

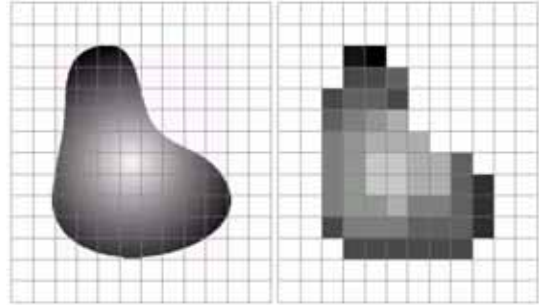
VFX 上課筆記 2005/03/09

✚ Image Fundamentals

◆ Image Formation



說明：Light source 透過 Lens 投影到 Image device 上面，接著，在 Spatial (domain) 和 Intensity (range) 上做 Sampling 的動作，記錄成一個非連續的二維陣列(如右圖所示)



◆ What is an image

我們可以把 Image 想成一個函數, $f: \mathbb{R}^2 \rightarrow \mathbb{R}$; $f(x, y)$ 的值代表的是位置 (x, y) 的顏色值。

◆ A digital image

A digital image 是經過兩個步驟而來，先在空間裡取 sample, 在對每個 sample 的值做 quantization, 以式子來表示就是

$$f[i, j] = \text{Quantize}\{f(i D, j D)\}$$

最後，一張 image 就可以被表示成一個二維的整數矩陣。

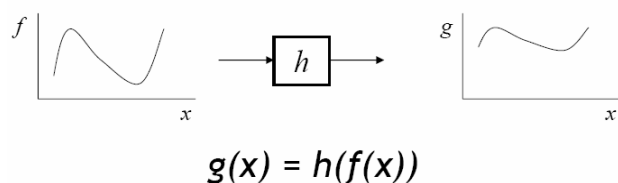
◆ Aliasing

當 Sample 時的頻率和影像的頻率是倍數或因數的關係時，會產生 Aliasing effect，如圖中右邊所示，可以看出明顯的 Pattern 存在。



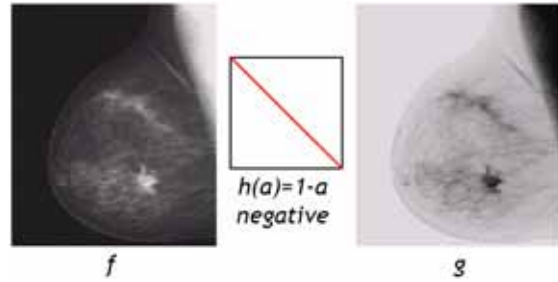
✚ Image processing

我們通常會對影像做一些處理，一般常見的方法多在 spatial domain 及 frequency domain 上做處理，以下的介紹則以 spatial domain 上為主。

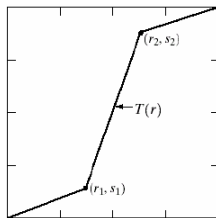
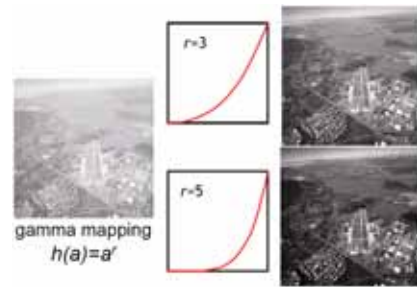


- ◆ Point processing 是只針對單點作處理，而不考慮其他點和統計特性，這樣的處理比較簡單，可是在某些目的有用，譬如 Image enhancement、contrast stretching...

