

Image feature extraction

一張圖形影像包含了多少意義?也就是我們要分析理解的目標。<應用>上在理解圖形的重要特徵(primitive characteristic/attributes) 可以幫助我們去分割類似的圖形、辨識物體、自動化等功能等等，但是圖形的特徵(feature) 究竟有哪幾類?

在前兩章學到的邊緣分析下，這是分辨物體，也是人眼分類的方式，利用 Amplitude feature 去辨別圖片(也是 spatial domain process)，其中我們除了之前所學的線性濾波器外還有接下去要利用統計學的分析方式(非線性濾波器)。還有人眼也會利用 Frequency feature 去辨別圖片(freq domain process)，也就是紋理(texture)特徵的分析方法(Figure9)。

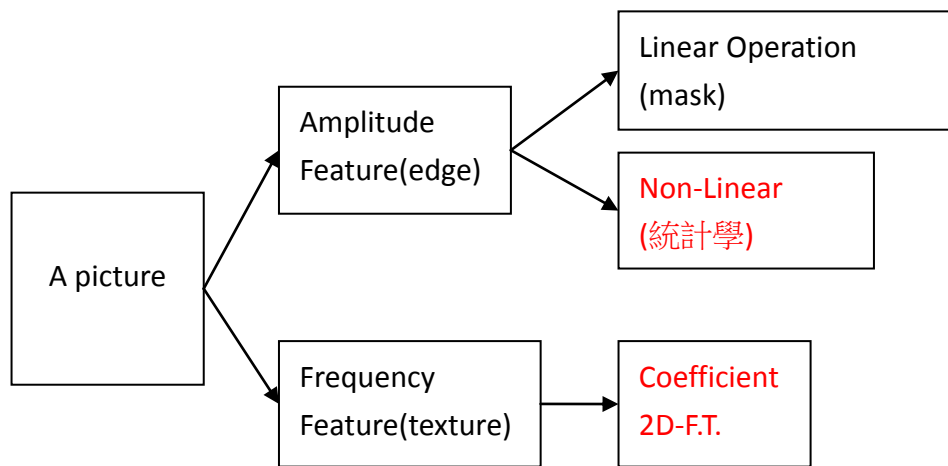


Figure9. 圖形特徵的分析方法

舉例來說影像如同聲音感受，受體的眼睛正如耳朵，耳朵能分辨同一人在不同音量大小所表示的意義 (好比老師上課遇到重點會跟同學大聲提醒，而大小聲可以利用微分方式擷取加強重點內容)，這如同眼睛所能分析的 Amplitude feature。同理耳朵能分辨同樣音量卻是不同人所發出的聲音，這是因為音色的不同。相同音色即使音量不同，但是 freq domain 中所擁有的基底是相同的，這如同眼睛所能分析的 Frequency feature，也稱為 texture。

Amplitude feature

人眼所能分辨 **Amplitude** 是因為強度的突然改變(**edge**)為特徵重點，其中利用線性濾波器的方式在前面描述過，除此之外還可以利用非線性的統計方式去分析 **spatial domain process**。<定義>

$$\text{Mean } S_M = \sum_{b=0}^{L-1} bP(b)$$

$$\text{n-th mean } \mathcal{M}_n = \sum_{b=0}^{L-1} (b-m)^n P(b)$$

$$\text{Standard deviation } \sigma_b = \sqrt{\mathcal{M}_2} = \sqrt{\sum_{b=0}^{L-1} (b-m)^2 P(b)}$$

$$\text{Energy } S_N = \sum_{b=0}^{L-1} |P(b)|^2$$

$$\text{Entropy } S_N = -\sum_{b=0}^{L-1} P(b) \log_2(P(b))$$

而這邊還有一種稱為 **Histogram** 的統計工具。<定義>gray level 的機率分佈圖形。

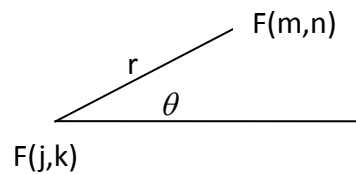
First-order Histogram

$$P(b) = \Pr\{F(j, k) = r_b\}$$

$$P(b) \approx N(b) / M$$

而且還有 **second-order histogram** 定義了(joint-probability distribution of pair pixels)

。其<定義>如下



$$P(a, b) = \Pr\{F(j, k) = r_a, F(m, n) = r_b\}$$

以及統計的運算子(measurements)

$$\text{Autocorrelation } S_A = \sum_{a=0}^{L-1} \sum_{b=0}^{L-1} abP(a, b)$$

$$\text{Covariance } S_C = \sum_{a=0}^{L-1} \sum_{b=0}^{L-1} (a-\bar{a})(b-\bar{b})P(a, b) \quad \bar{a} = \sum_{a=0}^{L-1} \sum_{b=0}^{L-1} aP(a, b) \quad \bar{b} = \sum_{a=0}^{L-1} \sum_{b=0}^{L-1} bP(a, b)$$

$$\text{Inertia } S_I = \sum_{a=0}^{L-1} \sum_{b=0}^{L-1} (a-b)^2 P(a, b)$$

$$\text{Absolute value } S_V = \sum_{a=0}^{L-1} \sum_{b=0}^{L-1} |a-b| P(a, b)$$

$$\text{Inverse difference: } S_F = \sum_{a=0}^{L-1} \sum_{b=0}^{L-1} \frac{P(a, b)}{1 + (a-b)^2}$$

$$\text{Energy: } S_G = \sum_{a=0}^{L-1} \sum_{b=0}^{L-1} |P(a, b)|^2$$

Entropy: $S_T = -\sum_{a=0}^{L-1} \sum_{b=0}^{L-1} P(a,b) \log_2(P(a,b))$

在<物理意義>這是分析在某固定向量上顏色的分布情況，有點類似輕微的紋理分析。

Transform coefficient feature

這是紋理分析的第一部，我們把圖形利用 Fourier transform 轉換到 freq domain 表示一張圖形是由多少種不同頻率的影像(Basis)組合而成。這些係數分佈

$$\mathbb{F}(w_x, w_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(x, y) e^{-i(w_x x + w_y y)} dx dy$$

