

# 繪圖處理器系統層級之功耗模型設計

## System-level Power Analysis Modeling for GPU

賴俊宏 馬明達 黃迅倫 田胡治之 盧俊銘  
Chun-Hung Lai, Arthur Marmin, Hsun-Lun Huang, Haruyuki Tago, Juin-Ming Lu

### 中文摘要

隨著行動裝置對圖形處理的需求越來越高，因此設計一個繪圖應用程式於繪圖處理器 ( Graphics Processing Unit, GPU ) 上，以估算執行時其相對應的功耗值，在進行單位功耗效能最佳化上，佔有舉足輕重的角色。有鑑於此，本論文提出一種可藉由繪圖API之執行記錄，來推導功耗，且不受限於硬體設計，並容易移植至各種不同繪圖處理器架構的通用型繪圖處理器功耗模型。本論文所提出的功耗模型可分為兩大階段，第一階段主要針對某種特定的繪圖處理器架構進行功耗模型訓練，第二階段則使用待分析程式之繪圖API執行記錄，以進行功耗預估。實驗結果顯示，使用具公信力之測試程式glmark2-es2來進行功耗預估，本論文所提出之功耗模型可達到95%之平均精確度。

### Abstract

Since 3D graphic applications have reached the field of mobile devices and known as a tremendous expansion, graphic processors' power consumption has become a major challenge in mobile SoC design. A high-level hardware agnostic model endorsed by GPU energy consumption measurements under specific 3D graphic applications running is proposed in this paper. The modeling method of the proposed design comprises of two phases, the first phase is GPU hardware dependent and the graphics API is presented in the second phase to trace the related information with only per a target application. The experiment results show that the proposed model can catch the average relative accuracy benefit about 95 % in a run-time of 2 seconds for the well-known benchmark glmark2.

### 關鍵詞(Key Words)

繪圖處理器 (Graphics Processing Unit ; GPU)

功耗模型 (Power Model)

非結構化仿真 (Structure-agnostic)

## 1 · 前言

隨著對行動裝置上畫面顯示的細緻度及華麗度之追求，繪圖處理器 ( Graphics Processing Unit · GPU ) 已被廣泛的用於加速應用程式上

3D場景的繪製，然而受限於行動裝置上有限的電池容量，繪圖應用程式開發者必須能在開發階段很快的了解到3D場景的複雜度是如何影響功耗，以取得效能、3D場景表現、及功耗之間的平衡等，因此，估算出一個繪圖應用程式

於繪圖處理器上執行時之相對應的功耗值，在進行單位功耗之效能最佳化時佔有舉足輕重的角色。

針對繪圖處理器的功耗預估，傳統方法大多針對一般用途繪圖處理器 (General Purpose Graphics Processing Unit, GPGPU) 的特性來建立功耗模型[1]-[3]，只有少部分的文獻針對嵌入式繪圖處理器中的繪圖管線 (Graphics Pipeline) 特性來建立功耗模型。此外，大多數的文獻都是仰賴繪圖處理器內部的硬體事件計數器 (Hardware Event Counters)或是其它的硬體訊號/狀態變動資訊來估算功耗，然而，這部分的資訊常常必須相依於硬體架構設計，且繪圖處理器供應商並不一定有提供方法來取得這些訊號/狀態變動資訊，這樣的特性，往往造成建立的功耗模型，只能限制運用在某種特定架構上而無法快速移植至其它種的繪圖處理器硬體架構。另一方面，由於是採用相依於硬體架構的資訊來建立功耗模型，導致繪圖應用程式開發者，可能無法對繪圖處理器架構有一定的認知，因此，很難直接從這些硬體變動資訊了解到是哪一段程式碼或哪一個3D場景會造成大量的功耗，進一步限制了最佳化的進行。

有鑑於此，本篇論文提出一套以繪圖應用程式開發者所熟悉的高階參數，來建立出通用型繪圖處理器功耗模型的方法，此高階參數意指，可藉由分析繪圖應用程式原始碼或是繪圖應用程式設計介面 (Application Programming Interface, API)，如OpenGL ES 2.0 API之執行記錄所推導而來的參數值，而不需實際存取硬體(或硬體模擬器)，來獲得且不相依於硬體架構的參數值。本功耗模型不僅使用貼近於繪圖應用程式開發者的高階參數，使其在程式開發階段就能直接觀察到這些高階參數對功耗的影響，並可輕易的找出哪個3D場景有大量的功耗並進行改善；同時由於使用不相依於繪圖處理器硬體架構之高階參數，來進行功耗預估，因此，可輕易移植至不同繪圖處理器架構；此外，本功耗模型可根據待分析的繪圖應用程式之特性，動態選用最合適的權重值，以適應繪圖處理器中各主要元件/繪圖管線的負載不平衡情況，進一步提高功耗預估的準確度。

