

TR-84-002

從多方面來鑑別一三度空間物體

計畫主持人：黃 俊 雄

參與人員：助理員：洪 德 成

研究生助理：李 文 進

計畫編號：NSC 72-o4o4-E001-R02

執行期限：七十二年三月一日至七十三年二月二十九日

中研院資訊所圖書室



3 0330 03 00045 4

0045

書 考 參
—
借 外 不

摘 要

本報告提出一三度空間物體辨認的一新原理：即從各方面來照像而由此多張照片綜合用統計方法來辨認。爲了實際驗證，我們用一個六面體做實驗。先研究各特徵，如邊長、角度等，隨不同視角而轉變，再把此轉變或然率化而得到特徵的理論分佈函數。此分佈函數與一未知物體多次照像後得的特徵樣本比較看是否類似，如類似則判定此物體爲原來六面體。類似的準則是用 Cramer-von Mises statistic 來判定。本研究所發展出的方法很容易推廣到其他多面物體。

目 錄

一、前言	1
二、計畫目標	1
三、研究成果	2
四、六面物體的照像及影像處理	2
1 照像	2
2 影像處理	4
3 特徵抽取	5
五、六面物體之理論特徵的或然率化	5
1 理論特徵	6
2 或然率化	14
3 計算機自動轉換	14
六、辨認	15
1 統計上有關的辨認方法	15
2 少數樣本的修正辨認	17
七、實驗報告	17
八、結論	29
九、參考書目	30
十、附錄一	31
附錄二	
附錄三	

一、前言：

三度空間物體的辨認太難了，至今仍為人工智慧的研究主題。一般方法是用知識基系統加上適當建立的影像模式來解說 (Ballard and Brown [1], Barr and Feigenbaum [2], Brady [3])。最先進的研究是模擬人的視神經操作 (Marr [4]) 可惜還沒有實際有效的成果出現。傳統的統計識別方法在三度空間物體幾乎難予處理，至今沒人提出有效方法。Invariant Moments 沒有效力，Fourier Descriptor 亦是好不了多少 [5,6]。本研究提出另一種有效的統計方法來處理此一問題，方法很合乎一般直覺的看法：即要了解一個三度空間物體就要從各個角度去看它，然後把看到的結果綜合聯繫起來。

本研究提出的方法是看一個物體的一些特性，隨著不同視角而轉變，並把此轉變記錄下來，然後用或然率的方法加以隨機化而得到變化的分配函數。此分配函數可與照許多張像後得到的某物體特性相比較，而判斷出照片中的某物體是否為原來考慮的物體。有關此方法的理論細節已在 Huang and Tou [7] 內解說了，本研究報告旨在說明實驗的各項細節及所遭遇的困難及其成果。依據此次實驗研究本方法有其特定價值並可用於工業產品檢定之用，只要將 resolution of images 提高即可 (實驗是用 128x128 resolution，照明設備不好故所得結果有些地方不理想，然而大致說來是不錯的)。

二、計畫目標：

本計畫目標是對一平行六面體做一辨認實驗，先探討其特徵的理論分佈再大規模做實驗，看不同六面體是否能區分開；原來六面體是否能接受不區分。

三、研究成果：

本研究已把區分六面體所需要的技術發展成熟，程式附在報告後面。實驗共做了十六次，每次三十張照片，特徵共有九個。不同物體皆可區分，原來物體有時不能接受，主要是因照片的 resolution 太低及照像處理技術不理想以致干擾太大。

四、六面物體的照像及影像處理：

資訊研究所計算機設備為 PDP-11/70 為 minicomputer，每一 user 只有 64 K bytes，用此有限記憶來處理照片是顯然太少的，我們是為此困難浪費不少時間。

1 照像：一個用木塊自鋸的平行六面體放在地上，然後用 Hamamatsu T.V. Camera 由上傾斜往下照它，如圖 1 所示。在圖 1 中， $y_0 = 20.2$ cm, $z_0 = 115.5$ cm, $f = 16$ cm。令六面體之下面中心點為坐標之原點，其八個端點為

$$\begin{aligned} V_1 &= (0, -b, h), & V_2 &= (c, -d, h), \\ V_3 &= (0, b, h), & V_4 &= (-c, d, h), \\ V_5 &= (0, -b, 0) & V_6 &= (c, -d, 0), \\ V_7 &= (0, b, 0) & V_8 &= (-c, d, 0). \end{aligned}$$

此處 $b=c=4.5$ cm, $d=1$ cm, $h=4.3$ cm 照片之 resolution 為 128×128 ，每個 pixel 佔一個 byte 為 256 gray level。相機內有些小污點不致影響相片太大。照明設備是用 200 燭光的燈斜照，背景為藍底。平行六面體上有些面塗上黑色以利於取像，由於相機是放在主機房內，常有人走動，照像的干擾較大。利用公式 $\frac{1}{p} + \frac{1}{g} = \frac{1}{f}$ 及鏡頭修正， $p=233$, $g=18.5$ 得 $f=17.14$ 經由實驗 12.3 cm 對應 128 條，故兩相

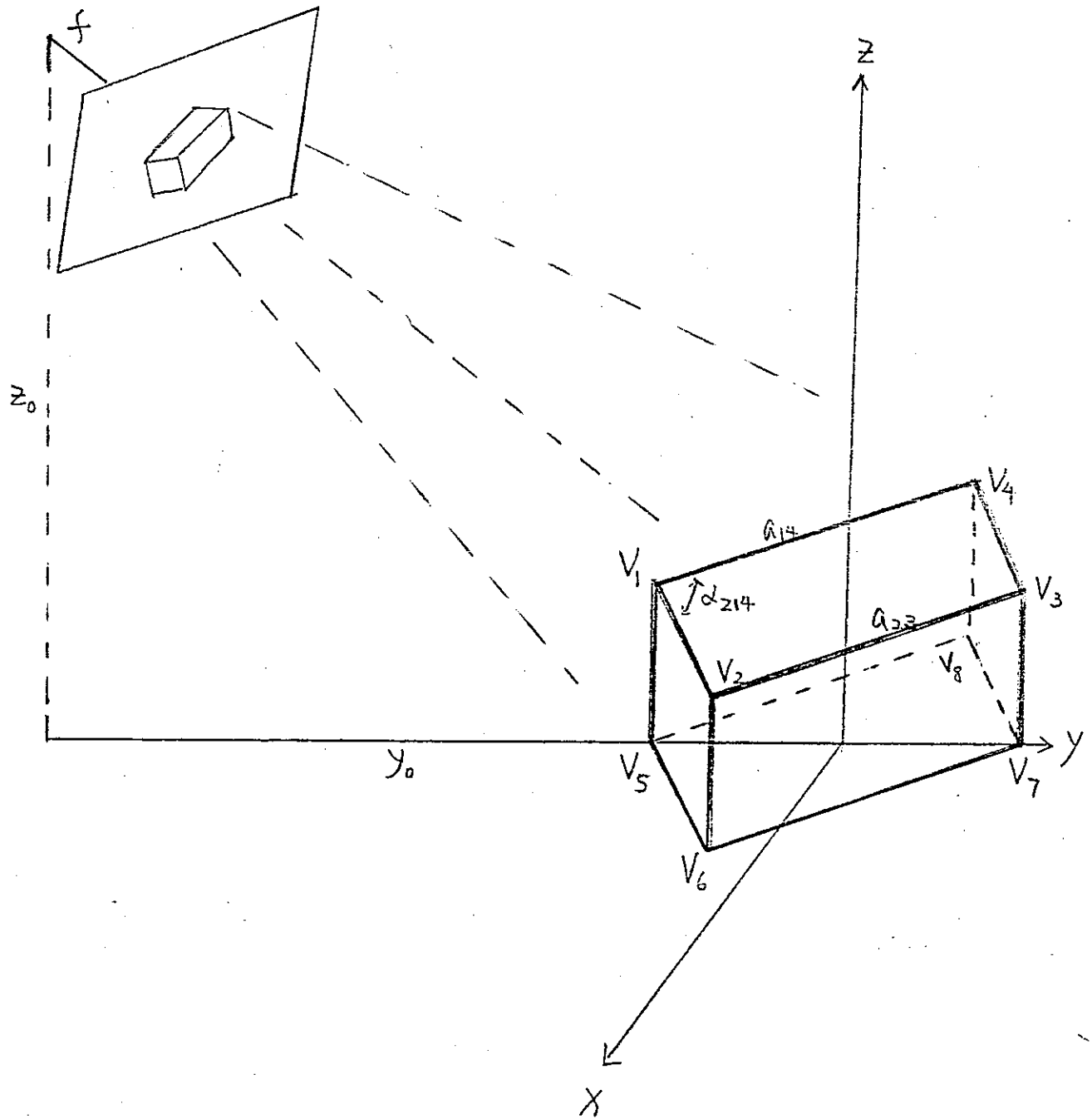


圖 1. 平行六面體之照相。

鄰 pixel 之距為 0.0076297cm 。Hamamatsu 是用 DMA 接到 unibus, 16bits 傳送一次 (即兩 pixels 一起 packed), 其手冊上有詳細說明 interface 及隨帶之照像 routine。最高 resolution 為 1024×1024 通常為 256×256 及 512×512 。又此相機可取 binary pictures, driver 還在 test 中。

在圖 1 中, 以 Z 為軸旋轉六面體得一隨機視角而照像, 這樣共照了 30 張, 每張佔 128×128 bytes, 須相當大的空間來存放。

2 影像處理: 由於只用 128×128 resolution, 故有些誤差可能顯著, 這是由於 PDP-11/70 的限制使然。影像處理的工作程序是: Take raw picture \rightarrow Edge detection \rightarrow Thinning \rightarrow Corner finding \rightarrow Line fitting \rightarrow Vertice calculating.

要找出六面體影像的邊及端點是不簡單的工作。首先如果燈光明明不好, 則影像內有物體之陰影, 或干擾反光等不良現象會發生。再次由於 resolution 低, 找邊誤差大。我們用 gradient 方法來做 edge detection, threshold 自調, 找出的 edge 常很粗故需加以 thinning。Thinning 方法見 Huang and Lee [8]。找出 edge 的另一方法也試過, 是用 Nevatia and Babu [9] 之 templates, 效果較精確, 但所需記憶體太大, 不適合在 11/70 上跑。

接著要找 Corner, 我們用 Davis [10] 之方法。其計算步驟如下:

(1) 用 chain code 算出每點的相鄰角度的差, 如差接近零則視為直線 (2) 計算 $\cos\theta$ 值, θ 為相鄰角差對任意 $\cos\theta > 0.96$, threshold, 之 pixel, 向左找個 range 計算在這個區域內所有 pixel 的 $\cos\theta$, 找出 $\max(\cos\theta)$, 如果 $\max(\cos\theta) \geq \text{threshold}$ 表示該 pixel 附近沒有 corner 發生。(3) 去掉假的 corner: 在可能發生曲折的位置

，附近也常有其他的點會使人誤會為曲折點發生。那就在適當區域內找出 $\cos\theta$ 最大值而確定這一值的發生位置才是真正 corner 產生的位置(4)找出三線相交的中間點是較易找(5)經過 thinning 後之 edge 常彎曲不理想，故須加以修正所得之 corner 點。修正方法是用直線適合法 (Line Fitting)。

由上面找出的 corner 點來劃分直線區域，每線區域並不成一直線故用最小平方法來做直線適合計算。由這些新的估計直線來重新算出其交點，這些交點即為修正後的 corner 點，這些點較為精確可靠。

3. 特徵抽取：特徵抽取是要花點時間考慮的，因為抽取的特徵要能反映出物體的特性，同時要能容易算出其隨不同視角而變化的情形。抽出的特徵如下：

- (1) 頂面之四邊長和。
- (2) 頂面之任兩鄰邊長之差之絕對值之和。
- (3) 頂面之任兩對邊長之差之絕對值之和。
- (4) 頂面之任兩鄰角之差之絕對值之和。
- (5) 頂面之任兩對角之差之絕對值之和。
- (6) 下面之兩底邊長和（如只看到一底邊就取一值）。
- (7) 下面之兩底邊長之差之絕對值（如只有一邊就取 0 值）。
- (8) 下面垂直邊長之和。
- (9) 下面兩相鄰垂直邊長之差之絕對值之和。

本章所提之各種計算皆列在附錄一。

五、六面物體之理論特徵的或然率化：

或然率化爲一抽象的觀念，實際上的應用是在大規模做實驗時才有。我們考慮從隨機的視角來看六面體，就需要照許多張照片了。從統計觀點看照多少張才算足夠呢？這看運氣及實驗設計了。我們在此由統計理論上應該每次要照六七十張，但最後只照三十張，這是因爲機器小且實驗環境不理想。

1 理論特徵：由圖 1 及 Huang and Tou [7] 我們知由 V_1 及 V_2 所成之邊長 $|a_{12}|$ 其在照片中之投影長爲 $|a_{12}^i|$ 且

$$|a_{12}^i| = \left\{ f_0^2 \frac{A_1^2 + B_1^2}{[(b \cos\theta + y_0) \cos\phi - (h - Z_0) \sin\phi]^2 [(c \sin\theta + d \cos\theta + y_0) \cos\phi - (h - Z_0) \sin\phi]^2} \right\}^{\frac{1}{2}}$$

where $\phi = \frac{-\pi}{6}$ and θ is rotational angle of object,

$$A_1 = [bc + (c \cos\theta + (b-d) \sin\theta) y_0] \cos\phi - [b \sin\theta + c \cos\theta - d \sin\theta] (h - Z_0) \sin\theta,$$

$$B_1 = (h - Z_0) [-c \sin\theta + (b-d) \cos\theta].$$

$|a_{23}^i|$ \equiv 將 $|a_{12}^i|$ 內之 b 改成 $-b$ ，對應的 A_1, B_1 改成 A_2, B_2 。

$|a_{34}^i|$ \equiv 將 $|a_{12}^i|$ 內之 θ 換成 $\theta + \pi$ ，對應的 A_1, B_1 改成 A_3, B_3 。

$|a_{14}^i|$ \equiv 將 $|a_{23}^i|$ 內之 θ 換成 $\theta + \pi$ ，對應的 A_2, B_2 改成 A_4, B_4 。

由這四個邊長之公式我們很容易寫成計算程式再用這程式來計算前五個理論特徵：

(1) 頂面之四邊長之和

$$f_1(\theta) = |a_{12}^i| + |a_{23}^i| + |a_{34}^i| + |a_{14}^i|。$$

(2) 頂面之任兩鄰邊長之差之絕對值之和

$$f_2(\theta) = ||a_{12}^i| - |a_{23}^i|| + ||a_{23}^i| - |a_{34}^i||$$

$$+ ||a_{34}^i| - |a_{14}^i|| + ||a_{14}^i| - |a_{12}^i||。$$

(3) 頂面之任兩對邊長之差之絕對值之和

$$f_3(\theta) = || a_{12}^i | - | a_{34}^i || + || a_{23}^i | - | a_{14}^i ||$$

由公式可計算夾角之 cos 值。令 $\alpha_{123}^i = \cos$ (angle between a_{12}^i and

$$a_{23}^i) = \frac{A_1 A_2 + B_1 B_2}{\sqrt{(A_1^2 + B_1^2)(A_2^2 + B_2^2)}};$$

$$\text{同理 } \alpha_{214}^i = \frac{A_1 A_4 + B_1 B_4}{\sqrt{(A_1^2 + B_1^2)(A_4^2 + B_4^2)}},$$

$$\alpha_{234}^i = \frac{A_2 A_3 + B_2 B_3}{\sqrt{(A_2^2 + B_2^2)(A_3^2 + B_3^2)}},$$

$$\alpha_{341}^i = \frac{A_3 A_4 + B_3 B_4}{\sqrt{(A_3^2 + B_3^2)(A_4^2 + B_4^2)}}.$$

$$(4) \text{ 頂面鄰角差之和 } f_4(\theta) = | \alpha_{214}^i - \alpha_{123}^i | + | \alpha_{234}^i - \alpha_{341}^i | \\ | \alpha_{123}^i - \alpha_{234}^i | + | \alpha_{341}^i - \alpha_{412}^i |.$$

$$(5) \text{ 頂面對角差之和 } f_5(\theta) = | \alpha_{214}^i - \alpha_{234}^i | + | \alpha_{123}^i - \alpha_{341}^i |.$$

從圖 1 中知 $|a_{56}^i| \equiv |a_{12}^i|$ 中令 $h=0$;

$|a_{58}^i| \equiv |a_{14}^i|$ 中令 $h=0$; $|a_{67}^i| \equiv |a_{23}^i|$ 中令 $h=0$;

$|a_{78}^i| \equiv |a_{34}^i|$ 中令 $h=0$. 又

$$|a_{15}^i| = \text{SQRT} \left\{ f_0^2 \left[\frac{A_9^2 + B_9^2}{((b \cos \theta + y_0) \cos \phi - (h - z_0) \sin \phi)^2 ((b \cos \theta + y_0) \cos \phi + z_0 \sin \phi)^2} \right] \right\}$$

where $A_9 = h b \sin \theta \sin \phi$ and $B_9 = (b \cos \theta + y_0) h$.

$|a_{37}'| \equiv |a_{15}'|$ 內 改成 $-b$.

$$|a_{26}'| = \text{SQRT} \left\{ f_0^2 \left[\frac{A_{11}^2 + B_{11}^2}{((c \sin\theta + d \cos\theta + y_0) \cos\phi - (h - z_0) \sin\phi)^2 ((c \sin\theta + d \cos\theta + y_0) \cos\phi + z_0 \sin\phi)^2} \right] \right\}$$

where $A_{11} = (c \cos\theta - d \sin\theta) h \cos\phi$,

$B_{11} = (c \sin\theta + d \cos\theta + y_0) h$.

$|a_{48}'| \equiv |a_{26}'|$ 內之 c 改成 $-c$; d 改成 $-d$ 。

當我們轉動六面體時，相機內只看到某些面及邊而看不到另一面及邊；轉到某角度時相機只看到兩面而非三面，再轉一下又可看到三面了。這某角度可由下公式計算出來：由圖2之幾何原理得角度 θ 滿足

$$f(\theta) = \sin\delta - \frac{y_0}{b} \sin(\theta + \delta) + \frac{y_0}{\sqrt{c^2 + d^2}} \sin\theta = 0$$

where $\delta = \tan^{-1}(c/d)$. 這可用 Newton method 解根得 $\theta = -2.2496$ 及 $.9275$ 。

即當原本轉角在 5.3557 及 5.3912 中間或在 3.8131 及 3.8416 之間時相機只看到兩面。同樣由圖2之物體再旋轉約 90 度，又會發生同樣只見兩面，此時解類式上面公式之根 $\theta = -0.7, 2.4701$ 。

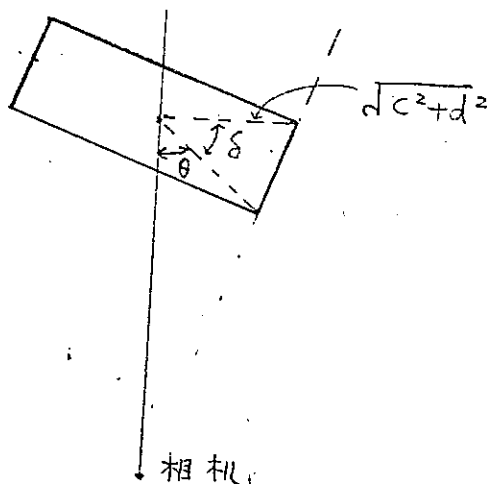
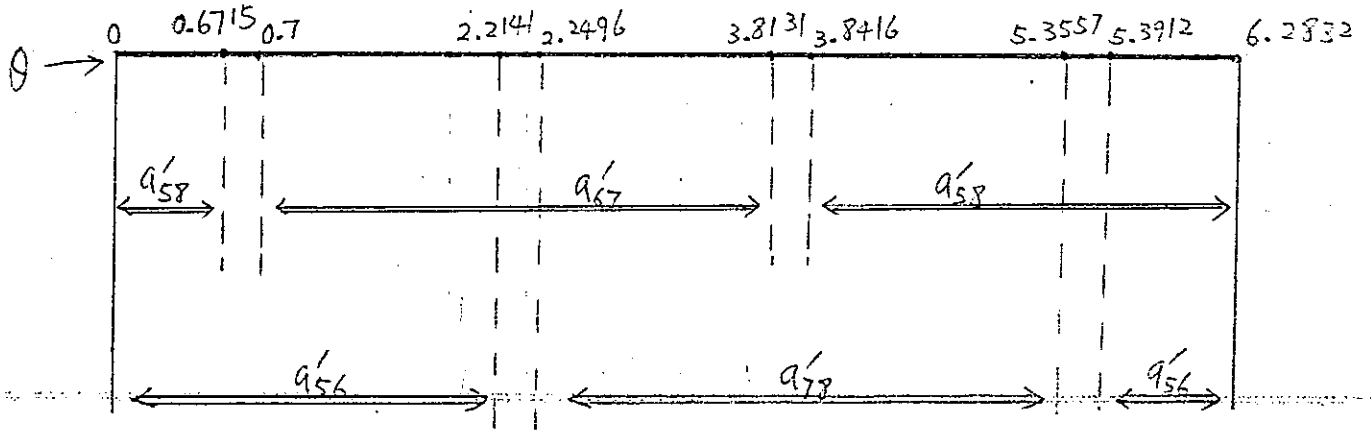


圖2. 計算只見兩面之轉角。

當 θ 變動時，相機只見到某些邊，這可由下表表示出來。



舉例：當 θ 在 0.7 和 2.2141 中時，相機只看到 a'_{67} 及 a'_{56} ，當 θ 在 2.2141 及 2.2496 中間時，相機只看到 a'_{67} 。由上表我們可看出對應的垂直邊在那些角度看得到在那些視角看不到。如此我們可計算下面之特徵了：

(6) 下面之兩底邊長和

$$f_6(\theta) = \begin{cases} |a'_{58}| + |a'_{56}| & \theta \text{ in range 1} \\ |a'_{56}| & \theta \text{ in range 2} \\ |a'_{67}| + |a'_{56}| & \theta \text{ in range 3} \\ |a'_{67}| & \theta \text{ in range 4} \\ |a'_{67}| + |a'_{78}| & \theta \text{ in range 5} \\ |a'_{78}| & \theta \text{ in range 6} \\ |a'_{58}| + |a'_{78}| & \theta \text{ in range 7} \\ |a'_{58}| & \theta \text{ in range 8} \\ |a'_{58}| + |a'_{56}| & \theta \text{ in range 9} \end{cases}$$

where range1 — range9 are from intervals of θ in the above table.