也就成了這張圖的特徵(feature)，如同聲音中的“音色”，進而能分析這張圖形的 texture。

同理我們也可以把係數(Coefficient)轉換至 polar domain 等等不同的 domain 之中去了解其 freq 所代表的意義。

$$\mathfrak{M}(w_x, w_y) = |\mathbb{F}(w_x, w_y)|^2 \quad \text{能量強度 PSD domain}$$

$$N(\rho) = \int_0^{2\pi} \mathfrak{M}(\rho, \theta) d\theta \quad \text{polar 轉成圓環分析 domain}$$

$$P(\theta) = \int_0^{\infty} \mathfrak{M}(\rho, \theta) d\rho \quad \text{polar 轉成 angle 分析 domain}$$

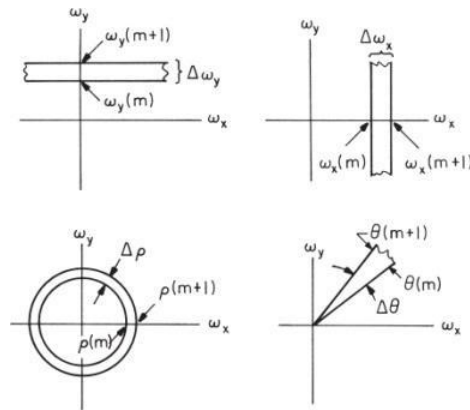


Figure10.freq domain 之下的各種分析紋理的方式。

Image feature evaluation

在一張圖中我們常常需要找到的特徵 feature 包含了振幅的 edge 以及頻譜的 texture 兩種特徵，通常一張圖形經由 edge classification 或是 segmentation 切成具有相同類似紋理的 texture regions，我們再藉由 texture analysis 去自動辨識紋理的特徵。所以一影像處理流程圖必須包含多集的分析處理。

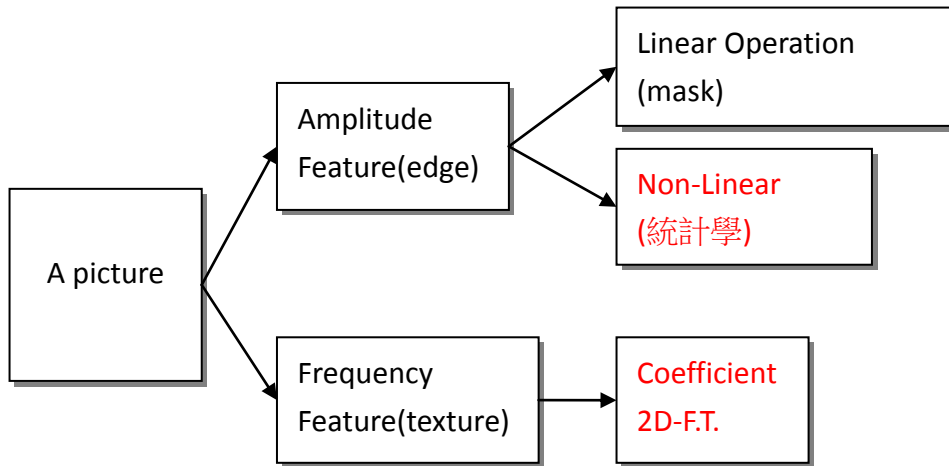


Figure2 圖形特徵處理的方法

在 edge analysis 中切割有很多變數、參數達到最佳化，其中最大的特色是不管是否在線性或非線性 Filters 的處理方式下，都有定義流程或是數學描述式(好比定義 edge、 $1'st/2'st$ 微分等等)。在 texture analysis 該如何用數學定義呢？

在 texture analysis 中分成兩種方式。第一方式是由一個對照組求出其紋理特徵函數，然後和其他實驗組去互相比較相似性 similarity。(這種方式好比 F.T./edge detection method/second-order histogram)做的分析方式，結果出來都是紋理特徵函數的比較。

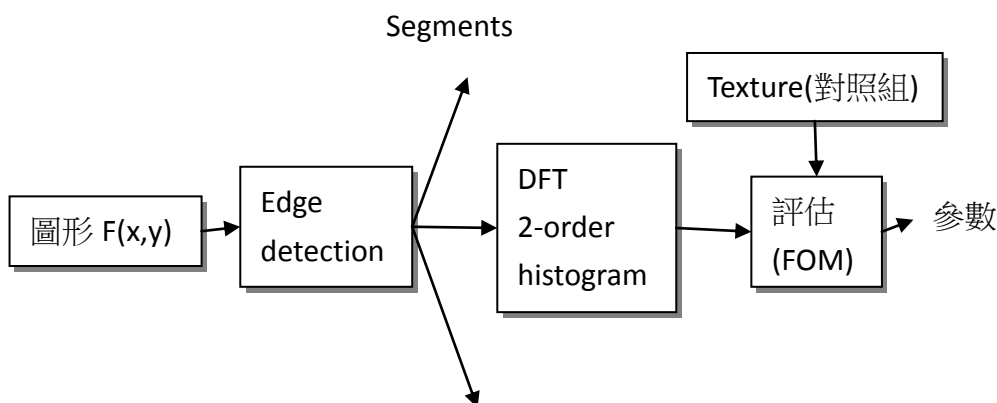


Figure3. texture analysis 的第一種方式(在不同 basis 下的特徵)

，但是這明顯的缺點是“對照組”該從何而來？沒有定義。

texture analysis，另一種方式直接定義了兩種 quantities，就是希望能從頭至尾都是用數學去描述 texture 的邏輯化處理。首先，必須先利用 discrete stochastic fields 定義許多 texture images