## 2. 通用型繪圖處理器功耗模型

### 2.1 功耗模型方程式與高階繪圖參數

本論文所提出的通用型繪圖處理器功耗模型是根據OpenGL繪圖管線的運作以及Imagination TBDR (Tile Based Deferred Rendering)繪圖處理器架構，並配合前置的功耗剖析 (Power Profiling)推導出與繪圖處理器功耗具有高度關聯性的6個高階繪圖參數，同時選定5個繪圖處理器之主要元件來進行功耗模型的建立，此6個高階繪圖參數對這5個主要元件的功耗影響整理如表1所示。

表 1 高階繪圖參數與各主要元件之關係

Components Parameters	VS	FS	TA	TU	ISP
Number of Vertices	Yes	Yes	Yes	No	Yes
Number of Textures	No	Yes	No	Yes	No
Texture Filtering	No	Yes	No	Yes	No
Lighting and Shading	Yes	Yes	Yes	No	No
Overlapping Objects	No	Yes	No	Yes	Yes
Window Resolution	No	Yes	Yes	Yes	Yes

表1中，每個主要元件之功耗都會受到一些高階參數的影響，如VS (Vertex Shader)之功耗會受到頂點個數 (Number of Vertices)及打光及陰影模式(Lighting and Shading)兩個高階參數的影響，它們之間的關聯性可以為線性方程式，多項式方程式，指數方程式，或是對數方程式，而不同的高階參數可以選擇不同的關聯性方程式來逼近真實的功耗值。

另一方面，本論文所提出的功耗模型，整體的功耗是由各主要元件之功耗值配合動態權重 (Dynamic Sets of Weights)的計算而組成；在此權重值的使用，主要用來表現出不同繪圖應用程式的特性，所造成的各主要元件對整體功耗的相對貢獻值之差異，使功耗模型可根據不同繪圖應用程式的特性，動態的從事先定義好的權重值集合中，選用一組最合適的各主要元件之權重值來進行功耗預估，功耗模型方程式如(1)所示。

$$P_{total} = \sum_{j \in \{Components\}} P_{component}(j) \dots (1)$$

$$P_{component}(j) = \sum_{i \in \{Parameters\}; j \in \{Components\}} \alpha_i \cdot c_i + W_j \cdot P_{ave}(j)$$

其中的  $P_{total}$  為總功耗值；而  $P_{component}(j)$  為各主要元件之功耗值，是由會影響此元件功耗的各個選定參數之多項式關係所組成。 $c_i$  代表會影響繪圖處理器各主要元件功耗的各個高階參數之數值。 $\alpha_i$  為各個選定參數每單位的數值對耗能之貢獻值，稱之為多項式係數 (Polynomials Coefficients)。  $W_j$  則代表各主要元件的權重值，以權重值來調整各主要元件的功耗值。主要特色是定義了一系列的權重值集合，代表應用程式對於各個主要元件的工作負載差異所造成的不同功耗，最後， $P_{ave}(j)$  則代表各主要元件的平均功耗值，配合權重值的使用，可以得到各主要元件在不同工作負載下的基本功耗值。

段的工作只需針對要評估的繪圖處理器架構進行一次即可，以決定出和不同硬體架構相依的所有參數值。另一方面，功耗預估階段 (Online Phase) 的工作是完全不相依於硬體架構，只需使用藉由分析 OpenGL ES 2.0 API 之執行記錄所推導而來的高階參數值，就能預估出待分析的繪圖應用程式於特定繪圖處理器架構上執行時相對應的功耗值。

### 2.2.1 訓練階段 (Offline Phase)

訓練階段的第一個步驟為訓練出連結高階參數與功耗之間關係的多項式係數。為了達到這個目的，我們採用客製化的 OpenGL ES 測試樣本程式，藉由一次只變動一個目標高階參數值，而其餘高階參數值不變的方式，來設計測試程式；藉由改變此要評估的高階參數其值的方式，進行多次的執行與功耗的量測，最後再用不同種的關聯性方程式，來逼近會影響此元件功耗的各個高階參數與此元件功耗之間的關係，並推導出各主要元件的功耗方程式中每個高階參數的多項式係數值  $\alpha_i$ 。如圖 1 所示，本篇論文採用正交多項式回歸 (Polynomial Regression) 的方式來推導出多項式係數值。

