

不同的壓縮技術和不同的圖形格式之比較

一、摘要

以前的人類們在記錄訊息的時候會用石板雕刻訊息在上面、或者是用口耳相傳的方式來傳遞訊息，到了現代科技進步發展快速，現在的生活中我們會能夠使用圖片、影像的方式來記錄訊息，這些東西大部分都需要軟體方面以及硬體方面的結合，這些圖片或是影像在設備中大多數都會以數位訊號的方式來保存，保存的手段還能分為各式各樣的格式，除了我們常見常聽到的 JPEG、PNG 和 GIF 等等，如今還有越來越多焦點聚集在 AVIF 上面。

需要關注這些格式的結構中支援的功能像是 PNG 支援 alpha channel 透明通道等，在這些功能中可壓縮又能保持影像訊號的品質是最為重要的，這些格式的結構直接影響到壓縮能力，像是 JPEG 支援有損壓縮，壓縮是透過資料一致性來達到壓縮的目的。透過兩大類無損和有損壓縮的相關技術簡介和實作其中的技術驗證其特性。

根據一系列比較結果發現興新的 AVIF 支援功能的完整性在現在因網速及螢幕品質上升，動不動就要求及高畫質的時代的確更具有未來性，但是受限於普及性不佳。

二、前言

我們在日常生活中常常會接觸到許多不同格式的資料，其中圖片資料常的儲存格式有 JPEG(Joint Photographic Experts Group)，此外還有常見的 PNG(Portable Network Graphics)，還有一個也是很常見的格式 GIF(Graphics Interchange Format)格式，他們彼此之間的關係、差異、實用性、功能和可壓縮性之間有甚大的差異，透過在丁建均老師的高等數位訊號處理的課程中了解到壓縮的原理並且也了解到 JPEG 格式中的 4:2:0 的壓縮技術，但是其中有些課堂上提到過的技術像是 Huffman code，雖然了解其運作原理但並沒有實作的經驗，因此希望能 Matlab 作為工具實作壓縮技術並加以討論結果。除此之外希望能解析未來的興新格式 AVIF(AV1 Image File Format)又有哪些優勢。這些圖形格式都是常見的格式，壓縮方法也是有許多廣為人知的技術，我希望我能透過此書面報告同時當作一個 final project 來重新實作後相信我能對於各個格式之間的定位有著更深一層的了解。

