

遙測影像邊緣偵測即時處理之系統晶片設計

馬清軒

國立嘉義大學資訊工程系

s0932781@mail.ncyu.edu.tw

徐超明

國立嘉義大學資訊工程系

rchs@mail.ncyu.edu.tw

摘要

本論文在設計即時 (real-time) 影像邊緣偵測之系統晶片，將其應用在需即時資料處理之衛星遙測影像系統，以達成後續之影像壓縮資料即時傳輸。系統晶片 (system on a chip, SoC) 是將含有百萬個以上的電晶體電路整合設計在同一晶片上，來形成一個完整的系統，且具有輕巧、價廉、低耗電、性能佳等優點，而且系統晶片已隨著積體電路製程技術的快速演進而逐漸醞釀成熟。本論文設計功能強大的 On-Board SoC 於人造衛星，利用 SoC 設計遙測影像即時處理之能力，讓圖像能夠直接以即時方式在衛星上處理，利用二元化與邊緣偵測以擷取有遙測影像重要特徵，並減少資料的傳輸量，進而減少傳輸的延遲，如此便可達到即時性的 On-Board 遙測影像處理。

關鍵詞：數學形態學、邊緣偵測、SoC 設計、影像處理、衛星遙測影像

一、序言

本論文為設計應用於衛星遙測影像之即時影像處理的 SoC 系統晶片。遙測影像是指用間接的方式獲得遠端目標物的狀態與影像資料，一般而言，都是從飛機或者是人造衛星對地面觀測，以取得目標訊息並加以分析，其優點在於能短時間內，收集大量資料。此項技術已廣泛用於軍事、航太、氣象、航海...等領域，代替人類前往危險或者是無法到達的地方觀測，其中尤以軍事、航太...等訊息的即時性可能造

成重大影響之領域，最廣為人所熟知。雖然目前衛星艙上處理器 (on-board processor) 速度相當快，但對於資料量大的衛星遙測影像處理與傳輸卻無法達到即時性，主要是因為傳輸所需要的延遲時間遠大於處理時間。所以雖然地面站的電腦運算速度快，整體效能仍依賴於人造衛星與地面站電腦的近即時 (near real-time) 傳輸。因此，本論文提出一個系統晶片 (SoC) 解決辦法，可以透過裝置功能強大的 On-Board SoC 於人造衛星，提供人造衛星遙測影像即時處理之能力，讓圖像能夠直接以即時 (real-time) 方式在衛星進行處理，以擷取有用之影像資料特徵，並減少資料的傳輸量，進而減少傳輸的延遲，如此便可達到即時性，而且由於影像已於外太空處理完畢，地面上的接收站裝置不用再自己進行運算，故可以簡化接收裝置，降低耗電量，並提高接收設備攜帶性與遙測資料可分析性與利用度。

二、文獻探討

影像分割 (Image Segmentation) 技術的應用廣泛，例如：遠端遙測、視訊編碼...等，等等，早已為人所熟知，然而依據各種影像應用的特性，就有其不同的需求，例如：數位視訊，所使用的影像分割技術必須能夠即時處理；遠端遙測，就必須同時滿足精確性及即時性。因此，分割方法的複雜度是一個相當重要的參考依據，因為被廣泛使用的工具皆依賴於其運算的效率 [9]。影像分割的目的在於抽取出影像中的物體或者是結構，將現實的立體物件投影到二維平面 [2] 以取得影像之重要特徵。概括而言，影像分割的方法主要分為

兩種類型[6,12]，區域導向 (Region-based) 跟輪廓導向 (Contour-based)，區域導向的方法優勢在於處理含有雜訊的圖形，但此方法需要較高的實作成本；輪廓導向的方法，雖然運算成本低，但處理結果對原影像品質較敏感。