(7) 下面之兩底邊長差之絕對值

$$f_7(\theta) = \begin{cases} |a_{58}| - |a_{56}| & \theta \text{ in range 1} \\ 0 & \theta \text{ in range 2} \\ |a_{67}| - |a_{56}| & \theta \text{ in range 3} \\ \vdots & \vdots \end{cases}$$

(8) 下面垂直邊長之和

$$f_8(\theta) = \begin{cases} |a_{48}| + |a_{15}| + |a_{26}| & \theta \text{ in range 1} \\ |a_{15}| + |a_{26}| & \theta \text{ in range 2} \\ |a_{15}| + |a_{26}| + |a_{37}| & \theta \text{ in range 3} \\ |a_{26}| + |a_{37}| & \theta \text{ in range 4} \\ |a_{26}| + |a_{37}| + |a_{48}| & \theta \text{ in range 5} \\ |a_{37}| + |a_{48}| & \theta \text{ in range 6} \\ |a_{37}| + |a_{48}| + |a_{15}| & \theta \text{ in range 7} \\ |a_{48}| + |a_{15}| & \theta \text{ in range 8} \\ |a_{48}| + |a_{15}| + |a_{26}| & \theta \text{ in range 9} \end{cases}$$

(9) 下面兩相鄰垂直邊長之差之絕對值之和

$$f_9(\theta) = \begin{cases} \left| |a_{48}| - |a_{15}| \right| + \left| |a_{15}| - |a_{26}| \right| & \theta \text{ in range 1} \\ \left| |a_{15}| - |a_{26}| \right| & \theta \text{ in range 2} \\ \left| |a_{15}| - |a_{26}| \right| + \left| |a_{26}| - |a_{37}| \right| & \theta \text{ in range 3} \\ \vdots & \vdots \end{cases}$$

以上九個特徵其隨 θ 而變化之情形對於圖 3。

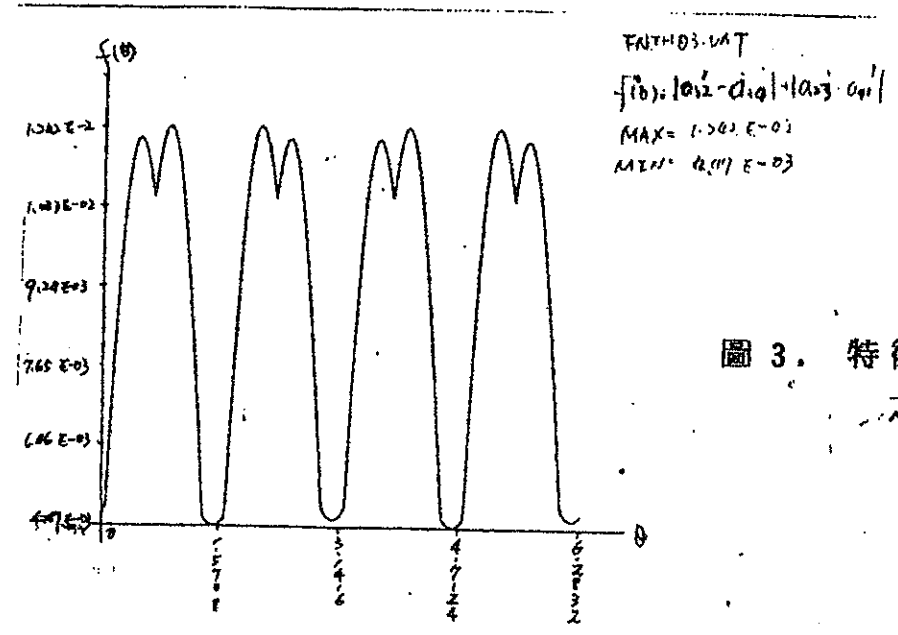
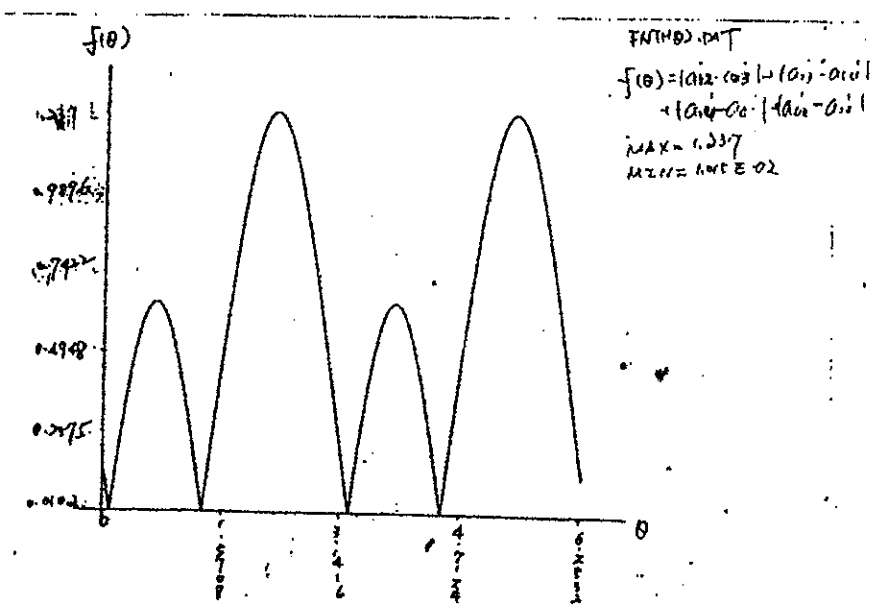
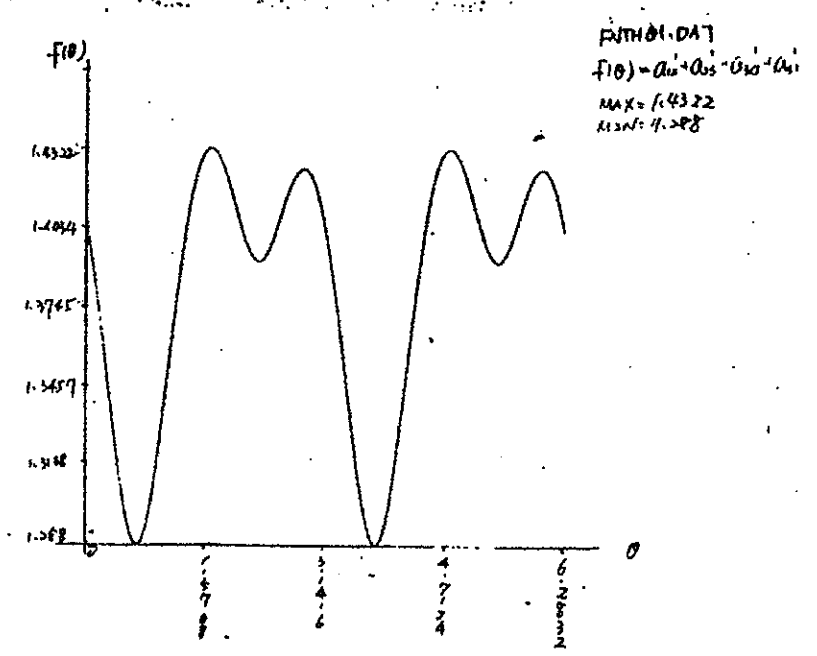


圖 3. 特徵隨轉角之變化。

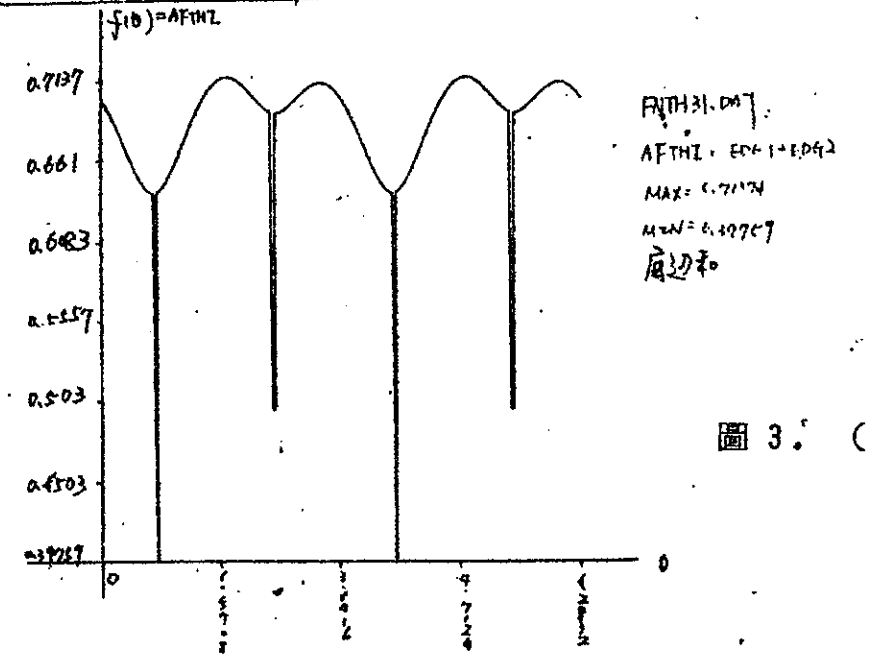
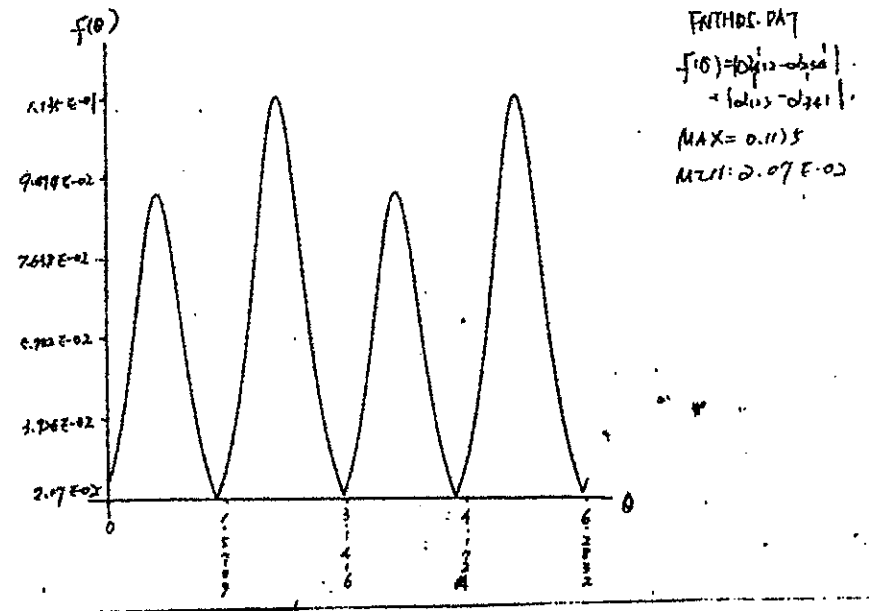
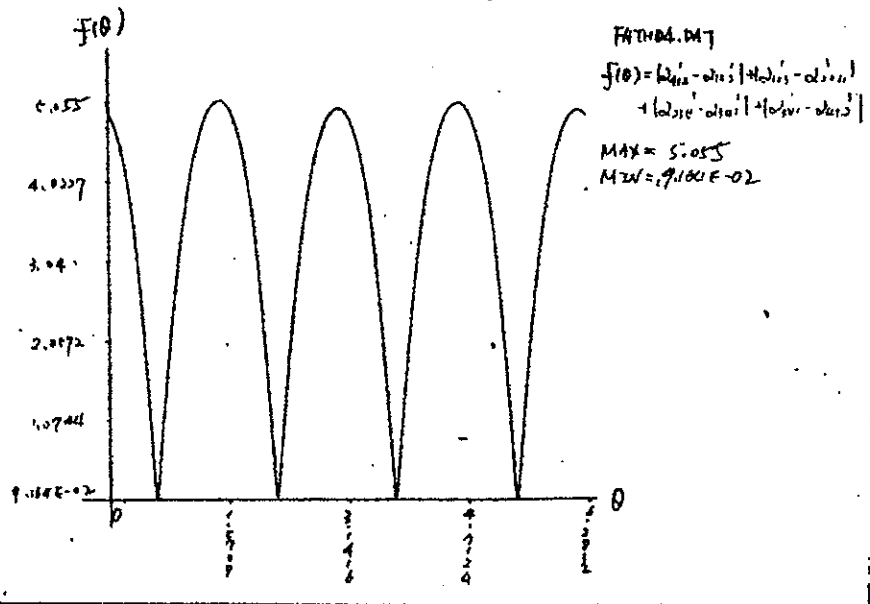
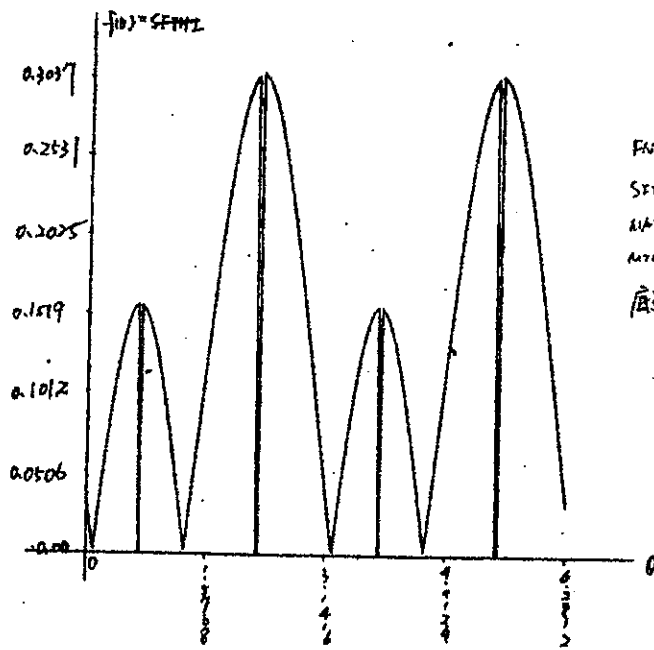
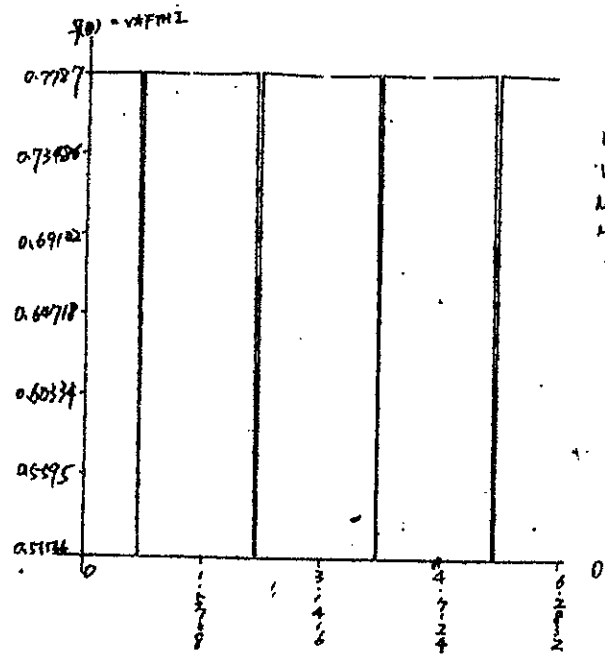


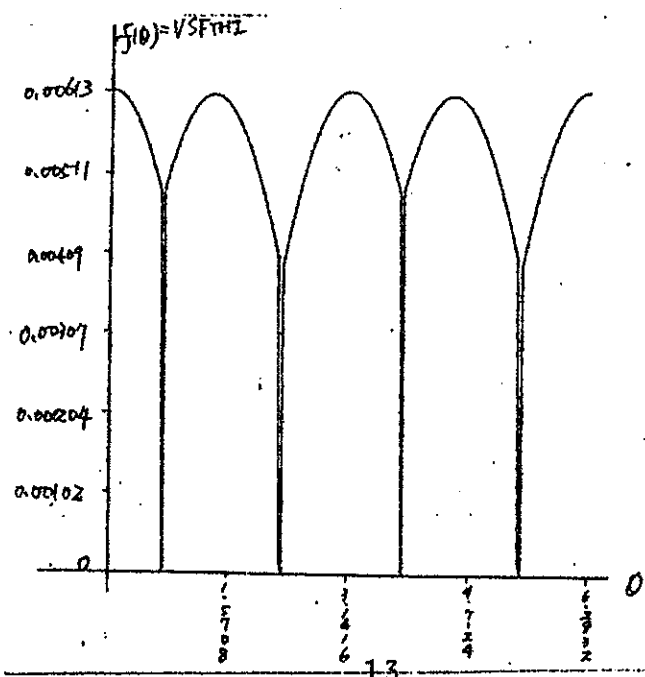
圖 3. (續)



FNTH33.DAT
 $SFTHZ = \text{ABS}(2061 - \text{EPR}2)$
 $\text{MAX} = 0.1127$
 $\text{MIN} = 0.0000$
 底辺差



FNTH33.DAT
 $VAFTHZ = \text{LEB}1 - \text{LEB}2 + 203$
 $\text{MAX} = 0.7796$
 $\text{MIN} = 0.5166$
 底辺差



FNTH34.DAT
 $VSFTHZ = \text{ABS}(V201 - V202)$
 $\quad + \text{ABS}(V202 - V203)$
 $\text{MAX} = 6.13 \text{E}3$
 $\text{MIN} = 7.10 \text{E}-07$
 底辺差

圖 3. (續)

2 或然率化：由圖 3 我們可看出 θ 由 0 變到 $2\pi=6.2832$ 時各個特徵的變化。因為變化是多重曲線故要或然率化很難，只有用數值方法再借助計算機才達得到。令 θ 具有 uniform distribution，即旋轉角是隨機的均勻性亂轉（或觀測視角是隨機地定下）。由或然率學得 θ 之 density

$$P(\theta) = \begin{cases} \frac{1}{2\pi} & \text{if } 0 \leq \theta < 2\pi, \\ 0 & \text{otherwise.} \end{cases}$$

由理論特徵 $Y_i=f_i(\theta)$, $i=1, \dots, 9, Y_i$ 的分配函數為

$H_i(y_0)=P[Y_i \leq y_0] = \frac{1}{2\pi} \times \{ \text{所有對應 } y \leq y_0 \text{ 之 } \theta \text{ 點之集合之長度} \}$ 。

既然假設 θ 具有均勻分配，那麼在實際應用上我們要取多少樣本點才可得到一樣本分配，使樣本分配與均勻分配誤差小於一定範圍，比如百分之一？此問題之解答可由 Berry-Essen 定理 [11, P. 542] 令誤差 = ϵ ，則由定理， $\sigma^2 = \frac{\pi^2}{3} = 3.2897$, $\rho = \frac{\pi^3}{4} = 7.7516$ 。誤差滿足公式 $\epsilon = \frac{3\rho}{\sigma^3 \sqrt{n}}$ ，即 $n = \frac{9\rho^2}{\sigma^6 \epsilon^2}$ 。當 $\epsilon=0.01$ 時 $n=150000$ ，此數目太大實際上做不到。由其在我們做照片實驗 n 一定越小越好，這點是很難解決的困擾。