三、 圖片格式

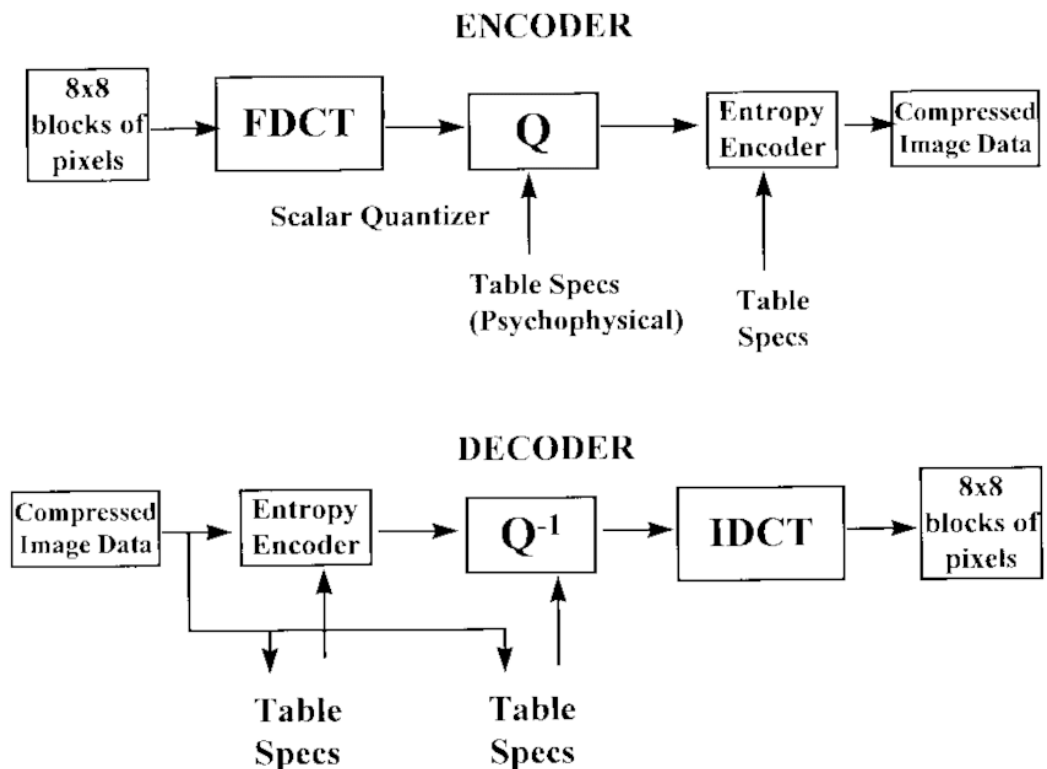
(一) JPEG

在 RGB 顏色中各有 8 個 bits 可以儲存，所以每個通道能記錄的數值都能夠介在 0~255 之間，形成(0~255,0~255,0~255)共 $3 \times 8 = 24$ 個 bits 來儲存的空間。JPEG[1]是 1992 年被提出的儲存格式，是目前最為廣泛使用來儲存圖片的格式之一，常常在網路上、智慧型手機或是數位相機拍照的時候圖片格式，是一種有損壓縮的方法。其中有一些新的技術提出，如 JPEG2000，其原理是利用離散小波轉換為基礎來壓縮圖片的方法，成為了新一代的 JPEG 的格式，可以提供比原本 JPEG 演算法更高品質的圖片。

[2]

圖一為 JPEG 格式常用的壓縮技術的流程圖[3]，流程如下：

1. 將圖片進行 8 by 8 的分割
2. 進行 Discrete Cosine Transform(DCT)
3. 量子化
4. 熵編碼



圖一、JPEG 壓縮和解壓縮流程

(二) GIF

此格式是早期在網路上廣泛使用的點陣圖(raster graphics)，最早是由 CompuServer 公司定義其標準，能以高解析度和品質被不同硬體使用。此格式的索引顏色共有 256 種即為 8 位元。曾經因為使用的 LWZ 壓縮法

專利問題而被 PNG 取代，最特別的是因為可以同時儲存多影格，出現許多 GIF 的動圖。下方的圖是 GIF 的檔案結構主要能分成三大區塊。除此之外較為特別的是擁有 Global Color Table 作為預設 RGB 色彩表。

1. Header 和其他資訊

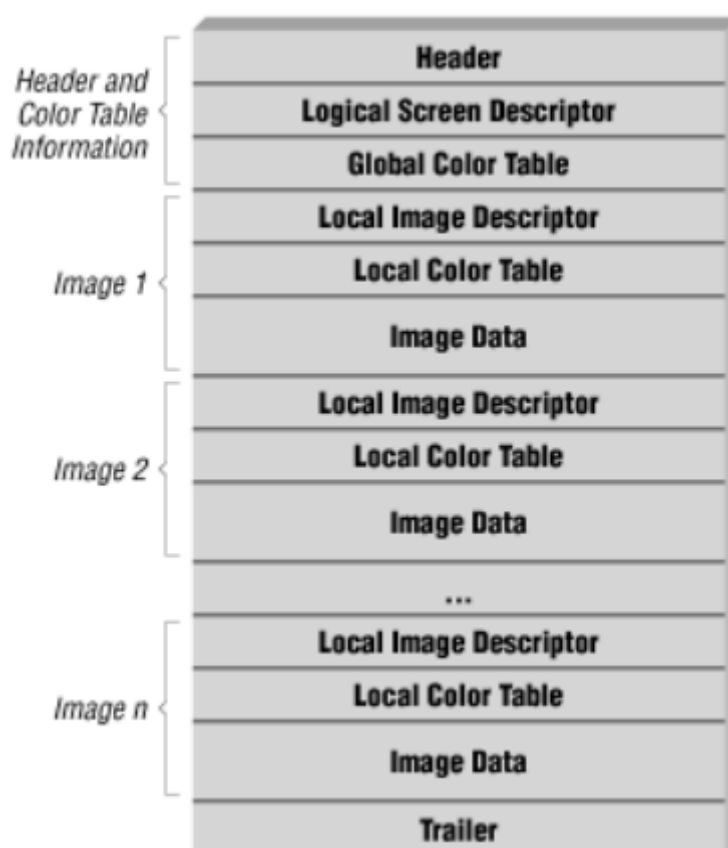
包含了 GIF 的署名(Signature)確認其為 GIF 檔、版本資訊、Logical Screen Descriptor 和 Global Color Table