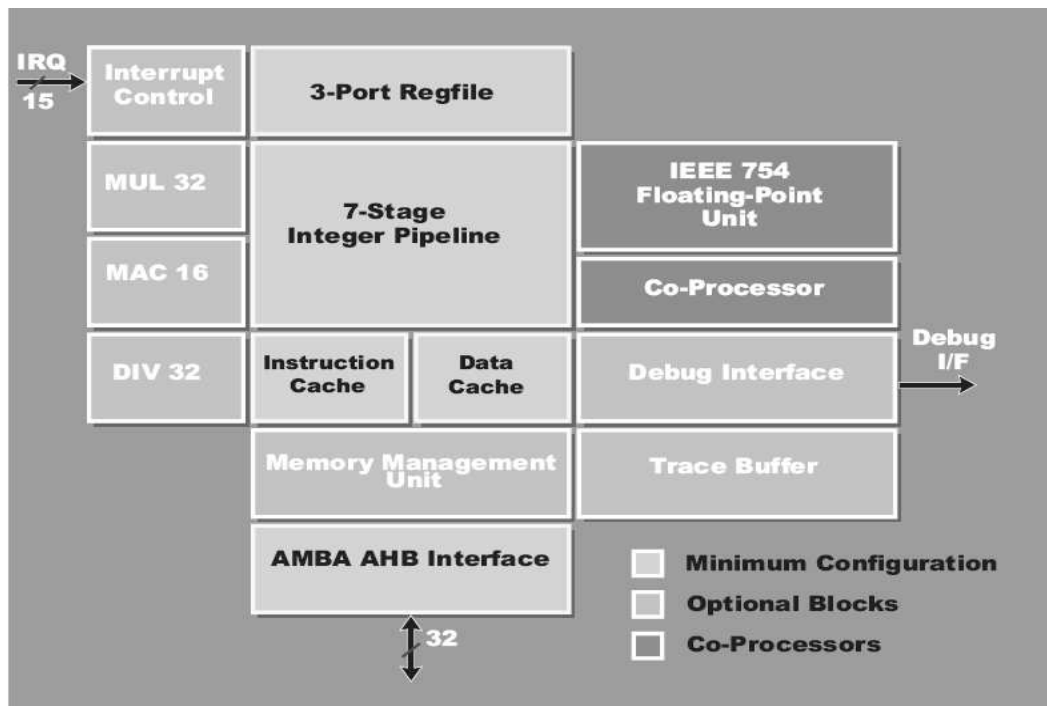
因為大量的影像資料及複雜的影像處理運算，尤其是很多的影像處理之應用，需要每個像素都進行運算，導致每秒需要大量的運算，因此即時影像處理不容易以循序處理器 (serial processor) 達到。有一個解決辦法就是使用 FPGA，FPGA 在容量及功能上持續地成長，而且具有對特定演算法開發之低成本及可重複設定之特性 [3]，所以非常適合應用在影像處理，特別是即時影像處理之用途 [10]。

本論文在於對遙測影像進行運用影像分割方法之 SoC 實現，首先將影像 I 進行二值化方法 [2] 運算，得到二元影像 B，再經由數學型態學 (Mathematical Morphology) 的處理，抽取出物件的邊緣。本論文將此演算法以 SoC 平台實現，並透過 FPGA 最佳化方法 [4] 達到即時處理之特性。

三、SoC 系統架構與演算法

SoC 系統架構

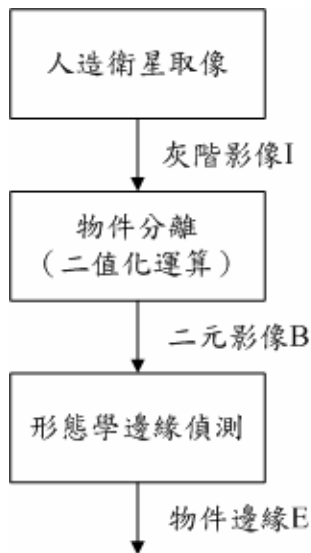
一個典型的 SoC 包含處理器、記憶體及其週邊設備，而 IP core 以標準的匯流排架構傳送、接收資料 [8]。因此可以透過開發 IP core，加入到 SoC 系統晶片，達到特殊應用之目的。以 VHDL 撰寫的 LEON3 是本論文所使用的 Processor IP Core，LEON3 是一個歐盟太空總屬 (European Space Agency) 設計開發的 32-bit Processor IP 核心。如圖一為 LEON3 系統架構，它是 Open source IP core，與 SPARC V8 架構相容，符合 IEEE-1754 架構；擁有 Harvard 架構的 7-stage pipeline，分開的指令、資料快取；支援多重處理器及 on-chip debugger，且修改容易，具有容錯 (fault-tolerance) 功能；IP 之間溝通的部份，使用 AMBA 匯流排，具備 plug&play 的 AHB arbiter/multiplexer、AHB/APB Bridge [13]。根據以上的特點，顯示 LEON3 Processor 很適合用於衛星遙測影像處理之 SoC 的開發與應用。