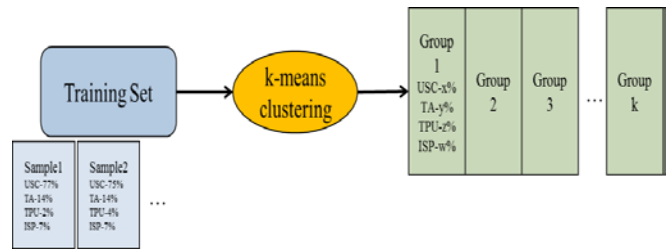
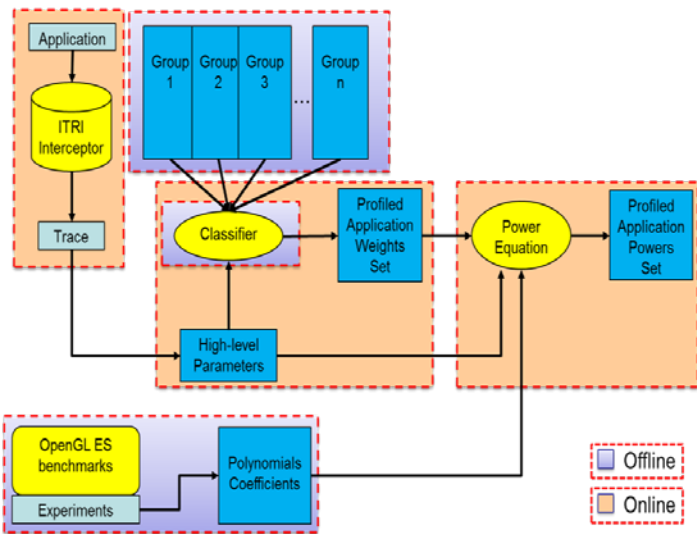


圖 2 分群及每個群組之各主要元件的權重值建立方式

圖 1 通用型繪圖處理器功耗模型之整體架構

### 2.2 功耗模型架構

圖 1 所示為本論文所提出的功耗模型整體架構，主要可分為兩大階段；訓練階段 (Offline Phase) 主要是以客製化的測試樣本程式，配合針對某一種特定的繪圖處理器架構，所實際量測出之功耗值，來訓練出各主要元件功耗與高階參數之間的關聯性，並對測試樣本程式進行分群 (Clustering & Grouping)，以推導出每個群組 (Group) 中各主要元件的權重值，訓練階

為了能使待分析的繪圖應用程式能有合適的主要元件之權重值 (Component Weight,  $W_j$ )，以協助更精確的耗能評估，因此，訓練階段的第二個步驟則以相同的測試樣本程式來進行繪圖應用程式的分群 (Clustering & Grouping)，同時推導出每個群組 (Group) 中各主要元件的權重值，以定義出一系列的候選權重值集合。分群的目的在於：將行為及特性相似的繪圖應用程式歸納在同一個群組中，在此我們使用了分群演算法 (Clustering Algorithm)，例如 K-means 來定義候選的權重值

集合。K-means是一種聚類(Cluster)的方法，其能根據我們指定的群組個數K，將n個測試樣本程式所產生的n組各主要元件之權重值劃分成K個群組，概念如圖2所示。最後在功耗預估階段 ( Online Phase ) 時，就能使用歸類器(Classifier)以判斷待分析的應用程式適用於哪一個候選群組(Group)之各主要元件的權重值

分群演算法的輸入值為：(1) 預建立的群組個數K；(2) n個測試樣本程式所實際量測出之功耗值，以及其相對應的各主要元件之權重值，即為各主要元件對整體功耗的貢獻比率值；而其輸出值則為K個群組，以及這K個群組中各自相對應的各主要元件之權重值，即為功耗模型方程式(1)中的 $W_j$ 。我們將各主要元件的平均功耗值 $P_{ave}(j)$ 定義為擁有"1"的權重值，因此，如果權重值小於1則表示目前功耗低於平均值，如果權重值大於1則表示目前功耗是大於平均值，如此一來，從各個群組之權重值數據就可以了解其特色。除了產生候選群組之各主要元件的權重值外，分群的另一個輸出結果，則是各個測試樣本程式各是歸納於哪個候選群組的這層關係，之後就可以依這層關係並配合功耗模型的各個輸入參數值(即為高階繪圖參數值或是硬體事件計數值)來進行歸類器(Classifier)的訓練。

訓練階段的第三個步驟為歸類器之訓練，目的是用來在功耗預估階段，判斷待分析的繪圖應用程式，適合使用哪一個候選群組之權重值。在此，歸類器是根據機器學習演算法來建立，我們使用了類神經網路演算法(Neural Network Algorithm)來實現；類神經網路是一種透過學習過程而擁有推論結果之能力的人工智慧機器，而所謂的學習過程指的是要有人告訴它在什麼樣的情況會得到什麼樣的結果，在本架構中是以各個測試樣本程式中高階參數值及其所屬的群組來做學習，最後在功耗預估階段便能透過此歸類器來判斷待分析的應用程式屬於哪個候選群組，幫助我們挑選合適的各主要元件的權重值。