2.影像資訊

包含 Local Color Table 等等

3.Trailer

結束標誌



圖二、GIF 結構

(三) PNG

是一種可無損壓縮的點陣圖格式，最開始是為了抵抗 GIF 格式的專利而產生，希望能取代 GIF 格式的功能，同時也包含了許多 GIF 未支援的功能。

GIF 未支援功能

1. 每像素的真彩影像高達 48 位元
2. 每像素的灰階高達 16 位元
3. 支援 alpha channel 表示透明程度

4. 包含 gamma information 與影像中輝度(luminance)與三色刺激值(tristimulus values)有關[4]

(四) 如何選擇要使用哪種影像格式？

透過以上 JPG、PNG、GIF 的大致上的介紹，我們會發現在使用不同個格式上我們會有不同的優缺點，那我們到底該要選擇使用哪種影像格式呢？

大致上可以把儲存圖片的格式有以下的分類：

JPG：用在尺寸較大的照片，以減少文件大小。

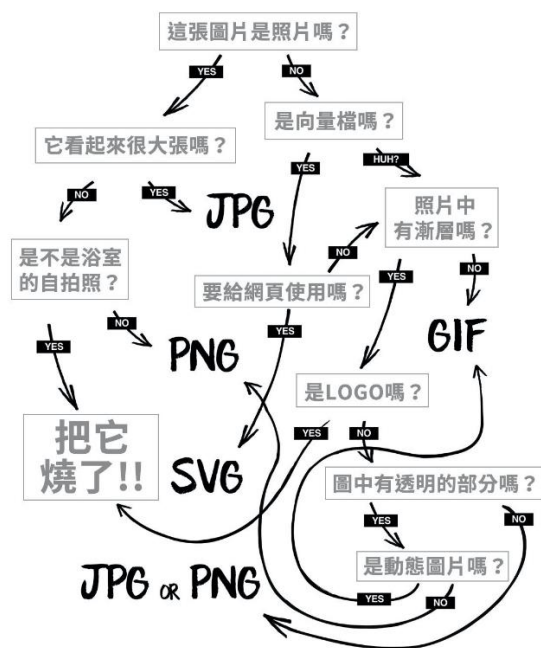
PNG：有效減少圖片檔案大小以及保留照片透明元素。

SVG：向量檔，縮放不失真。

GIF：可以製作小型動態圖片檔案。

而在選擇要用哪種影像格式來儲存的時候我們可以依據圖三[5]來做判斷

JPG ● PNG ● GIF ● SVG

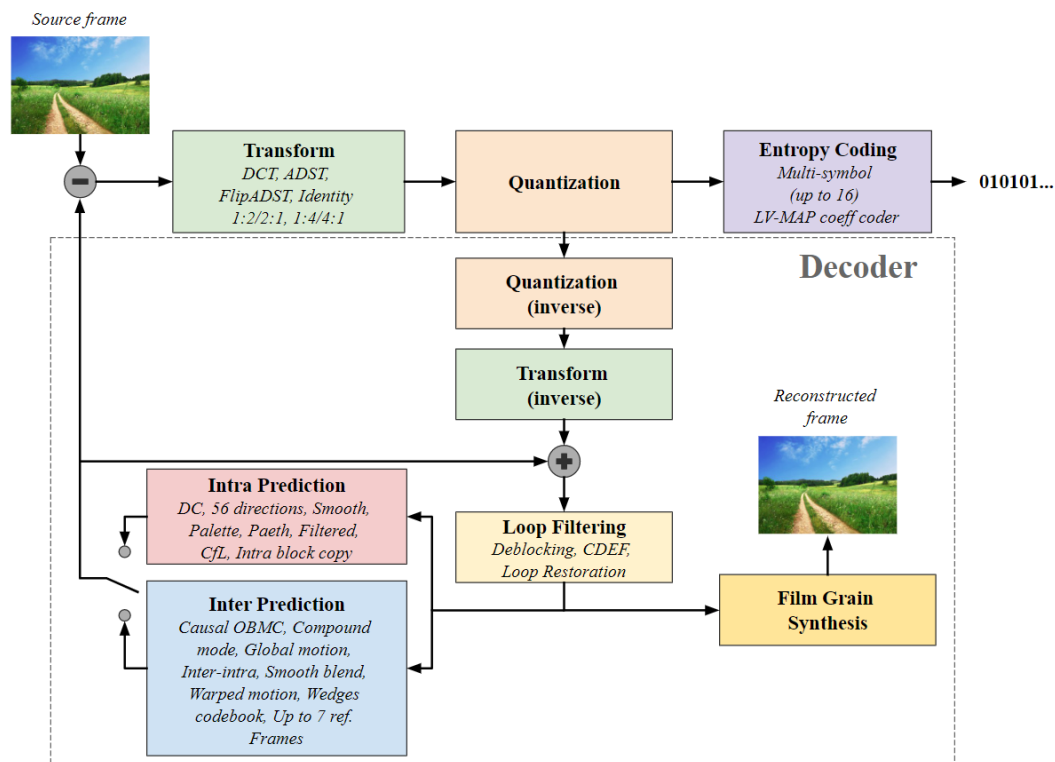


圖三、影像格式選擇流程圖[5]

(五) AVIF