3 計算機自動轉換：我們把區間 $[0, 2\pi]$ 分成 1000 等分，叫 $\{\theta_i = i = 0, 1, \dots, 999\}$ ，再對每一特徵 k 計算 $\{f_k(\theta_j), j=0, \dots, 999\}$ 求 $\{f_k(\theta_j)\}$ 之 max 及 min。再把區間 $[\min, \max]$ 分成 91 等分，叫 $y_0 = \min, y_1, \dots, y_{91} = \max$ ，然後寫一程式來算 $H_k(y)$ 。計算的步驟如下：

(1) 假設 f_k 是先上升（ f_k 是先下降亦類似處理）； θ_j 一步一步增加。

- (2) 對每一 i 找出 j 使 y_i 落在 $f_k(\theta_j)$ 及 $f_k(\theta_{j+1})$ 之間。記 $d_i = \theta_j$;
如對某 i , 找不到 j 則增加 i , $d_i = d_{i-1}$.
- (3) 當對某 j 值使 $f_k(\theta_j) > f_k(\theta_{j+1})$ 時則 f_k 是向下降。令 j 繼續增加。
- (4) 每一 i (i 由大變小) 找出 j 使 y_i 落在 $f_k(\theta_j)$ 及 $f_k(\theta_{j+1})$ 之間, 記下 $e_i = \theta_j$; 如對某 i , 找不到 j , 則減小 i , $e_i = e_{i+1}$.
- (5) 當對某 j 使 $f_k(\theta_j) < f_k(\theta_{j+1})$ 時, f_k 是變成向上升。
- (6) 每一 i (由小變大) 找出 j 使 y_i 落在 $f_k(\theta_j)$ 及 $f_k(\theta_{j+1})$ 之間, 記下 $f_i = \theta_j$; 如對某 i , 找不到 j 則增加 i , $f_i = f_{i-1}$.
- (7) 計算 $d_i = d_i + f_i - e_i$ (對所有 i) 。
- (8) 假如 $\theta_j < 2\pi$ 則 GO TO(3) 不然 GO TO(9) .
- (9) 印出 d_i 及 y_i , $i=1, \dots, 91$ 完畢。

我們如此找出的 $H_k(y)$ 是間斷的 91 點, 不是連續, 所以用線性內插法來算連續的 $H_k(y)$ 。請參見圖 4 。

本章所討論的各式方法, 其程式是放在附錄二。

六、辨認：

辨認的理論很多, 有統計判別法、語法法、人工智慧法(語意法)等, 此處所談的統計判決法推廣到三度物體。

- 1 統計上有關的辨認方法: 統計的模式識別法理論可見 Tou and Gonzalez [12] 。本研究是把物體的每一特徵或然率化而得到特徵的分配函數。看特徵的分配函數就可區別不同物體。區別方法很多, 已在 Huang and Tou [7] 內討論過。

我們在此只用 Cramer-von Mises statistic 來區別物體, 由前我

Feature 1

Feature 3

Feature 5

Feature 7

1.28768003	0.00000000	0.00447000	0.00000000	0.02070000	0.00000000	v.vvvvvv0000	0.00000000
1.28928542	0.03692385	0.00455833	0.04509417	0.02173122	0.02212246	0.00337611	0.03119411
1.29089081	0.05108862	0.00464667	0.12907439	0.02276244	0.04615476	0.00675222	0.0399477
1.29249620	0.06302521	0.00473500	0.16822638	0.02379367	0.07002796	0.01012833	0.0518843
1.29410160	0.07305196	0.00482333	0.19305450	0.02482489	0.09406038	0.01335044	0.0611152
1.29570699	0.08291952	0.00491167	0.19910236	0.02585611	0.11602364	0.01688056	0.0722561
1.29731238	0.09119557	0.00500000	0.20308125	0.02688733	0.13607717	0.02025667	0.0821237
1.29891777	0.09724346	0.00508833	0.20721927	0.02791855	0.15613066	0.02363278	0.0931052
1.30052316	0.10504204	0.00517667	0.21119815	0.02894977	0.17634328	0.02700889	0.1021771
1.30212855	0.11108990	0.00526500	0.21517700	0.02998600	0.19416864	0.03038500	0.1142728
1.30373394	0.11825185	0.00535333	0.21931502	0.03101222	0.21215309	0.03376111	0.1232173
1.30533934	0.12414059	0.00544166	0.22329392	0.03204344	0.23045588	0.03713722	0.1343581
1.30694473	0.13002929	0.00553000	0.22711362	0.03307467	0.24589379	0.04051333	0.1451806
1.30855012	0.13607717	0.00561833	0.23109251	0.03410589	0.26133180	0.04388944	0.1560031
1.31015551	0.14117013	0.00570666	0.23507130	0.03513711	0.27533740	0.04726556	0.1671439
1.31176090	0.14721800	0.00579500	0.23920932	0.03616834	0.29141206	0.05064167	0.1773299
1.31336629	0.15215179	0.00588333	0.24334744	0.03719956	0.30414441	0.05401778	0.1873565
1.31497169	0.15708561	0.00597166	0.24732628	0.03823078	0.31926411	0.05739389	0.1983383
1.31657708	0.16217856	0.00606000	0.25130510	0.03926201	0.33247387	0.06077000	0.2091607
1.31818247	0.16822638	0.00614833	0.25512481	0.04029323	0.34552458	0.06414611	0.2203016
1.31978786	0.17220527	0.00623666	0.25910369	0.04132446	0.35921192	0.06752222	0.2312833
1.32139325	0.17713900	0.00632500	0.26324171	0.04235568	0.37130761	0.07089833	0.2422650
1.32299864	0.18223196	0.00641333	0.26722059	0.04338690	0.38324425	0.07427444	0.2530874
1.32460403	0.18605168	0.00650166	0.27119941	0.04441813	0.39534003	0.07765055	0.2640690
1.32620943	0.19225872	0.00659000	0.27533740	0.04544935	0.40632167	0.08102667	0.2750507
1.32781482	0.19623758	0.00667833	0.27931631	0.04648057	0.41714412	0.08440278	0.2873057
1.32942021	0.20021646	0.00676666	0.28329518	0.04751180	0.42844412	0.08777889	0.2982873
1.33102560	0.20515026	0.00685500	0.28727403	0.04854302	0.43926662	0.09115500	0.3103831
1.33263099	0.21024321	0.00694333	0.29109377	0.04957424	0.45040745	0.09453111	0.3224788
1.33423638	0.21404629	0.00703166	0.29523179	0.05060547	0.45947927	0.09790722	0.3333013
1.33584177	0.21820091	0.00711999	0.29936981	0.05163669	0.47141591	0.10128333	0.3452379
1.33744717	0.22329387	0.00720833	0.30334860	0.05266792	0.47969189	0.10465945	0.3574028
1.33905256	0.22711360	0.00729666	0.30732739	0.05369914	0.49131021	0.10803556	0.3703845
1.34065795	0.23220652	0.00738499	0.31210205	0.05473036	0.49942702	0.11141167	0.3821619
1.34226334	0.23618543	0.00747333	0.31719500	0.05576159	0.51056784	0.11478778	0.3966450
1.34386873	0.24143751	0.00756166	0.32133302	0.05679281	0.51948053	0.11816389	0.4095364
1.34547412	0.24525724	0.00764999	0.32515275	0.05782403	0.52950728	0.12154000	0.4225871
1.34707952	0.24923611	0.00773833	0.32929078	0.05885526	0.53762418	0.12491611	0.4364335
1.34868491	0.25321496	0.00782666	0.33342880	0.05988648	0.54637766	0.12829222	0.4507574
1.35029030	0.25926286	0.00791499	0.33820346	0.06091771	0.55575185	0.13166833	0.4646039
1.35189569	0.26340085	0.00800333	0.34329644	0.06194893	0.56563538	0.13504444	0.4794053
1.35350108	0.26722050	0.00809166	0.34934428	0.06298015	0.57454807	0.13842055	0.4954797
1.35510647	0.27135861	0.00817999	0.35332313	0.06401137	0.58346075	0.14179666	0.5133053
1.35671186	0.27533737	0.00826833	0.35746115	0.06504259	0.59173679	0.14517277	0.5314489
1.35831726	0.28027120	0.00835666	0.36144003	0.06607381	0.60064936	0.14854889	0.5505472
1.35992265	0.28425005	0.00844499	0.36525977	0.06710503	0.60956204	0.15192500	0.5766488
1.36152804	0.28822893	0.00853333	0.37114853	0.06813625	0.61751974	0.15530111	0.6075246
1.36313343	0.29427683	0.00862166	0.37735546	0.06916747	0.62659156	0.15867722	0.6336262
1.36473882	0.29841483	0.00870999	0.38133436	0.07019869	0.63550425	0.16205333	0.6395192
1.36634421	0.30239370	0.00879832	0.38547242	0.07122991	0.64362115	0.16542944	0.6446077
1.36794960	0.30621338	0.00888666	0.38945130	0.07226113	0.65269291	0.16880555	0.6497007
1.36955500	0.31130636	0.00897499	0.39724988	0.07329235	0.66176480	0.17218167	0.6555894
1.37116039	0.31624013	0.00906332	0.40138996	0.07432357	0.67051828	0.17555778	0.6606824
1.37276578	0.32037821	0.00915166	0.40536678	0.07535479	0.67959005	0.17893389	0.6665710
1.37437117	0.32531199	0.00923999	0.40950474	0.07638601	0.68754780	0.18231000	0.6716646
1.37597656	0.32945001	0.00932832	0.41746253	0.07741723	0.69661963	0.18568611	0.6771190
1.37758195	0.33517960	0.00941666	0.42128226	0.07844845	0.70457739	0.18906222	0.6836007
1.37918735	0.33931759	0.00950499	0.42733014	0.07947967	0.71364915	0.19243833	0.6896484
1.38079274	0.34441054	0.00959332	0.43337792	0.08051089	0.72351670	0.19581445	0.6945822
1.38239813	0.34934431	0.00968166	0.43735680	0.08154211	0.73258859	0.19919056	0.7007893
1.38400352	0.35427815	0.00976999	0.44356382	0.08257333	0.74166836	0.20256667	0.7057231
1.38560891	0.35841617	0.00985832	0.44929340	0.08360455	0.75248209	0.20594278	0.7116119
1.38721430	0.36414564	0.00994666	0.45550042	0.08463577	0.76378280	0.20931889	0.7178189
1.38881969	0.36939776	0.01003499	0.46043420	0.08566699	0.77285463	0.21269500	0.7237075
1.39042509	0.37433156	0.01012332	0.46759620	0.08669821	0.78272223	0.21607111	0.7307103
1.39203048	0.41762158	0.01021165	0.47348493	0.08772943	0.79577297	0.21944723	0.7377132
1.39363587	0.44849756	0.01029999	0.47921434	0.08876065	0.80977851	0.22282334	0.7437618
1.39524126	0.47252995	0.01038832	0.48542145	0.08979187	0.82457995	0.22619945	0.7498090
1.39684665	0.49353835	0.01047665	0.49131021	0.09082309	0.84893060	0.22957556	0.7558569
1.39845204	0.51343268	0.01056499	0.49958614	0.09185431	0.86198127	0.23295177	0.7617455
1.40005744	0.53141713	0.01065332	0.50547487	0.09288553	0.86977977	0.23632768	0.7697033
1.40166283	0.54959261	0.01074165	0.51343262	0.09391675	0.87375867	0.23970389	0.7757512
1.40326822	0.56853193	0.01082999	0.51948053	0.09494797	0.87789667	0.24308001	0.7838681
1.40487361	0.58572060	0.01091832	0.52552843	0.09597919	0.88188757	0.24645612	0.7897568
1.40647900	0.60354602	0.01100665	0.53746492	0.09701041	0.88585442	0.24983223	0.7977145
1.40808439	0.62073475	0.01109499	0.55544949	0.09804163	0.88999242	0.25320834	0.8058313
1.40968978	0.63760501	0.01118332	0.57359308	0.09907285	0.89397132	0.25658444	0.8128342
1.41129518	0.65670365	0.01127165	0.59364659	0.10010407	0.89779103	0.25996053	0.8196778
1.41290057	0.67580223	0.01135999	0.60956204	0.10113529	0.90176982	0.26333663	0.8287496
1.41450596	0.69665134	0.01144832	0.62961555	0.10216651	0.90781778	0.26671273	0.8376624
1.41611135	0.71684405	0.01153665	0.65046483	0.10319773	0.91195571	0.27008882	0.8457791
1.41771674	0.73755413	0.01162499	0.67370147	0.10422895	0.91593462	0.27346492	0.8548510
1.41932213	0.75983584	0.01171332	0.69757456	0.10526017	0.92198247	0.27684101	0.8647185
1.42092752	0.78768784	0.01180165	0.72176611	0.10629139	0.92787123	0.28021711	0.8749045
1.42253292	0.81681329	0.01188998	0.74818569	0.10732261	0.93391907	0.28359321	0.8858860
1.42413831	0.85787511	0.01197832	0.77969837	0.10835383	0.93901199	0.28696930	0.8968678
1.42574370	0.89989197	0.01206665	0.81566721	0.10938505	0.94601482	0.29034540	0.9099186
1.42734909	0.91389745	0.01215498	0.87009805	0.11041627	0.95381337	0.29372150	0.9239245
1.42895448	0.92901707	0.01224332	0.91386569	0.11144749	0.96193027	0.29709759	0.9407944
1.43055987	0.95002562	0.01233165	0.94394583	0.11247871	0.97402608	0.30047369	0.9629170
1.43216002	1.00000000	0.01242000	0.99996823	0.11351000	0.99996823	0.30385000	1.0000001

們知第 i 特徵之理論分配為 $H_i(y)$ ，而從 n 張照片上取出的第 i 特徵數值為 y_1, y_2, \dots, y_n 。令 $x_1 \leq x_2 \leq \dots \leq x_n$ 為 y_1, \dots, y_n 之 order statistic 及令 $Z_j = H_i(X_j)$ 。則 Cramer-von Mises statistic

$$W^2 \text{ 爲 } W^2 = n \int_{-\infty}^{\infty} (F_i(t) - H_i(t))^2 dH_i(t) = \sum_{i=1}^n \left(Z_i - \frac{2i-1}{2n} \right)^2 + \frac{1}{12n},$$

此處 $F_i(t) = \frac{\text{number of } X_j \leq t}{n}$ 。 W^2 之效力與其他方法類似。

2 少數樣本的修正辨認：由於照片不能取太多張，即 n 小，故 W^2 須加以修正。修正為

$$W_0^2 = \left(W^2 - \frac{0.4}{n} + \frac{0.6}{n^2} \right) \left(1 + \frac{1}{n} \right).$$

當第一型誤差之或然率 α 為 0.05 時，我們接受照片之物體為原來具有 $H_i(t)$ 之物體，假如

$$W_0^2 \leq 0.461 \quad \text{對所有 } i=1, \dots, 9 \text{ 皆成立。}$$

只要有一個 i 不成立則不接受照片之物體為原來之物體。假如 α 定為 0.10 則當 $W_0^2 \leq 0.347$ 對所有 $i=1, \dots, 9$ 皆成立時，我們接受照片之物體為原來之物體。

假如 n 足夠大，比如 $n=100$ 或 200 且照片之視角確為 random 則 α 取 0.10 應很適當。但我們因 n 很小，小於 50 且影像處理時某些角度看到的物體之特徵取不出來，加以有誤差故 α 通常取 0.05。

本章所提的方法其計算程式在附錄三。

七、實驗報告：

由前面圖 1 中平行六面體之規格為 $h=4.3$, $b=c=4.5$, $d=1$ (註： $y_0=202$, $Z_f=115.5$, $f_0=17.14$)。以此物叫 A 物為準，我們利用第五章之程算出九個特徵之分佈函數，其部份資料如圖 4 所示。

再次我們利用第四章所寫之程式對準 A 物照像，每次旋轉 A 物一小角度，共照了三十二張照片，跑了程式以後二張照片因干擾太大而報廢，我們得到九個特徵的數值（每個三十）。圖 5 (a) 是 A 物照了像及做 edge detection 後的結果。九個特徵的數值建成一個 file 再輸入到 CRAMER 程式內，這程式把每一特徵的 Creamer-von Mises statistic W_0^2 算出來並做檢定。結果是九個特徵內有 2 個在 $\alpha=.05$ 內接受 7 個在不接受。追究這 7 個特徵之不接受的原因是 resolution 太低及做影像處理物體背後見到的邊之誤差大。特徵 1 是頂面四邊和，而特徵 6 是底邊和，兩者誤差大。做 edge detection 時每個邊皆有誤差，尤其斜邊誤差較大，累積起來誤差就顯著，但相減時誤差就消掉。

再次考慮 B 物是一扁的平行六面體， $b=5.65, c=5.4, d=1.15, h=3.4$ ，這個物體太大，如 $f=16$ 時螢幕放不下，故調 $f=15$ 時才照得下完全物體。我們照了三十張照片（參見圖 5(b)）。跑了 CRAMER 取 $\alpha=0.05$ 。其結果是特徵 2, 4, 5, 7 接受，其他拒絕。特徵 2 是鄰邊差，這個接受是因為 B 物與 A 物之頂面同為接近正方形。特徵 4 及 5 是鄰邊角度差及對邊角度差，它們被接受是因為 B 物與 A 物之頂面角度同為直角。特徵 7 是底邊差，相減結果是與原 A 物之底邊差相似。其他特徵被拒絕表示 B 物與 A 物大有不同。

再次考慮 C 物是一梯形方塊，體積比 A 物大一些，它的八個頂是 $(0, 6.56, 4.0), (-2, 3, 1.9, 4.0), (0, -6.56, 4.0), (5, 8, -2.55, 4.0), (0, 6.56, 0), (-2.3, 1.9, 0), (0, -6.56, 0), (5.8, -2.55, 0)$ 。我們共照了 30 張照片（ $f=15$ ，其他參數不變）。圖 5 (c) 是它照了像及做 edge detection 的結果，跑了 CRAMER 後定 $\alpha=.05$ 除了特徵 6 其餘特徵被拒絕。由此結果可判定這 C 物與 A 物是差別太大。