在本論文中是使用3層結構的類神經網路，如圖3所示，包含輸入層(Input layer)、輸出層(Output layer)、隱藏層(Hidden layer)和其

中的神經元。神經元與神經元間的連線名為突觸(Synapse)，每個突觸都有一個權重值，資料通過突觸時會乘以突觸的權重值作為神經元的輸出，並以激活函數(activation function)控制輸出大小，輸入層接收功耗模型的各個輸入高階參數值資料，並在隱藏層中傳輸、分析、權衡，通過測試樣本程式的輸入與結果對各個突觸的權重值進行校正，最後在輸出層輸出所屬的群組資訊。

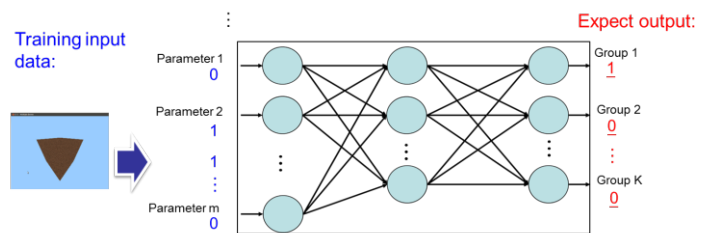


圖 3 以類神經網路來實現歸類器並進行歸類器的訓練

輸入層中的m個輸入神經元是相對應於每一個高階參數值，而輸出層中的K個輸出神經元代表的是分群時所建立的K個群組，其會輸出一個介於0~1之間的值，以代表此測試樣本程式和K個群組之間的接近關係，其中1表示最接近且關聯性最高，0表示無關聯性。因此，在訓練歸類器時，我們是以用於繪圖應用程式分群時所使用的相同測試樣本程式中的高階繪圖參數值為輸入層之輸入，並以各個測試樣本程式所屬的候選群組關係當成輸出層之輸出來訓練歸類器。假設 $App_{vert}$ 已知是歸納於群組1，因此我們就將其所有的高階參數值輸入至歸類器，並以預期其為群組1(即輸出層Group 1的輸出值為1)的方式來訓練，同樣的假設 $App_{frag}$ 已知是歸納於群組2，因此我們就將其所有的高階參數值輸入至歸類器，並以預期其為群組2(即輸出層Group 2的輸出值為1)的方式來訓練，以依類推，輸入足夠數量的測試樣本後，即可完成歸類器的訓練。訓練完後產生相對應的類神經網路組態檔，此即為神經元與神經元間連線的權重值。

訓練階段三大步驟所使用的演算法及其所需之輸入資料，以及最終產生的輸出資料整理如表2所示。因此完成訓練階段後所輸出的(1)各主要元件的功耗方程式中每個高階參數的多項式係數值；(2) K個候選群組之各主要

元件的權重值；(3)類神經網路組態檔，即為針對某種特定繪圖處理器架構所建立的功耗預估模型，之後繪圖應用程式開發者只需進行功耗預估階段的工作，就可以直接觀察到這些高階參數對繪圖處理器功耗的影響。

表 2 各階段所需的演算法及其輸入/輸出

Algorithm	Required Input	Output
Polynomial regression	- Graphics parameters - Measured power Value	- Polynomials coefficients
K-means	- Measured power value and component weight	- Weight set of each group - Group assignment for each training sample
Neural network	- Graphics parameters - Group assignment for each training sample	- Configuration file of classifier

### 2.2.2 功耗預估階段 (Online Phase)

訓練階段完成功耗模型的建立後，接著只要使用待分析的繪圖應用程式之高階參數值，即可進行功耗預估。在此，我們使用ITRI所開發的OpenGL ES Interceptor，其可在繪圖應用程式執行過程中擷取出所有的OpenGL ES 2.0 API函式呼叫並儲存成一個API追蹤檔 (API Trace)，藉由分析API追蹤檔，即可得到每一幀畫面中事先選定的各個高階參數值，並利用這些高階參數值當成通用型繪圖處理器功耗模型的主要輸入。獲得了待分析的繪圖應用程式之各個高階繪圖參數值，以及在訓練階段中的第二步驟所訓練完成的歸類器後，將這些擷取出的高階參數輸入歸類器後，即可對此待分析的繪圖應用程式進行歸類，並從候選權重值集合中選用一組最合適的各主要元件之權重值來進行功耗預估。