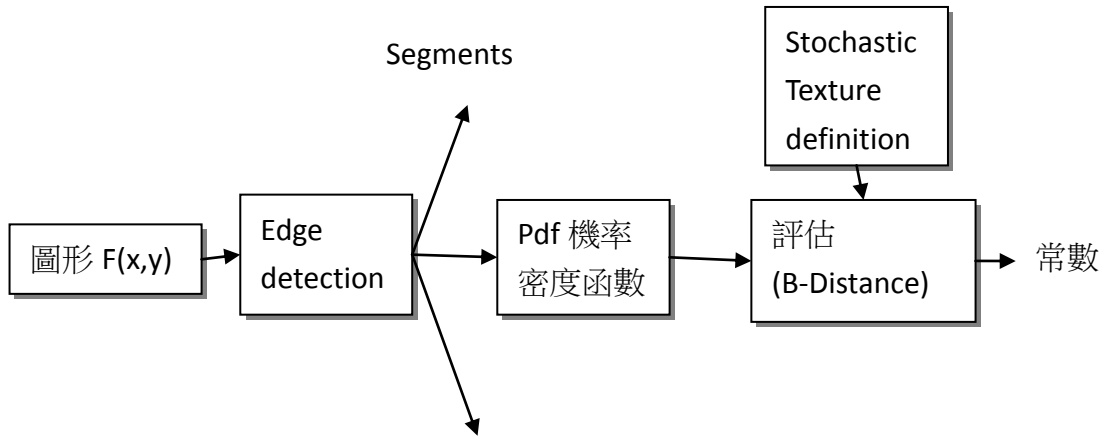


Figure4. texture analysis 的第二種方式(stochastic)

· 這種特殊方式是隨機信號的，代表很多圖形都符合某種機率分布條件，再者將經過評估(Figure of merit)方式 (Bhattacharyya distance)去比較兩個圖形機率分布(pdf)的相似性(similarity)。結果出來就是一常數的評分(和 F.T.分析不同)。這種隨機程序處理方式有很複雜的定義，並且在運算量上都非常龐大。

Bhattacharyya distance (B-distance)

我們在統計學上，<目的>是常被用來測量兩個機率密度函數(pdf)的相似性。由於兩函數的相似性、相異性，分布情況都可藉由許多參數(correlation/covariance/variance...)去了解，在此的 B-distance 是統計上幫助非函數圖形(只有機率分佈定義)之間的相似性。

<定義> B-distance

$$D_B(p, q) = -\ln(BC(p, q))$$

$$BC(p, q) = \sum_{x \in X} \sqrt{p(x)q(x)}$$

而常用的兩個高斯分佈圖形之間的 B-distance 可以當作常用公式。

$$\begin{aligned} \lambda_1 &\sim G(\mu_1, \sigma_1^2) \\ \lambda_2 &\sim G(\mu_2, \sigma_2^2) \end{aligned} \quad D_B(\lambda_1, \lambda_2) = \frac{1}{8}(\mu_1 - \mu_2)^T \left(\frac{\sigma_1^2 + \sigma_2^2}{2} \right)^{-1} (\mu_1 - \mu_2) + \frac{1}{2} \ln \frac{|\frac{1}{2}(\sigma_1^2 + \sigma_2^2)|}{|\sigma_1^{\frac{1}{2}}|^2 |\sigma_2^{\frac{1}{2}}|^2}$$

Transform coefficient feature

這是屬於第一種的 texture analysis(在不同 basis 下的特徵)。<方式>利用 frequency transform 找出一區塊圖形的紋理(texture)函數，了解此一紋理在頻譜上的特徵。在 Amplitude feature 的基底是(X-Y axis)藉由 edge 的分析分辨物體。而 frequency feature 的基底(弦波 $\sin \omega / \cos \omega$)是藉由 texture 的分析分辨物體。

首先必須抽取出任一圖形的 frequency feature。Lendaris, Stanley 在研究光學系統(optical system produced an electric filed radiation pattern)時候發現在 2D-image 上的 Fourier transform<定義>

$$F(\omega_x, \omega_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(x, y) e^{-i(\omega_x x + \omega_y y)} dx dy$$

其中 $F(\omega_x, \omega_y)$ 和原本 image $F(x, y)$ 是 unique 的對應函數。而且任意 image 也必滿足 Dirichlet rule

$$\int_{-\infty}^{\infty} |F(x, y)| < \infty$$

，所以我們可以將此公式當作所有 image frequency transform。

其中 $F(\omega_x, \omega_y)$ 參數是代表了在 frequency domain 上的基底函數大小(weight)，而基底(2D basic functions)是根據 Euler formula $e^{i\theta} = \cos \theta + i \sin \theta$ 是表示兩個弦波($\sin \omega / \cos \omega$ 的合成函數)。

當然在<應用>上要利用 F.T. 來做 texture analysis 必須先找到對照組再進行函數的比較分析。

通常在應用上有兩個地方還需要<注意>。第一是影像是 discrete 函數，必須再定義(redefine)給離散訊號專用的 DFT 方式。第二是影像的 Frequency-domain 仍是 2D 圖形，常常我們只需要比較某些特定區域的不同，將之化簡成 1D 的函數(好比 polar form (ρ, θ))再做比較。

離散訊號影像重新<定義>transform 函數，<Q>因為 image $F(x, y)$ 是離散非週期訊號，其 Fourier Coefficient $F(\omega_x, \omega_y)$ 是週期連續訊號，這會導致電腦無法處理連續函數。<解決>但只要我們把原本圖形 $F(x, y)$ 轉換成週期訊號

$F(x + N, y + N) = F(x, y)$ ，其 F.T. 結果就會變成離散的訊號(這好比將原本連續訊號作取樣(sample)的結果

$$F(u, v) = F(n\omega_x, n\omega_y) \quad , \quad n = 1 \dots N$$

我們定義離散 image 的 DFT 的<公式>

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} F(x, y) e^{-\frac{2\pi i}{N}(ux+vy)}$$

並且化簡 $M(\omega_x, \omega_y) = |F(\omega_x, \omega_y)|^2$ 稱之 intensity of radiation pattern，類似通信中的能量密度函數。 $M(\omega_x, \omega_y)$ 雖然對於 $F(\omega_x, \omega_y)$ 並非 1-1 函數(unique)，但是當影像 shift 位置時候，並不會影響到 $M(\omega_x, \omega_y)$ 函數，也是在應用上常使用的函數。

