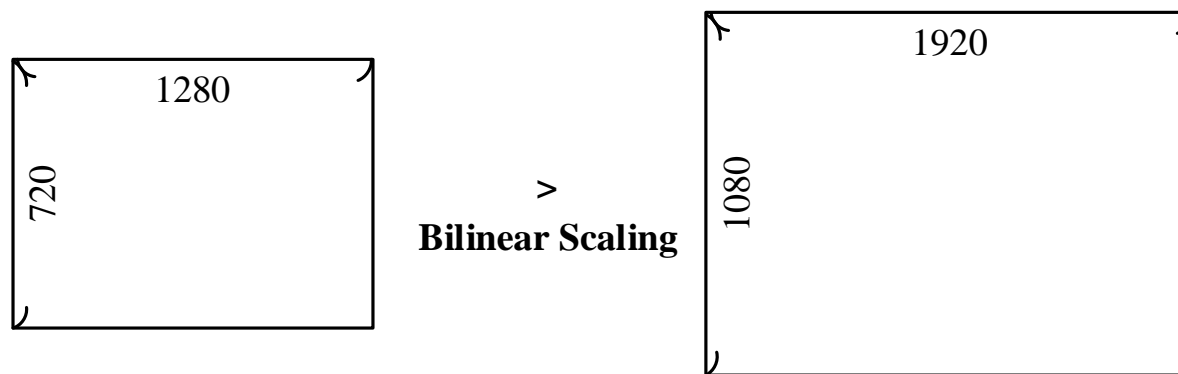


LAB12

Bilinear Scaling

Introduction

- 影像縮放(Image Scaling)是影像處理最常見的任務之一。不管是相片預覽、影像串流調整畫質、或者驅動不同顯示面板以及作為現在深度學習的前處理...等等。當我們將影像從一個解析度縮放到另一個解析度時，最核心的問題在於「如何在新的像素位置上，根據原始像素值推估出合理的顏色值」，也就是所謂的**插值(Interpolation)**問題。
- 本次LAB需要你們讀入720p / 60Hz的螢幕訊號並將該輸入影像透過Bilinear Interpolation進行放大並輸出1080p / 60Hz的螢幕訊號



Interpolation

- 在離散數學中，插值指在離散數據的不連續點上補插連續函數，使得該連續曲線**能經過**全部的離散數據點(Fig.1)。
- 利用插值可根據函數在有限個離散點的取值情況進而推估出函數在其他未知位置的近似值。由此可知所謂的插值即是利用已知的數據來估計其他未知位置的數值，至於在影像處理中為何需要插值則可以從Fig.2所示，若我們僅有一個4*4的圖片，而希望他可以顯示在一個有7*7解析度的螢幕，那會有33個未知的像素值，此時即可使用前面所介紹的插值原理來還原出該33個Pixel的值應該為多少。