當通用型繪圖處理器功耗模型所需的三個輸入值都準備就緒，即(1)待分析的繪圖應用程式之高階參數的數值；(2)所選用的候選群組之各主要元件的權重值；及(3)於訓練階段所推導出的多項式係數值，並將其代入功耗模型方程式(1)中，即可得出繪圖處理器的總體功耗值以及繪圖處理器中各主要元件的功耗值。

## 3. 以硬體事件計數器為基礎之繪圖處理器功耗模型

除了以高階參數來建立通用型繪圖處理器功耗模型外，我們也建立了另一種以硬體事件計數器為基礎之繪圖處理器功耗模型，其藉由內含在真實硬體或硬體模擬器中的事件計數器來進行功耗的評估，此種方法由於需實際存取硬體來獲得硬體狀態變動資訊，因此功耗模型的設計會相依於硬體架構，而無法快速移植至特定的硬體架構，雖然如此，但以硬體事件計數器為基礎之繪圖處理器功耗模型，可讓我們瞭解到，如果考量更底層的硬體資訊後，功耗預估的精確度是否可以進一步的提昇。本篇論文根據繪圖處理器繪圖管線中最耗能之區塊，挑選出10個硬體事件計數器建立出功耗模型，此10個硬體事件計數器對表1中的5個繪圖處理器主要元件的功耗影響整理如表3所示。

表 3 硬體事件計數器與各主要元件之關係

Component	Counter	Description
VS/FS	#1	執行過的指令個數
	#2	指令issue的次數
	#3	浮點數運算次數
	#4	CoALU運算次數
	#5	迭代抓取Pixel Values的次數
	#6	Common Store存取次數
TA	#7	經過的vertex個數
	#8	產生出的Tile Object個數
TU	#9	運算過的Texel個數
ISP	#10	運算過的Pixel個數

由表3可以得知，由於Vertex Shader (VS)/Fragment Shader (FS)是整個繪圖管線中功耗最高的元件，因此採用了6個硬體事件計數器來進行VS/FS這兩個主要元件的功耗預估，其餘的Tile Accelerator (TA), Texture Unit (TU), 以及Image Synthesis Processor (ISP)則分別採用1~2個硬體事件計數器來進行功耗預估，其功耗模型方程式如(2)所示。

$$P_{total} = \sum_{j \in \{components\}} \left( \sum_{i \in \{counters\}} \alpha_{i,j} * c_{i,j} + \beta_j \right) \dots (2)$$

其中的 $P_{total}$ 為總功耗值， $c_{i,j}$ 代表元件 $j$ 中的各個硬體事件計數器之數值， $\alpha_{i,j}$ 為各個硬體事件計數器每單位的數值對功耗之貢獻值， $\beta_j$ 則代表元件 $j$ 的基礎功耗。

在進行功耗模型訓練時，主要的輸入資料為：每個測試樣本程式所量測出來的功耗值以及硬體事件計數值，其中各個測試樣本程式之功耗值，可在真實發展板上新增探測電路並使用儀器進行量測來獲得，而硬體事件計數值，可藉由在高階硬體模擬器上執行測試樣本程式的方式來獲得。因此，功耗模型的建立是以測試樣本程式之硬體事件計數值以及各主要元件之功耗量測值配合線性回歸(Multiple Linear Regression, MLR)的方式，來訓練出連結硬體事件計數值與功耗之間關係的多項式係數值。利用上述觀念為基礎，整個繪圖處理器分成5個主要元件區塊，分別為VS、FS、TA、TU、ISP，以各別元件的功耗值對上會影響此元件的多個硬體事件計數值分別作線性回歸計算，並以最小平方方法求出各別元件相對應的多項式係數值 $\alpha_{i,j}$ 及基礎功耗 $\beta_j$ ，如方程式(3)所示：

$$\min \sum_i (Y_i - \beta - \alpha_1 c_{i,1} - \dots - \alpha_p c_{i,p})^2 \dots (3)$$

其中 $Y_i$ 為第 $i$ 個測試樣本程式的功耗量測值， $c_{i,p}$ 為第 $i$ 個測試樣本程式的第 $p$ 個硬體事件計數值，以線性回歸方式建立功耗模型的方法為使用硬體事件計數值作為自變數，並以功耗量測值作為應變數，再來以最小平方方法求得每個硬體事件計數器的功耗貢獻值(即為多項式係數值 $\alpha_{i,j}$ )及基礎功耗值(即為 $\beta_j$ )。如此即完成以硬體事件計數器為基礎的繪圖處理器功耗模型之建立。

## 4 · 實驗結果

### 4.1 實驗環境設定

為了驗證本論文所提出之通用型繪圖處理器功耗模型，因此我們選定的目標繪圖處理器硬體架構為採用TBDR(Tile Based Deferred Rendering)繪圖管線架構[4]之PowerVR SGX540繪圖處理器，其內建於Panda Board ES Rev B3發展板中，軟體發展環境則是建立在

