Ultra Lightweight Dehaze Methods for Robot Vision Using HLS-專題演講筆記

2025/12/23

壹、 前言：一場理論與實踐的完美結合

本篇心得旨在記錄與反思聆聽國立臺北大學宋啟嘉教授主講「Ultra Lightweight Dehaze Methods for Robot Vision Using High-Level Synthesis」後的心得與啟發。作為一名專攻嵌入式系統與電腦視覺的資訊工程研究生，這個主題完美地擊中了我的研究興趣核心。

在當前 AIoT 與機器人視覺應用蓬勃發展的時代，如何克服邊緣裝置的運算資源限制，實現高效能的即時影像處理，始終是我們面臨的關鍵挑戰。宋教授的演講不僅深入剖析了前沿的去霧演算法，更將其與 FPGA 硬體加速及高階合成（High-Level Synthesis, HLS）開發流程緊密結合，為我們展示了一條從軟體演算法到硬體 IP 實現的清晰路徑，其內容兼具學術深度與工程實用性，帶來了極大的啟發。

貳、 奠定基石 — 現代 FPGA 的演進與高階合成的崛起

當代 FPGA 的技術趨勢與戰略價值：

演講開宗明義地揭示了現代 FPGA 技術的驚人進展。其技術指標不僅是單純的數字，更代表著一種戰略性的運算典範轉移。當前最先進的製程已達 7nm FinFET+，單一 Virtex Ultrascale+ 晶片上的電晶體數量高達 350 億個，其規模令人咋舌。更關鍵的是，邏輯閘的成本效益出現了指數級的提升，從 1990 年的每個 $15 美元驟降至 2020 年的 $0.01 美元。這種巨大的成本優勢與無與倫比的平行運算能力，解釋了為何像 AWS 這樣的雲端巨頭自 2018 年起便大規模採用 FPGA 作為其雲端運算加速的核心組件之一。FPGA 已不再是傳統意義上的原型驗證工具，而是高效能異構運算中不可或缺的一環。

FPGA 架構的演化脈絡：

為了更清晰地理解 FPGA 的發展，宋教授梳理了其架構的演進歷程。從最初單純的邏輯閘陣列，到今日高度整合的系統單晶片，FPGA 的每一次迭代都標誌著功能與整合度的躍升。

時期	架構類型	核心特徵
1983 年以來	傳統 FPGA	主要由邏輯區塊（Logic Blocks）和大量 Block RAM 組成，使用硬體描述語言（HDL）進行設計，彈性高但開發週期長。
2004-2012 年	SOPC	可在 FPGA 內部配置一個 32 位元的軟核 CPU，在 uClinux 環境下進行 C 語言編程，但 CPU 速度受限於邏輯閘。
2012-2017 年	SoC FPGA	硬體整合了雙核心 ARM CPU（667MHz-1000MHz），顯著提升處理效能與系統整合度，成為主流的嵌入式系統開發平台。
2019 年至今	Cloud FPGA	專為資料中心設計，透過 PCIe 高速介面與伺服器連接，提供比 CPU/GPU 更高效能的特定應用加速。
2017 年至今	RFSoC	整合了高效能的類比數位轉換器(ADC)與數位類比轉換器(DAC)，為 5G 通訊、雷達等射頻應用提供了單晶片解決方案，大幅降低功耗與複雜度。

整合了高效能的類比數位轉換器（ADC）與數位類比轉換器（DAC），為 5G 通訊、雷達等射頻應用提供了單晶片解決方案，大幅降低功耗與複雜度。

高階合成（HLS）作為關鍵推動力的角色：

傳統 FPGA 開發最大的門檻在於冗長且複雜的硬體描述語言（RTL）設計流程。高階合成（HLS）的出現，正是為了解決此一痛點。HLS 允許開發者使用熟悉的 C/C++ 等高階語言來描述演算法邏輯，再由工具鏈自動編譯成底層的 RTL 電路。

