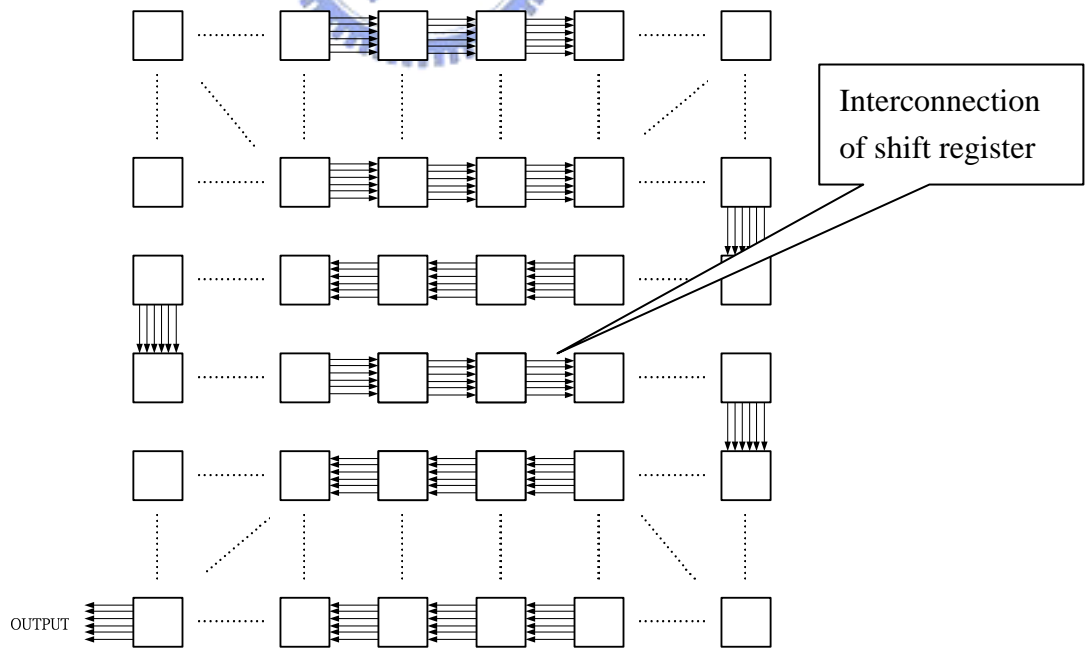


圖 3.25 陣列間 shift-in 與 shift-out 之連結圖(以 4X4 陣列為例)