而在介紹以上幾種算是已經歷史悠久的圖片格式，接下來要介紹的是一種在 2018 年由各個巨頭(Netflix、Google、Facebook、微軟、三星、華為、騰訊)組成的開發媒體聯盟所開發出來的一種儲存演算法，此格式甚至還被定義為下一世代圖片格式，是在 HEIF(High Efficiency Image File Format)下利用 AV1 技術壓縮。AV1 壓縮技術順序是先分區將圖形分成小

圖 128 乘 128 最小到 4 乘 4 像素分區的方式相較其他技術更加靈活、轉換如 DCT 之後量子化再熵編碼再解碼的部分加入的預測的步驟包含了多影像間的預測跟單影像自己內部的預測例如運動預測值或是預測角度範圍等技術，如下圖所示。[6]



圖四、AV1 壓縮和解壓縮流程

HEIF 是由 MPEG 所定義，較特別的是可以儲存連續影像，其結構大致如下圖所示。

主要結構分為四層：

1. ftyp

描述文件類型和相容性

2. moov

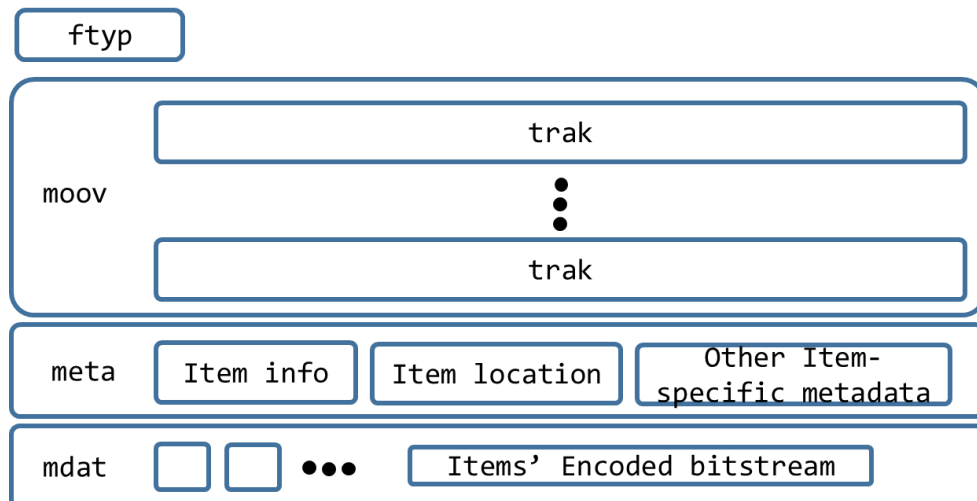
可包含多個 track 儲存影像序列

3. meta

包含影像的資訊如位置

4. mdat

影像存放的位置，有些技術會將原本的一張在此部分切割成小圖儲存增加平行運算的速度和節省記憶體並且更好裁切。



圖五、HEIF 結構

四、壓縮方法

圖片為什麼要壓縮?如同老師上課所提到的如果不使用一些壓縮方法的話，我們儲存一張原始的圖片可能會需要幾 10MB 的儲存空間，單單一張圖片就需要花費 10MB 的儲存空間，聽起來將會是一件蠻恐怖的事情。而壓縮的方法如同老師上課所提到過的一樣，可以分成兩類:無損壓縮及有損壓縮，無損壓縮代表資料能夠不失真還原成原本的樣子，而有損壓縮則代表資料在還原後會有一些資料的缺失。