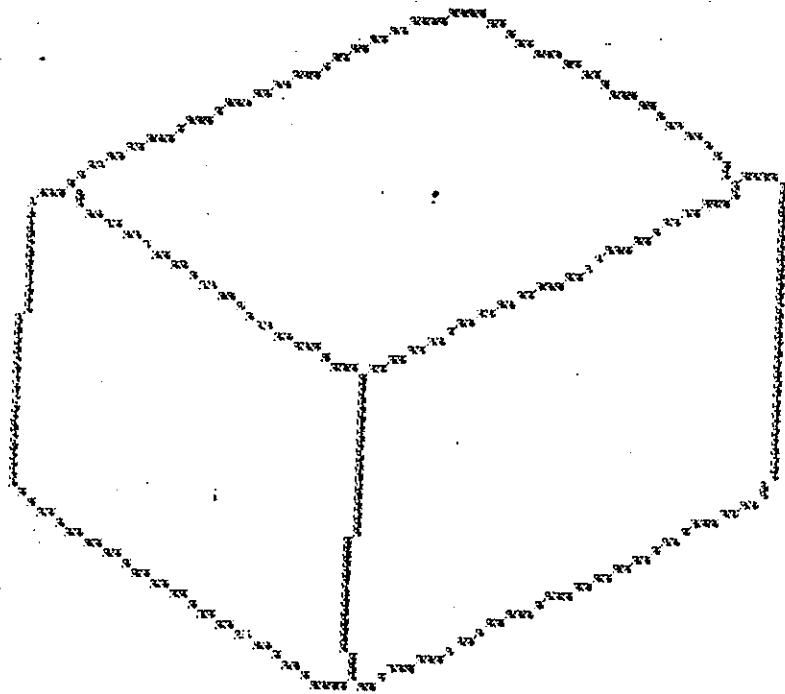
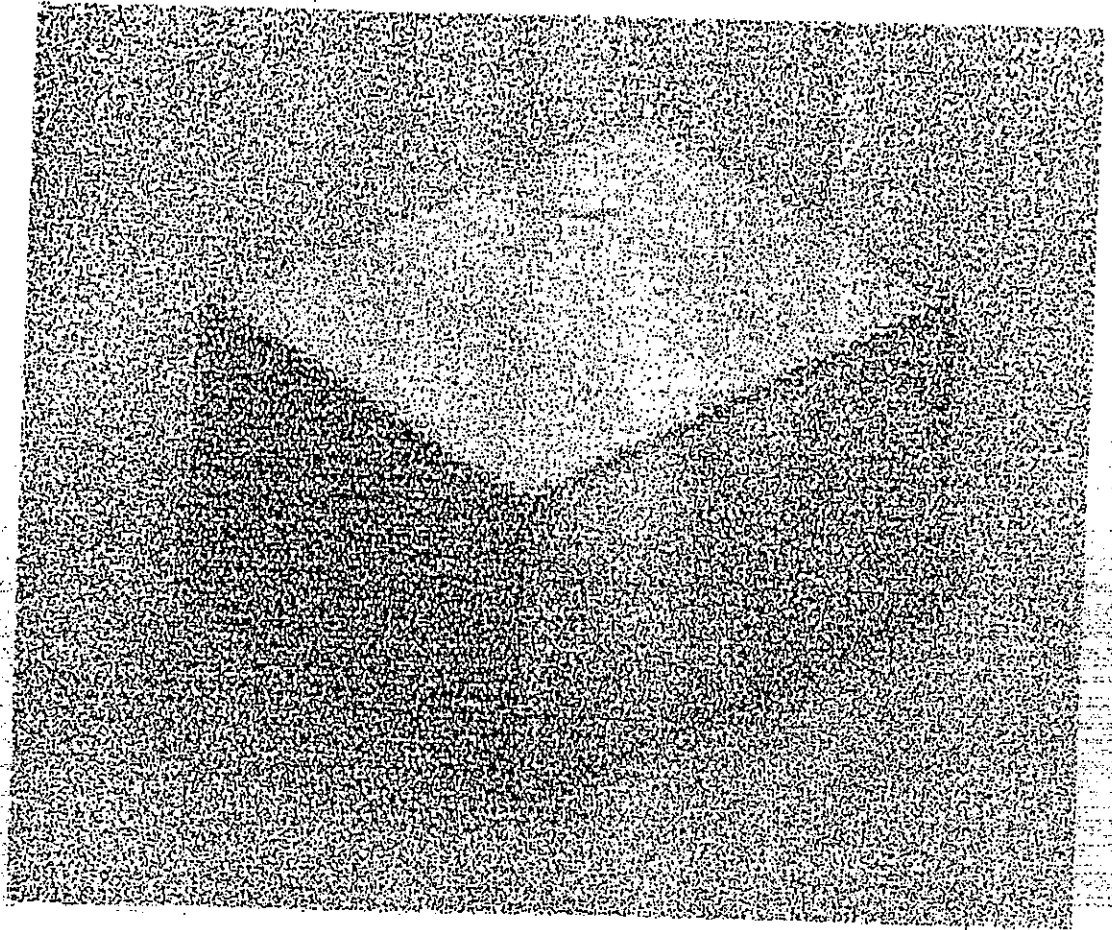
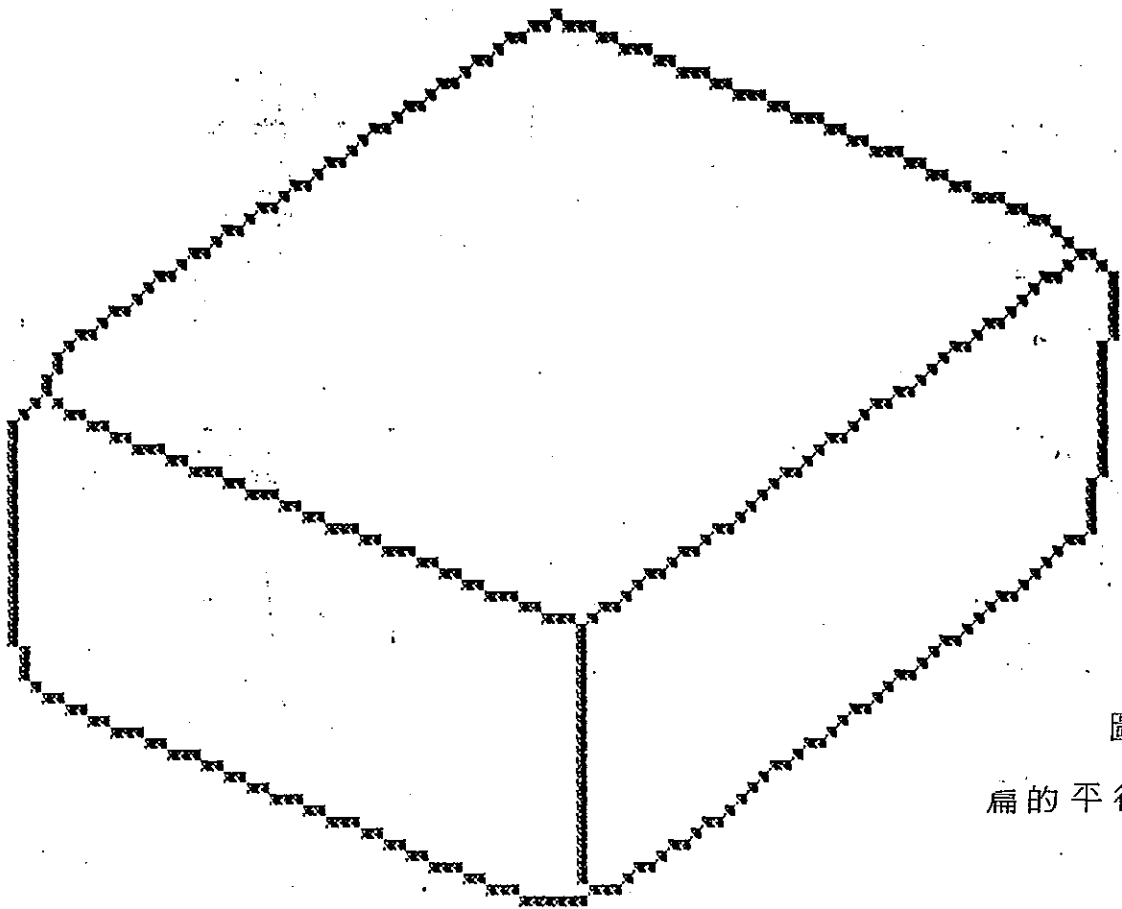
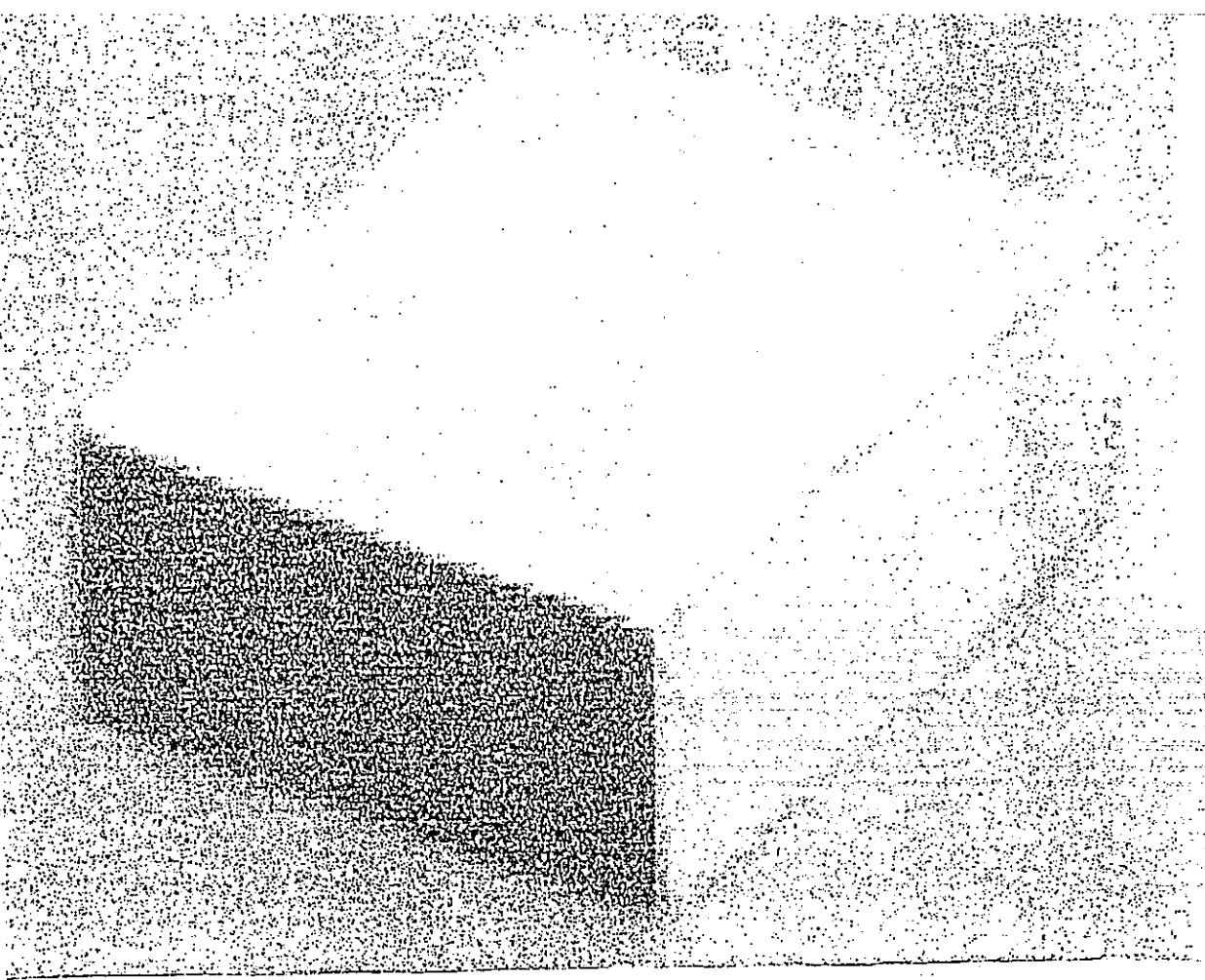


圖5(a). 原來平行六面體。



圖(b).

扁的平行六面體。

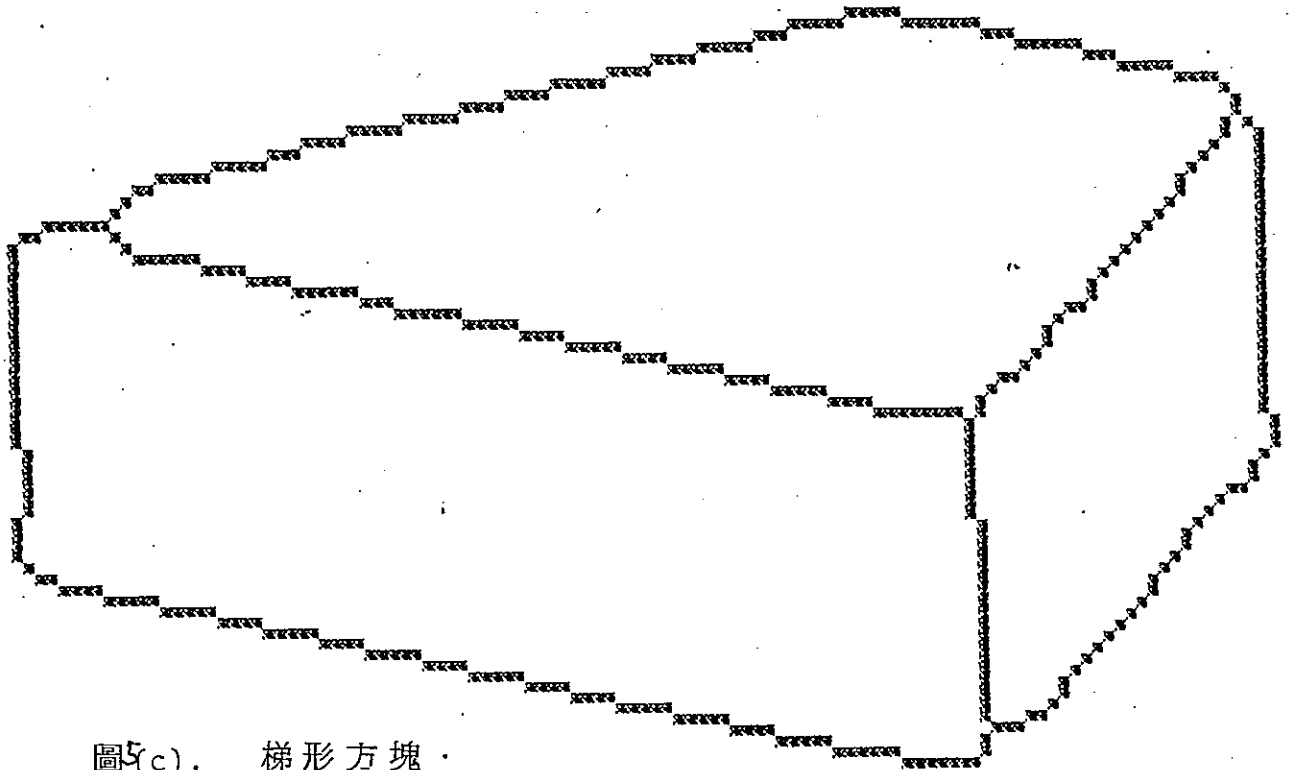
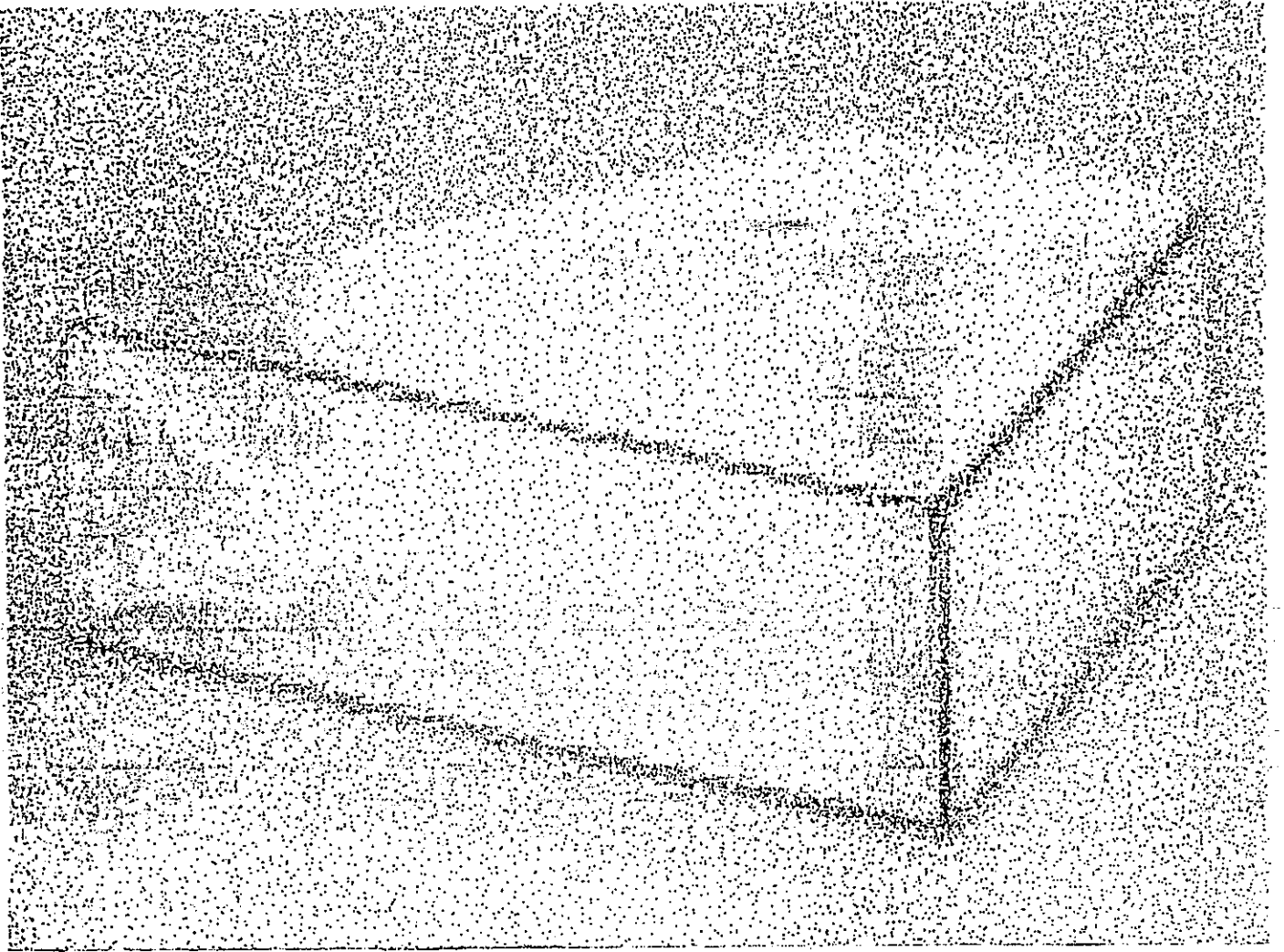


圖5(c). 梯形方塊。

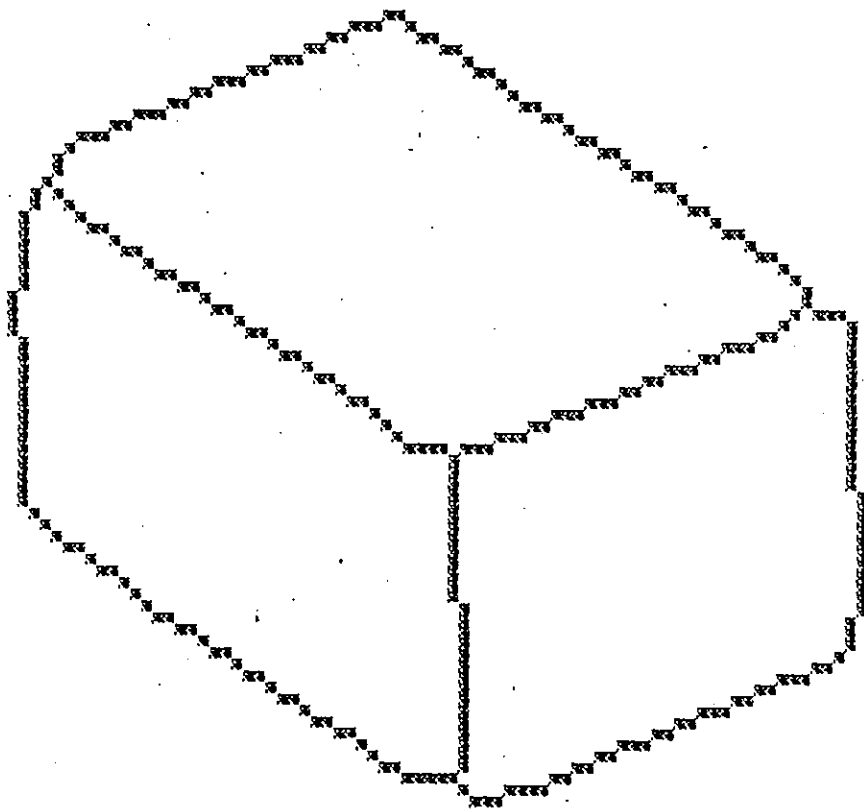
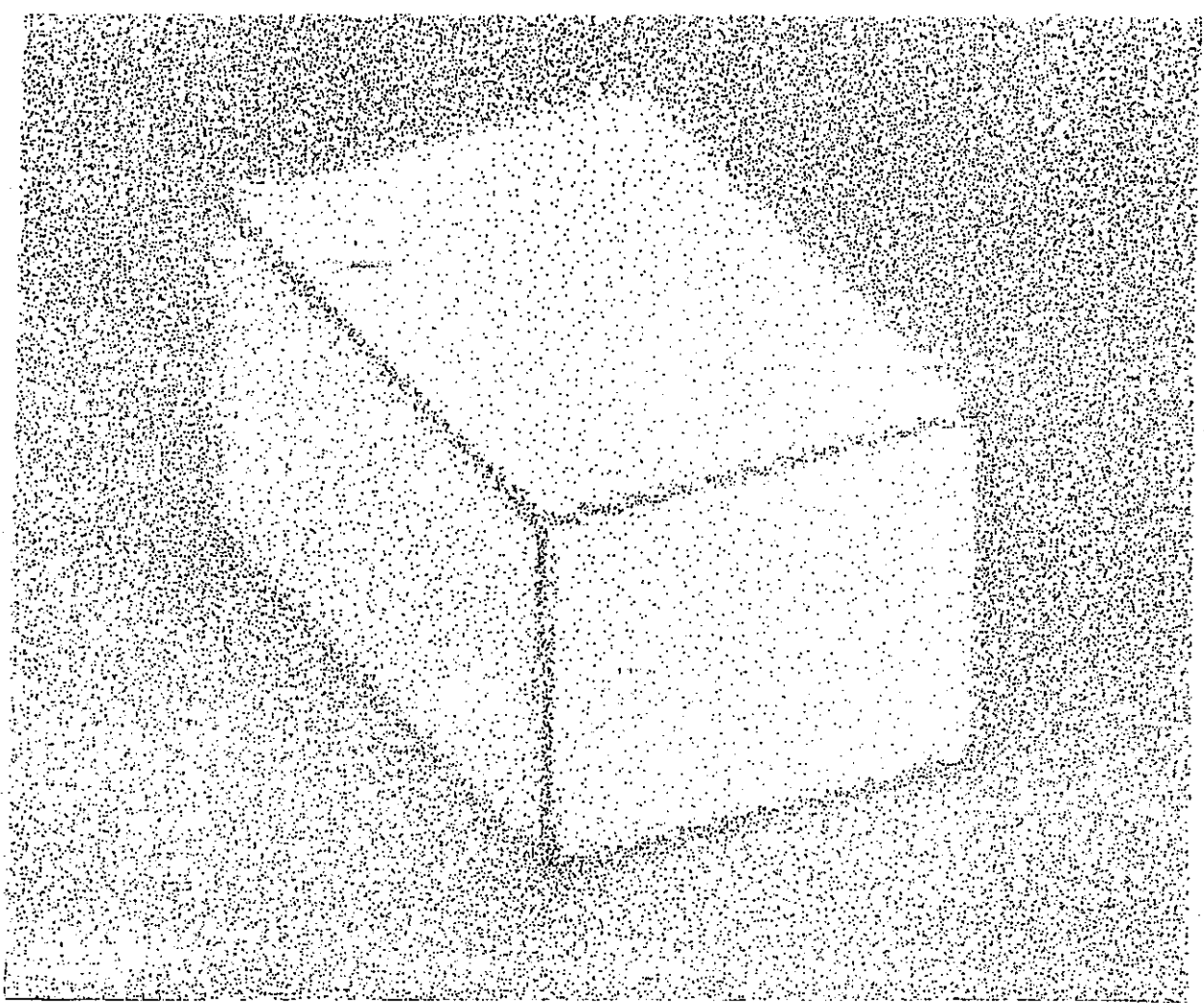


圖5(d). 類似原來平行六面體.