圖一、LEON3 系統架構區塊圖

影像處理演算法

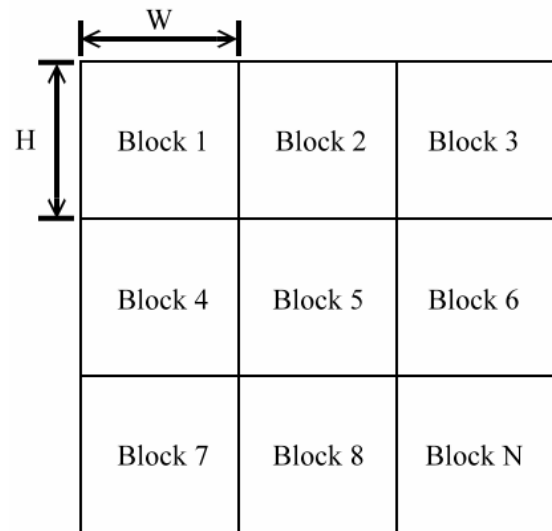
本論文之影像分割的 SoC 處理流程圖如圖二，首先利用『強健適應性限度方法』(Robust adaptive thresholding)[2]將影像 I 進行二值化方法運算，得到二元影像 B，再經由數學型態學 (Mathematical Morphology) 的處理，抽取出物件的邊緣 (boundary extraction)。就資料量來分析，灰階影像 I 遠大於二元影像 B，而二元影像 B 會大於物件物件 E，所以本文件所提出的解決辦法，是透過減少需傳輸的資料量來降低傳輸的延遲時間，進而達到資料處理的即時性。



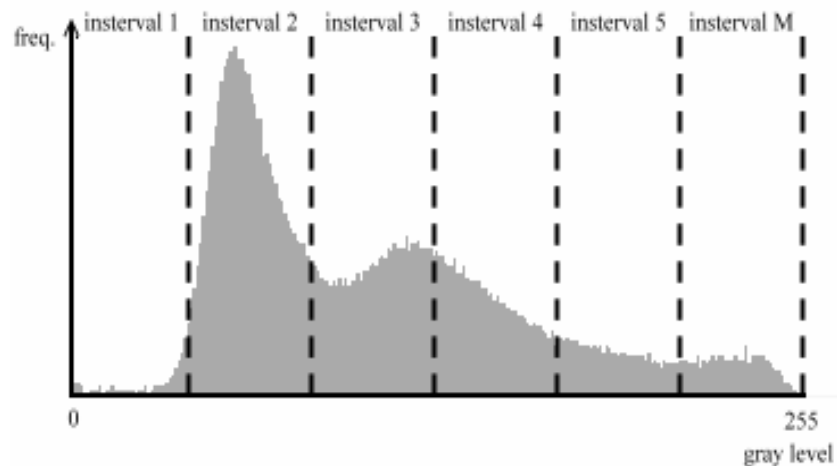
圖二、遠端遙測影像之流程圖

(一) 強健適應性限度方法

傳統的影響二元化[1,4,7,11]一般採取限度 (thresholding) 的方法，首先將影像分割為 N 個相等區塊(如圖三)，計算出各個區塊的直方圖 (histogram)，接著將直方圖分成 M 個部份(如圖四)，決定各個部份裡頻率最高的灰階值 $G_{p_i}, i \in (1...M)$ ，並算出各 N 個區塊的直方圖之灰階平均值 μ_B 。



圖三、影像區塊圖(blocks)



圖四、影像灰階分佈圖

最後，使用所有區塊各自的 μ_B 跟 $G_{Pi}, i \in (1 \dots M)$ ，利用函式(Eq.1)算出平均值，得到閾值 T ，再經由函式(Eq.3)的運算，運算出二元影像 B 。

$$T = \frac{\sum_{k=1}^N \left(\sum_{i=1}^M (G_{Pi}) + \mu_{Bk} \right)}{N \cdot M + N} \quad (1)$$

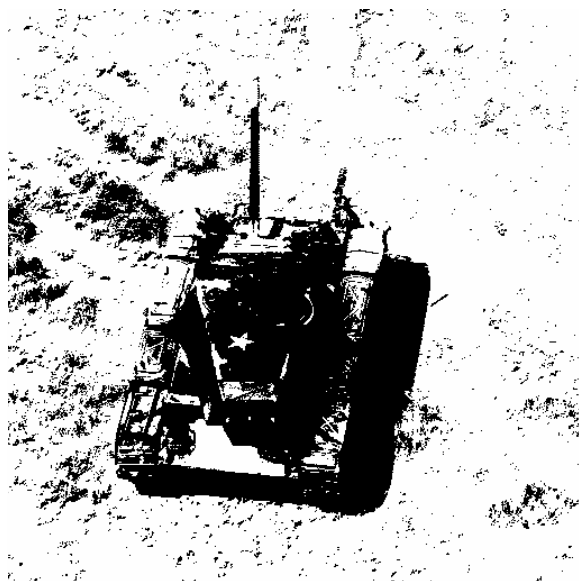
$$\mu_{Bk} = \frac{\sum_{x=1}^W \left(\sum_{y=1}^H I(x, y) \right)}{W \cdot H} \quad (2)$$

$$B(x, y) = \begin{cases} 1: I(x, y) > T \\ 0: I(x, y) < T \end{cases} \quad (3)$$