Linux kernel 3.4.0-1491-omap4 及 Ubuntu 12.04.5 LTS，並採用OpenGL ES 2.0及OpenGL ES GLSL ES 1.0 為此環境上的繪圖應用程式介面(Graphics API)。

#### 4.1.1 繪圖處理器功耗量測

首先要介紹繪圖處理器功耗值的量測方式。我們是採用實際電路量測的方式來獲得發展板上繪圖處理器的功耗值，為了達此目的，我們在繪圖處理器核心之電壓輸入端加入了內含電流放大器的探測電路，以得到繪圖處理器之電流及電壓值；此外為了將電流及電壓值，可以和繪圖應用程式中的每幀畫面繪製的開始及結束進行時間上的同步，每當繪圖應用程式要開始或結束一幀畫面的繪製時，我們會透過Panda Board ES Rev B3發展板上的一般用途輸出入腳位(General Purpose Input/Output, GPIO)，來傳送每幀畫面繪製的開始及結束的訊號，以將此訊息傳送至發展板上的探測電路中，並透過National Instrument之DAQ進行量測，以獲得繪圖處理器電流及電壓值，並推算出相對應的繪圖處理器功耗值，繪圖處理器功耗量測的實驗環境如圖4所示。

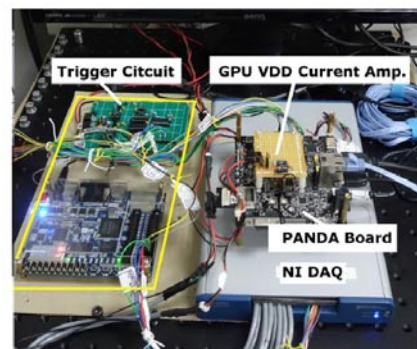


圖 4 繪圖處理器功耗量測實驗環境

繪圖處理器功耗之實際量測結果如圖5所示，在此範例中我們進行兩幀畫面的功耗量測，紅色線標示出來的地方表示一幀畫面開始繪製，綠色線標示出來的地方表示一幀畫面結束繪製，由於藉由GPIO來傳送每幀畫面繪製的開始/結束之同步訊號，因此，在此環境上也可以量測出每幀畫面的繪製時間，從圖5可看到當繪圖處理器開始進行繪製時，其功耗值約550~750 mW；而當繪圖處理器閒置時，其功耗值約500 mW。必須注意的是：每個測試樣本程

式會執行1000幀畫面的繪製，並對這1000幀畫面所量測出的繪圖處理器功耗值取平均，以當成各個測試樣本程式之功耗值來進行功耗模型的訓練。

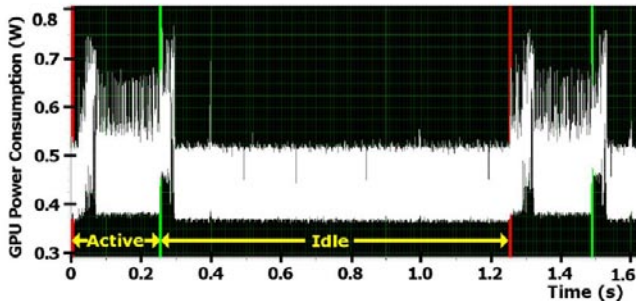


圖 5 繪圖處理器功耗量測結果

#### 4.1.2 功耗模型訓練用之測試樣本程式

為了訓練出各主要元件的功耗方程式中每個高階參數的多項式係數值以及一系列候選群組之各主要元件的權重值，我們使用客製化的OpenGL ES測試樣本程式來進行功耗模型的訓練。這些測試樣本程式涵蓋了在一個場景中使用簡單的3D物件(如Box的繪製)，到使用複雜的3D物件(如Fractals)。為了分析各個高階參數對功耗值的影響，我們以一次只變動一個高階繪圖參數的方式來客製化這些測試樣本程式。本論文中使用如表1中的6個高階參數來建立功耗模型，因此測試樣本程式會針對頂點個數、貼圖個數、貼圖過濾模式、打光及陰影模式、物件被遮蔽率、以及畫面解析度來各別作調變。以頂點個數為例，即是以不同的頂點數目來構成BOX；以畫面解析度為例，則是以不同解析度的方式來輸出BOX的繪製。於訓練階段所使用之客製化OpenGL ES測試樣本程式及其輸出結果如圖6所示。