這個流程不僅僅是語言的轉換，它帶來了三大核心效益：

1. 降低開發門檻： 讓不熟悉硬體設計的軟體工程師與演算法科學家也能夠利用 FPGA 的強大運算能力。

2. 加速設計迭代： 大幅縮短開發週期。演講中引用的「QRD」專案，開發時間從傳統方法的 12 週驚人地縮短至 1 週，這在分秒必爭的產業環境中具有決定性的優勢。

3. 優化資源使用： HLS 工具能進行設計空間探索（Design Exploration），在效能與資源間取得平衡。例如，「Sphere Decoder」的案例顯示，透過 HLS，LUTs 資源減少了 11%，Registers 減少了 31%。

總結而言，深入理解 FPGA 從硬體架構到開發方法的演進，特別是掌握 HLS 這座串連軟體與硬體的橋樑，是設計現代高效能異構運算系統的必經之路。而宋教授接下來分享的去霧演算法，正是這一理念的最佳實踐。

參、 核心應用剖析 — 超輕量級即時去霧演算法

問題背景與重要性：

在機器人視覺、自動駕駛等領域，影像品質是所有後續分析的基石。然而，在真實世界中，霧、霾、雨天或低光照等惡劣天候條件，會導致影像對比度下降、細節丟失，進而嚴重影響物件偵測、語義分割等高階電腦視覺任務的準確性。因此，開發一個能夠即時、高效地去除這些環境干擾的演算法，具有迫切的應用需求。

演算法的核心目標與設計思路：

針對上述挑戰，宋教授團隊的研究目標清晰而宏大，可歸納為以下四點：

- ✓ 超越現有技術： 開發一個在性能上與現有先進演算法（無論是傳統方法或基於深度學習的方法）相當甚至更好的超高效能演算法。
- ✓ 最小化缺點： 針對現有方法可能存在的色彩偏移、細節丟失等問題進行改善。
- ✓ 最小化運算需求： 設計一個極度輕量化的模型，使其能夠在資源受限的邊緣裝置上流暢運行。
- ✓ 硬體實現與整合： 利用 HLS 將演算法實現為一個可重用的 IP 核心，並無縫整合至即時的 HDMI 影像處理管線中。

➢ 去霧演算法的技術流程：

該去霧演算法的流程設計精巧，構成了一個從分析到重建的完整管線。首先，演算法同時對輸入影像進行暗通道提取（Dark channel extraction）亮通道提取（Bright channel extraction），以分別估計霧的濃度與大氣光。接著，結合這兩項資訊與一個可調的自適應霧氣因子（Adaptive Fog Factor），初步估算出傳輸圖（Transmission map）。

為了提升傳輸圖的品質與準確性，演算法進一步採用卷積（Convolution）非等向性擴散濾波器（Anisotropic Diffusion Filter）恢復場景（Scene recovery）。恢復後的影像還會經過一道對比度延展（Contrast Stretching）的後處理步驟，最終生成清晰、自然的輸出影像。

➢ 演算法的性能綜合評估：

該演算法的性能表現極為出色，不僅在傳統影像品質指標上領先，更在實際

電腦視覺任務中展現了其應用價值。

首先，在 Dehazing Metrics on RESIDE 資料集的標準評測中，其處理速度遠超其他方法。

Metrics	DCP [4]	BCCR [14]	MAPD [16]	**Proposed**
FPS	0.23	4.39	2.46	**384.37**
PSNR	17.780	17.382	22.006	**22.181**
SSIM	0.821	0.768	0.887	**0.889**

從上表可見，該演算法在維持頂尖的 PSNR 與 SSIM 指標的同時，達到了 384.37 FPS 的驚人處理速度，這意味著它完全有能力處理高幀率的即時影像串流，而這是許多複雜演算法難以企及的。值得注意的是，此方法不僅超越了如 DCP 等經典演算法，其效能指標在與 Dehaze-Net、Dehaze-Former 等多種近期提出的先進方法相比時，同樣展現出顯著的競爭優勢。