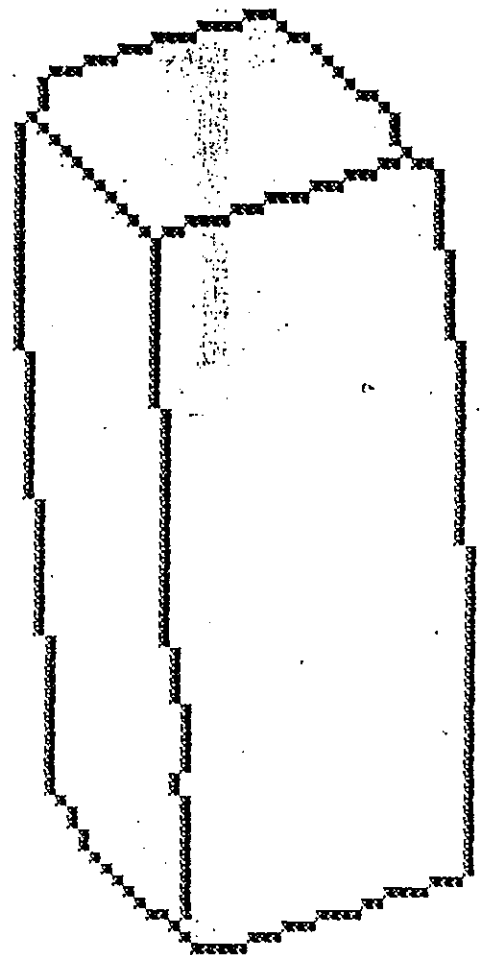
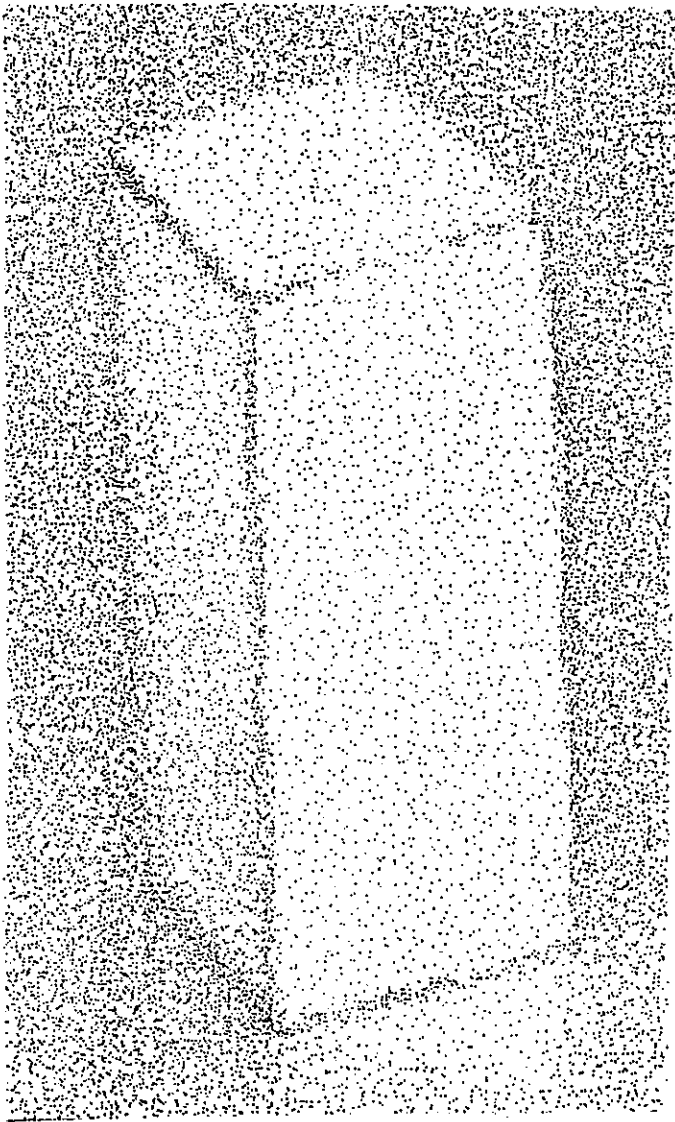
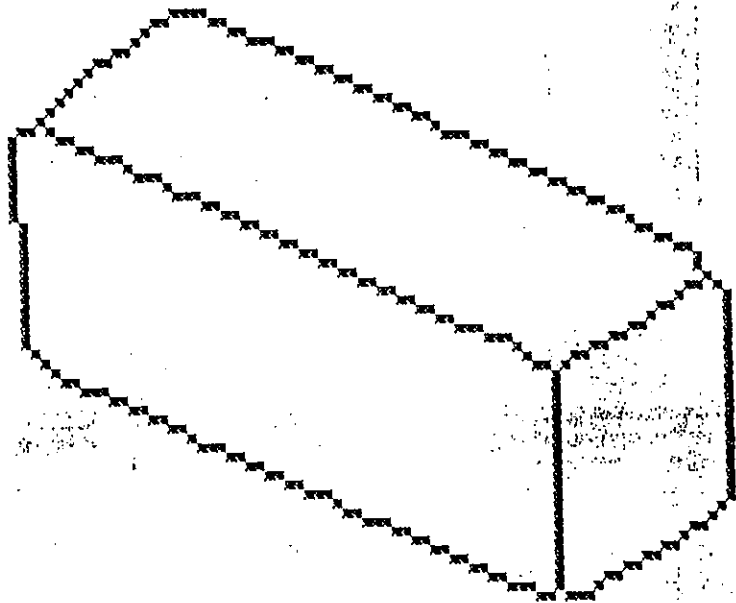
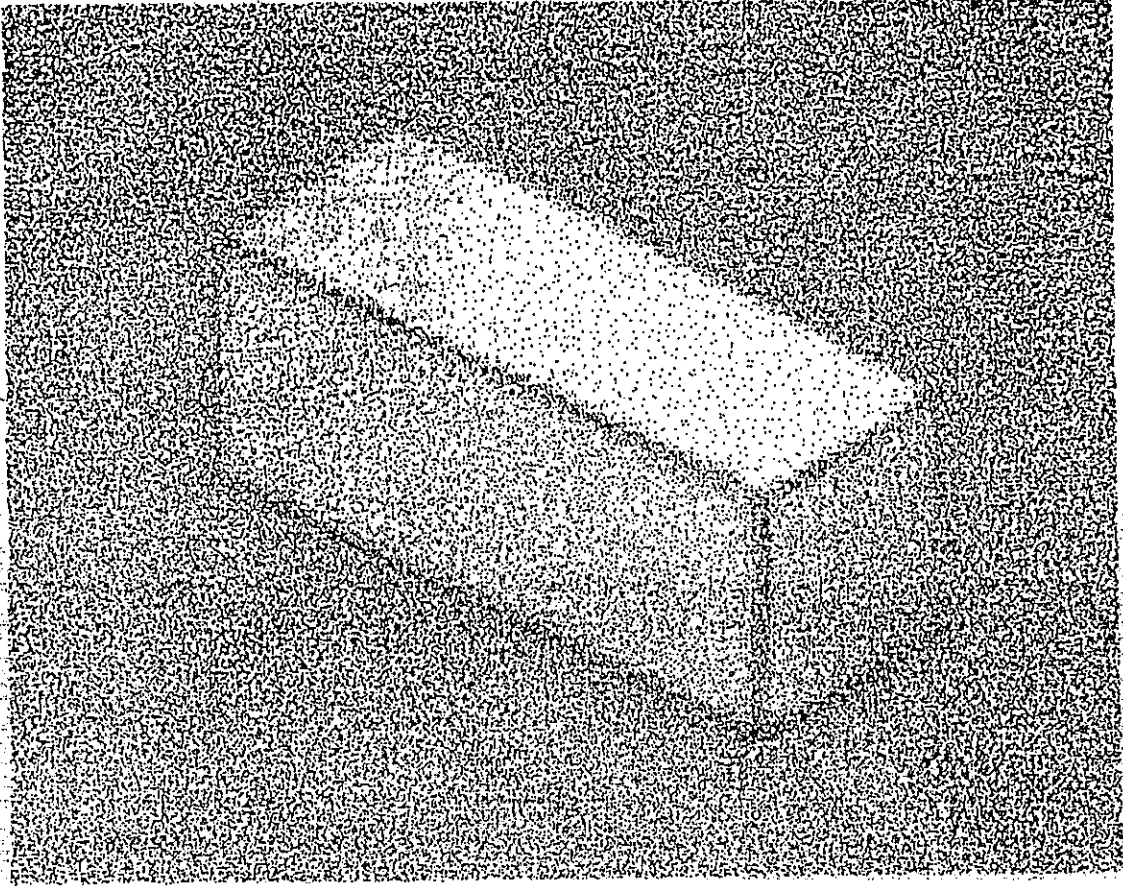


圖5(e). 長柱體.



圖(f). 倒長柱體.

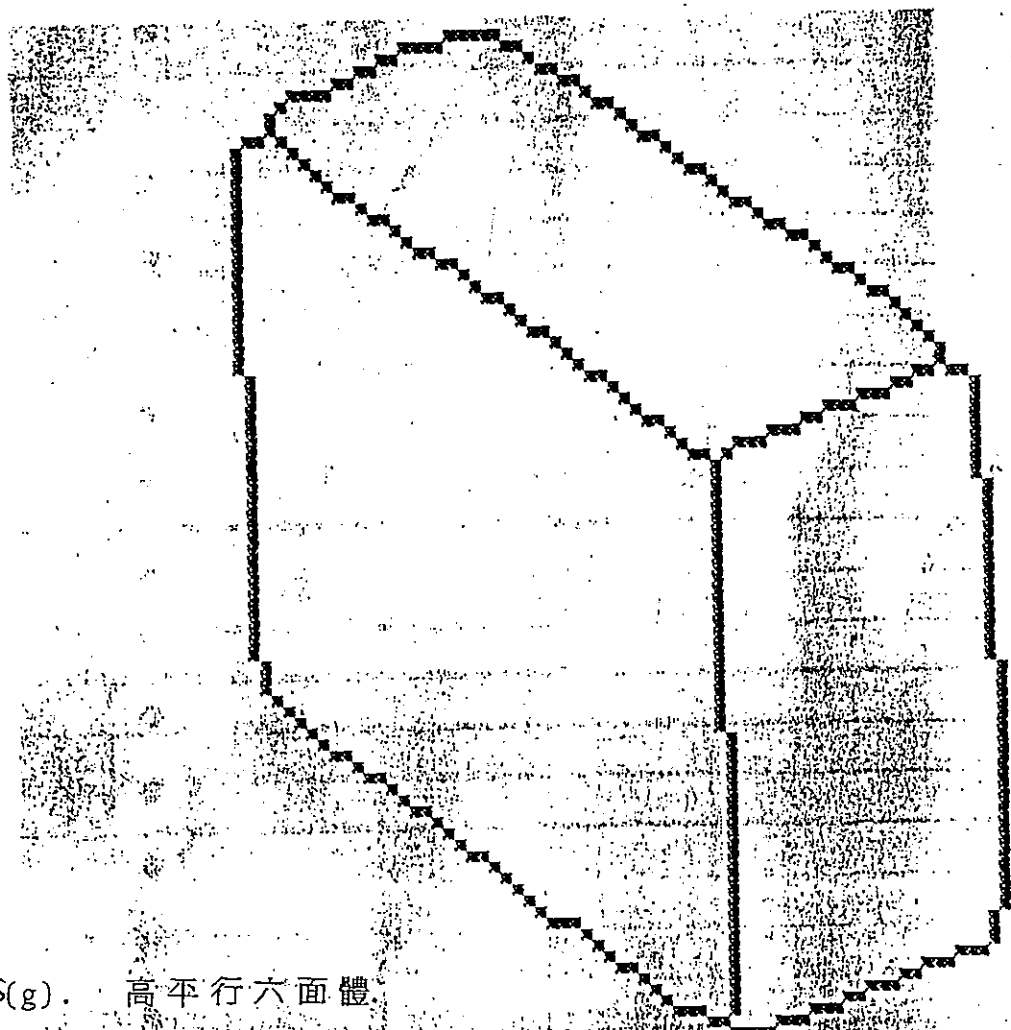
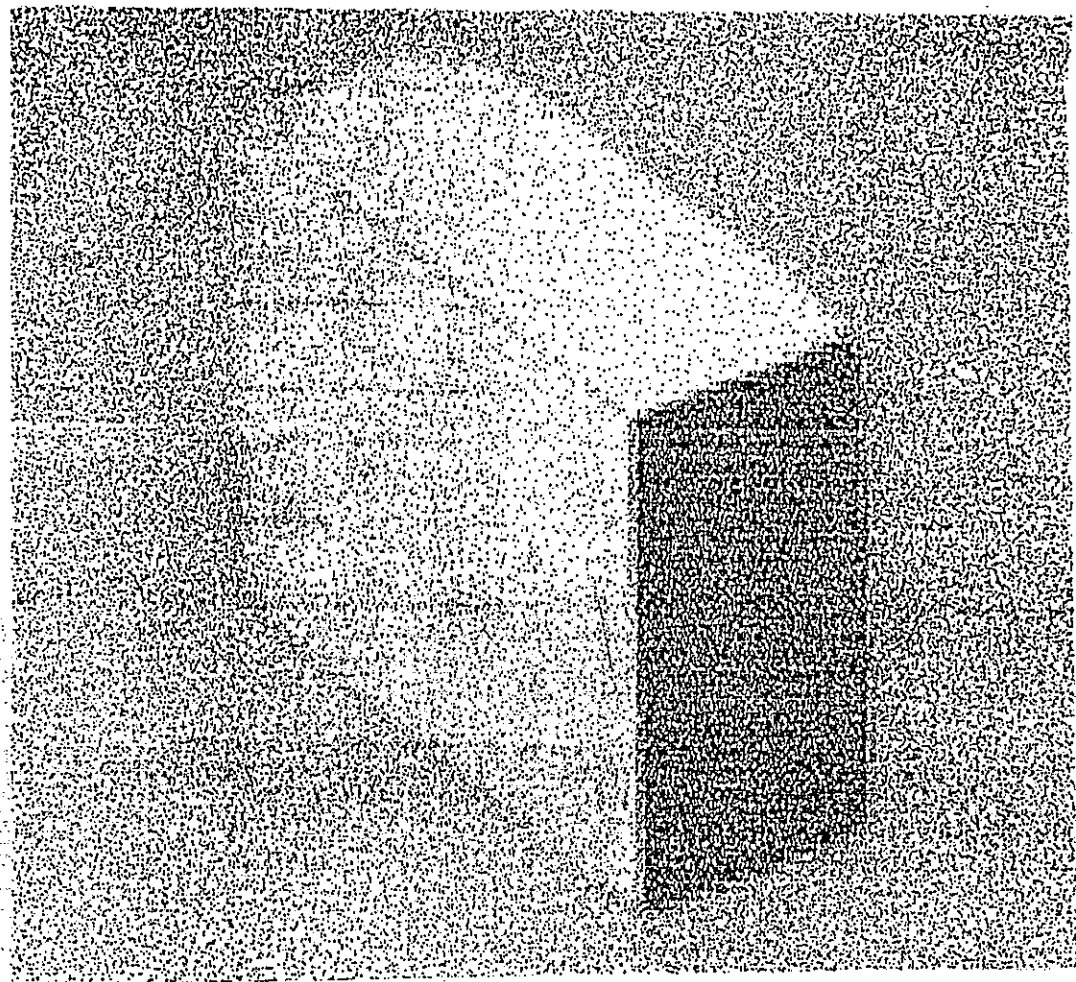


圖5(g). 高平行六面體

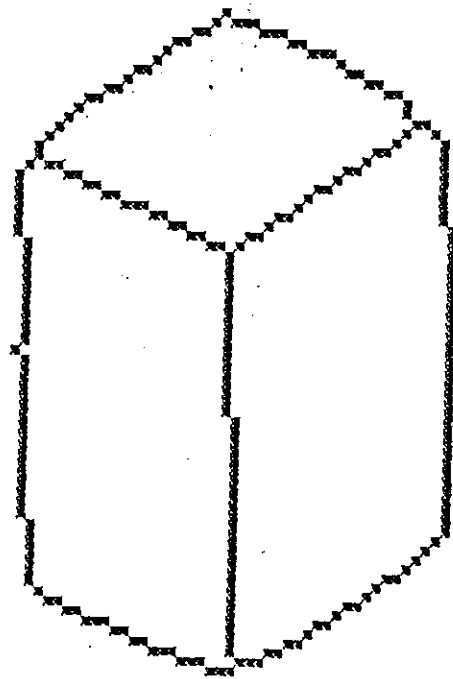
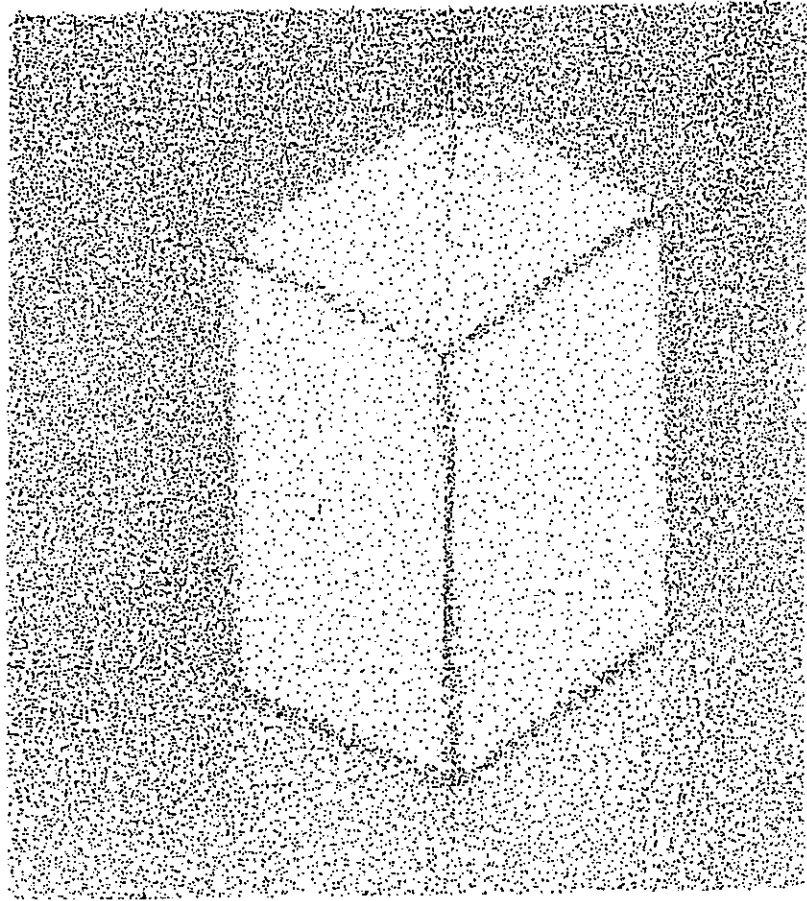


圖5(h). 小方塊.