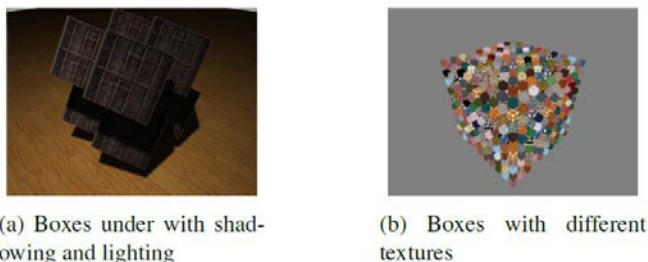


圖 6 OpenGL ES測試樣本程式

## 4.2 實驗結果與比較

本論文以選定5個繪圖處理器主要元件，6個高階參數，以及將測試樣本程式的各主要元件之權重值群聚成4個候選群組的方式，來進行功耗模型的建立與驗證。此6個高階參數對這5個主要元件的功耗影響整理如表1所示。

首先於訓練階段藉由在Panda Board上實際量測出的功耗值以及ITRI OpenGL ES Interceptor所擷取出的6個高階參數值(或是在高階硬體模擬器上，執行測試樣本程式所獲得的10個硬體事件計數值)，來進行功耗模型的訓練；進入功耗預估階段只要將待分析繪圖應用程式的6個高階參數值 $c_i$ ，歸類器選用的各主要元件權重值 $W_j$ 及多項式係數 $\alpha_i$ 代入功耗模型方程式(1)中，即能預估出此待分析繪圖應用程式所產生的平均功耗值。本論文使用glmark2-es2測試程式來驗證功耗模型的精確度，其一部分的內容及輸出結果如圖7所示。

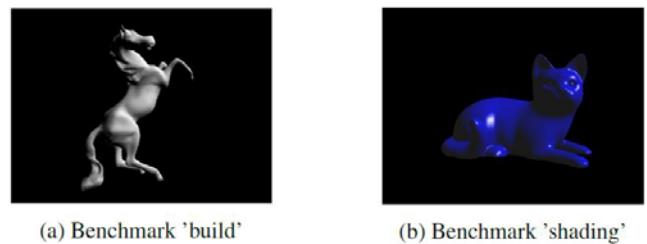


圖 7 glmark2-es2內容及其輸出結果

圖8顯示glmark2-es2中的7個不同場景之功耗量測值與功耗預估值，其中的Measured表示功耗量測值；Structure-Agnostic表示使用通用型繪圖處理器功耗模型的預估結果；Counter-Based表示使用以硬體事件計數器為基礎之繪圖處理器功耗模型的預估結果。圖8中的值皆代表繪製一幀畫面的平均功耗(單位：mW)。可以觀察到本論文所提出的Structure-Agnostic方式其功耗預估值皆逼近實際的功耗量測值，但Counter-Based方式則會有低估功耗的趨勢。低估的現象可歸因於：本篇論文只針對5個繪圖處理器主要元件(VS, FS, TA, TU, ISP)來進行功耗模型的建立，但沒有考量所有的硬體元件；或是沒有使用足夠多的高階參數/硬體事件計數器。基本上我們所挑選出

來的這5個主要元件，其平均約代表80%的繪圖處理器功耗值，雖然已涵蓋到大部分的功耗值來源，但如果要進一步提高精確度則可以考量更多的硬體元件。

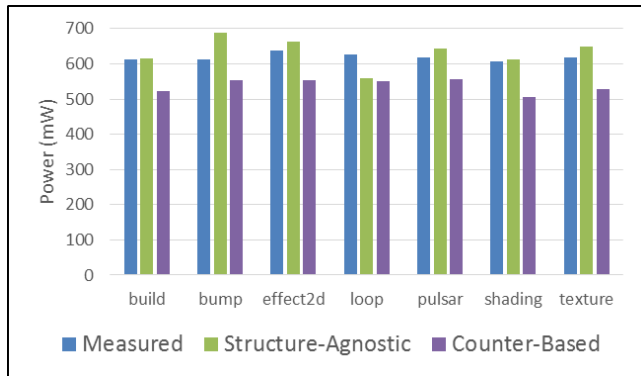


圖 8 glmark2-es2 功耗預估結果比較

表4進一步的比較功耗模型的精確度，可以看到 Structure-Agnostic 方式的精確度介於 88% ~ 99%，並達到 95% 的平均精確度；而 Counter-Based 方式的精確度則介於 83% ~ 90%，並擁有 87% 的平均精確度。Structure-Agnostic 方式之所以會比 Counter-Based 方式有更高的精確度，主要是歸因於動態權重的使用，其對預分析的程式進行歸類，並動態選用最合適的權重值來進行預估，以權重值來調整各主要元件的功耗預估值，來適應繪圖管線的負載不平衡情況，進一步提高預估準確度。