第二個問題在於需要某些特定區域的 F.T. 函數分布，可以想像成把原本 2D Fourier Coefficient $M(\omega_x, \omega_y)$ 轉換成 1D 的 Coefficient。有許多<公式>可以取出不同的特定區域。

$$\langle \text{Horizontal slits} \rangle \quad S_1(m) = \int_{-\infty}^{\infty} \int_{\omega_y(m)}^{\omega_y(m+1)} M(\omega_x, \omega_y) d\omega_x d\omega_y$$

$$\langle \text{Vertical slit} \rangle \quad S_2(m) = \int_{\omega_x(m)}^{\omega_x(m+1)} \int_{-\infty}^{\infty} M(\omega_x, \omega_y) d\omega_x d\omega_y$$

$$\langle \text{Ring} \rangle \quad S_3(m) = \int_{\rho(m)}^{\rho(m+1)} \int_0^{2\pi} M(\rho, \theta) d\rho d\theta$$

$$\langle \text{Sector} \rangle \quad S_4(m) = \int_0^{\theta(m+1)} \int_{\theta(m)}^{\theta(m+1)} M(\rho, \theta) d\rho d\theta$$

除此之外，影像可以在後級處理上加入 \log 函數 $g(x, y) = \log(f(x, y))$ 轉換，讓 **high frequency** 的特徵部分加強凸顯。

這種 **Fourier Coefficient** 紋理分析方式是依照人眼所辨識圖形的“粗糙度”(coarse to fine)來做的，但是在一開始區塊的分割(spatial segmentation in noise environment)、或是 overlap(相同紋理的重疊，好比通信中 ISI 現象)、或是光影的方向都會影響到係數 $F(\omega_x, \omega_y)$ 的變化，要如何做出良好的 texture analysis 在後級處理上仍需要很多努力。

Edge detection methods

也是第一種 texture analysis 方式(在不同 basis 下的特徵)由 Rosenfeld, Troy 提出分析圖形 $F(x, y)$ 每一 pixel 的 Neighborhoods 有多少 edge 數量。具有相同類似數量的 edges 的點可以視為紋理相同，其<物理意義>也像是分析 image 的 gradient $G(j, k)$ 的分佈，利用人眼辨識粗糙度是依靠 edge 的密度來實做。其公式為

$$T(j, k) = \frac{1}{W^2} \sum_{m=-w}^w \sum_{n=-w}^w E(j+m, k+n)$$

$E(j, k)$ 是經由 edge detection 後的邊界圖形。

Autocorrelation methods

第一種的 texture analysis(在不同 basis 下的特徵)，其<物理意義>不像是上述兩例利用“人眼分辨粗糙度”的方式進行，而是另一種人眼(human viewer)對圖形 mean/variations/autocorrelation..等統計學參數的敏感性以作為比較的參數。可以分別<定義>如下

$$\langle \text{mean} \rangle \mu = E[x_0]$$

$$\langle \text{variation} \rangle \sigma^2 = E[(x_0 - \mu)^2]$$

$$\langle \text{correlation} \rangle R(x, y) = \frac{E[(x - \mu_x)(y - \mu_y)]}{\sigma^2}$$

其中以 Autocorrelation 舉例，這很類似 F.T.的<觀念>，把 image transform 至新的 domain(只是 Basis function 不同)來看其紋理變化。而 autocorrelation 的基底就是位移 shift 的大小。

$$\langle \text{定義} \rangle A_f(m, n) = \sum_j \sum_k F(j, k)F(j - m, k - n)$$

在 coarse region 中 autocorrelation 結果因 shift 對此運算影響不大，常出現類似 uniform distribution。而 fine region 中常就是非常多變。Faugeras/Pratt 提出新的 autocorrelation spread measure 方式。不僅減少 1/2 運算量(computation)，而且定義評分標準(FOM)去比較紋理結果。<定義>

$$S(u, v) = \sum_{m=0}^T \sum_{n=-T}^T (m - \mu_m)^u (n - \mu_n)^v A_f(m, n)$$

$$\eta_m = \sum_{m=0}^T \sum_{n=-T}^T m A_f(m, n) \quad , \quad \eta_n = \sum_{m=0}^T \sum_{n=-T}^T n A_f(m, n)$$

<profile spread> $S(2, 0)$ <cross-relation> $S(0, 2)$ <second degree spread> $S(1, 1)$

Decorrelation method

我們建立流程圖

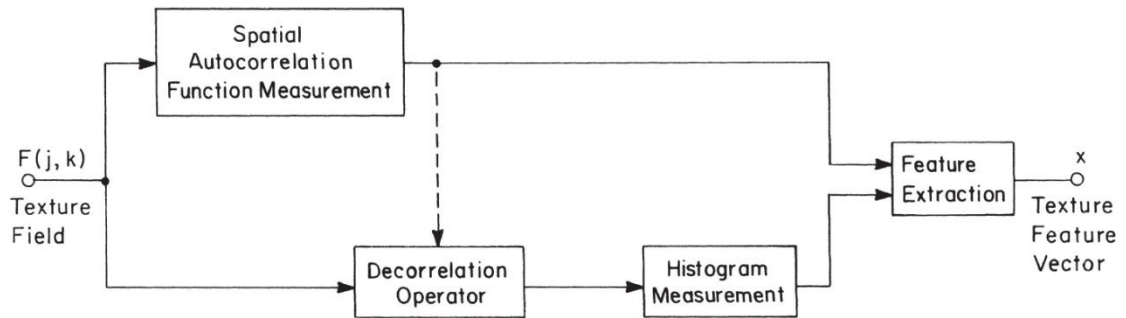


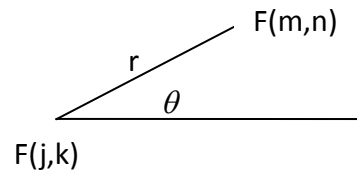
Figure5. Decorrelation method of texture feature extraction

將 texture field 區塊求出其 autocorrelation 函數，再將 texture field 發展成 whitening filter $w(j,k) = F(j,k) \otimes H_w(j,k)$ ， $w(j,k)$ 成了 uncorrelated whiten field，求出其 histogram 分布就可以分析人眼對“autocorrelation”的分辨法。其中由於在求 feature extraction(B-distance)的運算量龐大，可以利用 Laplacian/Sobel gradient 運算子簡化運算量。

