

Ultra Lightweight Dehaze Methods for Robot Vision Using

HLS-專題演講心得

2025/12/23

壹、序言

近日有幸參與國立臺北大學宋啟嘉教授主講的「Ultra Lightweight Dehaze Methods for Robot Vision Using High-Level Synthesis」專題演講，這是一場深度結合硬體加速、高階合成（High-Level Synthesis, HLS）與前瞻電腦視覺應用的知識饗宴。身為一名主攻電腦視覺與嵌入式系統的資工研究生，這個主題幾乎完美地擊中了我所有學術興趣的核心。在當前這個 AI 模型日益龐大、算力需求不斷攀升的時代，如何將複雜的演算法高效地部署到資源受限的邊緣裝置（如機器人）上，已成為學術界與工業界共同面臨的關鍵挑戰。宋教授的研究，正是在這個交會點上，提出了一個兼具優雅理論與卓越效能的解決方案。

本次心得將不僅是對演講內容的摘要，我更希望能以一名研究生的視角，深入剖析其技術演進的宏觀脈絡、HLS 在其中扮演的革命性角色，以及這項研究成果對於解決真實世界問題的深遠意義。為此，讓我們先從這一切的基礎——FPGA 的發展趨勢談起。

貳、宏觀趨勢：為何 FPGA 是當代計算的明日之星？

在深入探討特定的除霧演算法之前，理解講者選擇 FPGA 作為技術平台的深層原因至關重要。演講開頭對 FPGA 宏觀趨勢的分析，為整個研究的動機與價值提供了堅實的背景。FPGA 不再是過去那個僅限於硬體專家的神秘領域，而是當代異構計算中一顆冉冉升起的明星。

演講中提到了三大趨勢，共同推動了 FPGA 的普及與發展：

- 製程與成本效益：技術的飛速進展令人驚嘆。當前的 FPGA 已採用頂尖的 7nm FinFET+ 製程，單顆如 Virtex Ultrascale+ 的晶片便能容納高達 350 億個電晶體。更關鍵的是成本的戲劇性下降：單一邏輯閘的成本從 1990 年的約 15 美元，驟降至 2020 年的 0.01 美元。這使得 FPGA 從昂貴的原型驗證工具，轉變為具備大規模部署潛力的計算平台。
- 雲端數據中心的採用：自 2018 年以來，以 AWS 為首的雲端服務巨頭，其採用的 FPGA 實例已超過 50%。這標誌著一個重要的產業轉變：FPGA 憑藉其可重構性與高能效比，在數據中心加速任務中（如機器學習、影像處理）找到了不可或缺的定位，不再僅僅是 CPU 和 GPU 的附庸。
- 異構計算的崛起：現代 FPGA 已演化為高度整合的 SoC (System-on-Chip)。這種「異構硬體加速」（Heterogeneous Hardware Acceleration）架構，將強大的

ARM CPU 核心、高速 ADC/DAC，以及可程式化邏輯陣列整合在單一晶片上，為處理複雜的混合訊號與計算任務提供了前所未有的彈性。

宋教授以一張清晰的演進圖，描繪了 FPGA 從 1980 年代至今的架構演化歷程：

- FPGA (Since 1980s): 以邏輯區塊 (Logic blocks) 和少量記憶體 (Block RAM) 為核心，使用硬體描述語言進行開發。
- SOPC FPGA (2004-2012): 可配置軟核心 CPU (如 Altera NIOS) 於 FPGA 中，開啟了軟硬體協同設計的時代。
- SoC FPGA (2012-2017): 整合硬核心的 32 位元雙核心 CPU (如 Xilinx Zynq、Altera Cyclone V)，性能與整合度大幅提升。
- MPSoC FPGA (2017-Today): 進入多核心處理器時代，整合 64 位元四核心 ARM CPU，適用於更複雜的資料處理與 AI 應用。
- Cloud FPGA (2018-Today): 專為數據中心設計，提供比 CPU/GPU 更高的能效，並透過虛擬化技術加速雲端應用。
- RFSoC (2017-Today): 終極的異構整合，將射頻等級的 ADC/DAC 直接整合進晶片，徹底改變了通訊與雷達等領域的設計典範。