然而傳統的 thresholding 的方法 [1,4,7,11]，無法辨識亮點集合代表的是物件或者是背景，例如有一張坦克遙測影像(如圖五)，其背景為草原，物件為坦克，但使用一般二值化方法，背景被分類成亮點集合，物件分類成暗點集合(如圖六)，可能會造成之後使用的形態學運算造成誤判。因此，可以使用一個簡單的偵測方式[2]，統一讓亮點的集合代表物件。



圖五、坦克遙測影像



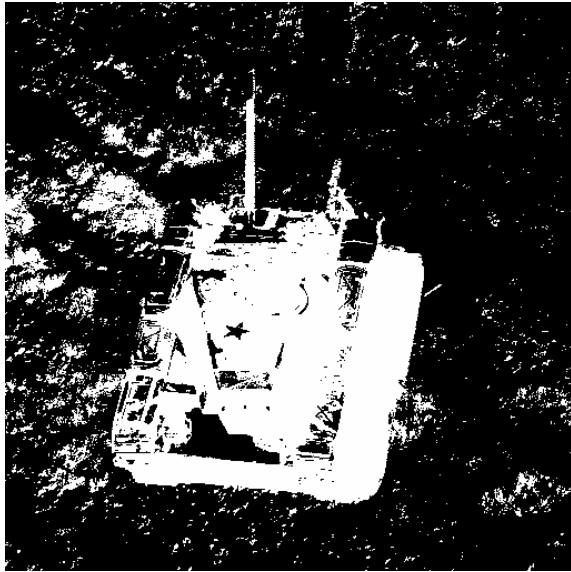
圖六、一般二值化結果

首先必須計算二值化後的亮點集合 Br 的元素個數及暗點集合 Dr 的元素個數，因為可以由大部份的影像觀察得知，背景在影像中所佔據了絕大部分的區域，所以可透過利用此觀察結果，比較集合裡元素的個數，決定集合所要代表的明亮度，因此可以將函式(Eq.3)改寫成函式(Eq.4)。

$$\begin{aligned} &\text{if NumOf}[Br] > \text{NumOf}[Dr] \\ &B(x, y) = \begin{cases} 1: I(x, y) > T \\ 0: I(x, y) < T \end{cases} \end{aligned} \quad (4)$$

$$\begin{aligned} &\text{if NumOf}[Dr] > \text{NumOf}[Br] \\ &B(x, y) = \begin{cases} 0: I(x, y) > T \\ 1: I(x, y) < T \end{cases} \end{aligned}$$

加入此偵測方法，可以將二值化影像導正回來，使得背景統一為暗點集合，物件為亮點集合(如圖七)，讓之後使用的形態學運算能夠正確判斷，提高運算出來之影像的正確率。



圖七、導正後的二元影像

(二) Edge Detection

對於二元影像之處理，數學形態學 (Mathematical Morphology) 提供了一個簡單而有效的方法[2]，其使用邊緣偵測函式則如 (Eq.5)

$$\begin{aligned} E_e &= B - \varepsilon[B, SE], \\ E_d &= B - D[B, SE] \end{aligned} \quad (5)$$

其中 B 代表二元影像，E 代表邊緣影像， ε 代表 Erosion 運算子，D 代表 Dilation 運算子，而 SE 代表 Structuring element。一般的形態學 Erosion 跟 Dilation 使用的 SE 為 3x3 矩陣，由 SE 定義要參與型態學運算的 pixel，而常用的 SE 為 3x3 十字矩陣 (如圖八)。每次的 Erosion 或者是 Dilation 會設定 $\varepsilon[B, SE](X_5)$ 為黑點或者是白點 (位置標示如圖九)，也就是說，每個 3x3 的區塊皆可以算出其中間點的值。

(三) FPGA 最佳化

根據數學形態學的理论，本論文設計所使用的基本處理 module，結合此 module 與型態學的運算元[4]，進而提昇 FPGA 運算效能。傳統處理方法之 FPGA 區塊如圖

十，傳統的處理方法是移動 SE window，使其對齊各個 pixel，取代掉 window 裡的所有 pixel，因此每次處理都必須從記憶體讀取 window 大小的資料，可見得此方法不適合用於需即時處理之應用，因為需要浪費相當多的 clock cycle。