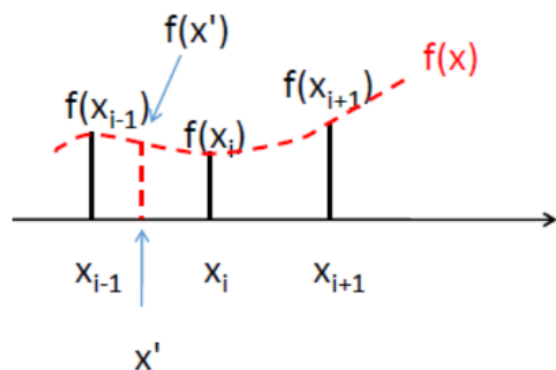


Fig.1

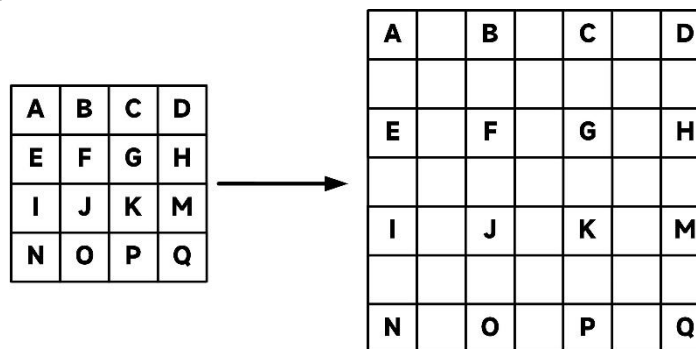
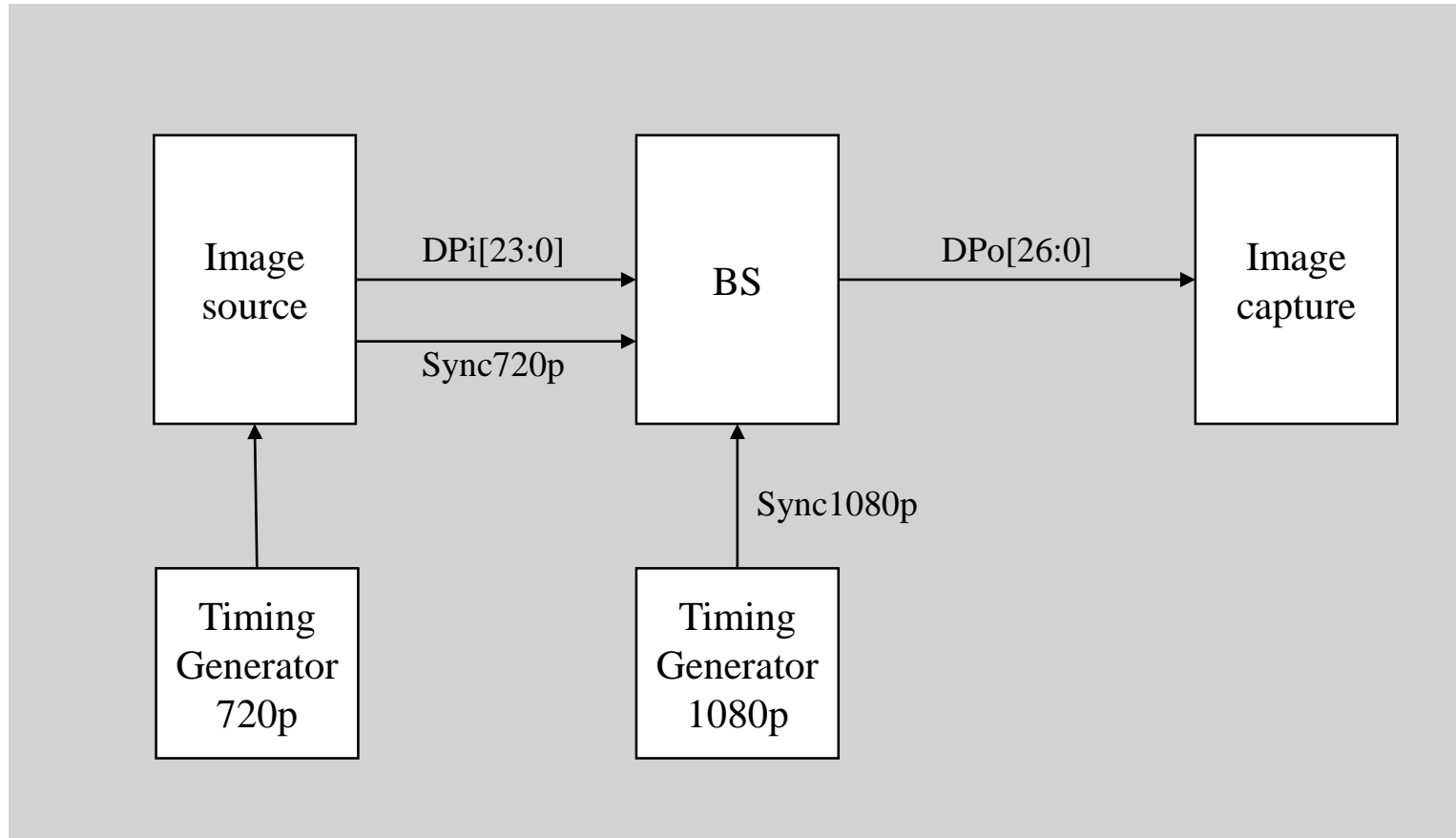