(一) 無損壓縮

● DCT

離散餘弦變換是壓縮圖形中 JPEG 用的一種變換，透過這樣的變換可以將圖形中的像素轉換到頻域，可以將其中人眼部敏感的區域加以去除，其中的 8*8 公式如下:

$$F_{(u,v)} = \frac{1}{4} C_{(u)} C_{(v)} \left[\sum_{x=0}^7 \sum_{y=0}^7 f_{(x,y)} * \cos \frac{(2x+1)ux}{16} \cos \frac{(2x+1)v\pi}{16} \right]$$

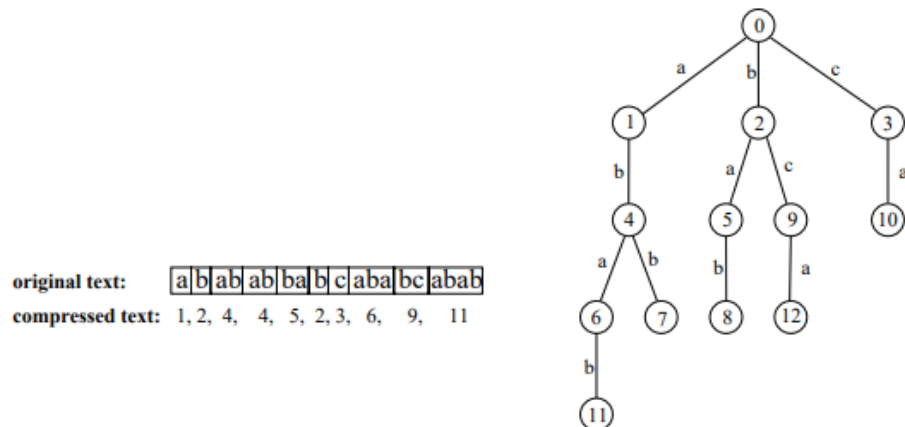
$$f_{(x,y)} = \frac{1}{4} \left[\sum_{u=0}^7 \sum_{v=0}^7 C_{(u)} C_{(v)} F_{(u,v)} \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2x+1)v\pi}{16} \right]$$

$$\text{where: } C_{(u)}, C_{(v)} = \frac{1}{\sqrt{2}} \text{ for } u, v = 0$$

$$C_{(u)}, C_{(v)} = 1 \text{ otherwise}$$

- LZW

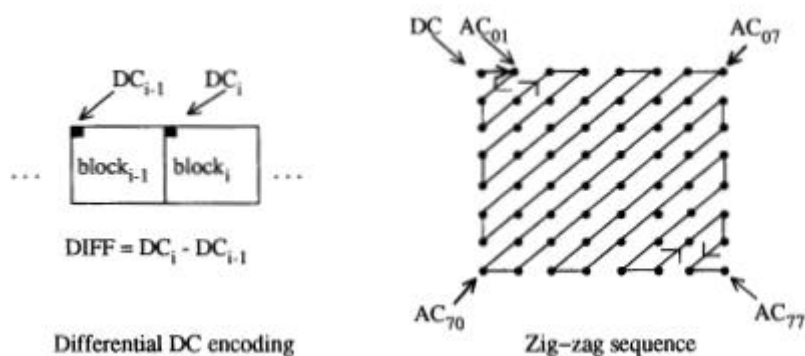
是一種建立類似字典樹的概念來進行編碼的演算法，第一步驟是需要先建立字典，將每一個單獨的字用一編碼表示，這樣重複的字就可以用相同的編碼來代替。如下方的圖，代表建立字典的過程，這個演算法的更進一步的變形有常見的 LZ77 或是 LZ78，但是因為並未如下方提到的 Huffman code，對資料進行分析所以並不算是最節省空間的作法，在 GIF 壓縮中常常使用。



圖六、LZW 編碼示意圖

- Zig-zag sequence

在 JPEG 中經過 DCT 和量子化後會產生一矩陣，其中的數值又可以分成較大和較小的數值，較大的數值被稱之為 DC 係數，若有多個 DC 係數是利用差分編碼進行儲存，如下圖中的左圖。較小的數值稱之為 AC 係數，兩者皆會利用 Z 字型排列的方式進行儲存，如下圖中的右圖，這樣的好處是會先儲存到低頻的信號，而高頻的信號大多已在前面的壓縮處理中被刪除因人眼對於高頻訊號的分辨能力有限，所以後方高頻區域的數值多為零，只需要利用 EOB(end of block)的符號加以標示代表後方值皆為零。