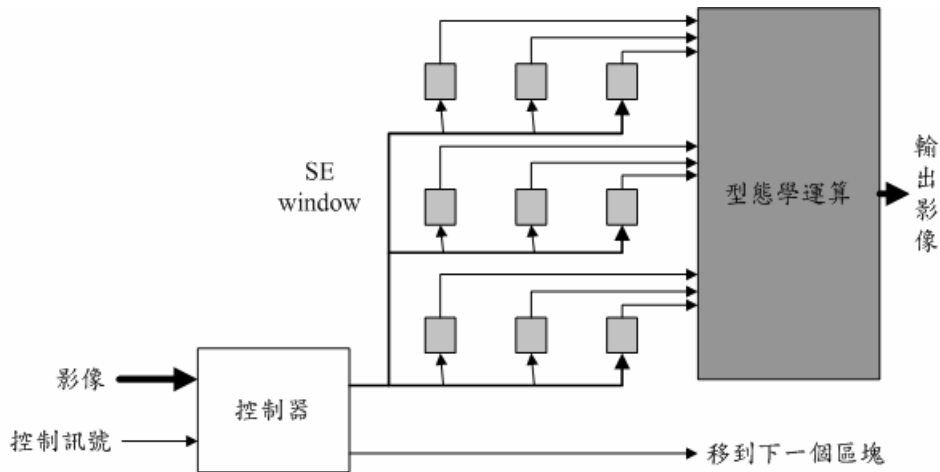
0	1	0
1	1	1
0	1	0

圖八、十字矩陣

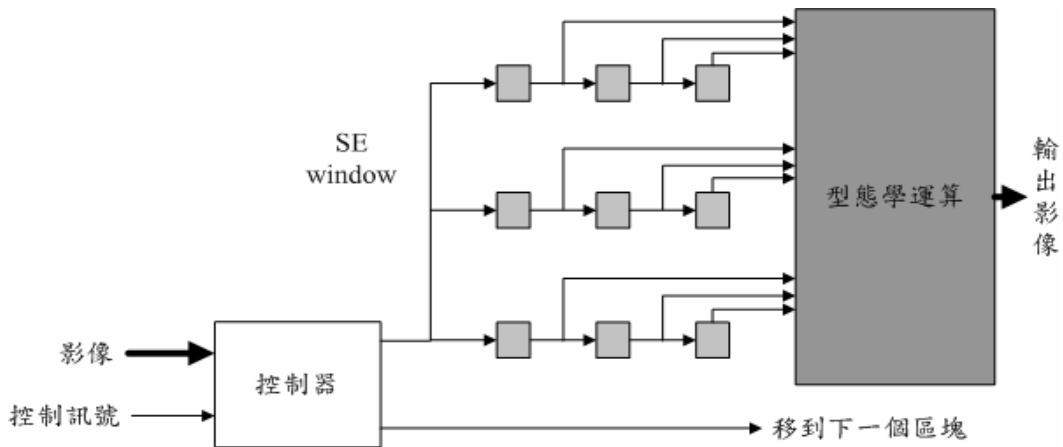
X ₁	X ₂	X ₃
X ₄	X ₅	X ₆
X ₇	X ₈	X ₉

圖九、矩陣示意圖

相反地，如果使用的方法是，移動被處理的圖片，使得 SE window 對齊各個 pixel，而不是移動 SE window 的話，就可以利用部份 pixel 重複的特性，減少從記憶體讀取資料的次數，本論文所提出的處理方法之 FPGA 區塊如圖十一。



圖十、傳統處理方法之 FPGA 區塊



圖十一、本論文所提出的處理方法之 FPGA 區塊

若以常用的 3x3 大小的 SE window 來看，每次進行型態學運算，使用傳統的 FPGA 區塊需要讀取 9 個 pixel 才能進行型態學運算，但若使用本文件所提出的 FPGA 區塊，除了影像每一行(Row)的第一次運算，每次只需讀取 3 個 pixel 就能進行運算，可以很明顯地提昇型態學運算之效能。