Dependency matrix method

第一種的 texture analysis(在不同 basis 下的特徵), 是 Haralick 提出以 joint amplitude histogram(2-order histogram) of pairs of pixel 的紋理分析。<物理觀念>是一 transform 把 texture 投影至以 (ρ, θ) 為固定距離參數的色彩變化。也是人眼辨識紋理的一種方式(某一距離色彩重複週期出現), 雖然距離 (ρ, θ) 對於任意圖不確定, 但是只要良好的話, 很容易藉由 2-order histogram 凸顯出紋理特徵。

Dependency matrix method 又稱(gray scale dependency matrix/co-occurrence matrix)其<公式>



$$P(a, b) = \Pr\{F(j, k) = r_a, F(m, n) = r_b\}$$

是代表統計上每一點的相關性(statistical dependency of pixel pairs)。

求出了(2-order histogram)接著要評估 2 functions 之間的相關性(similarity)必須藉由之前學過統計上的參數 (autocorrelation/covariance/inertia/entropy/energy)...去簡化 function 的特徵或比較。好比透過<inertia>公式

$$T(j, k, \rho, \theta) = \sum_{a=0}^{L-1} \sum_{b=0}^{L-1} (a-b)^2 P(a, b, j, k, \rho, \theta)$$

去評分(FOM) image 的特徵常數。

Texture define

在很多時候風景圖(nature scene image)缺少 sharp edge，但是包含了很多不同的類似紋理區塊(texture)，而 texture analysis 可以幫助圖形的 segment/classification...作用。其中在 edge detection 困難之處在於 threshold/mask (linear filter) 決定，或是 ideal edge define(non-linear)。同樣的問題出現在 texture analysis，<Q>在於定義 texture 的數學方式為何？

Hawkins 首先提出紋理 texture 的邏輯化描述

- (1) some local 'order' is repeated over a region which is large in comparison to the order's size
- (2) the order consists in the nonrandom arrangement of elementary parts
- (3) the parts are roughly uniform entities having approximately the same dimensions everywhere within the textured region.

除了要把這邏輯敘述定義成數學(math measurement)的困難外，我們發現兩個<問題>，image 的 size 很重要，在越小區塊的 texture 測量(small numerical texture measurement)越精細，反之區塊越大的紋理得到結果會越粗糙。所以在前級的 region segment(edge detection)必須先找出類似的區塊紋理(uniformly texture)才能分析。還有一點，即使 image 做出完美(ideal)的分割，texture 圖形卻是包含了許多 noise/semi-repetitive...等等，和我們定義的紋理必有不同的地方，也是我們努力改善的前景。

所以在 texture analysis 的第二種方式，就是(Figure2) 如圖所示先定義了 texture(artificial)，然後再者將經過評估 (Figure of merit)方式(Bhattacharyya distance)去比較兩個圖形機率分布(pdf)的相似性(similarity)。

Math definition of Texture by stochastic

數學方式定義了紋理。我們利用離散隨機程序(Discrete Stochastic Field)<定義>an array of numbers that are “randomly” distributed in amplitude and governed by some joint probability density。去設定 image 的機率密度函數，然後反推回(inverse transform)原本圖形(因為定義是 pdf 所以滿足條件的圖形很多)，接著討論人眼對不同 DSF 的感受以發展及了解紋理特徵。

<歷史>在 1960's，Julesz 定義了 Discrete Stochastic Field 的重要參數，Purks, Richards/ Pratt 發展出人眼感受的機制 (mechanism of human view perception)對於自動化辨識做出的貢獻。

Discrete Stochastic Field 制定的流程圖

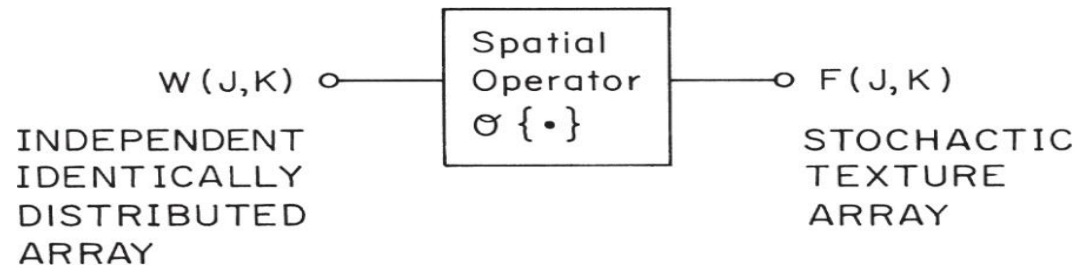


Figure6. Stochastic texture field generation model.

如 Figure6。其中制定出的 stochastic texture array 其機率分佈又分成 n-th order density<定義>

$$P(x_0 | Z_1, \dots, Z_J) = \frac{P(x_0, Z_1, \dots, Z_J)}{P(Z_1, \dots, Z_J)}$$

其中 $Z_1 \sim Z_J$ 表示任一點(pixel)的 neighbors(不一定 9 點)。若 $J=0$ 稱 first order density $P(x_0)$ [無條件]、 $J=1$ 稱 2-order density $P(x_1)$...以此類推<制定>出由機率密度函數和每一點跟周遭 neighbors 獨立性關係所產生的 texture。

Julesz texture field

因為由 Discrete Stochastic Field 制定出的 Texture，然後利用 B-Distance 求出和其他圖形的 pdf 差異。<流程圖 Figure2> 所述，這種由 pdf 去比較紋理差異性的方式雖然十分數學邏輯化，但是相同的 pdf 卻可能產生不同紋理的人眼感受卻是值得深入探討的。