正是因為 FPGA 變得如此強大、普及且易於取得，才為接下來要討論的、旨在弭平軟硬體開發鴻溝的高階合成技術鋪平了康莊大道。

參、核心技術：HLS 如何弭平硬體與軟體的鴻溝

本章節將聚焦於本次演講的技術核心——高階合成 (High-Level Synthesis, HLS)。宋教授強調，HLS 不僅僅是一種工具，更是一種徹底改變嵌入式系統開發思維的革命性方法。對於像我這樣熟悉軟體開發，但對傳統硬體描述語言 (HDL) 感到卻步的研究者而言，HLS 無疑是開啟硬體加速大門的鑰匙。

HLS 的核心概念是將高階程式語言 (如 C/C++) 的程式碼，透過編譯與指令 (Directives) 引導，自動轉換為底層的硬體描述語言 (RTL)。這帶來了兩大無可比擬的優勢：

1. 降低開發門檻 (Lowers HLS entrance barrier)：演算法開發者可以專注於演算法邏輯本身，而無需從頭學習複雜的 Verilog 或 VHDL 語法與硬體時序設計，大幅縮短了學習曲線。
2. 減少設計成本 (Alleviates design costs)：HLS 能夠快速進行設計探索與迭代。演講中引用了一個 QRD (QR 分解) 的案例，指出使用 HLS 相較於傳統手寫 RTL，「節省了 2 至 3 週的工程時間」，這個數據具體地展現了 HLS 在加速開發流程上的驚人效益。

為了量化 HLS 的優化能力，演講中展示了一個案例。The following table, based on the "Sphere Decoder" case study, compares the resource usage of an HLS-generated design against an expert's manual RTL design:

指標 (Metric)	傳統 RTL 專家 (RTL Expert)	HLS (Autopilot)	差異 (Diff %)
LUTs	32,708	29,060	-11%
Registers	44,885	31,000	-31%
DSP48s	225	201	-11%
BRAMs	128	99	-26%

從上表可見，HLS 不僅提升了開發效率，其生成的硬體在資源利用上甚至比專家手動設計更為精簡。然而，要發揮 HLS 的最大潛力，關鍵在於掌握其優化技巧。宋教授詳細解析了五種核心優化指令，它們可以被視為是「程式設計師的工具包，用以明確指示編譯器如何將序列式的軟體邏輯，映射到並行化的硬體結構上，而這正是 FPGA 加速的主要來源」。

- Optimization I: Loop unrolling (迴圈展開)

其原理是將迴圈內的多次迭代操作，在硬體上複製成多個並行的運算單元。一個原本需要 4 個時脈週期才能完成的迴圈，展開後可以在 1 個週期內完成，以面積換取速度，是提升運算密集型任務效能的利器。

- Optimization II: Array partitioning (陣列分割)

在軟體中，陣列儲存於單一記憶體區塊。但在硬體上，這會造成存取瓶頸。陣列分割技術可以將一個大陣列拆分成多個小的記憶體單元（可分為 block, cyclic, complete 三種模式），從而允許多個運算單元同時對其進行並行存取，大幅提升記憶體帶寬。

- Optimization III: Resources (資源分配)

HLS 允許開發者透過指令 (directive) 來明確指定某個運算（如乘法）應該使用哪一種硬體資源來實現。例如，一個乘法可以被實現為組合邏輯、多級流水線乘法器等，開發者可以根據對面積和速度的需求進行權衡。

- Optimization IV: Loop Pipelining (迴圈流水線)

這是 HLS 中最核心的優化之一。它將迴圈中的操作分解成多個階段，讓不同迭代的操作可以重疊執行。這好比一條工廠的裝配線；雖然一台車從頭到尾的製造時間 (延遲, Latency) 可能更長，但工廠每分鐘都能產出一台新車 (吞吐率, Throughput)，遠比一次只造一台車高效。流水線技術極大地提升了整體吞吐率，能夠在每個時脈週期都產出一個結果。

- Optimization V: DataFlow (資料流)

如果說流水線是在迴圈內部進行優化，那麼資料流就像是更高層次的流水線。它作用於函數 (function) 或迴圈之間，將生產者-消費者模式的任務串聯起來，自動建立 FIFO (先進先出佇列) 來緩存數據，使得整個處理流程可以像水流一樣不間斷地高效運作。

掌握這些 HLS 優化技巧，是將高階演算法高效移植到 FPGA 上的不二法門。而接下來，我們將看到這些技術如何被應用於解決一個棘手的實際問題。

肆、應用實戰：以輕量化除霧演算法解決機器視覺痛點

理論與工具的價值最終體現在解決實際問題上。本章節將深入探討宋教授的研究團隊如何巧妙地運用前述的 FPGA 與 HLS 技術，來應對機器視覺在戶外惡劣環境下面臨的嚴峻挑戰。