再次考慮 D 物是一與 A 物大小類似，同為平行六面體但頂面較為長方形化。 $b=2.7$, $c=2.75$, $d=0.6$, $h=6.25$ 。 ($f=16$, 其他不變)，共照了 32 張照片 (參見圖 5 (d))。跑了 CRAMER 後，全部特徵被拒絕 $\alpha=.05$ 。只有特徵 4 及 7 在 $\alpha=.01$ 時被接受。特徵 4 是鄰邊角度差，同為直角故沒有問題。特徵 7 可能是由於誤差所引起。

再次考慮 E 物是一長柱體， $b=2.4$, $c=2.45$, $d=0.15$, $h=8.65$ 。共照了 32 張照片 (參見圖 5 (e))。跑了 CRAMER 後定 $\alpha=.05$ ，全部特徵除了 4 之外被拒絕。特徵 4 是因為角度為直角的原因。

再次考慮 F 物是把 E 物長柱體倒下照， $b=4.7$, $c=3.5$, $d=2.4$, $h=3.25$ ，共照了 33 張照片 (參見圖 5 (f))。定 $\alpha=.05$ 再跑了 CRAMER 後得到除了特徵 4 外其餘皆被拒絕。

再次考慮 G 物，是一高平行六面體，大小與 A 物類似，但較高。共照了 30 張照片 (參見圖 5 (g))。跑了 CRAMER 後在 $\alpha=.05$ 時大都特徵被拒絕，只有特徵 4，為角度，被接受。特徵 8，垂直邊和，最顯著區別出來。

再次考慮 H 物是一小方塊，每邊皆比原平行六面體小。共照了 31 張照片 (參見圖 5 (h))。跑了 CRAMER 後在 $\alpha=.05$ 時大都特徵被拒絕，只有特徵 4 及 5 與角度有關，被接受。

再把 A 至 H 物重做一次實驗。重新照像 (取視角，即旋轉角，不一樣) 其結果是大同小異，都是特徵 7 被接受 (在 $\alpha=.01$)，特徵 4 有時被接受。例如梯形方塊再照了 31 張照片結果是 $\alpha=.05$ 時全部特徵皆被拒絕，只有特徵 7 在 $\alpha=.01$ 時被接受。

以上共做了十六次實驗。每次照像的個數只有三十左右，這是因為 Disk Memory 限制使然。三十左右的個數並不能提供非常充分的訊息來有效區別大小類似的物體。圖 6 是跑了 CRAMER 的部份結果。

```

RUN CRAMER
..... WHAT'S THE INPUT FILE-NAME .....
FEATUR.DAT\TAD.\4.DAT
.. INPUT NUMBER OF PICTURES : NP
30
.... READ FEATUR DATA : DATAM
.... PLEASE INPUT THE THEORETICAL FILE NAME
DIST01.DAT
... INPUT CRITICAL VALUES ....
.45
STATIS= 6.126081 .... CRITIC= 0.4500000
**** REJECT FEATURE : 1
.... PLEASE INPUT THE THEORETICAL FILE NAME
DIST02.DAT
... INPUT CRITICAL VALUES ....
.45
STATIS= 1.030505 .... CRITIC= 0.4500000
**** REJECT FEATURE : 2
.... PLEASE INPUT THE THEORETICAL FILE NAME
DIST03.DAT
... INPUT CRITICAL VALUES ....
.45
STATIS= 10.32024 .... CRITIC= 0.4500000
**** REJECT FEATURE : 3
.... PLEASE INPUT THE THEORETICAL FILE NAME
DIST04.DAT
... INPUT CRITICAL VALUES ....
.45
STATIS= 1.248215 .... CRITIC= 0.4500000
**** REJECT FEATURE : 4
.... PLEASE INPUT THE THEORETICAL FILE NAME
DIST05.DAT
... INPUT CRITICAL VALUES ....
.45
STATIS= 10.32024 .... CRITIC= 0.4500000
**** REJECT FEATURE : 5
.... PLEASE INPUT THE THEORETICAL FILE NAME
DIST31.DAT
... INPUT CRITICAL VALUES ....
.45
STATIS= 0.3256763 .... CRITIC= 0.4500000
... ACCEPT FEATURE : 6
.... PLEASE INPUT THE THEORETICAL FILE NAME
DIST333\33\2.DAT
... INPUT CRITICAL VALUES ....
.45
STATIS= 3.038112 .... CRITIC= 0.4500000
**** REJECT FEATURE : 7
.... PLEASE INPUT THE THEORETICAL FILE NAME
DIST33.DAT
... INPUT CRITICAL VALUES ....
.45
STATIS= 6.907508 .... CRITIC= 0.4500000
**** REJECT FEATURE : 8
.... PLEASE INPUT THE THEORETICAL FILE NAME
DIST34.DAT
... INPUT CRITICAL VALUES ....
.45
STATIS= 10.32024 .... CRITIC= 0.4500000
**** REJECT FEATURE : 9
TT5 -- STOP

```

圖 6. 梯形方塊跑 Cramer 之結果

八、結論：

這次研究實驗收益良多，比如需要隨機化才能使用統計上的檢定方法。雖然照像的個數少於33而使誤差大，但由這些實驗仍可看出檢定方法的效力及各特徵的變化情形。我們對不同物體仍很有效的加以區分出來。對同一物體則必須照像個數增加及影像處理技術改進（解像度增加）才能有好結果。

由這次研究我們亦發現另一有效方法來區別物體，即將物體按一定旋轉方向慢慢旋轉而照像。我們追蹤物體特定端點的軌跡看其變化來判別物體。由照像取出的端點並不是連續變化，我們可用 B-Spline 來連起來。最後用 Fourier Descriptor 來取特徵。解決 phase 的移動可用係數比等方法。這個方法比本次研究方法更改進，因為隨機化會損失不少訊息。

九、參 考 書 目

1. Ballard D. H. and Brown C. M., 1982, Computer Vision, Prentice-Hall Inc., New Jersey, USA
2. Barr A. and Feigenbaum E. A., 1981, The Handbook of Artificial Intelligence, Vol. 1,2,3.
3. Brady M., 1982, Computational approaches to image understanding, Computing Surveys, Vol. 14, No. 1, March.
4. Marr D., 1982, Vision, Freeman Co., San Francisco, USA
5. Wong E. and Steppe J., 1969, Invariant recognition of geometric shapes, in Methodologies of Pattern Recognition, Watanabe, ed., Academic Press, N.Y.
6. Wallace T. P., 1981, Three dimensional shape analysis using local shape descriptors, IEEE Pattern Analysis and Machine Intelligence, Vol. PAMI-3, p. 310-322.
7. Huang J. S. and Tou J. T., 1982, Statistical approach to the identification of a three dimensional object viewed from random directions, Technical report , TR-82-020, Institute of Information Science, Academia Sinica, Rep. of China.
8. Huang J.S. and Lee W.J., 1983, A new thinning algorithm: removing noise-spurs and retaining end-points, Proceedings of National Computer Symposium, Rep. of China, p. 471-477.
9. Nevatia R. and Babu K.R., 1980, Linear feature extraction and description, Computer Graphics and Image Processing, 13, p.257-269.
10. Davis L. S., 1977, Understanding shape: angles and sides, IEEE Trans. on Computer, Vol. c-26, p.236-242.
11. Feller W., 1971, An Introduction to Probability Theory and Its Applications, Vol. 2, John Wiley and Sons Inc., USA
12. Tou J. T. and Gonzalez R. C., 1973, Pattern Recognition Principles, Addison-Wesley Inc., Massachusetts, USA

***** 十、附錄一 *****

```

C*
C* PURPOSE : FIND ONT ALL VERTICES OF A BINARY PICTURE OF A
C* CUBIC AND THEN CALCULATE THE LENGTH AND ANGLES
C* OF ALL THE VERTICES.
C* BY MEANS OF COLLECTING THESE NUMERIC DATAS FROM
C* ARBITRARY DIRECTIONS WE CAN IDENTIFY THIS CUBIC
C*
C* BNDARY.FTN : FIND THE I,J LIMITATION
C* CAUCUL.FTN : CAUCULATE THE CHAIN CODE OF OBJECT
C* ADS.FTN : FIND ALL THE CORNERS OF OBJECT
C* APPROX.FTN : USE MEAN-SQUARE METHOD TO APPROACH ALL THE
C* EDGES OF THE OBJECT.
C* RCORNR : THE OUTSIDE CORNERS OF THE OBJECT
C* ICTER : THE POSITION OF THE INSIDE CORNERS OF THE
C* OBJECT IN THE ARRAY OF RCORNR
C* MEANSQ.FTN : THE SUBROUTINE OF MEAN-SQUARE ALGORITHM
C* LNAG.FTN : CALCULATE LENGTHS AND ANGLES OF LINES
C* IBND : NO. OF BOUNDARY PIXELS
C* ISLOPE : SLOPE F EACH PIXEL
C* ICOORX : COORDINATES OF PIXELS
C* ICTER : POSITION OF 3-LINE-CROSS VERTEX IN CORNR
C* CORNR : LOCATION THAT CORNERS OCCUR
C* IAG : NO. OF CORNERS
C* FEATUR.FTN : FEATURES EXTRACTION
C* FIGUR : SUM OF EDGES OF TOP,SIDE,FACE REGION
C* DIFFERENCE OF EDGES OF TOP,SIDE,FACE REGION
C* FACDIF : DIFFERENCE OF FACING EDGES ON TOP REGION
C* DIFFERENCE OF FACING ANGLES ON TOP REGION
C* VLNTH : SUM OF VERTICAL EDGES
C* BTLNTH : SUM OF BOTTOM'S EDGES
C*

```

```

C PROGRAM HONCOR.FTN
  BYTE IM(128,128),INFNM(20),ICOORX(2,1000),ISLOPE(1000)
  INTEGER ICTER(5,5),CORNR(20,2),IVEX(20,2),IAB(20,2)
  INTEGER ICORNR(200),RCORNR(20,2)
  REAL AN(20),INA(5,5,3),LNPTH(20),INLNTH(5,5),SIDANG(5,2)
  REAL TOPFGR(2,2),FACDIF(2),VLNTH,BTLNTH
  COMMON /A/IM,ICOORX
  COMMON /B/IUP,IDN,ILF,IRT
  COMMON /C/LNPTH,INLNTH,AN,INA,SIDANG
C .....> SET INATIAL VALUE <.....
  DO 11 I=1,2
  DO 11 J=1,20
    CORNR(J,I)=0
    IVEX(J,I)=0
    IAB(J,I)=0
  11 RCORNR(J,I)=0
  DO 12 I=1,2
  DO 12 J=1,1000
  12 ICOORX(I,J)=0
  DO 13 I=1,1000
  13 ISLOPE(I)=0
  DO 14 I=1,5
  DO 14 J=1,5
  14 ICTER(I,J)=0
  CALL ASSIGN(1,'FEATUR.DAT')
C
C .....> MAIN PROCESS <.....
C

```

TYPE *, *** PLEASE TYPE IN THE THE INPUT FILENAME ***
 ACCEPT 100,(INFNM(I),I=1,19)

```

.. WHAT'S THE STEPIING POSITION ? ..
TYPE *,
ACCEPT *,IIB
100 FORMAT(80A1)
105 IF(INFNM(IIB).NE.'B')GOTO 110
      INFNM(IIB-1)=INFNM(IIB-1)+1
      INFNM(IIB)='0'
110 CALL ASSIGN(3,INFNM)
      TYPE 101,INFNM
101 FORMAT(//////////1X, 80A1)
      READ(3,1000,ERR=999)((IM(I,J),J=1,128),I=1,128)
      CALL CLOSE(3)
C
      TYPE *,' ****      SUBROUTINE OF BNDARY.FTN      ****'
      CALL BNDARY
C
      TYPE *,' ****      SUBROUTINE OF CALCUL.FTN      ****'
      CALL CALCUL(IBND,ISLOPE)
C
      TYPE *,' ****      SUBROUTINE OF ADS.FTN      ****'
      CALL ADS(IBND,IAG,ISLOPE,CORNR,IVEX,IVX,ICORNR)
C
      TYPE *,' ****      SUBROUTINE OF APPROX.FTN      ****'
      CALL APPROX(ICORNR,RCORNR,IAG,IVEX,ICTER,IVX,IBND)
C
      TYPE *,' ****      SUBROUTINE OF LNAG.FTN      ****'
      CALL LNAG(RCORNR,ICTER,IAG,IAG+1)
C
      TYPE *,' ****      SUBROUTINE OF FEATUR.FTN      ****'
      CALL FEATUR(TOPFGR,FACDIF,VLNTH,BTLNTH,IVX)
C
      INFNM(IIB)=INFNM(IIB)+1
      GOTO 105
999  TYPE *,'----- END OF JOB -----'
      CALL CLOSE(1)
1000 FORMAT(1X,128I1)
      STOP
      END

```

```

C*****
C*
C*      SUBROUTINE BNDARY.FTN
C*      SUBJECT : FIND THE SQUARE BOUND OF OBJECT
C*
C*      IDX : THE PARAMETERS OF EACH SIDE OF LIMITATION
C*      IUP,IDN : LIMITATIONS OF I
C*      JLF,JRT : LIMITATIONS OF J
C*
C*****

```

```

      SUBROUTINE BNDARY
      INTEGER IDX(3,2)
      BYTE IM(128,128),ICORX(2,1000)
      COMMON /A/IM,ICORX
      COMMON /B/IUP,IDN,ILF,IRT
      DATA IDX/1,128,1,128,1,-1/
C
      DO 150 IFLAG=1,2
10   DO 100 I=1,2
          DO 200 M=IDX(1,I),IDX(2,I),IDX(3,I)
          DO 200 N=IDX(1,I),IDX(2,I),IDX(3,I)
              GOTO (210,220)IFLAG
210  IF(IM(M,N).EQ.0)GOTO 200
          IF(I.EQ.1)IUP=M
          IF(I.EQ.2)IDN=M
          GOTO 100
220  IF(IM(N,M).EQ.0)GOTO 200
          IF(I.EQ.1)ILF=M
          IF(I.EQ.2)IRT=M

```

```

GOTO 100
200 CONTINUE
100 CONTINUE
150 CONTINUE
TYPE *, 'IUP=', IUP, 'IDN=', IDN, '., ILF=', ILF, 'IRT=', IRT
RETURN
END
C*****
C#
C# -----> CAT.FTN : CHECK FOR CONTINUITY OF EDGE <-----
C#
C*****
SUBROUTINE CAT(I,J,IC)
BYTE IM(128,128),ICDORX(2,1000)
COMMON /A/IM,ICDORX
IC=0
DO 100 IK=0,1
IF(IM(I-1,J+IK).NE.IM(I-1,J+IK-1))IC=IC+1
IF(IM(I+1,J-1K).NE.IM(I+1,J-1K+1))IC=IC+1
IF(IM(I+IK,J+1).NE.IM(I+IK-1,J+1))IC=IC+1
IF(IM(I-1K,J-1).NE.IM(I-1K+1,J-1))IC=IC+1
100 CONTINUE
RETURN
END
C*****
C#
C# PURPOSE : CALCULATE THE CHAIN CODE AND SLOPE
C#
C# CAT : CHECK FOR THE CONTINUITY OF EDGES
C# ADJUST : READJUST THE SECOND 3-LINE VERTEX'S LOCATION
C#
C*****
C PROGRAM CALCUL.FTN
SUBROUTINE CALCUL(L,ISLOPE)
BYTE ICDORX(2,1000),ISLOPE(1000)
BYTE IM(128,128)
COMMON /A/IM,ICDORX
COMMON /B/IUP,IDN,ILF,IRT
BYTE IX(11),IY(11),IS(11),ITX,ITY
DATA IX/0,1,1,1,0,-1,-1,-1,0,1,1/,IY/-1,-1,0,1,1,1,0,-1,-1,-1,0/
IS/4,5,6,7,0,1,2,3,4,5,6/
1 L=0
IVX=1
C ... SEARCH THE FIRST BOUNDARY WE MEET
DO 10 I=IUP,IRT
DO 10 J=ILF,IRT
IF(IM(I,J).EQ.0)GOTO 10
TYPE *, '... HERE IS THE FIRST BOUNDARY ...(',I,J,')'
GOTO 25
10 CONTINUE
C
25 L=L+1
ICDORX(1,L)=I
ICDORX(2,L)=J
C
C FIND THE SUCCESSIVE BOUNDARY BY STARTING AT WESTERN
C ORIENTATION IN 8 ORIENTATIONS
C
II=1
JJ=10
IF(ICNT.GT.4.AND.IVX.GT.2)II=3
CALL CAT(I,J,ICNT)
IF(ICNT.LT.6)GOTO 200
II=3
JJ=10
IF(IY-ED.1)II=1

```

```

200 DO 30 I2=I1,JJ
      I3=I+IX(I2)
      J3=J+IY(I2)
      IF(I3.EQ.ICDORX(1,1).AND.J3.EQ.ICDORX(2,1))GOTO 100
      IF((IM(I3,J3).EQ.0).OR.(IM(I3,J3).EQ.2))GOTO 30
          DO 32 I4=L-3,L
              IF(I3.EQ.ICDORX(1,I4).AND.J3.EQ.ICDORX(2,I4))GOTO 3
32      CONTINUE
          I=I3
          J=J3
          ISLOPE(L)=IS(I2)
          GOTO 25
30      CONTINUE
C
TYPE *, '----- NOISE OCCUR -----'
TYPE *, '... I=',I,'... J=',J,'...L=',L
DO 50 K=1,L
      IIX=ICDORX(1,K)
      IIY=ICDORX(2,K)
50      IM(IIX,IIY)=0
      I=ICDORX(1,1)
      GOTO 1
C
100      ISLOPE(L)=I2
      ICDORX(1,L+1)=I3
      ICDORX(2,L+1)=J3
      TYPE *, '..... L =',L
      RETURN
      END

```

```

DIMENSION THI(1000),Y(1000),A(91),D(91),E(91),F(91),G(91)
BYTE OUTFNM(20),INFNM(20),AGAIN
999 TYPE *, '..... PLEASE IN THE INPUT-FILENAME ..... '
ACCEPT 3000,(INFNM(I),I=1,19)
TYPE *, '..... PLEASE INPUT THE OUTPUT-FILENAME ..... '
ACCEPT 3000,(OUTFNM(I),I=1,19)
3000 FORMAT(80A1)
CALL ASSIGN(2,INFNM)
CALL ASSIGN(3,'DISOUT.DAT')
CALL ASSIGN(7,OUTFNM)
C
... Y IS THE FUNCTION VALUE DEPENDS ON THI
1 READ(2,1)(THI(I),Y(I),I=1,1000)
FORMAT(1X,F9.5,1X,F9.5)
CALL MAX(Y,A1,A2)
A(91)=A1
A(1)=A2
D(1)=0
D(91)=0
H=(A1-A2)/90
C
.... A(I) IS Y-AXIS OF DISTRIBUTION
C
.... D(I) IS DISTRIBUTION FUNCTION OR PROBABILITY OF (X<=I)
DO 1000 1000 I=2,90
      D(I)=0
1000      A(I)=A(I-1)+H
      I2=91
      J=1
7 TYPE *, '..... READ JSH..... '
ACCEPT *,JSH
DO 103 K=1,JSH
DO 70 I=1,I2-1
70      E(I)=0
105 FORMAT(20F4.3)

```

```

I=I2-1
E(I2)=THI(J)
J1=J+1
C
50 ..... Y DECREASE .....
DO 20 J=J1,1000
IF(Y(J+1).GT.Y(J))GOTO 8
IF(Y(J+1).LT.A(I) .AND. Y(J).GE.A(I))GOTO 40
IF(Y(J1).LT.A(I))GOTO 41

20 CONTINUE
40 E(I)=THI(J)
J1=J+1
GOTO 74

41 E(I)=E(I+1)
74 IF(I.EQ.1)GOTO 8
I=I-1
GOTO 50

8 E(I)=THI(J)
I1=I
DO 9 I=1,91
9 F(I)=0
WRITE(3,105)(E(I),I=1,91)
I=I1
F(I)=THI(J)
I=I+1
J1=J+1

C
15 .... Y INCREASE .....
DO 12 J=J1,1000
IF(Y(J1).GT.A(I))GOTO 16
IF(Y(J).LT.A(I) .AND. Y(J+1).GE.A(I))GOTO 14
IF(Y(J+1).LT.Y(J))GOTO 17

12 CONTINUE
14 F(I)=THI(J)
J1=J+1
IF(J.EQ.1000)GOTO 17
GOTO 71

16 F(I)=F(I-1)
71 IF(I.EQ.91)GOTO 17
I=I+1
GOTO 15

17 I2=I
F(I)=THI(J)
IF(I2.EQ.91)GOTO 29
DO 28 I=I2+1,91
F(I)=F(I-1)

28 CONTINUE
29 WRITE(3,105)(F(I),I=1,91)
DO 18 I=1,91
G(I)=F(I)-E(I)
18 D(I)=D(I)+G(I)
E(I2)=F(I2)
IF(I2.EQ.91)GOTO 103
DO 30 I=I2+1,91
30 E(I)=E(I-1)
103 CONTINUE
J1=J+1
I=I2-1

5 DO 2 J=J1,999
IF(Y(J+1).LT.A(I) .AND. Y(J).GE.A(I))GOTO 4
IF(Y(J1).LT.A(I))GOTO 3

2 CONTINUE
4 E(I)=THI(J)
J1=J+1
GOTO 73

3 E(I)=E(I+1)
73 IF(J.GE.999)GOTO 77