四、實驗結果

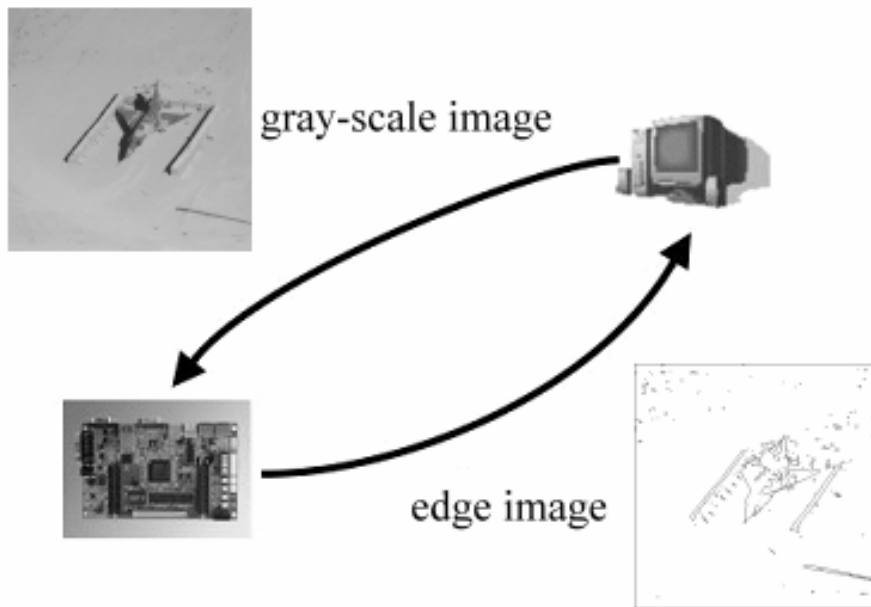
(一) 開發與測試環境

本論文之開發與測試環境如下列：
 FPGA tool: Xilinx ISE Foundation 6.3i、
 Xilinx ISE WebPACK 9.1i、
 GRMON evaluation version

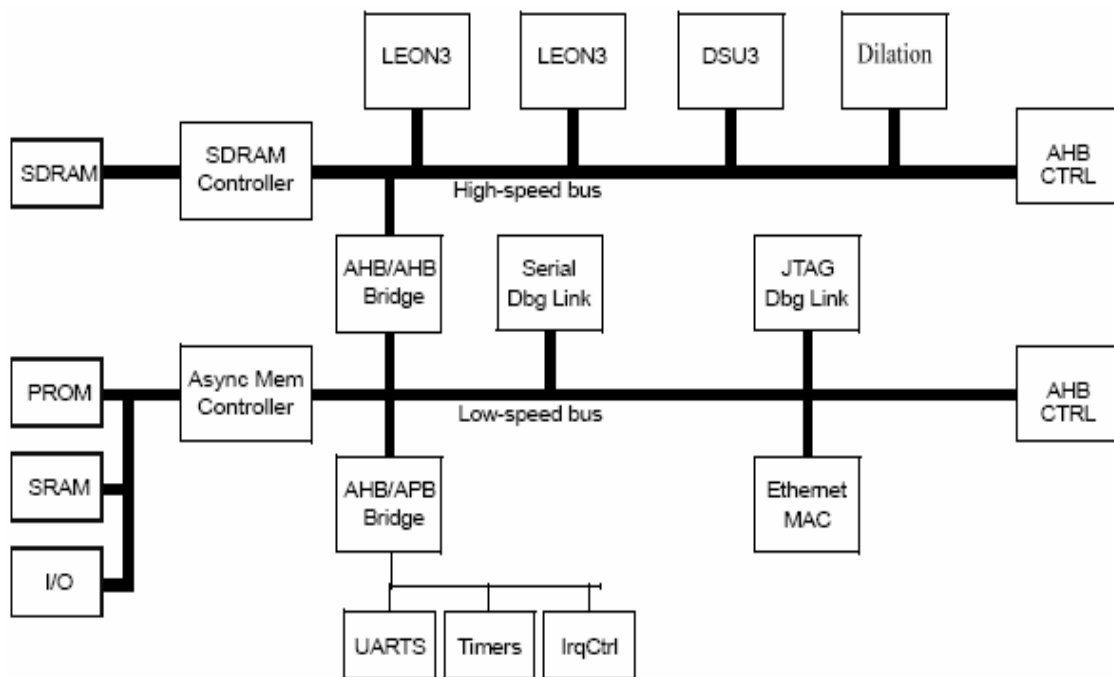
FPGA: GR-XC3S-1500 (Xilinx XC3S1500 development board for LEON)
 IMAGE: 大小為 512x512，每像素 8 位元
 SE: diameter 為 3x3 的十字矩陣

(二) 模擬方式

模擬系統的運作圖如圖十二，電腦透過 RS232 的傳輸介面，將一張 512x512 的灰階影像圖片傳給 FPGA 開發板，FPGA 開發板上所設計的系統晶片依照特定的 protocol 與電腦溝通，將整張圖片儲存在 SDRAM 的特定區塊，接著開始進入我們所關注的影像處理的階段，SDRAM 上的資料會送入設計好的演算法之系統晶片進行處理，並將處理好的影像放回 SDRAM，依照 protocol 回傳影像，最後由電腦端驗證運算結果。