1) image negative 如右圖所示，通常用於在黑暗影像中加強白色或灰色的細節，反之亦然。

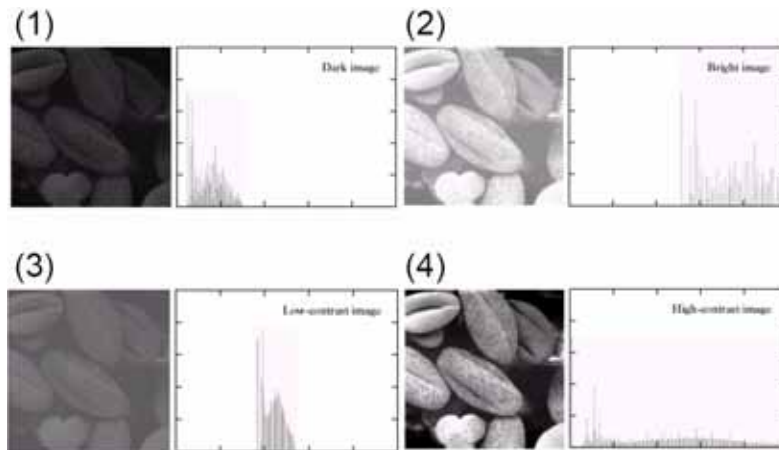


2) gamma mapping, 當 gamma 值小於一的時候，是將影像的暗的部份的對比拉開來強調暗部細節，大於一時則相反。例如 CRT 的螢幕本身會產生一個 1.8~2.5 的 gamma mapping，所以在輸入影像之前必須先透過一個 gamma correction 的動作來使得輸出結果變得正常



3) contrast stretching，可用於強調 original image 特定範圍內 pixel value 的點，如左圖所示，value 介在 r_1, r_2 間的點的值被拉開了。

◆ Histogram processing (對整張影像的統計特性做處理)：



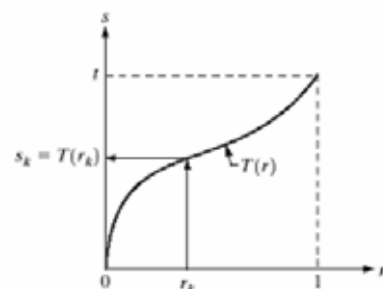
由 histogram 中可以看出圖形中 intensity 的分布情況，如圖一 histogram 整體偏左，所以圖形偏暗；圖二偏向右側，所以圖形偏亮；理想的情形應該是如圖四中整體平均分布，對比較為明顯。對於對比不明顯的影像通常有以下兩種處理方式

1) Histogram equalization：

把 histogram 畫成累積機率的分布(如右圖)，之後對每個點的 pixel value r_k 重新對應到 s_k 上，如此可得到分佈較為平均的影像。

2) Histogram matching：

把某圖的 accumulated histogram 變成和目標圖形的



accumulated histogram 一樣。例如攝影時使用 auto exposure,會使相鄰的 image 的亮度不同,所以要使用 histogram matching 讓它們的 histogram 分佈一致,對於之後的 video processing 更為正確

◆ Neighborhood processing (也叫 filtering,考慮鄰近的點) :

如果把 image 裡面的 pixel 做隨機排列,並不會改變其 histogram,在這種情況下不論是做 point processing 或是 histogram processing 都不會跟直接對原圖做有差別,that doesn't make sense,所以應該把鄰近的 pixel 考慮進來,一般的 neighborhood processing 有下列幾種 :

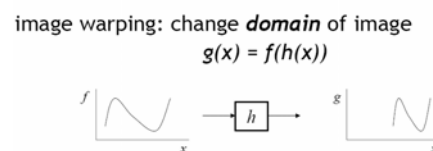
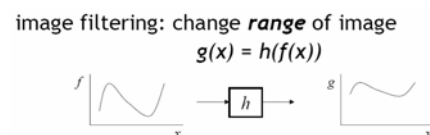
- 1) medean filter(取附近 pixel 總和的平均值),
- 2) median(取附近 pixel 中的中位數),
- 3) Gaussian filter(利用 gaussian function)

🌈 Image Warping

Image processing 是改變 range,而 Image Warping 是改變 domain,如右圖所示。

◆ Global warping (又叫 Parametric)

全部的點都用同一個 warping function 也因此可以用很簡單的幾個 parameter 來表示整個動作,下表列出了一些常見的 operation。輸入值為 pixel 在原圖中座標所在位置,輸出值為在新圖中該點對應到的座標;值得注意的是,這些 function 都有 inverse function。