圖七、DC 係數差分編碼和 AC 係數 Z 字排列

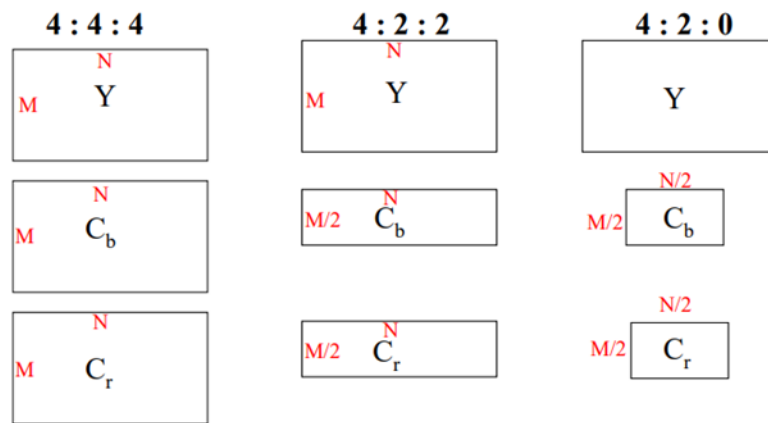
- Huffman code

是一種熵編碼演算法，第一個步驟是先將訊號中所有可能性的出現機率進行排列，之後第二步驟是建立案出現機率排序的字典稱為 Huffman tree，原則是越常出現的訊號用越短的編碼代替達到減少整體訊號平均長度和期望值，以二進位制觀察會發現訊號被壓縮，且能依照前面建立的字典無損恢復訊號。在前面的 JPEG 壓縮步驟第四步中有提到，此 Huffman code 可以當作這一步的其中一種方法，但不僅限於此種編碼。

(二)有損壓縮

- 4:2:0

這個是利用空間上的一致性進行壓縮，先將影像的 RGB 像素轉為 YCbCr 再利用人眼對彩度不敏感的特性將減少 Cb 和 Cr 的取樣率達到壓縮的目的，但是會對訊號造成損耗，如下圖最右側 4:2:0 的部分。



圖八、4:2:0 示意圖

- 量子化

在 JPEG 格式中是根據量子化表將已 DCT 過後的矩陣除以量子化表上的相對應位置，解量子化則是反過來乘以量子化表對應位置，因為對於分整數有四捨五入會造成有損的壓縮結果。

可以參考以下的公式：

$$F_{(u,v)}^Q = \text{Integer Round}\left(\frac{F_{(u,v)}}{Q_{(u,v)}}\right)$$

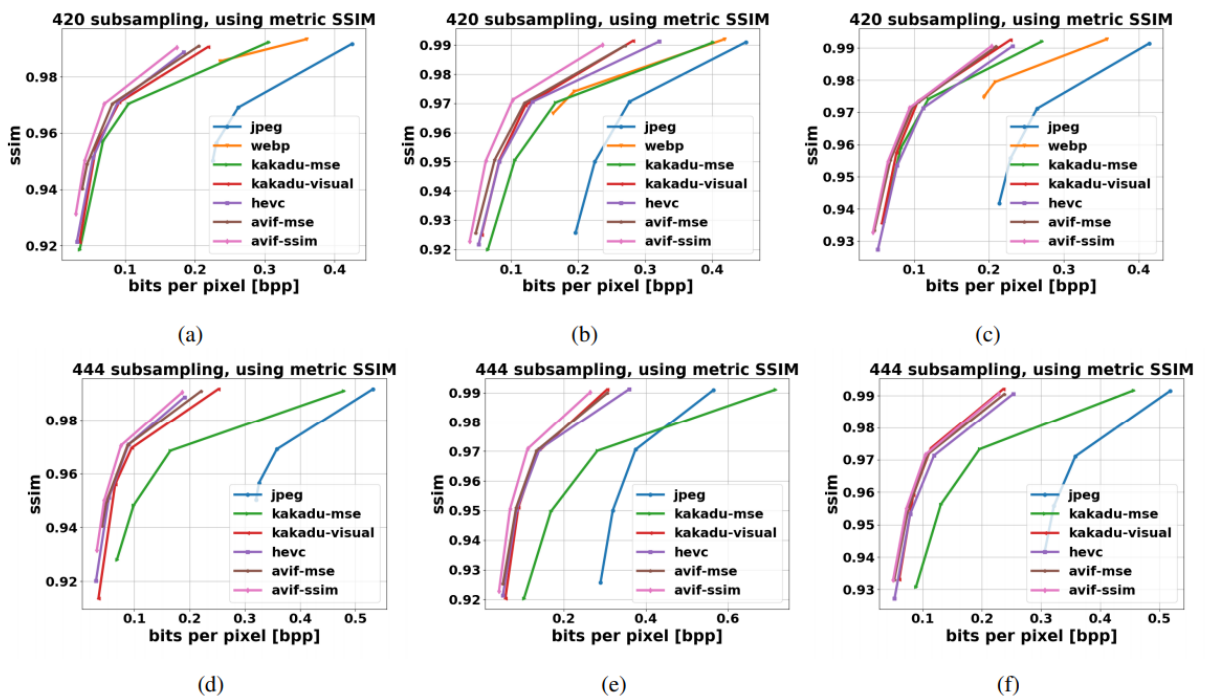
$$F_{(u,v)}^{Q'} = F_{(u,v)}^Q * Q_{(u,v)}$$

