Lab1

104021219 鄭余玄

Proj02-02

實作方式

原圖是 256 (2^8) 色階的灰階圖片，也就是使用 8 bits 表示一張圖片。要減少 intensity level 到 level 2^n 的方法，就是只使用 n bits 表示一張圖片，其中 n 從 2 到 8。

實作方式是將每個點的 intensity 整除以 $256/level$，就可以直接減少 bit 表示數量。最後每個點再乘以 $256/level$ 重新對應回 256 色中表示的顏色。

Proj02-03

實作方式

題目 a 小題假設縮放倍率為整數，也因此在這個假設下可以做一些優化。縮小（`scalingFactor < 1`）的實作方式，可以直接使用 `Matlab` matrix indexing 方式，跳著取樣原圖的點，就可以快速達到縮小。

放大（`scalingFactor` $\geq$ `1`）的實作方式，是我突發奇想想到的。假設 $A \in M_{n\times m}$ 並且 `scalingFactor` $= p \in \mathbb{Z}$，則使用 pixel replication 放大結果是 $A \otimes J_p \in M_{(n\times p)\times(m\times p)}$，其中 $\otimes$ 是 tensor product，$J_p = (1)_{p\times p}$ 是矩陣元素全部為 1。因為在進行 image replication 的過程，就非常類似計算 Kronecker product (tensor product)，所以讓我想到這個做法。

此外，我也有實作非整數倍的縮放。因為上述兩種方法，在小數 rounding 取樣點會不同，所以非整數倍只能透過 `for-loop` 方式，將新座標（縮放過的圖）對應回就舊座標（原圖），取得該點數值。在 `Matlab` 執行上就會相對慢許多。

(c) Explain the reasons for their differences.

由結果可以看出，使用 image replication 縮小再放回原本大小，會讓線條看似變粗糙，斜直線尤其明顯。

Proj02-04

實作方式

依據老師上課講義的推倒方式，將新座標（縮放過的圖）對應回舊座標（原圖），並且找到最近的 4 個座標點，再使用公式計算出新的 intensity 數值。

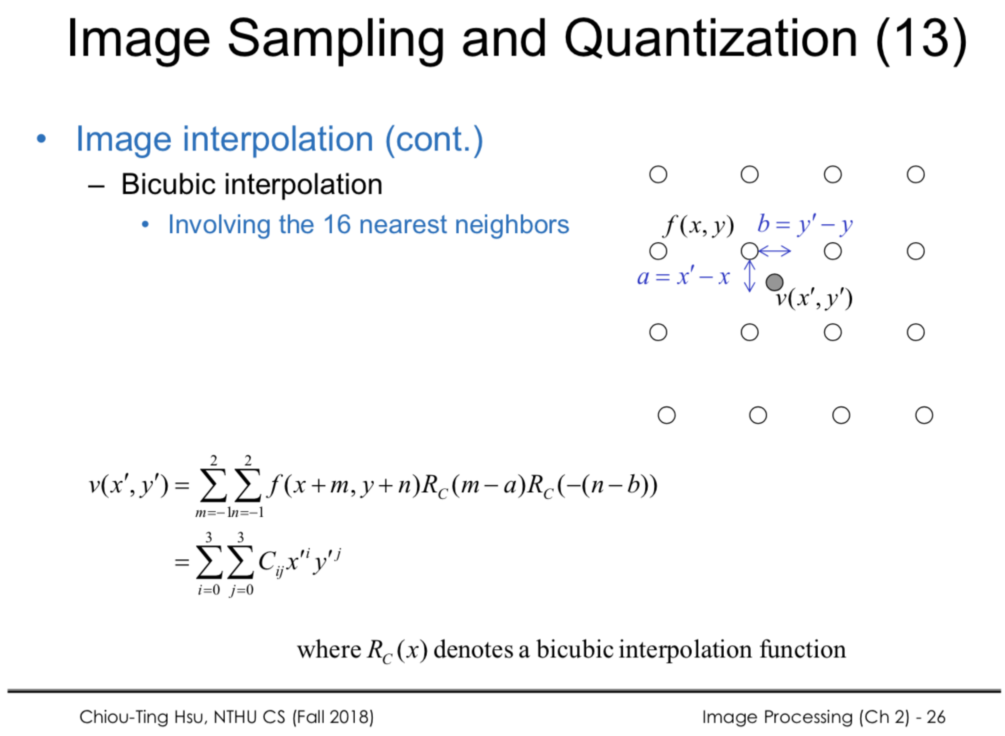
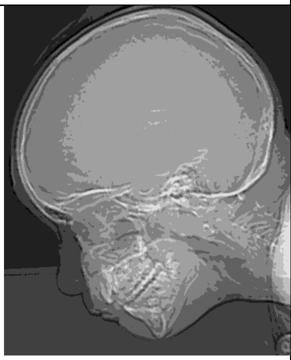
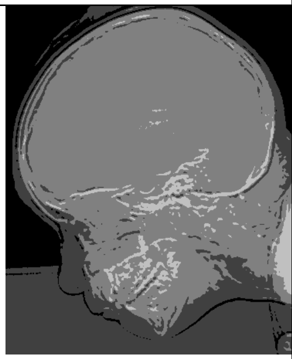


Figure 1: Bilinear Interpolation

(c) Explain the reasons for their differences.

由結果可以看出，使用 bilinear interpolation 會讓圖看起來更模糊，數字尤其明顯。數字外觀並不適合只透過 diagonal neighbors 重建，因此在縮小放大的過程中，丟失重要的資訊，像是數字中間的空洞和開口位置。直線也相對 replicaton 方法 intensity 更低，diagonal neighbors 更容易取到非直線上的像素。

Proj02-02			
			
a.tif	b.tif	c.tif	d.tif
			
e.tif	f.tif	g.tif	h.tif

Proj02-03

Proj02-04