表 4 功耗模型精確度比較

Models Patterns	Structure-Agnostic	Counter-Based
build	99%	86%
bump	88%	90%
effect2d	96%	87%
loop	89%	88%
pulsar	96%	90%
shading	99%	83%
texture	95%	86%
Average	95%	87%

相較於文獻[5]其同樣針對繪圖管線進行功耗預估並於 Panda Board 進行精確度驗證，然而

其只能達到 92% 的平均精確度而略低於本論文所提出的 Structure-Agnostic 功耗模型，另一方面文獻[1]使用了和本篇論文近似的群組歸類方式來進行一般用途繪圖處理器 ( GPGPU ) 的功耗預估，然而其精確度僅介於 85% ~ 90%，同樣低於本篇論文所提出的方式。Structure-Agnostic 功耗模型除了擁有精確度外，由於不需透過模擬即可進行功耗預估，因此其執行時間相當快速。於功耗預估階段只需使用低於 2 秒的執行時間，然而，若是使用將功耗模型加入高階硬體模擬器的方式，來進行則需耗費數分鐘的執行時間；而使用商用功耗模擬器 ( 如 PrimeTime PX ) 配合閘層級的硬體電路方式來獲得功耗值更需耗費數小時，且這兩種方式的執行時間會隨著測試程式複雜度的提高而增加。

## 5 . 結論

本論文提出一種以可藉由繪圖 API 之執行記錄，來推導功耗的通用型繪圖處理器功耗模型建立技術。由於使用了繪圖應用程式開發者所熟悉的高階繪圖參數來建立功耗模型，因此可以在不需對繪圖處理器架構有相當認知程度的情況下，就可以很輕易的瞭解到自己設計的程式碼或是 3D 場景對功耗所造成的影響。此外，由於使用了不相依於硬體架構的高階繪圖參數來進行功耗預估，因此本功耗模型可輕易的被移植至不同繪圖處理器架構。

透過實驗我們驗證了單純藉由分析繪圖 API 之執行記錄來推導功耗，仍然可以具有 88% ~ 99% 的高精確度表現；同時在不需透過模擬的情況下，只需使用數秒的執行時間即可完成功耗預估，使程式開發者可即時獲得不同 3D 場景的功耗變化趨勢，以利單位功耗之效能最佳化的進行。未來預計分析其它的高階繪圖參數對功耗的影響，並擴充本功耗模型的概念使其能相容於其它種類型的 API，如 GPGPU 所使用的 OpenCL，DSP 所使用的 OpenAL，或是 DirectX 繪圖 API，以達到通用的目標。

## 參考文獻

- [1] G. Wu, J. N. Greathouse, A. Lyashevsky, N. Jayasena, and D. Chiou, "GPGPU performance and power estimation using



machine learning,” IEEE 21st International Symposium on High Performance Computer Architecture USA, pp. 564–576, February 2015.

- [2] C. Luo, and R. Suda “A performance and energy consumption analytical model for GPU”IEEE Ninth International Conference on Dependable Autonomic and Secure Computing Australia, pp. 658–665, December 2011.
- [3] J. Lucas, S. Lal, M. Andersch, M. Alvarez-Mesa, and B. Juurlink “How a single chip causes massive power bills GPUSim-Pow: a GPGPU power simulator,” IEEE International Symposium on Performance Analysis of Systems and Software USA, vol. 1, pp. 97–106, April 2013.
- [4] Imagination Technologies Limited “PowerVR hardware architecture overview for developers ,” March 2015.
- [5] C. H. Huang, Y. A. Chung, P. S. Huang and S. L. Tsao “High level energy consumption model of embedded graphic processors,” IEEE International Conference on Digital Signal Processing Singapore, pp. 105–109, July 2015.

黃迅倫



現任職於工研院資通所系統架構設計部技術副理。國立成功大學電機工程碩士。專長於電子系統層級設計流程開發。

田胡治之



現任職於工研院資通所系統架構設計部。專長於低功耗系統晶片架構研究。同時擁有多年的微處理器與系統晶片設計經驗並任職於國內外大廠的豐富經歷。

盧俊銘



現任職於工研院資通所嵌入式系統與晶片技術組副組長。聘兼系統架構設計部經理。國立清華大學電機工程研究所碩士。專長為系統架構設計、超大型積體電路設計與測試。

## 作者簡介

賴俊宏



現任職於工研院資通所系統架構設計部。國立中山大學資訊工程博士。專長於軟硬體協同設計、繪圖處理器架構設計及功耗分析。

馬明達



現任職於工研院資通所系統架構設計部。專長於繪圖處理器架構設計與嵌入式系統。目前從事繪圖處理器功耗分析及架構最佳化。