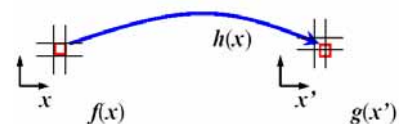


Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\begin{bmatrix} I & t \\ & \end{bmatrix}_{2 \times 3}$	2	orientation + ...	
rigid (Euclidean)	$\begin{bmatrix} R & t \\ & \end{bmatrix}_{2 \times 3}$	3	lengths + ...	
similarity	$\begin{bmatrix} sR & t \\ & \end{bmatrix}_{2 \times 3}$	4	angles + ...	
affine	$\begin{bmatrix} A \\ & \end{bmatrix}_{2 \times 3}$	6	parallelism + ...	
projective	$\begin{bmatrix} \tilde{H} \\ & \end{bmatrix}_{3 \times 3}$	8	straight lines	

- translation: $\mathbf{x}' = \mathbf{x} + \mathbf{t}$ $\mathbf{x} = (x, y)$
- rotation: $\mathbf{x}' = \mathbf{R} \mathbf{x} + \mathbf{t}$
- similarity: $\mathbf{x}' = s \mathbf{R} \mathbf{x} + \mathbf{t}$
- affine: $\mathbf{x}' = \mathbf{A} \mathbf{x} + \mathbf{t}$
- perspective: $\underline{\mathbf{x}}' \cong \mathbf{H} \underline{\mathbf{x}}$ $\underline{\mathbf{x}} = (x, y, 1)$
 ($\underline{\mathbf{x}}$ is a homogeneous coordinate)

◇ forward warping :

是指 original image 中的 pixel 經過 warping function 運算後,把原先的 pixel 直接 copy 到新的座標位置。



萬一 Pixel 在經過 forward warping function 運算之後所得到的值並不是整點,而是落在數個 pixel 之間,這時候 splatting 就是可使用的處理方式之一。

(說明: splatting -

以所得到的新座標值為中心向四周圍擴散,所以每個整點的值都是經由周圍的點擴散所得到的結果,例如可使用 Gaussian function 來做擴散。在實作上擴散範圍到什麼程度,以及由每個點擴散來的 weighting 要如何去設計是需要考慮的。)

◇ Inverse warping :

和 forward warping 相反，目標座標當作 input 輸入 inverse warping function 去反查所對應到的 original image 的座標位置。同樣地，得到的座標不一定在整點上，這時候可以使用內插法得到對應的值。

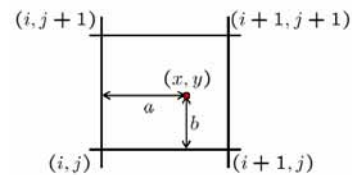
1) Nearest neighbor :

選取最近的點，這種方法得到的圖會很不自然

$$g(\mathbf{n}) = f\{\text{INT}[a_1(n_1, n_2) + 0.5], \text{INT}[a_2(n_1, n_2) + 0.5]\}$$

2) Bilinear interpolation :

不管經過 inverse warping function 後得到的座標是多少，其一定會介於原圖中某四個點之間，如右圖所示，透過這四個點進行 interpolation 後即為所求



3) Bicubic interpolation :

方法和 bilinear interpolation 類似，只是將參考點從周圍四點提高到周圍十六個點

$$f(x, y) = (1-a)(1-b) f[i, j] + a(1-b) f[i+1, j] + ab f[i+1, j+1] + (1-a)b f[i, j+1]$$

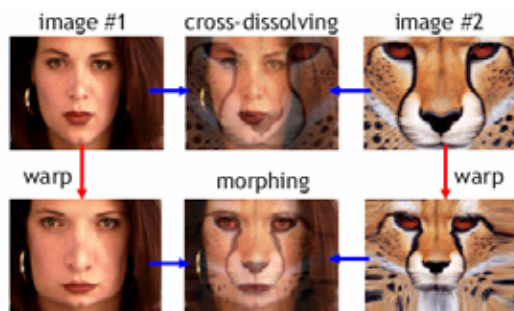
<http://astronomy.swin.edu.au/~pbourke/colour/bicubic/>