1. *Notation.* Let $x_n = F(j, k - n)$ denote a row neighbor of pixel x_0 and let $P(m)$, for $m = 1, 2, \dots, M$, denote a desired probability generating function.
2. *First-order process.* Set $x_0 = m$ for a desired probability function $P(m)$. The resulting pixel probability is

$$P(x_0) = P(x_0 = m) = P(m) \quad (16.5-2)$$

Im.F

VISUAL TEXTURE DISCRIMINATION 549

3. *Second-order process.* Set $F(j, 1) = m$ for $P(m) = 1/M$, and set $x_0 = (x_1 + m) \text{MOD}\{M\}$, where the modulus function $p \text{MOD}\{q\} = p - [q \times (p \div q)]$ for integers p and q . This gives a first-order probability

$$P(x_0) = \frac{1}{M} \quad (16.5-3a)$$

and a transition probability

$$p(x_0|x_1) = P[x_0 = (x_1 + m) \text{MOD}\{M\}] = P(m) \quad (16.5-3b)$$

4. *Third-order process.* Set $F(j, 1) = m$ for $P(m) = 1/M$, and set $F(j, 2) = n$ for $P(n) = 1/M$. Choose x_0 to satisfy $2x_0 = (x_1 + x_2 + m) \text{MOD}\{M\}$. The governing probabilities then become

$$P(x_0) = \frac{1}{M} \quad (16.5-4a)$$

$$p(x_0|x_1) = \frac{1}{M} \quad (16.5-4b)$$

$$p(x_0|x_1, x_2) = P[2x_0 = (x_1 + x_2 + m) \text{MOD}\{M\}] = P(m) \quad (16.5-4c)$$

我們可以看出這隨機程序在 pixel pairs along a row are independent and spatial uncorrelated，<物理觀念>也是說在一隨機分佈(random distribution)的圖形(很像 noise)中，若是滿足一些特有條件機率，則成了會讓人眼有感受(sensitive)的特殊 textures。

Pratt ,Fangeras, Gagalowicz texture field

前人利用 Julesz texture field 再加上 Gaussian process · 把條件機率密度函數(conditional pdf)轉換成新的<定義>

$$p(x_0|z_1, \dots, z_J) = \frac{\left[(2\pi)^{J+1} |\mathbf{K}_{J+1}| \right]^{-1/2} \exp \left\{ -\frac{1}{2} (\mathbf{v}_{J+1} - \boldsymbol{\eta}_{J+1})^T (\mathbf{K}_{J+1})^{-1} (\mathbf{v}_{J+1} - \boldsymbol{\eta}_{J+1}) \right\}}{\left[(2\pi)^J |\mathbf{K}_J| \right]^{-1/2} \exp \left\{ -\frac{1}{2} (\mathbf{v}_J - \boldsymbol{\eta}_J)^T (\mathbf{K}_J)^{-1} (\mathbf{v}_J - \boldsymbol{\eta}_J) \right\}}$$

where

$$\mathbf{v}_J = \begin{bmatrix} z_1 \\ \vdots \\ z_J \end{bmatrix} \quad (16.5-6a)$$

(16.5-6b)

$$\mathbf{v}_{J+1} = \begin{bmatrix} x_0 \\ \mathbf{v}_J \end{bmatrix} \quad (16.5-6c)$$

以做改善 texture definition ◦

Microstructure Methods

屬於第一種 texture analysis(在不同 basis 下的特徵)。我們將 texture 實驗組圖形和對照組圖形做比較並偵測出其 correlation 圖形。和 autocorrelation 不同之處是在於 autocorrelation 是利用已切割成區塊的 texture region 求出其特徵函數。而對照組和實驗組的 correlation 是將未切割的原圖和紋理圖形做運算，把特徵相似的地方標示出的方法。

而 Microstructure Methods <做法>是原影像和紋理的 correlation 運算(若對稱則相等於 convolution)，求出其相似紋理的位置。同樣的可以將不同紋理($H_i[j,k]$)和原圖做 correlation 運算求出位置。甚者我們可以把 correlation 後的 image 再求其 standard deviations measurement 作為更精確的紋理特徵(如圖一所示步驟(Figure1))。

$$M_i(j,k) = F(j,k) \otimes H_i(j,k)$$

$$M_i(j,k) = F(j,k) \otimes H_i(j,k)$$

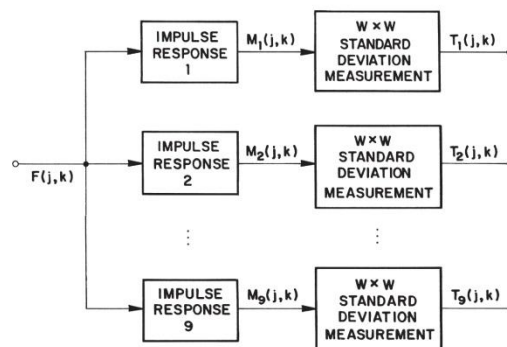


Figure1. Laws microstructure texture feature extraction method

可以<注意>的是第一 impulse response $H_i[j,k]$ 是之前的 Linear 運算子求 edge detection(1'st/2'st)相似，但是當 mask 較大或是有特殊排列時候，其 convolution 運算就是在找原圖中和 mask 紋理相似(similarity)的地點。第二則是 standard deviation measurement 也是在 edge detection 也會用到，但是它是一個 non-linear 運算子。

而在利用 microstructure detection 方法時用到的 mask 仍會有類似之前的<問題>產生。包括:” pyramid average, the Sobel horizontal and vertical gradients, the weighted line horizontal and vertical gradients and the cross second derivative...等等。

Haralick 則利用 Chebyshev Basis 的 edge detection mask 作為 texture measurement。因為 Chebyshev Basis 把圖形轉換到彼此正交函數的圖形上。可以想像成一圖形是由許多方向的紋理特徵圖所組合而成。而且可以將 Chebyshev Basis(3x3) 擴大變成更多樣的紋理特徵分析。

Gabor Filter Methods

由於 microstructure method 的紋理特徵不容易量測(scalable)。Bovik et al.利用 Gabor Filter(線性濾波器)作為量測紋理特徵的平均值。而 Gabor filter 的 impulse response($H[x, y]$)是由一群同調(harmonic)的 Gaussian functions 相乘組合。其<公式>如下

$$H(x, y) = G(x', y') e^{2\pi i f x'} \quad i = \sqrt{-1}$$

$$G(x, y) = \frac{1}{2\pi\lambda\sigma^2} e^{-\frac{(\frac{x}{\lambda})^2 + y^2}{2\sigma^2}} \quad (x', y' \triangleq) \quad x(\cos\phi - s y \sin\phi) + y(\sin\phi + s x \cos\phi)$$