Fig.2

System Architecture



Hardware Description BS

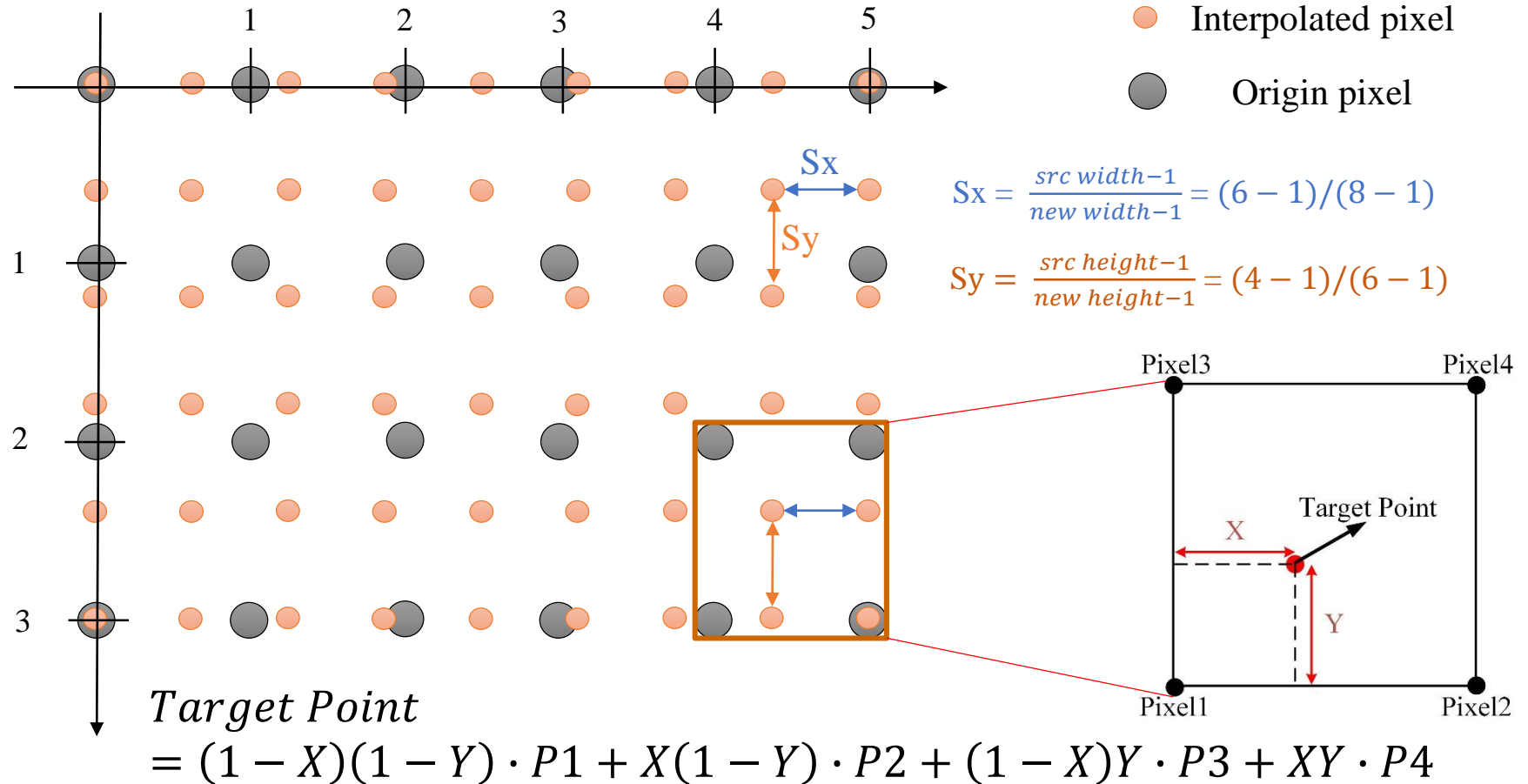
Signal	I/O	length	Description
clk1	I	1	Clock signal(720p)
clk2	I	1	Clock signal(1080p)
rst_n	I	1	Reset signal active low
Sync720p	I	3	Sync signal 720p
Sync1080p	I	3	Sync signal 1080p
DPi[23:0]	I	24	Input pixel
DPo[26:0]	O	27	Output pixel with sync

Bilinear Interpolation

- **雙線性插值法 (Bilinear Interpolation)** 屬於一階插值技術，其名稱來源於先後在水平方向與垂直方向各執行一次線性插值的過程。
- 具體做法是：對於待插值點，先找出其周圍最接近的 2×2 個像素，並根據這些像素與目標點在水平和垂直方向上的距離分配權重；距離越近的像素，權重越大。最後，將這四個像素的值按相應權重加權平均，得到新像素值。這種加權方式能夠在影像放大時實現平滑的過度，減少鋸齒現象。

Bilinear Interpolation

□ For 6x4 scale up to 9x6:



Criteria

- ❑ 請注意throughput能否對上(當輸入的幀率為60fps；你所輸出的幀率也應該為60fps)
- ❑ Memory 最多使用三條
- ❑ 請勿使用mod operator “%” 發現以0分計算
- ❑ 請勿使用division operator “/” 發現以0分計算
- ❑ 因為輸入的720p訊號其Clk與輸出的1080p Clk並不相等，需要切兩組Clock Domain，請謹慎解決該CDC issue.
- ❑ 輸出結果會和golden做驗證，誤差容忍範圍為正負1
- ❑ 電路不須合成

❑ Grading Policy

Not 60 fps	(0%)
Use mod operator	(0%)
Use division operator	(0%)
Use more than 3 memories	(0%)
Only pass one image	(25%)
Pass two images	(50%)
ALL PASS	(100%)

Criteria

Situation	Command
RTL simulation with $\text{IMG}\{X=1,2,3\}$	<code>make vcs IMG=X</code>
Dump Waveform	<code>make vcs IMG=X WV=1,2</code>
Launch nWave	<code>make wave</code>
Delete waveform files	<code>make clean</code>
Compress homework to tar format	<code>make tar</code>