◆ Non-parametric image warping :

由於 global warping 對於局部區域的 distortions 上表現並不理想，所以產生了和 global warping 相對，每一個 pixel 都有自己的 warping function。首先，對於兩張 image 給定一個 landmarks 的 set 後便可以得到其 Delaunay triangulation，之後便可以對每一塊區域作處理。

✚ Image morphing

Morphing 的目的是將一張 image 經過很連續的過程轉換成另一張 image。Cross dissolving 是個常用的方法，但是這個方法通常不夠好，因為會產生 Ghosting Effect。這是因為 feature points 沒有對齊的關係。所以 Morphing 要做的動作就是 warping + cross-dissolving，其中 warping 的部分是做 shape(geometric)的轉換而 cross-dissolving 的部分是做 color(photometric)的轉換。如下圖所示





左圖是一百張少棒選手和結婚照使用 cross - dissolving 直接做平均的結果

右圖是用 image morphing 的方式做出來的平均人臉。



◆ Image Morphing 的方法有以下幾步:

Create a morphing sequence: for each time t

1. Create an intermediate warping field (by interpolation)
2. Warp both images towards it
3. Cross-dissolve the colors in the newly warped images



左圖是一張理想的morphing的例子
最左邊 $t=0$ 的部分和 $t=1$ 的部分，分別是原圖及目標的圖形，而 $t=0.5$ 就是morphing的中間圖。

◆ Warp specification

在做image morphing時需要將兩張圖warp到相同的field，而在指定特徵的方法有以下的兩種，分別為mesh warping 和 field warping。

在mesh warping 的部分，是指定相對應的 spline control points，再利用內積的方法到相對應的warping function(參考barycentric coordinate)。較為容易實做，但在使用者的方便性較為缺乏。

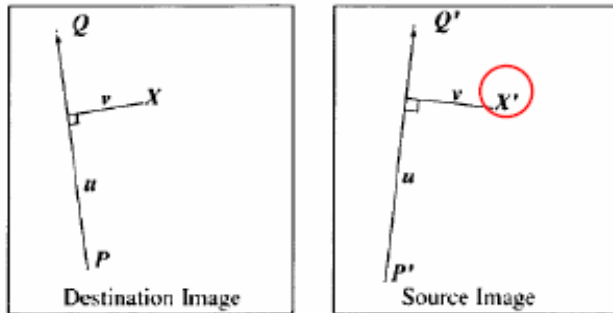
而在field warping的部分，是指定相對應的向量線條，利用Beier & Neely的演算法用內積的方式到相對應的warping function，方法較複雜但對使用者來說較為容易操作。

以下分別是mesh warping(左圖) 和 field warping(右圖)的範例。

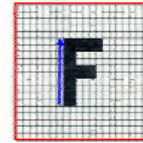


◆ Beier&Neely (SIGGRAPH 1992)的演算法如下

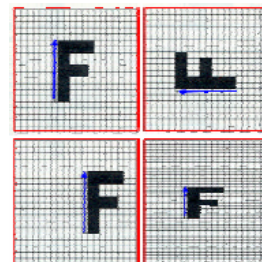
• Single line-pair PQ to P'Q':



• Examples:



Affine transformation



$$u = \frac{(X - P) \cdot (Q - P)}{\|Q - P\|^2} \quad (1)$$

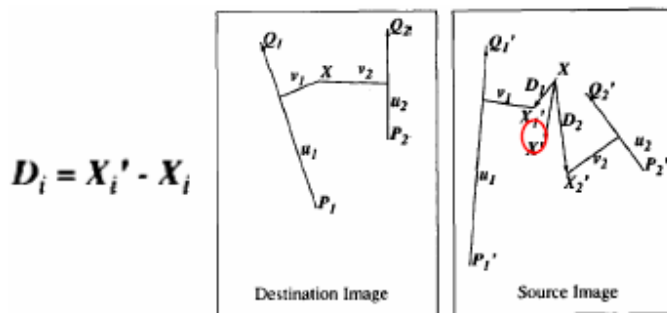
$$v = \frac{(X - P) \cdot \text{Perpendicular}(Q - P)}{\|Q - P\|} \quad (2)$$

$$X' = P' + u \cdot (Q' - P') + \frac{v \cdot \text{Perpendicular}(Q' - P')}{\|Q' - P'\|} \quad (3)$$