圖十二、模擬系統運作圖



圖十三、SoC 平台硬體架構圖

(三) SoC 架構

本論文所使用 SoC 平台之硬體架構如圖十三，在運算上，使用雙核心之 LEON3 及自行設計之型態學運算 Dilation IP，以 AHB 匯流排將高速運算之 IP core 連接起來，在對外傳輸之 IP 上有 UARTS 的 RS232 及 Ethernet MAC 的乙太網路，在除錯上，可以使用 JTAG Debugger Link。

整體 SoC 架構為，先由 LEON3 進行

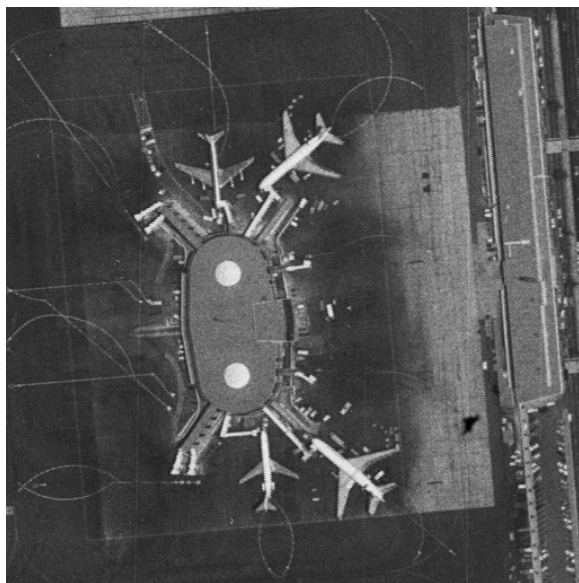
二值化之運算，並將計算後的二元影像 B 交由我們所設計型態學運算之 Dilation IP，最後將 Dilation 後的影像與二元影像 B 作互斥運算(XOR)，得到邊緣影像。

(四) 測試結果

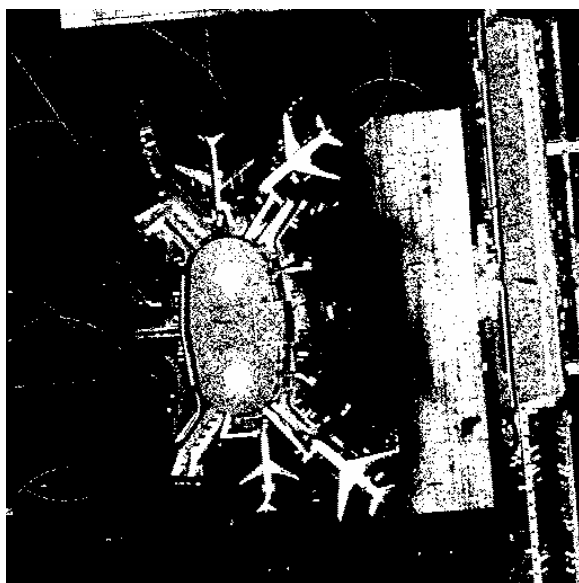
所使用影像為機場遙測影像圖(如圖十四)，主要物件為四台飛機與停機埠，原始影像之灰階分佈圖如圖四所示。

我們的目標是將原始的機場遙測圖在進行形態學邊緣偵測，因此必須先進行二值化處理。此影像經過二值化運算[2]後，我們可以看到運算的結果圖(如圖十五)，主要的物件(飛機與停機埠)為亮點集合，背景為暗點集合，成功地將物件與背景分離，完成形態學運算之前處理。

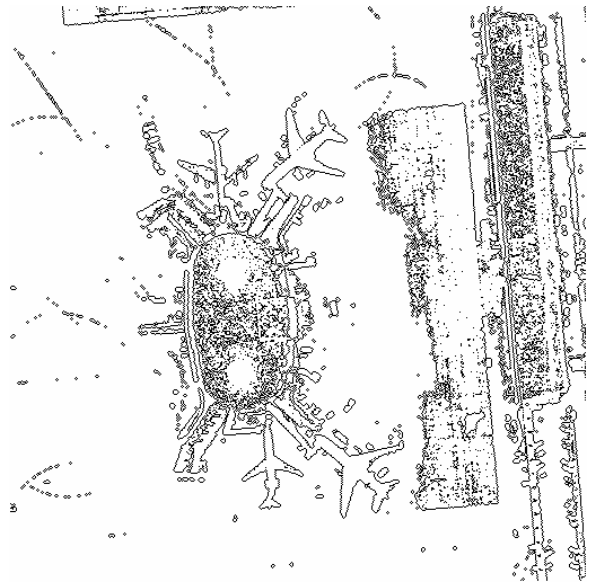
邊緣偵測後的影像(如圖十六)，可以清楚地看到四台飛機與停機埠的邊緣，提高人眼及電腦視覺對影像中物件的辨識率，並減少影像所需要的儲存量，進而減少影像傳輸的延遲時間。