下圖 Figure2 顯示了 impulse response($H[x, y]$)的實部和虛部的強度圖形。

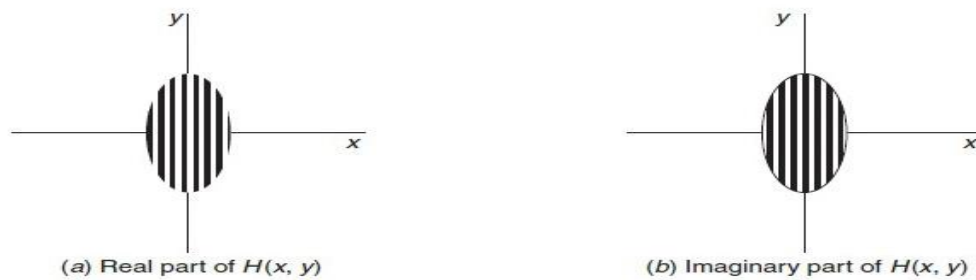


Figure2 . Impulse response array of Gabor Filter $H(x,y)$

而其 Gabor filter 的 F.T. $H(u, v)$ 變成如圖(figure3)下

$$H(u, v) = e^{-2\pi^2\sigma^2[(u-F)^2 + (v)^2]}$$

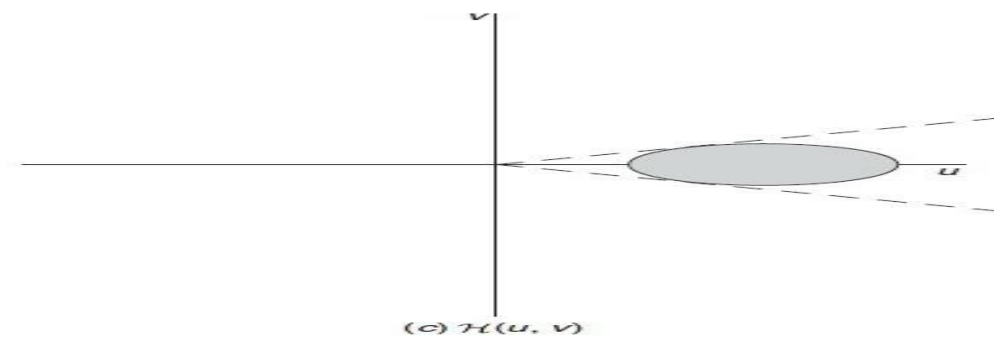


Figure3 . transfer function of a Gabor filter

Gabor Filter 具有<特徵>是可調整方向(tunable orientation)以及調整中心頻率城旁頻帶濾波(passband and tunable center frequency)，在特殊參數下還可以變成"daisy petal filter"，以其形狀而命名(其葉片由中心點發散出去)。

而<後續發展>Grigorescu et al.展示和比較了圖形經 Gabor Filter 後的紋理特徵。以及由 Randen, Husoy 發展的紋理分類(texture classification method)。

Transform and Wavelet Methods

屬於第一種 texture analysis(在不同 basis 下的特徵)。之前學過的 F.T.是很直觀的將 image 轉換到頻域上的一種通則(generalized transform)。但是缺點是在實做上以免資源的耗費甚鉅，我們先將原圖(NxN)形切割成多個 MxM(好比 Jpeg 的 M=8)的大小，再去做 F.T.的紋理分析。但是<Q> MxM size 的區塊(region)究竟能否包含一個週期以上的紋理讓我們找到仍是隱藏的問題。

Mallat 用 Haar's Wavelet transform(as a means of generating texture feature vectors)分析。而 Unser 改善了 8x8 的 mask 效能。Randen, Husoy 利用 Daubechies transforms 將 mask size 提升至 10x10。

<結論>是 Wavelet transform 提供了比 F.T.更細膩的分類(classification for many samples)

Singular-Value Decomposition Methods

Ashjari 提出由 SVD 理論的 texture measurement。由 SVD 分解法告訴我們任一個(MxN)矩陣(A)可以被分解成

$A = U \Sigma V^T$ 其中的 U 和 V 是 unitary matrix(滿足 $UU^T = I$ 特徵), 而且也是 transform 的 Basis。Σ 則是由 singular

values(AA^T 的 eigenvalues)組合成的 A 矩陣特徵值。

在<方法>上任一大小的 image 切割成 MxM 小區塊以免資源的浪費。分析其 region texture 的 singular values。若是 singular values 趨向 uniform distribution, 則在圖形中的意義是“spatial uncorrelated”; 反之, singular values 趨向集中(skewed), 則在圖形中的意義是“highly structured”。

Chris Harris, Mike Stephens "the General radial Hilbert transform and its application to 2-D edge detection"

1. The edge tracking problem

2. Moravec detector

3. Problems of Moravec detector & solutions

3.1 <sol> Noisy response due to a binary window function

3.2 <sol> Only a set of shifts at every 45 degree is considered

3.3 <sol> Only minimum of E is taken into account

1. The edge tracking problem

在 edge detection 之後的圖片，我們希望能有更多的資訊使得<目標>輪廓(contour)能凸顯，但是在已知的 filter 或做法之中常常都因資訊不足使得偵測出的 edge 無法連接而成一 closed form(contour)，這點我們希望能藉由一些演算法或更多分析技術(分辨 edge, corner 再連接而成 curve)而達到<目標>tracking edge connectivity

造成輪廓(contour)難以辨別的問題是來自於，

1. 連續的類比訊號轉為數位離散訊號之後，我們不好定義兩線段"連續"的意義，導致在非水平或垂直線段上求其 edge 而遺失輪廓的情況。

2. 在某些 SNR 很小的圖片中，很多 edge 在 response function 結果後非常接近 threshold 使得 detector 誤判情況。

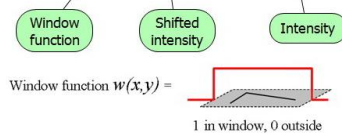
要<解決>這兩點問題作者提出要把圖形的 edge 給分類成 corner 以及 edge 兩種，讓 tracking edge 能輕易由 corner & edge 連接而成。