五、 討論

在以上提到的格式中發現 PNG 和 GIF 都不支援有損壓縮，下方表格是 Overview of the High Efficiency Image File Format 報告中整理個格式之間的比較，其中的 heic 就是指前面提到的 HEIF 格式其中一種擴展格式。可以發現多數傳統的影像格式除了 GIF 外，對於多圖片都沒有支援；在在延伸至其他編碼程式和縮圖的欄位傳統影像格式則是除了 JPEG 外都沒有此功能；在透明度上除傳統影像格式則是除了 PNG 外都沒有。在這些比較下可以清楚發現能有效壓縮並支援各種功能的下一代格式的重要性，但是這些 HEIF 格式雖然功能強大，有一個致命的缺點就是普及性的問題，需要大多數的硬體及軟體設備都可兼容格式的實用性才會上升。

	.heic	JPEG/Exif	PNG	GIF (89a)	WebP	JPEG-XR / TIFF	JPEG-XR / JPX	BPG
Formats and extensibility								
Base container file format	ISOBMFF	TIFF	-	-	RIFF	TIFF	- ⁴	-
Lossy compression	Yes (HEVC)	Yes (JPEG)	No	No	Yes (VP8)	Yes	Yes	Yes (HEVC ¹⁰)
Lossless compression	Yes (HEVC)	Yes (TIFF Rev 6.0)	Yes (PNG) ¹	Yes (GIF) ¹	Yes (VP8L)	Yes	Yes	Yes (HEVC ¹⁰)
Extensible to other coding formats	Yes	Yes ⁸	No	No	No	Yes ⁸	Yes ⁹	No
Metadata format (on top of internal)	Exif, XMP, MPEG-7	Exif	-	-	Exif, XMP	Exif, XMP	JPX, (XMP) ⁶	Exif, XMP
Extensible to other metadata formats	Yes	No	No	No	No	No	Yes (XML-based)	Yes
Other media types (audio, text, etc.)	Yes	Audio ²	No	No	No	No	Yes ⁷	No
Multi-picture features								
Multiple images in the same file	Yes	No ¹¹	No	Yes ³	Yes ³	No	Yes	Yes ⁹
Image sequences / animations	Yes	No	No	Yes	Yes	No	Yes	Yes
Inter coding	Yes	No	No	No	No	No	No	Yes
Derived images								
Multiple-of-90-degree rotations	Yes	Yes	No	No	No	Yes	Yes	No
Cropping	Yes	No	No	No	No	No	Yes	No
Tiling/overlaying	Yes	No	No	No	Yes	No	Yes	No
Extensible to other editing operations	Yes	No	No	No	No	No	No	No
Auxiliary picture information								
Transparency (alpha plane)	Yes	No	Yes	No ¹²	Yes	Yes	Yes	Yes
Thumbnail image	Yes	Yes	No	No	No	Yes	Yes	Yes