其次，為了評估對下游任務的實際影響，研究團隊在 COCO 資料集上合成了不同濃度的霧氣，並使用 YOLOv7-tiny 進行物件偵測。結果顯示，在濃霧條件下（例如霧氣係數 β=3.0），經過該演算法處理後的影像，物件偵測的平均精確率（mAP）提升了高達 7.9%。這項數據有力地證明，該演算法不僅是「看起來」更清晰，而是能實質性地提升 AI 模型的感知能力。

如此卓越的性能，其背後必然離不開針對硬體特性進行的精巧 HLS 優化。接下來，我們將深入探討實現這一切的關鍵技術。

肆、 關鍵技術洞察 — 五大 HLS 優化策略

要將一個 C/C++ 描述的演算法高效地轉譯為平行執行的硬體電路，單純的編譯是遠遠不夠的。開發者必須使用一系列的優化指令（Directives）來指導 HLS 工具生成期望的硬體架構。宋教授在演講中分享了五大核心優化策略，正是將演算法性能最大化的精髓所在。

✓ Optimization I: Loop Unrolling (迴圈展開)

- 原理與目的： 該技術將迴圈的多次迭代展開，轉化為多個可以同時執行的獨立運算單元，以此來發掘迭代之間的平行性。
- 性能影響： 演講中的範例顯示，一個原本需要 4 個時脈週期才能完成的迴圈，在完全展開後，僅需 1 個週期即可完成，大幅降低了延遲。

- ✓ Optimization II: Array Partitioning (陣列分割)
  - ➢ 原理與目的： 在硬體中，陣列通常被實作為有存取埠限制的 Block RAM。因此，若要讓「迴圈展開」的平行運算潛力完全發揮，就必須透過陣列分割來解決記憶體頻寬瓶頸。陣列分割將一個大陣列拆分成多個小型的獨立記憶體，每個都有自己的讀寫埠，從而支援真正的平行存取。
  - ➢ 性能影響： 這是實現高吞吐率平行處理的基礎，若無此優化，Loop Unrolling 的效果將大打折扣。
- ✓ Optimization III: Resources (資源約束)
  - ➢ 原理與目的： 開發者可以透過指令明確指定特定運算（如乘法、加法）所使用的硬體資源類型與數量。例如，可以選擇使用組合邏輯實現的快速乘法器，或是節省面積但延遲較高的流水線乘法器。
  - ➢ 性能影響： 這一策略賦予了設計者在速度、面積和功耗之間進行權衡取捨的彈性，以達到最佳的系統設計目標。
- ✓ Optimization IV: Loop Pipelining (迴圈流水線)
  - ➢ 原理與目的： 流水線技術將迴圈中的單次迭代操作拆分成多個階段，並讓不同迭代的各個階段重疊執行，就像工廠的生產線一樣。
  - ➢ 性能影響： 雖然單次迭代的總延遲（Latency）可能會略微增加，但系統的吞吐率（Throughput）會大幅提升。演講範例中，未使用流水線的設計吞吐率為每 3 個週期產出一個結果，而使用後則能達到每 1 個週期就產出一個結果，實現了三倍的效率提升。
- ✓ Optimization V: Dataflow (資料流)
  - ➢ 原理與目的： Dataflow 是一種更高層次的流水線技術，它作用於函式（Function）或任務（Task）層級。當一個系統由多個依序執行的函式組成時，Dataflow 優化可以讓這些函式像流水線一樣平行運作。一旦前一個函式處理完一部分資料並輸出，後一個函式就可以立即開始處理，無需等待前一個函式完全結束。
  - ➢ 性能影響： 極大地提升了整個處理鏈路的吞吐率，是實現高效能影像處理管線的關鍵技術。

總結來說，這五大優化技術的靈活組合與運用，是將理論演算法轉化為高效能、低延遲硬體實現的關鍵所在。這些技巧不僅適用於影像去霧，其應用潛力更遍及所有需要在 FPGA 上實現的高效能運算領域。