```



```

C MADE FOR ALL THE ANGLE INCLUDING INTEGER AND ALL THE
C LENGTH OF EDGE INCLUDING INNER EDGE
C INPUT:
C VERTEX:AN ARRAY THAT CONTAINS THE POINT'S COORDINAES
C INPNT: AN ARRAY THAT ITS FIRST ELEMENTS IS THE INNER
C POINTS AND OTHERS WHICH CONNECTED WITH THE
C INNER POINT .
C THE ARRAY MUST ENDED WITH 0
C BPNT: AN INTEGER INDICATE HOW MANY POINTS AT OUTER
C GRAPH.
C PNT: AN INTEGER INDICATE HOW MANY POINTS OF THE ALL
C GRAPH.
C OUTPUT:
C LNGTH:AN ARRAY CONTAIN THE LENGTH OF OUTER EDGE
C WHICH CORRESPONDENCE TO VERTEX
C INLNTH:AN ARRAY CONTAIN THE LENGTH OF INNER EDGES
C WHICH CORRESPONDENCE TO INPNT
C AN: AN ARRAY WHICH CONTAIN THE MEASURE OF ANGLE WHICH
C CORRESPONDANCE TO VERTEX.
C INA: AN ARRAY WHICH CONTAIN THE MEASURE OF INNER ANGLE
C CORRESPONDANCE TO INPNT.
C

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE LNAG(VERTEX,INPNT,BPNT,PNT)
INTEGER BPNT,PNT,VERTEX(20,2),PNTEND,INPNT(5,5)
REAL LNGTH(20),INLNTH(5,5),AN(20),INA(5,5,3),SIDANG(5,2)
COMMON /C/LNGTH,INLNTH,AN,INA,SIDANG
DO 5 I=1,5
DO 5 J=1,2
5 SIDANG(I,J)=0
IF(BPNT.LT.2)RETURN
LNSTRT=1
LNEND=2
PNTEND=3
TYPE *,'\\' THE LENGTH AND ANGLE OF OUTER BOUNDARIES. //'
200 CALL DSTNT(VERTEX,LNSTRT,LNEND,DSTANT)
LNGTH(LNSTRT)=DSTANT
TYPE *,'...' LNGTH(' ,LNSTRT,')=' ,DSTANT
CALL ANGE(VERTEX,LNSTRT,LNEND,PNTEND,ANG)
AN(LNSTRT)=ANG
TYPE *,'...' ANG(' ,LNSTRT,')=' ,ANG
LNEND=LNEND+1
LNSTRT=LNSTRT+1
PNTEND=PNTEND+1
IF(LNEND.GT.BPNT)LNEND=1
IF(PNTEND.GT.BPNT)PNTEND=1
IF(BPNT.GE.LNSTRT)GOTO 200
IF(BPNT.GE.PNT)RETURN
TYPE *,'----' THE LENGTH AND ANGLE OF INNER VERTEX ----'
DO 900 J=1,PNT-BPNT
LNSTRT=INPNT(J,1)
DO 800 K=1,5
LNEND=INPNT(J,K+1)
PNTEND=INPNT(J,2)
IF(INPNT(J,K+2).NE.0)PNTEND=INPNT(J,K+2)
CALL DSTNT(VERTEX,LNSTRT,LNEND,DSTANT)
INLNTH(J,K)=DSTANT
TYPE *,'...' INLNTH(' ,J,K,')=' ,DSTANT
CALL ANGE(VERTEX,LNEND,LNSTRT,PNTEND,ANG)
INA(J,K,1)=ANG
TYPE *,'...' INA(' ,J,K, ,1)=' ,ANG
II=LNEND+1
IF(II.GT.BPNT)II=1
CALL ANGE(VERTEX,LNEND,II,LNSTRT,ANG)

```



```

                IF(II,LE.0)II=BPNT
                CALL ANGE(VERTEX,II,PNTEND,LNSTRT,ANG)
                INA(J,K,3)=ANG
                IF(INPNT(J,K+2),EQ.0)GOTO 900
800          CONTINUE
900          CONTINUE
C
IF(VERTEX(4,1),EQ,VERTEX(7,1))RETURN
TYPE *, '*** THE INSIDE ANGLE OF 3-LINE CROSS LINE *****'
DO 300 I=2,5
  IF(INPNT(1,I),EQ.0)GOTO 400
  TYPE *, '/// THE',I,'TH INSIDE ANGLE'
  I1=INPNT(1,I)-1
  I2=INPNT(1,I)+1
  IF(I1,LE.0)I1=6
  CALL ANGE(VERTEX,I1,INPNT(1,I),INPNT(1,1),ANG)
  SIDANG(I,1)=ANG
  TYPE *, '---- THE LEFT SIDE ANGLE OF',I,'TH =',ANG
  CALL ANGE(VERTEX,INPNT(1,1),INPNT(1,I),I2,ANG)
  SIDANG(I,2)=ANG
  TYPE *, '---- THE RIGHT SIDE ANGLE OF',I,'TH =',ANG
300          CONTINUE
400          SIDANG(I,1)=0
              SIDANG(I,2)=0
              RETURN
              END
C*****
C*
C*          PURPOSE : FIND CORNERS          METHODS OF L.S.DAVIS
C*
C*          IDIFF   : CHANGE OF TWO CONTINUOUS SLOPES
C*          CORNR   : MATRIX THAT STORES THOSE CORNERS
C*          IHEAD   :
C*          ITAIL   : RANGE THAT FOR CALCULATING COSINES
C*          ISTEP   : STEP OF NEXT PIXEL BE CALCULATING
C*          THRESH  : THRESHOLD TO DELETE FALSE CORNER
C*
C*****
C          SUBROUTINE ADS(L,I2,ISLOPE,CORNR,IVEX,IVX,ICORNR)
C          PROGRAM ADS.FTN
          INTEGER ICORNR(200),CORNR(20,2),IVEX(20,2)
          BYTE IM(128,128),ICORRX(2,1000),ISLOPE(1000),IDIFF(1000)
          COMMON /A/IM,ICORRX
          COMMON /B/IUP,IDN,ILF,IRT
          LOGICAL*1 AGAIN
          REAL ANGL(100)
          DATA ANGL/100*1.0/
C
C          FIND THE 3-LINE-CROSS VERTECES
          CALL VERTEX(IVEX,IVX)
C
C          CALCULATE THE CHANGE RATE OF SLOPE
C
          DO 10 I=2,L+1
            I1=I-1
            IDIFF(I1)=ISLOPE(I)-ISLOPE(I1)
10          CONTINUE
11111 TYPE *, '... PLEASE KEY IN THE RANGE OF DOMAIN IHEAD/ITAIL'
C          ACCEPT *,IHEAD,ITAIL
          IHEAD=5
          ITAIL=20
C          TYPE *, '... PLEASE KEY IN THE STEP,INCREMENT IN DOMAIN'
C          ACCEPT *,ISTEP
          ISTEP=6

```

THRESH=0.982

II=0
DO 20 I=1,L
IF(IDIFF(I),EQ.0)GOTO 20
IP=I+1

C
C
C
C
CALCULATE THE COSINE OF POINTS WHICH ARE THE LOCAL MAXIMUM OF
DIFFERENCE

ALOCAL=-1
DO 30 K1= IHEAD,ITAIL,ISTEP
IR=IP+K1
IF(IR.GT.L)IR=IR-L
IAX=ICORX(2,IR)-ICORX(2,IP)
IAY=ICORX(1,IR)-ICORX(1,IP)
DO 30 K2= IHEAD,ITAIL,ISTEP
IL=IP-K2
IF(IL.LT.1)IL=L+IL
IBX=ICORX(2,IP)-ICORX(2,IL)
IBY=ICORX(1,IP)-ICORX(1,IL)
COS=COSINE(IAX,IAY,IBX,IBY)
IF(COS.LT.ALOCAL)GOTO 30
ALOCAL=COS

30 CONTINUE
IF(ABS(ALOCAL).GT.THRESH)GOTO 20
II=II+1
ICORNR(II)=I+1
IF(I.EQ.L)ICORNR(II)=1
ANGL(II)=ALOCAL

20 CONTINUE

C

IS=(IHEAD+ITAIL)*0.6
DO 50 I=1,II
DO 55 I1=1,II/3
IN1=I+I1
IF(IN1.GT.II)IN1=IN1-II
NB1=ICORX(1,ICORNR(I))-ICORX(1,ICORNR(IN1))
NB2=ICORX(2,ICORNR(I))-ICORX(2,ICORNR(IN1))
NB=IABS(NB1)+IABS(NB2)
IF(NB.GT.IS)GOTO 50
IF(ABS(ANGL(I))-ABS(ANGL(IN1)))56,57,57
56 ANGL(IN1)=2
GOTO 55
57 ANGL(I)=2
GOTO 50

55 CONTINUE

50 CONTINUE

I2=0
DO 60 I=1,II
IF(ANGL(I).EQ.2)GOTO 60
I2=I2+1
ICORNR(I2)=ICORNR(I)
CORNR(I2,1)=ICORX(1,ICORNR(I))
CORNR(I2,2)=ICORX(2,ICORNR(I))
TYPE *,'===> ICORNR(',I2,')=',ICORNR(I)
TYPE *,'ICORX(1,',I2,')=',CORNR(I2,1)
TYPE *,'ICORX(2,',I2,')=',CORNR(I2,2)
TYPE *,'

60 CONTINUE

ICORNR(I2+1)=ICORNR(1)

C TYPE *,'*** DO YOU WANT TO TRY AGAIN ? Y/N'

C ACCEPT 500,AGAIN

AGAIN='N'

```
IF (MODIN .EQ. 1) GO TO 1111
RETURN
END
```

```
C*****
C*
C*          CALCULATE THE INNER PRODUCT OF IA AND IB
C*
C*****
```

```
FUNCTION COSINE(IAX,IAY,IBX,IBY)
  DOT=IAX*IBX+IAY*IBY
  DISTX=IAX*IAX+IAY*IAY
  DISTX=SQRT(DISTX)
  DISTY=(IBX*IBX+IBY*IBY)
  DISTY=SQRT(DISTY)
  IF (DISTX.EQ.0 .OR. DISTY.EQ.0)COSINE=1
  IF(DISTX.NE.0.AND.DISTY.NE.0)COSINE=DOT/(DISTX*DISTY)
  RETURN
END
```

```
C*****
C#
C#          FIND THE 3-LINE-CROSS VERTECES
C#
C#          IVEX  : MATRIX OF THESE VERTECES
C#          IVX   : NO. OF VERTECES
C#
C*****
```

```
SUBROUTINE VERTEX(IVEX,II)
  INTEGER ID(10),IVTX(20,2),IVEX(20,2)
  BYTE IM(128,128),ICORX(2,1000)
  COMMON /A/IM,ICORX
  COMMON /B/IUP,IDN,ILF,IRT

C
  II=0
  DO 100 I=IUP,IDN
    DO 200 J=ILF,IRT
      IF(IM(I,J).EQ.0)GOTO 200
      ICNT=0
      CALL CAT(I,J,ICNT)
      IF(ICNT.LT.6)GOTO 200
      II=II+1
      IVTX(II,1)=I
      IVTX(II,2)=J

200    CONTINUE
100    CONTINUE
C    TYPE *, '+++++ TOTAL NO. OF VERTEX IS -->', II
    IF(II.LT.4)GOTO 400
    MA=MAX(IVTX(1,2),IVTX(2,2),IVTX(3,2))
    MI=MIN(IVTX(1,2),IVTX(2,2),IVTX(3,2))
    DO 300 I=1,II
      IF(IVTX(I,2).EQ.MA)ID(4)=I
      IF(IVTX(I,2).EQ.MI)ID(2)=I

300    CONTINUE
    DO 350 I=1,II-1
      IF(I.EQ.ID(4).OR.I.EQ.ID(2))GOTO 350
      ID(1)=I

350    CONTINUE
      ID(3)=4
      DO 380 J=1,II
        IVEX(J,1)=IVTX(ID(J),1)
        IVEX(J,2)=IVTX(ID(J),2)

C    TYPE *, 'IVEX(', J, ', 1) = ', IVEX(J,1), ' IVEX(', J, ', 2) = ', IVEX(J,2)
380    CONTINUE
    RETURN

400    MA=MAX(IVTX(1,2),IVTX(2,2))
    MI=MIN(IVTX(1,2),IVTX(2,2))
```

410

IF (IVTX(I,2).EQ.MI)J2=I

IVEX(3,1)=IVTX(J1,1)
IVEX(3,2)=IVTX(J1,2)
IVEX(2,1)=IVTX(J2,1)
IVEX(2,2)=IVTX(J2,2)
IVEX(1,1)=IVTX(J1,1)
IVEX(1,2)=IVTX(J1,2)
RETURN

END

C*****
C#
C# APPROX.FTN : FITTING EDGES BY MEAN-SQUARE METHOD
C#
C# INPT : MIDDLE POINTS AMONG TWO CORNERS
C# RCORNR : CORNERS OBTAINED FROM MEAN-SQUARE
C# ICORNR : ALL THE BOUNDARY POINTS
C#

SUBROUTINE APPROX(ICORNR,RCORNR,ICO,IVEX,ICTER,IVX,IBND)
BYTE IM(128,128),ICORX(2,1000)
INTEGER ICORNR(200),INPT(20),RCORNR(20,2),IVEX(20,2),ICTER(5,5)
REAL IAB(20,2)
COMMON /A/IM,ICORX

C

IBR=15
ICORNR(0)=1
DO 100 I=1,ICO
IF(ICORNR(I).NE.1)BREAK=(ICORNR(I)-ICORNR(I-1))/IBR
IF(ICORNR(I).EQ.1)BREAK=(IBND-ICORNR(I-1))/IBR
DO 200 J=1,IBR-1
INPT(J)=ICORNR(I-1)+J*BREAK
CONTINUE
CALL MEANSQ(INPT,IAB,I)

200

100

CONTINUE
IAB(ICO+1,1)=IAB(1,1)
IAB(ICO+1,2)=IAB(1,2)

C

DO 300 I=1,ICO
RCORNR(I,1)=(IAB(I+1,2)-IAB(I,2))/(IAB(I,1)-IAB(I+1,1))+0.5
RCORNR(I,2)=(IAB(I,1)*IAB(I+1,2)-IAB(I,2)*IAB(I+1,1))/(IAB(I,1)-IAB(I+1,1))+0.5

300

TYPE *,'. . . RCORNR(I,1/2)=' ,RCORNR(I,1),RCORNR(I,2)
CONTINUE
IF(IVX.LT.4)CALL TWOVEX(RCORNR,IVEX,ICO)
RCORNR(ICO+1,1)=IVEX(1,1)
RCORNR(ICO+1,2)=IVEX(1,2)
ICTER(1,1)=7
ICTER(1,2)=1
ICTER(1,3)=3
ICTER(1,4)=5
ICTER(1,5)=0
DO 400 I=1,ICO+1
TYPE *,'RCORNR(',I,',1/2)=' ,RCORNR(I,1),',',RCORNR(I,2)

400

CONTINUE
RETURN
END

C*****
C#
C# MEANSQA.FTN : MEAN-SQUARE APPROXIMATION
C#

SUBROUTINE MEANSQ(INPT,IAB,K)
BYTE IM(128,128),ICORX(2,1000)

```
REAL IAB(20,2),COV,VAR
COMMON /A/IM,ICOORX
```

C

```
TYPE *,'====> K=',K
IBR=14
EX=0
EY=0
DO 200 I=1,IBR
EX=EX+ICOORX(1,INPT(I))
EY=EY+ICOORX(2,INPT(I))
TYPE *,',, INPT(',I,')=',INPT(I),',,,EX/EY',EX,EY
```

200

```
CONTINUE
EX=EX/IBR
EY=EY/IBR
COV=0.
VAR=0.
DO 300 J=1,IBR
COV=COV+(EX-ICOORX(1,INPT(J)))*(EY-ICOORX(2,INPT(J)))
VAR=VAR+(EX-ICOORX(1,INPT(J)))**2
TYPE *,'COV=',COV,',, VAR=',VAR
```

300

```
CONTINUE
IF(VAR.NE.0)IAB(K,1)=COV/VAR
IF(VAR.EQ.0)IAB(K,1)=ICOORX(1,IFIX(EX))
IAB(K,2)=EY-IAB(K,1)*EX
TYPE *,',,IAB(K,1/2)= ',IAB(K,1),IAB(K,2)
RETURN
END
```

```
C#####
C#
C# TWOVEX.FTN : CASE WHEN ONLY TWO 3-LINE-VERVEX #
C#
C#####
```

```
SUBROUTINE TWOVEX(RCORN,IVEX,ICO)
INTEGER IVEX(20,2),RCORN(20,2)
```

C

```
ICO=ICO+2
IF(IVEX(2,1).LE.RCORN(1,1))GOTO 100
RCORN(6,1)=RCORN(1,1)
RCORN(6,2)=RCORN(1,2)
RCORN(5,1)=RCORN(4,1)
RCORN(5,2)=RCORN(4,2)
RCORN(4,1)=IVEX(3,1)
RCORN(4,2)=IVEX(3,2)
RCORN(1,1)=IVEX(2,1)
RCORN(1,2)=IVEX(2,2)
RETURN
```

100

```
RCORN(6,1)=RCORN(4,1)
RCORN(6,2)=RCORN(4,2)
RCORN(5,1)=RCORN(3,1)
RCORN(5,2)=RCORN(3,2)
RCORN(4,1)=IVEX(3,1)
RCORN(4,2)=IVEX(3,2)
RCORN(3,1)=RCORN(2,1)
RCORN(3,2)=RCORN(2,2)
RCORN(2,1)=RCORN(1,1)
RCORN(2,2)=RCORN(1,2)
RCORN(1,1)=IVEX(2,1)
RCORN(1,2)=IVEX(2,2)
RETURN
END
```

```
C#####
C#
C# FEATURE EXTRACTION : FEATUR.FTN #
C#
C# TOPFGR(2,2) #
```

```

C#*****
C#
C#     FEATURE EXTRACTION : FEATUR.FTN
C#
C#     FIGUR(2,2,3)
C#           <-----> RANGE : 1 : TOP,2:FACE,3:SIDE
C#           1 1-----> TYPE : 1:LENGTH,2:ANGLE
C#           1-----> MODE : 1:SUM,2:DIFFERENCE
C#     LNGTH      : LENGTH OF OUTER EDGES
C#     INLNTH     : LENGTH OF INNER EDGES
C#     AN         : ANGLE OF 2-LINE CROSS ANGLES
C#     INA        : ANGLE OF 3-LINE CROSS ANGLES
C#     SIDANG     : ANGLE OF INNER SIDE ANGLES
C#     VLNTH      : SUM ON LENGTH OF VERTICAL EDGES
C#     BTLNTH     : SUM ON LENGTH OF BOTTOM EDGES
C#     FACDIF     : 1 => DIFFERENCE OF FACING LENGTHS
C#                2 => DIFFERENCE OF FACING ANGLES
C#
C#     REMARKS :
C#           THIS PROGRAM TRIES TO FIND OUT 14 DIFFERENT TYPES
C#           OF FEATURES :
C#           1. THE SUM OF EDGES OF TOP,FACE,AND SIDE RANGE
C#           2. THE DIFFERENCE OF ADJACENT EDGES OF TOP REGION
C#           3. THE DIFFERENCE OF FACE EDGES OF TOP REGION
C#           4. THE SUM OF ANGLES OF TOP,FACE,AND SIDE REGION
C#           5. THE DIFFERENCE OF ADJANCE ANGLES OF TOP REGION
C#           6. THE DIFFERENCE OF FACE ANGLES OF TOP REGION
C#           7. THE SUM OF VERTICAL EDGES
C#           8. THE SUM OF BOTTOM EDGES
C#
C#*****
C#     SUBROUTINE FEATUR(FIGUR,FACDIF,VLNTH,BTLNTH)
C#     REAL LNGTH(20),INLNTH(5,5),AN(20),INA(5,5,3),SIDANG(5,2)
C#     REAL FIGUR(2,2,3),FACDIF(2),VLNTH,BTLNTH
C#     COMMON /C/LNGTH,INLNTH,AN,INA,SIDANG
C#
C#     ---->     FEATURES FOR TOP REGION     <-----
C#
C#     CALL CAL(LNGTH(6),INLNTH(1,1),SIDANG(2,1),INA(1,3,1),