圖九、圖形格式比較表



圖十、每像素的位元(bits per pixel, bpp)對應到的 SSIM 數值表現

因為表格並未比較 AVIF，其與 HEIF 的差異並非在功能上，而是在於使用的編解碼器是 AV1 是 Google 等多個公司聯盟旗下免費的工具，所以更有可能普及。上圖直接比較了 AVIF(棕最最小化 MSE、桃紅最大化 SSIM)、JPEG(藍)、JPEG2000(使用 kakadu 軟體，綠、紅)和 HEVC 各自在不同資料庫下每像素的位元(bits per pixel, bpp)對應到的 SSIM 數值表現，可反應出人眼看到的相似程度。從左到右分別是哪三個不同的圖庫第一個圖庫是有關於高解析度的自然圖檔，第二個圖庫是遊戲畫面相關的圖，最後則是從一種透過自動生成影片的資料庫 BVI-SynTex。上下則是分別採用 4:2:0 跟 4:4:4 色度採樣，可以發現 AVIF 格式的斜率是相當大，代表較少的 bpp 就能有不錯的 SSIM，代表其壓縮程度應該可以更大仍能保留一定便是程度的影像，但是對於各格式間的壓縮速度卻需要更進一步比較。

六、 結論

這些不同的格式適用的場景並不相同，因為彼此支援的功能不同，雖然能支援許多功能的圖形格式可以有更多應用的空間但是可能會儲存過多資訊會造成圖形越大，同時也會對於可壓縮性造成影響。在這之中格式定義的結構顯得更為重要，因為其定義不同的區塊會影響到他的兼容性，例如 GIF 就因為其結構得以出現動圖的形式，而 JPEG 特性則是能支援有損壓縮，透過一連串的處理，將人眼無法辨識的訊號去除或是將人眼辨識不敏感的彩度降迪取樣率達到利用資料空間上或是頻譜一致性來壓縮影像。PNG 是因 GIF 壓縮技術出現專利而出現較新的格式，和 GIF 很相似只支援無損壓縮，但是卻擁有更多功能，如更大的真彩空間和支援透明的 alpha channel 等 GIF 不支援的功能。

在畫質要求不斷上升的現代，興新的影像格式需求也日益上升，許多大公司也互相結盟希望能讓創造出的圖形格式成為新一代的標準，但是回過頭發現舊的格式之所以普及正是因為其壓縮和解壓縮技術免費，除此之謂需要有優秀的軟硬體兼容性，AVIF 格式就是在這樣的環境下應運而生，除了結合 Google 和蘋果等多家大公司的 AOMedia 聯盟其下免費的編碼和解碼技術其儲存結構採用 MPEG 定義的 HEIF 一樣的結構，支援多種影像儲存功能，如多影像的儲存或是壓縮，正因為免費和兼容的結合使得 AVIF 成為新一代影像格式的可能性增加。

分析 AVIF 的性能發現能以極低的 bbp 保持較高的 SSIM 在自然、遊戲或是合成影像都表現出明顯優於 JPEG 或是 JPEG 2000 的結果，但是對於壓縮的速率還需要更進一步的比較效率。

多種壓縮技術中除了 4:2:0 在作業三中被實作，其餘病媒又進一步測試，在實作了 Huffman 編碼後發現 JPEG 壓縮順序相當重要，按順序壓說能得到相當不錯的壓縮率，但是若僅對單一技術加以實作可能無法有效壓縮，但是經過 Huffman 編碼並解碼的影像的確並沒有發生損害。

七、 參考資料

1. Wallace, G.K., *The JPEG still picture compression standard*. IEEE Transactions on Consumer Electronics, 1992. **38**(1): p. xviii-xxxiv.
2. Skodras, A., C. Christopoulos, and T. Ebrahimi, *The JPEG 2000 still image compression standard*. IEEE Signal processing magazine, 2001. **18**(5): p. 36-58.
3. Haskell, B., et al., *Image and video coding - Emerging standards and beyond*. Circuits and Systems for Video Technology, IEEE Transactions on, 1998. **8**: p. 814-837.
4. Boutell, T., *Png (portable network graphics) specification version 1.0*. 1997.
5. 影像格式選擇流程圖. Available from:
<https://www.cool3c.com/article/146971>.
6. Barman, N. and M.G. Martini. *An evaluation of the next-generation image coding standard AVIF*. in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. 2020. IEEE.
7. "AV1," Wikipedia. Jun. 25, 2021. Accessed: Jul. 01, 2021. [Online]. Available:
<https://en.wikipedia.org/w/index.php?title=AV1&oldid=1030398853>
8. "AV1 Bitstream & Decoding Process Specification," p. 681.
9. "AV1 Image File Format (AVIF)." <https://aomediacodec.github.io/av1-avif/#biblio-heif> (accessed Jul. 01, 2021)
10. D. Jian-Jiun, "Advanced Digital Signal Processing," 2021