正如演講中所指出的，霧、雨、低光照等不良天候因素會嚴重「降低機器視覺系統的性能」（degrades performance of Robot Vision）。影像品質的下降會直接衝擊下游的 AI 任務，例如物件偵測（Object detection）、影像分類（Classification）、目標追蹤（Tracking）與語意分割（Segmentation），這對於自動駕駛或戶外機器人等安全攸關的應用而言，是亟待解決的痛點。

為此，該研究設定了清晰且務實的主要目標（Main Objective）：

- 開發一款超輕量級的影像增強演算法，能夠處理除霧、除雨、低光及水下等多種場景。
- 消除或最小化現有方法的缺陷，例如傳統暗通道先驗（DCP）演算法的色彩偏移問題。
- 將計算需求最小化，使其適合部署在資源受限的嵌入式平台。
- 使用 HLS 將演算法在 FPGA 上實現為一個可重用的 IP 核心，並整合到 HDMI 影像管線（HDMI pipelining）中。

其核心的除霧方法（Dehaze Method）建立在經典物理模型之上，但進行了諸多創新改良。根據演講中的流程圖，其詳細步驟如下：

1. 通道提取：同時進行暗通道（Dark channel）與亮通道（Bright channel）的提取，前者用於估計霧的濃度，後者用於修正天空等高亮區域的估計偏差。
2. 霧氣估算與透射率圖生成：根據通道資訊計算出一個「自適應霧氣因子」（Adaptive Fog Factor），接著透過「卷積」（Convolution）運算初步生成透射率圖（Transmission Map）。
3. 透射率圖精煉：為了平滑邊緣並保留細節，使用「各向異性擴散濾波器」（Anisotropic Diffusion Filter）對透射率圖進行精煉，使其更貼近真實場景。
4. 場景恢復與後處理：利用大氣散射模型，根據精煉後的透射率圖反解出清晰的場景影像。最後，透過「對比度拉伸」（Contrast Stretching）進一步增強影像的視覺品質，作為最終輸出。

當然，演算法的好壞需要由客觀數據來證明。接下來，我們將檢視該研究成果的全面性能評估。

伍、成果驗證：從影像品質到 AI 任務的全方位評估

任何學術研究的價值都必須建立在嚴謹且全面的量化評估之上。宋教授的團隊在這方面做出了絕佳的示範，他們從影像品質、下游 AI 任務影響，以及硬體資源消耗等多個維度，對其提出的方法進行了深入的驗證，充分展示了其研究成果的卓越之處。

◆ 影像品質指標分析

Performance was benchmarked on the RESIDE dataset against several conventional methods, with the following results:

指標 (Metrics)	DCP [4]	CAP [8]	NLD [15]	Proposed
FPS	9.23	12.27	2.97	384.37
PSNR	17.75	18.24	17.55	22.181
SSIM	0.821	0.785	0.841	0.871
BRISQUE	20.693	21.464	20.719	19.262

從數據中可以清晰地看到，「Proposed」方法不僅在峰值信噪比 (PSNR) 和結構相似性 (SSIM) 等傳統影像品質指標上取得了領先，更在處理速度 (FPS) 上達到了驚人的 384.37 帧，遠遠超越了其他基於 CPU 的演算法。這意味著該方法完全有能力進行即時 (real-time) 處理，突顯了其在硬體加速下的巨大優勢。
電腦視覺任務影響評估

影像增強的最終目的是服務於高階視覺任務。為此，研究團隊設計了一套精巧的實驗流程：他們使用 COCO 資料集和輕量化的 YOLOv7-tiny 物件偵測模型，來評估除霧演算法對下游 AI 任務的實際助益。

實驗數據顯示，在人工合成的濃霧（霧氣係數 $\beta=3.0$ ）條件下，YOLO 模型的平均精確度 (mAP) 從原始清晰影像的 36.1% 驟降至災難性的 11.1%。然而，在經過該團隊提出的除霧演算法處理後，mAP 能夠顯著恢復至 30.1%。演講總結道，這項技術「在非常濃的霧中帶來了高達 7.9% 的 mAP 提升」(up to 7.9% mAP improvement on very dense haze)，這強而有力地證明了該技術在真實 AI 應用中的巨大實用價值。