For each X in the destination image:

1. Find the corresponding u,v
2. Find X' in the source image for that u,v
3. Destination Image(X) = source Image (X')

當在multiple line-pair的情況下，我們可以用X₁'、X₂'來求出最終的X'，weighting的方式如下圖所示：



其中，

length 是指 line segment的長度

dist 是指到line segment的距離

參數 a, p, b 可自行調整

$$weight = \left(\frac{length^p}{(a + dist)^b} \right)$$

(當a越大，distance影響越不嚴重，所以結果會比較blur

當b越大，表示sharp function會被最近的線dominate

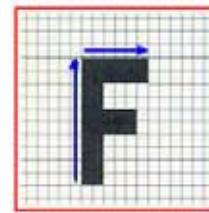
當p越大，越care line segment的長度)

完整的演算法如下:

```

For each pixel  $X$  in the destination
   $DSUM = (0,0)$ 
   $weightsum = 0$ 
  For each line  $P_i Q_i$ 
    calculate  $u, v$  based on  $P_i Q_i$ 
    calculate  $X'_i$  based on  $u, v$  and  $P_i' Q_i'$ 
    calculate displacement  $D_i = X'_i - X_i$  for this line
     $dist =$  shortest distance from  $X$  to  $P_i Q_i$ 
     $weight = (length^p / (a + dist))^b$ 
     $DSUM += D_i * weight$ 
     $weightsum += weight$ 
   $X' = X + DSUM / weightsum$ 
  destinationImage( $X$ ) = sourceImage( $X'$ )
  
```

原圖



結果



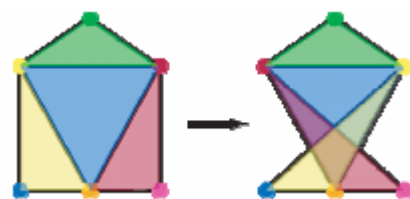
- ◆ 對於Animated sequences需要
 - Specify keyframes and interpolate the lines for the inbetween frames
 - Require a lot of tweaking
- ◆ 與mesh morphing的比較
 - 優點: more expressive
 - 缺點: speed and control
- ◆ Warp interpolation

How do we create an intermediate warp at time t ?

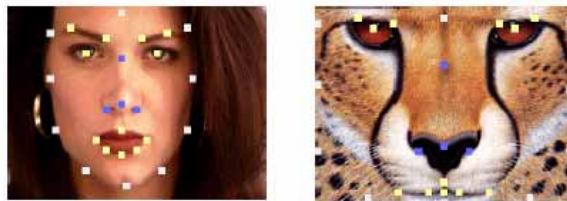
- For optical flow:
 - Easy. Interpolate each flow vector
- For feature point methods:
 - linear interpolation of each feature pair
- For Beier-Neely:

可以Interpolate line end-points, 但會碰到一種狀況, 當線旋轉180度時, 在 $t=0.5$ 的時後, 會變成一個點, 一個解決方式是interpolate line mid-point and orientation angle。

- ◆ Other Issues
 - Beware of folding
 - Can happen in any of the methods
 - You are probably trying to do something 3D-ish
 - Extrapolation can sometimes produce interesting effects - Caricatures



- ◆ Warp specification
 - How can we specify the warp
 - Specify corresponding points
 - interpolate to a complete warping function



- ◆ Transition control



如左圖所示，Morphing有時候無法得到最好的效果，如果人的鼻子warp到貓鼻子的速度可以更快的話，應該可以得到比較好的效果。

右圖就是做過Transition control之後得到的效果，很明顯看得出跟上一張有所不同。