伍、 延伸應用與未來展望

- 核心技術框架的廣泛適用性：

宋教授的研究最令人振奮的一點，在於其核心設計理念與技術框架具有極佳的延展性。團隊成功地將這套輕量級演算法設計與 HLS 優化方法，從去霧領域延伸至另外兩個充滿挑戰性的影像增強問題：

1. 低光照影像增強 (Low Light Enhancement): 針對夜間或光線不足場景，該方法同樣取得了優異成果。在相關的 Benchmark 數據中，其 FPS 高達 13.7，遠超許多傳統與深度學習方法，實現了即時處理，同時在 PSNR 等影像品質指標上也保持著高度競爭力。

2. 水下影像增強 (Under Water Image Enhancement): 水下影像因光線的選擇性吸收（特別是紅色光）而導致嚴重的色彩失真。該團隊的方法同樣能有效校正色彩偏差，恢復影像細節，在 UIEB 資料集的評測中，其視覺效果與量化指標均名列前茅。

這兩個成功的延伸應用證明，其背後的設計哲學——即「輕量化演算法 + HLS 硬體加速」——是一個強大且通用的解決方案，能夠應對多種因環境因素導致的影像品質下降問題。

- 未來研究方向：

演講最後，宋教授也揭示了團隊正在進行中的未來工作，為我們描繪了更廣闊的研究藍圖：

- ✓ 整合更強大的視覺模型： 計畫將增強後的影像輸入到更先進的 YOLO v7 與 v12 模型中，進行更複雜的物件偵測、實例分割與影像分類任務，以驗證其在更高階視覺任務上的助益。
- ✓ 生成合成訓練資料： 為了更好地訓練與評估低光照下的模型，團隊計畫使用 CycleGAN 來生成一個合成的黑暗場景 COCO 資料集，透過生成對抗網路創造出更多樣化、更逼真的訓練數據。

這些前瞻性的工作不僅展示了該技術框架的應用深度，也為我們後續的研究

提供了極具價值的參考方向。

陸、 結論：個人反思與啟發

宋啟嘉教授的演講如同一場及時雨，為我的研究帶來了深刻的啟發。總結而言，我認為本次演講有三個最核心的收穫：

1. HLS 的巨大潛力： 演講清晰地展示了 HLS 如何作為一座堅實的橋樑，彌合了軟體演算法與硬體實現之間的巨大鴻溝。特別是 Dataflow 與 Loop Pipelining 等指令，讓我們能將演算法的函式與迴圈轉化為高效的硬體流水線，這對於實現像即時影像處理管線這樣對吞吐率有極高要求的應用至關重要。

2. 輕量化演算法的關鍵價值： 在雲端 AI 模型日益龐大的今天，宋教授反其道而行，專注於設計極度輕量化且高效的演算法。這對於資源有限的機器人、無人機等邊緣運算裝置而言，才是真正實用且可部署的解決方案。

3. 系統性的評估方法： 研究團隊不僅使用 PSNR/SSIM 等傳統指標，更創新地將演算法與下游的物件偵測任務（如 YOLO）結合進行評估。這種直接衡量「對最終任務的貢獻」的評估方式，比單純的影像品質分數更具說服力，也更貼近實際應用場景。

對我個人而言，這次演講的內容與我的研究方向高度契合。我目前正致力於將電腦視覺演算法部署到嵌入式平台上，並時常受限於處理器的運算能力瓶頸。宋教授分享的五大 HLS 優化策略——特別是 Loop Pipelining 與 Dataflow 的概念——為我提供了一套具體的「方法論」，讓我思考如何重新審視我目前使用的演算法，並將其中的關鍵運算部分，透過 HLS 移植到 FPGA 上，構建一個 CPU+FPGA 的異構加速系統。未來，我計畫將這些技術應用於我的研究課題中，嘗試解決即時語義分割任務中的運算瓶頸，相信這將為我的研究開啟一扇新的大門。