2. Moravec detector

利用<統計>的方式，將 window shift 一些距離後做成的 auto-correlation，去比較其結果。

$$E(u, v) = \sum_{x, y} w(x, y) [I(x+u, y+v) - I(x, y)]^2$$

$$E(u, v) = \sum_{x, y} w(x, y) [I(x+u, y+v) - I(x, y)]^2$$



Four shifts: $(u, v) = (1, 0), (1, 1), (0, 1), (-1, 1)$
Look for local maxima in $\min\{E\}$

Fig 1. Moravec detector 的數學運算演算法

很容易理解其中(Fig 1) $E(u, v)$ 的概念，是當 (x, y) 是 corner 時， $E(u, v)$ 值 [shift (u, v) vector] 會非常的大；同理在 isolated point (noise) 也會非常大，但是 (x, y) 在 edge 時候，shift (u, v) 若是朝向切線 (tangent) 則變得很小，於是我們就利用這點 edge 和 corner 不同特性來分辨究竟 (x, y) 是 edge or corner [比較其 $\min\{E(u, v)\}$]

至於當 (x, y) 不是 corner or edge 的少量變化波動時候，其 $\min\{E(u, v)\}$ 都遠比 corner 來的小，可以很輕易的找到 corner 的地點。這種 Moravec detector 很像是在某一點附近作 local auto-correlation 的運算方式。

3. Problem of Moravec detector

(a) Noisy response due to a binary window function (因為 window function 權重係數沒有和 neighbor 距離相關。)

(b) Only a set of shifts at every 45 degree is considered (因為位移 (u, v) 只有數個 patterns。)

(c) Only minimum of E is taken into account (若是只分析 E(u,v) 最小值的比較法，很難辨別 "corner"、"edge"、"plane" 三大部分)

3.1 <sol> Noisy response due to a binary window function

因為 window is binary and rectangular function, 所以要將 Noise response 消除就利用 Gaussian 的 smooth circular window 增強權重係數。

$$w_{u,v} = e^{-\frac{(u^2+v^2)}{2\sigma^2}}$$

3.2 <sol> Only a set of shifts at every 45 degree is considered

- Consider all small shifts by Taylor's expansion

$$E(u, v) = \sum_{x,y} w(x, y) [I(x+u, y+v) - I(x, y)]^2$$

$$= \sum_{x,y} w(x, y) [I_x u + I_y v + O(u^2, v^2)]^2$$

$$\left\{ \begin{aligned} E(u, v) &= Au^2 + 2Cuv + Bv^2 \\ A &= \sum_{x,y} w(x, y) I_x^2(x, y) \\ B &= \sum_{x,y} w(x, y) I_y^2(x, y) \\ C &= \sum_{x,y} w(x, y) I_x(x, y) I_y(x, y) \end{aligned} \right.$$

根據線性代數的二項式公式 quadratic form 可以把 $E(u, v) \cong [u, v] M \begin{bmatrix} u \\ v \end{bmatrix}$ $M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$

3.3 <sol> Only minimum of E is taken into account

- A new corner measurement

接著在找到由 Tylor expansion 解決的 shift 角度問題後，若是只分析 E(u,v) 最小值的比較法，很難辨別 "corner"、"edge"、"plane" 三大部分，頂多能制定一個 threshold 去定義 corner 或是 non-corner。

我們發現當 M 線性代數中的二項式是影響 E(u,v) 大小的矩陣，若分析 M 的特徵值 $\lambda_1 \lambda_2$ 可以帶回去變化變數

$$E(u', v') = [u', v'] \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} u' \\ v' \end{bmatrix}$$

(等於旋轉了 u,v 座標軸)。馬上就知道

- (a) <物理意義> 就是當 shift(u,v) 時候，若 $\lambda_1 \lambda_2$ 都很大，表示 E(u,v) 不管 shift(u,v) 角度為何，都會讓 E(u,v) 很大，也就是 "corner" 的特徵。
- (b) 同理，當 eigenvalues $\lambda_1 \lambda_2$ 一大一小時，表示當 shift(u,v) 其中一變數會使得 E(u,v) 很小，也就是 "edge" 的特徵。
- (c) 同理，當 eigenvalues $\lambda_1 \lambda_2$ 都很小時，表示當 shift(u,v) 不管多少都會使得 E(u,v) 很小，也就是 "plane" 的特徵。

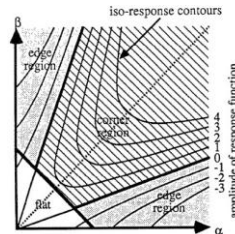
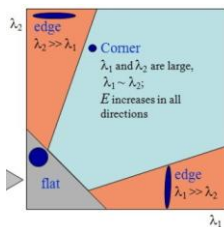


Fig 2. 特徵值的大小所代表 edge/corner/flat 意義 Fig 3 可以發現當 R>0 時是表示 (x,y) 接近 corner · R<0 時則是 edge or flat。

但是要<定義>這個 threshold 3 條線，就是由 $R = \det M - k(\text{trace } M)^2$

$$\left\{ \begin{aligned} \det M &= \lambda_1 \lambda_2 \\ \text{trace } M &= \lambda_1 + \lambda_2 \end{aligned} \right\} \quad (k - \text{empirical constant, } k = 0.04-0.06)$$



Matlab 寫作程式規範 & ImMaskProcess

```

%{
=====
寫作/修改時間&作者:
    [9/4] [9/5] 洗達
寫作理由&目標(摘要):

寫作方式&I/O& Display:

Rank (simple,med,hard) :

進階應用&改善:

所用到的關鍵函數:

=====
%}
Function []=xxx();    %名稱規範(目標[GetEdge]_等級[simple,med,hard])
%=第一步的事(越高階觀念越好)
    1.設參數(im_from_user_1)    %意義
    -----                    %參數意義
    --- ---
    ----
    %=====test/debug1=====
    ----
    %===== test/debug1 =====

%=第二步做的事(越高階觀念越好)
    %==更細節做的事
    -----
    --- ---
    ----
    %=====test/debug2=====
    ----
    %===== test/debug 2=====

%=第三步做的事(越高階觀念越好)
    ---

```