圖十四、原始的機場遙測影像圖



圖十五、機場遙測影像圖二值化後之影像



圖十六、機場遙測影像圖之邊緣圖

在此，我們進一步比較在 SoC 平台上比較不使用 Dilation IP、使用傳統的 Dilation IP、使用最佳化後之 Dilation IP 所需要的程式執行脈衝週期 (clock cycle)，以分析 SoC 平台上之型態學運算有無使用 Dilation IP 及 IP 有無最佳化的效能差異。如表一所示，可以看出在沒有 Dilation IP 下的 SoC 平台，由於必須使用高階軟體作運算，在效能上最差，而最佳化後的 Dilation IP 比傳統的 Dilation IP 效能上快了 25% 左右，更是比無 Dilation IP 之運算快了將近 40%。

表一、clock cycle 比較表

型態學運算 (Dilation)	耗時 (clock cycle)
C/C++ code	1,243,398
傳統 IP	1,076,450
最佳化 IP	784,085

五、結論

透過灰階影像二值化與型態學邊緣偵測運算，減少遙測影像資料量後，還可以進行編碼與壓縮等後處理，再次減少資料的傳輸量，進一步達成遙測影像即時處理與傳輸。在本文件中，我們建立了影像處理的 IP library，包含了二值化與型態學

運算，數學形態學在處理影像方面提供了相當簡單及有效的方法，再加上影像二值化前處理，這兩種方法的整合，將原本只能應用於黑白影像的形態學運算，延伸到處理灰階影像，擴大應用範圍，設計出具即時影像處理的系統晶片，對於航太、軍事...等各種需即時性訊息之領域的應用，提供即時、低耗能、高效率的影像處理功能。

六、參考文獻

- [1] A. Abutaleb, "Automatic thresholding of gray-level pictures using two-dimensional entropy," *Computer Vision, Graphics and Image Process.*, vol. 47, pp. 22-32, 1989.
- [2] A. Amer and E. Dubois, "Image segmentation by robust binarization and fast morphological edge detection. In *Proc. Vision Interface*, pages 357--364, Montreal, Canada, May 2000.
- [3] K. Appiah, A. Hunter and T. Kluge, "GW4: An FPGA-driven image segmentation algorithm," *Proc. 5th World Scientific and Engineering Academy and Society International Conference on Signal, Speech and Image Processing (WSEAS SSIP'05)*, 17-19 August, 2005, Corfu Island, Greece.
- [4] D. Baumann, J. Tinembart, "Mathematical Morphology Image Analysis on FPGA" *AISTA 2004*, Luxembourg, Septembre 2004.
- [5] J. Bernsen, "Dynamic thresholding of grey-level images," in *Proc. 8th Int. conf. on Pattern Recognition (ICPR8)*, (Paris, France), pp.1251-1255, Oct, 1986.
- [6] A. Chehikian, "Image segmentation by contours and regions cooperation," *Signal Process.*, vol. 78, pp. 329-347, Nov. 1999.
- [7] M. Dai, P. Baylou, L. Humbert, and M. Najim, "Image segmentation by a dynamic thresholding using edge detection based on cascaded uniform filters," *Signal Process.*, vol. 52, pp. 49-63, Apr. 1996.
- [8] J. R. Evans, T. Arslan, "The Implementation of an Evolvable Hardware System for Real Time Image Registration on a System-on-Chip Platform," eh, p. 142, 2002 *NASA/DoD Conference on Evolvable Hardware (EH'02)*, 2002.
- [9] L. Garrido, P. Salembier, and D. Garcia, "Extensive operators in partition lattices for image sequence analysis," *Signal Process.*, vol. 66, pp. 157-180, 1998.
- [10] K. Johnston, D. Gribbon and D. Bailey, "Implementing image Processing Algorithms on FPGAs," *Proceedings of the Eleventh Electronics New Zealand Conference*, Palmerston North, Nov. 2004.
- [11] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Mach. And Cybern.*, Vol. 9, no. 1, pp. 62-66, 1979.
- [12] S. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 884-900, Sept. 1996.
- [13] Gaisler research, LEON3 Processor, <http://www.gaisler.com/>