◆ 硬體資源利用率分析

輕量化不僅體現在演算法的計算複雜度上，也體現在硬體資源的消耗上。根據 HLS 工具鏈的報告，該除霧 IP 核心在 FPGA 上的資源使用情況如下：

- 查詢表 (LUT): 10%
- 正反器 (FF): 16%
- 區塊記憶體 (BRAM): 2%
- 數位訊號處理單元 (DSP): 15%

這些數據清楚地表明，該設計確實達成了「輕量化」的核心目標。它在 FPGA 晶片上僅佔用了極少的邏輯與記憶體資源，這意味著有充足的空間來整合更多其他功能，或將其部署在成本更低、尺寸更小的 FPGA 晶片上，為系統設計帶來了極大的彈性。

這個成功的框架不僅限於影像除霧，其核心思想更可以無縫擴展到其他充滿挑戰的視覺增強任務中。

陸、延伸與展望：從低光到水下，技術的無限可能

一個優秀的研究框架往往具有良好的可擴展性。在演講的後半部分，宋教授展示了如何將其核心的輕量化設計理念，成功應用於低光照和水下影像增強這兩個同樣棘手的領域，並對未來的研究方向進行了展望。

◆ 低光照增強 (Low Light Enhancement)

低光照環境下的影像通常存在噪點多、細節丟失和色彩失真等問題，是另一個類似的挑戰。研究團隊將其框架應用於此，並在標準的低光照 Benchmark 上進行了測試。結果顯示，他們的方法在 PSNR 指標上達到了 18.09 dB，超越了包括 EnlightenGAN、Reti-Diff 在內的多種主流方法。更重要的是，其處理速度達到了 13.7 FPS，在保證高品質的同時實現了接近即時的性能。

◆ 水下影像增強 (Underwater Image Enhancement)

水下環境因水體對光線的選擇性吸收（尤其是紅光衰減），會導致影像呈現嚴重的藍綠色偏和對比度下降。研究團隊利用其演算法框架，對水下影像的色彩平衡和細節進行恢復。在 UIEB 資料集的測試中，其方法在多項非參考指標 (Non-reference metrics，即無需標準答案即可評估影像品質的指標) 如 UCIQE 和 PCQI 上，均展現出與頂尖方法相當甚至更優的競爭力。

◆ 未來工作 (Undergoing Work)

展望未來，宋教授的研究並未止步。他提到目前正在進行的工作包括：

- 將影像增強模組與更先進的 YOLO v7/v12 模型深度結合，實現端到端的分割、偵測與分類任務。
- 利用 CycleGAN 等生成對抗網路技術，來生成更多樣化、更逼真的合成數據集（如合成的黑暗場景），以進一步強化模型的魯棒性。

總而言之，這項研究不僅出色地解決了當前的具體問題，更為未來的嵌入式 AI 視覺應用開闢了一條清晰且充滿潛力的新道路。

柒、總結與反思：一位研究生的學習之旅

回顧整場演講，宋啟嘉教授不僅展示了一項出色的研究成果，更為我們這些後輩學者上了一堂關於如何將理論與實踐完美結合的寶貴課程。作為一名資工研究生，我從中提煉出三點最為深刻的學習與反思：

1. HLS 的實踐力量：演講讓我真切地感受到 HLS 的變革性力量。它成功地將演算法設計與硬體實現解耦，讓像我這樣軟體背景的研究者，也能夠更親近底層硬體，將自己設計的演算法轉化為高效能的硬體加速器，而不必陷入傳統 HDL 漫長而陡峭的學習曲線中。

2. 系統性驗證的思維：我極為讚賞宋教授團隊對研究成果進行驗證的嚴謹態度。他們沒有僅僅停留在展示幾張效果驚豔的對比圖，而是從客觀的影像品質指標、對下游 AI 任務的實際影響，再到最終的硬體資源消耗，進行了完整且極具說服

力的多維度驗證。這種系統性的思維，對於如何進行嚴謹的學術研究，給了我巨大的啟發。

3. 理論與應用的連結：這場演講完美地詮釋了如何將前沿的學術理論（如改良的暗通道先驗）與先進的工程工具（如 HLS、PYNQ 平台）緊密結合，去解決一個非常具體且具有高度工業價值的問題——機器人視覺的環境適應性。這正是我們作為工科研究生所追求的最高目標。

最後，演講中分享的 PYNQ-ZU Tutorial 資源（[GitHub](#) 與 [YouTube](#) 連結）也極具價值。這類開源的教學與實作資源，為學生提供了親自動手、將理論付諸實踐的絕佳平台。