```

```

CALL CAL(INLNTH(1,1),INLNTH(1,3),INA(1,3,1),SIDANG(4,2),
-   FIGUR(1,1,1),FIGUR(2,1,1),FIGUR(1,2,1),FIGUR(2,2,1))
CALL CAL(INLNTH(1,3),LNGLTH(5),SIDANG(4,2),AN(6),FIGUR(1,1,1),
-   FIGUR(2,1,1),FIGUR(1,2,1),FIGUR(2,2,1))
CALL CAL(LNGLTH(5),LNGLTH(6),AN(6),SIDANG(2,1),FIGUR(1,1,1),
-   FIGUR(2,1,1),FIGUR(1,2,1),FIGUR(2,2,1))
C
C ---->   FEATURES FOR FACE REGION   <-----
C
CALL CAL(INLNTH(1,2),LNGLTH(3),SIDANG(3,2),AN(4),FIGUR(1,1,2),
-   FIGUR(2,1,2),FIGUR(1,2,2),FIGUR(2,2,2))
CALL CAL(LNGLTH(3),LNGLTH(4),AN(4),SIDANG(4,1),FIGUR(1,1,2),
-   FIGUR(2,1,2),FIGUR(1,2,2),FIGUR(2,2,2))
CALL CAL(LNGLTH(3),INLNTH(1,3),SIDANG(4,1),INA(1,3,1),
-   FIGUR(1,1,2),FIGUR(2,1,2),FIGUR(1,2,2),FIGUR(2,2,2))
CALL CAL(INLNTH(1,3),INLNTH(1,2),INA(1,3,1),SIDANG(3,2),
-   FIGUR(1,1,2),FIGUR(2,1,2),FIGUR(1,2,2),FIGUR(2,2,2))
C
C ---->   FEATURES FOR SIDE REGION   <-----
C
CALL CAL(LNGLTH(1),LNGLTH(2),SIDANG(2,1),AN(2),FIGUR(1,1,3),
-   FIGUR(2,1,3),FIGUR(1,2,3),FIGUR(2,2,3))
CALL CAL(LNGLTH(2),INLNTH(1,2),AN(2),SIDANG(3,1),FIGUR(1,1,3),
-   FIGUR(2,1,3),FIGUR(1,2,3),FIGUR(2,2,3))
CALL CAL(INLNTH(1,2),INLNTH(1,1),SIDANG(3,1),INA(1,1,1),
-   FIGUR(1,1,3),FIGUR(2,1,3),FIGUR(1,2,3),FIGUR(2,2,3))
CALL CAL(INLNTH(1,1),LNGLTH(1),INA(1,1,1),SIDANG(2,1),
-   FIGUR(1,1,3),FIGUR(2,1,3),FIGUR(1,2,3),FIGUR(2,2,3))
C
C ---->   FEATURE ON SUM OF VERTICAL LINES <-----
C ---->   FEATURE ON SUM OF BOTTOM LINES   <-----
C
VLNTH=LNGLTH(1)+LNGLTH(4)+INLNTH(1,2)
VDLNTH=ABS(LNGLTH(1)-LNGLTH(4))+ABS(LNGLTH(4)-INLNTH(1,2))
BTLNTH=LNGLTH(2)+LNGLTH(3)
C
C ---->   DIFFERENCE OF FACING EDGES ON TOP REGION <-----
C ---->   DIFFERENCE OF FACING ANGLES ON TOP REGION <-----
C
CALL CAL(LNGLTH(5),INLNTH(1,1),SIDANG(2,1),SIDANG(4,2),FACT1,
-   FACDIF(1),FACT2,FACDIF(2))
CALL CAL(LNGLTH(6),INLNTH(1,3),INA(1,3,1),AN(6),FACT1,FACDIF(1),
-   FACT2,FACDIF(2))
TYPE *,'====> THE SUM OF VERTICAL EDGES =',VLNTH
TYPE *,'====> THE SUM OF BOTTOM EDGES =',BTLNTH
TYPE *,'====> DIFFERENCE OF FACING LENGTH OF TOP REGION =',
-   FACDIF(1)
TYPE *,'====> DIFFERENCE OF FACING ANGLE OF TOP REGION =',
-   FACDIF(2)
CALL ASSIGN(1,'FEATUR.DAT')
WRITE(1,1000)FIGUR(1,1,1),FIGUR(2,1,1),FIGUR(1,2,2),FIGUR(2,2,1)
1000  FORMAT(1X,F9.5,1X,F9.5,1X,F9.5,1X,F9.5)
WRITE(1,2000)FACDIF(1),FACDIF(2)
2000  WRITE(1,2000)VLNTH,VDLNTH
      FORMAT(1X,F9.5,1X,F9.5)
CALL CLOSE(1)
RETURN
END
C.....
C.
C.   CALCULATE FOR FEATURES : CAL.FTN
C.
C.   L1,L2 : LENGTH OF TWO ARGMENTS
C.   A1,A2 : ANGLE OF TWO ARGMENTS
C.   F1,F2 : SUM AND DIFFERENCE OF LENGTH

```

C.....
SUBROUTINE CAL(L1,L2,A1,A2,F1,F2,F3,F4)
REAL L1,L2,A1,A2,F1,F2,F3,F4
F1=F1+(L1+L2)/2
F2=F2+ABS(L1-L2)
F3=F3+(A1+A2)/2
F4=F4+ABS(A1-A2)
RETURN
END

C##### 附錄二 #####

C#
C# SINIT.FTN : SET UP THE INITIAL PARAMETERS
C#

C#####
SUBROUTINE SINII(B,C,D,H,YO,ZO,FOC,PHI,IFNO,FLNM,ICHOSE,X1,X2,HZO)
BYTE FLNM(20)
TYPE *, '.... HOW MANY SECTORS DO YOU WANT ?

C#####
C#
C# SUBROUTINE OF STATIS.FTN TO CALL ASSIGN OR CALL CLOSE
C#

C#####
SUBROUTINE CHANEL(IFNO,FLNM,MODE)
BYTE FLNM(20)
INTEGER ILOG(16)
DATA ILOG/1,2,3,4,7,10,11,12,13,14,15,16,17,20,21,22/
C ... MODE : 0: CALL ASSIGN, 1: CALL CLOSE
IF(MODE.NE.0)GOTO 200
DO 10 I=1,IFNO
FLNM(6)=FLNM(6)+1
IF(MOD(I,10).NE.0)GOTO 20
FLNM(6)='0'
FLNM(5)=FLNM(5)+1
20 CALL ASSIGN(ILOG(I),FLNM)
10 CONTINUE
RETURN
200 DO 210 I=1,IFNO
210 CALL CLOSE(ILOG(I))
RETURN
END

CC
C FUCTION COSINE C
CC
FUNCTION COSIN(A1,A2,B1,B2)
REAL*8 DOT,SQA
DOT=A1*A2+B1*B2
SQA=(A1**2+B1**2)*(A2**2+B2**2)
IF(SQA.EQ.0)COSIN=1
IF(SQA.NE.0)COSIN=DOT/SQRT(SQA)
RETURN
END

```

C# SUBROUTINE OF FTHETR.FTN
C#
C#####
SUBROUTINE FTHETR(I,K,EDG,ALPHA,FTHI)
REAL EDG(10),ALPHA(10)
FTHI=0.
GOTO (10,20,30,40,50),I
10 DO 15 J=1,K
15 FTHI=FTHI+EDG(J)
RETURN
20 DO 25 J=1,K
25 FTHI=FTHI+ABS(EDG(J)-EDG(J+1))
RETURN
30 DO 35 J=1,2
35 FTHI=FTHI+ABS(EDG(J)-EDG(J+2))
RETURN
40 DO 45 J=1,K
45 FTHI=FTHI+ABS(ALPHA(J)-ALPHA(J+1))
RETURN
50 DO 55 J=1,2
55 FTHI=FTHI+ABS(ALPHA(J)-ALPHA(J+2))
RETURN
END

```

```

C#####
C#
C# WONEEG.FTN : WRITE THE FEATURES OF ONE EDGE
C#
C#####
SUBROUTINE WONEEG(B,C,D,H,YO,ZO,FOC,X1,X2,PHI,HZO)
INTEGER ILOG(16)
REAL*4 EDG(10),ALPHA(10)
DATA ILOG/1,2,3,4,7,10,11,12,13,14,15,16,17,20,21,22/
C .. PARAMETERS OF UP AND DOWN FACE ..
DO 200 THI=0.,X1,X2
CALL PARAME(B,C,D,H,YO,ZO,FOC,PHI,THI,EDG,ALPHA,HZO)
DO 230 IN=1,5
CALL FTHETR(IN,4,EDG,ALPHA,FTHI)
WRITE(ILOG(IN),1000)THI,FTHI

230 CONTINUE
200 CONTINUE
1000 FORMAT(1X,F5.3,1X,F9.5)
RETURN
END

```

```

C#####
C#
C# AAB.FTN : CALCULAT EDGE OF UP AND DOWN FACE
C#
C#####
SUBROUTINE AAB(B,C,D,H,YO,ZO,F,THI,PHI,AX,BX,EDG,HZO)
REAL*4 P1,AX,BX
BD=B-D
BCOSYO=B*COS(THI)+YO
AX=(B*C+(C*COS(THI)+BD*SIN(THI))*YO)*COS(PHI)
- (B*SIN(THI)+C*COS(THI)-D*SIN(THI))*HZO*SIN(PHI)
BX=HZO*(BD*COS(THI)-C*SIN(THI))
P1=((BCOSYO*COS(PHI)-HZO*SIN(PHI))**2) *
- (((C*SIN(THI)+D*COS(THI)+YO)*COS(PHI)-HZO*SIN(PHI))**2)
EDG=SQRT(F**2*(AX**2+BX**2)/P1)
RETURN
END

```

```

C#####
C#
C# VAAB.FTN : CALCULATE THE EDGE OF VERTICAL EDGE
C#
C#####

```

```

CDYO=(C*SIN(THI)+D*COS(THI))+YO
AX=(C*COS(THI)-D*SIN(THI))*H*SIN(PHI)
BX=CDYO*H
P1=((CDYO*COS(PHI)-HZO*SIN(PHI))*2) *
  ((CDYO*COS(PHI)+ZO*SIN(PHI))*2)
EDG=SQRT(F**2 * (AX**2 + BX**2)/P1)
RETURN
END

```

```

C#####
C#
C#   SUBROUTINE OF PARAME.FTN
C#
C#####
SUBROUTINE PARAME(B,C,D,H,YO,ZO,F,PHI,THI,EDG,ALPHA,HZO)
REAL   EDG(10),ALPHA(10),AX(10),BX(10)
PI=3.1416
CALL AAB(B,C,D,H,YO,ZO,F,THI,PHI,AX(1),BX(1),EDG(1),HZO)
CALL AAB(-B,C,D,H,YO,ZO,F,THI,PHI,AX(2),BX(2),EDG(2),HZO)
CALL AAB(B,C,D,H,YO,ZO,F,THI+PI,PHI,AX(3),BX(3),EDG(3),HZO)
CALL AAB(-B,C,D,H,YO,ZO,F,THI+PI,PHI,AX(4),BX(4),EDG(4),HZO)
AX(5)=AX(1)
BX(5)=BX(1)
EDG(5)=EDG(1)
IF(H.EQ.0)RETURN
DO 300 I=1,4
      ALPHA(I)=COSIN(AX(I),AX(I+1),BX(I),BX(I+1))
CONTINUE
ALPHA(5)=ALPHA(1)

RETURN
END

```

```

C#####
C#
C#   WTWOEG.FTN : WRITE THE FEATURES OF TWO EDGES
C#
C#####
SUBROUTINE WTWOEG(B,C,D,H,PHI,FOC,YO,ZO,X1,X2,HZO)
REAL EDG(10),VED(10),ALPHA(10)
INTEGER ILOG(16)
DATA ILOG/1,2,3,4,7,10,11,12,13,14,15,16,17,20,21,22/
DO 550 I=1,16
      THI=0.,X1,X2
      CALL PARAME(B,C,D,0.,YO,ZO,FOC,PHI,THI,EDG,ALPHA,HZO)
      CALL VAAB(0,B,H,YO,ZO,FOC,THI,PHI,A1,B1,VED(1),HZO)
      CALL VAAB(C,D,H,YO,ZO,FOC,THI,PHI,A1,B1,VED(2),HZO)
      CALL VAAB(0,-B,H,YO,ZO,FOC,THI,PHI,A1,B1,VED(3),HZO)
      CALL VAAB(-C,-D,H,YO,ZO,FOC,THI,PHI,A1,B1,VED(4),HZO)
      CALL SRANGE(THI,EDG,VED,AFTHI,SFTHI,VAFTHI,VSFTHI)
      WRITE(ILOG(1),1000)THI,AFTHI
      WRITE(ILOG(2),1000)THI,SFTHI
      WRITE(ILOG(3),1000)THI,VAFTHI
      WRITE(ILOG(4),1000)THI,VSFTHI
550 CONTINUE
1000 FORMAT(1X,F5.3,1X,F9.5)
RETURN
END

```

```

C#####
C#
C#   STATIS.FTN : STATISTICS FOR 3-D OBJECT'S PARAMETERS
C#
C#####
BYTE FLNM(20)

CALL SINIT(B,C,D,H,YO,ZO,FOC,PHI,IFNO,FLNM,ICHOSE,X1,X2,HZO)
..... MODE : 0: CALL ASSIGN, 1: CALL CLOSE
MODE=0
CALL CHANFI(TFNO,FLNM,MODE)

```

```

IF(ICHOSE,NE,1)CALL WTWOEG(B,C,D,H,PHI,FOC,YO,ZO,X1,X2,HZO)
MODE=1
CALL CHANEL(IFNO,FLNM,MODE)
END

```

```

C#####
C#
C# SRANGE.FTN : SUBROUTINE OF STATIS.FTN
C# CALCULATE THE RELATION OF TWO EDGES
C#
C#####

```

```

SUBROUTINE SRANGE(THI,EDG,VED,AFTHI,SFTHI,VAFTHI,VSFTHI)
REAL EDG(10),VED(10)
AFTHI=0.
SFTHI=0.
VAFTHI=0.
VSFTHI=0.
IF(THI.GE.0 .AND. THI.LE.0.6715)GOTO 1
IF(THI.GE.0.6715 .AND. THI.LE.0.7)GOTO 2
IF(THI.GE.0.7 .AND. THI.LE.2.2141)GOTO 3
IF(THI.GE.2.2141 .AND. THI.LE.2.2496)GOTO 4
IF(THI.GE.2.2496 .AND. THI.LE.3.8131)GOTO 5
IF(THI.GE.3.8131 .AND. THI.LE.3.8416)GOTO 6
IF(THI.GE.3.8416 .AND. THI.LE.5.3557)GOTO 7
IF(THI.GE.5.3557 .AND. THI.LE.5.3912)GOTO 8
IF(THI.GE.5.3912 .AND. THI.LE.6.2832)GOTO 9
TYPE *, '..... ERROR ON THETA-RANGE !'
1 AFTHI=EDG(4)+EDG(1)
SFTHI=ABS(EDG(4)-EDG(1))
VAFTHI=VED(1)+VED(2)+VED(4)
VSFTHI=ABS(VED(1)-VED(2))+ABS(VED(1)-VED(4))
RETURN
2 AFTHI=EDG(1)
SFTHI=0
VAFTHI=VED(1)+VED(2)
VSFTHI=ABS(VED(1)-VED(2))
RETURN
3 AFTHI=EDG(2)+EDG(1)
SFTHI=ABS(EDG(2)-EDG(1))
VAFTHI=VED(1)+VED(2)+VED(3)
VSFTHI=ABS(VED(1)-VED(2))+ABS(VED(2)-VED(3))
RETURN
4 AFTHI=EDG(2)
SFTHI=0
VAFTHI=VED(3)+VED(2)
VSFTHI=ABS(VED(3)-VED(2))
RETURN
5 AFTHI=EDG(2)+EDG(3)
SFTHI=ABS(EDG(2)-EDG(3))
VAFTHI=VED(3)+VED(2)+VED(4)
VSFTHI=ABS(VED(3)-VED(2))+ABS(VED(3)-VED(4))
RETURN
6 AFTHI=EDG(3)
SFTHI=0
VAFTHI=VED(3)+VED(4)
VSFTHI=ABS(VED(3)-VED(4))
RETURN
7 AFTHI=EDG(4)+EDG(3)
SFTHI=ABS(EDG(4)-EDG(3))
VAFTHI=VED(1)+VED(3)+VED(4)
VSFTHI=ABS(VED(3)-VED(4))+ABS(VED(1)-VED(4))
RETURN
8 AFTHI=EDG(4)
SFTHI=0
VAFTHI=VED(1)+VED(4)

```

```
AFTHI=EDG(4)+EDG(1)
SFTHI=ABS(EDG(4)-EDG(1))
VAFTHI=VED(1)+VED(2)+VED(4)
VSFTHI=ABS(VED(1)-VED(2))+ABS(VED(1)-VED(4))
RETURN
END
```

PASSWORD:

RSX-11M BL26 MULTI-USER SYSTEM

GOOD MORNING

10-MAR-84 09:49 LOGGED ON TERMINAL TT0:

Welcome to RSX/11M V3.2 Real-time system
Institute of Information Science, Academia Sinica

CC *Converting $f_i(\theta)$ into distribution function,*
>PIP TT0:=DISFUN.FTN

```
DIMENSION THI(1000),Y(1000),A(91),D(91),E(91),F(91),G(91)
BYTE OUTFNM(20),INFNM(20),AGAIN
999 TYPE *,'. . . . . PLEASE IN THE INPUT-FILENAME . . . . . '
ACCEPT 3000,(INFNM(I),I=1,19)
TYPE *,'. . . . . PLEASE INPUT THE OUTPUT-FILENAME . . . . . '
ACCEPT 3000,(OUTFNM(I),I=1,19)
3000 FORMAT(80A1)
CALL ASSIGN(2,INFNM)
CALL ASSIGN(3,'DISOUT.DAT')
CALL ASSIGN(7,OUTFNM)
C . . . . . Y IS THE FUNCTION VALUE DEPENDS ON THI
READ(2,1)(THI(I),Y(I),I=1,1000)
1 FORMAT(1X,F5.3,1X,F9.5)
CALL MAX(Y,A1,A2)
A(91)=A1
A(1)=A2
D(1)=0
D(91)=0
H=(A1-A2)/90
C . . . . . A(I) IS Y-AXIS OF DISTRIBUTION
C . . . . . D(I) IS DISTRIBUTION FUNCTION OR PROBABILITY OF (X<=I)
DO 1000 1000 I=2,90
D(I)=0
A(I)=A(I-1)+H
1000 I=2
J1=1
K=0
C . . . . . Y INCREASE
5 DO 2 J=J1,1000
IF(Y(J1).GT,A(I))GOTO 3
IF(Y(J).LT,A(I) .AND. Y(J+1).GE,A(I))GOTO 4
IF(Y(J).GT,Y(J+1))GOTO 77
2 CONTINUE
4 D(I)=THI(J)
J1=J+1
GOTO 73
3 D(I)=D(I-1)
73 IF(I,EQ,91)GOTO 77
I=I+1
GOTO 5
77 I2=I
I21=I2-1
E(I21)=D(I21)
DO 78 78 IS=I2,91
D(IS)=D(IS-1)
E(IS)=E(IS-1)
78 TYPE *,'. . . . . READ JSH. . . . . '
7 ACCEPT *,JSH
DO 70 103 K=1,JSH
DO 70 70 I=1,I2-1
70 E(I)=0
WRITE(3,105)(D(I),I=1,91)
105 FORMAT(20F6.3)
T-T2-1
```

```

C ..... Y DECREASE .....
50 DO 20 J=J1,1000
    IF(Y(J+1).GT.Y(J))GOTO 8
    IF(Y(J+1).LT.A(I) .AND. Y(J).GE.A(I))GOTO 40
    IF(Y(J1).LT.A(I))GOTO 41
20 CONTINUE
40 E(I)=THI(J)
    J1=J+1
    GOTO 74
41 E(I)=E(I+1)
74 IF(I.EQ.1)GOTO 8
    I=I-1
    GOTO 50
8 E(I)=THI(J)
    I1=I
    DO 9 I=1,91
9 F(I)=0
    WRITE(3,105)(E(I),I=1,91)
    I=I1
    F(I)=THI(J)
    I=I+1
    J1=J+1
C ..... Y INCREASE .....
15 DO 12 J=J1,1000
    IF(Y(J1).GT.A(I))GOTO 16
    IF(Y(J).LT.A(I) .AND. Y(J+1).GE.A(I))GOTO 14
    IF(Y(J+1).LT.Y(J))GOTO 17
12 CONTINUE
14 F(I)=THI(J)
    J1=J+1
    IF(J.GE.1000)GOTO 17
    GOTO 71
16 F(I)=F(I-1)
    IF(J.GE.1000)GOTO 17
71 IF(I.EQ.91)GOTO 17
    I=I+1
    GOTO 15
17 I2=I
    F(I)=THI(J)
    IF(I2.EQ.91)GOTO 29
    DO 28 I=I2+1,91
    F(I)=F(I-1)
28 CONTINUE
29 WRITE(3,105)(F(I),I=1,91)
    DO 18 I=1,91
    G(I)=F(I)-E(I)
18 D(I)=D(I)+G(I)
    E(I2)=F(I2)
    IF(I2.EQ.91)GOTO 103
    DO 30 I=I2+1,91
    E(I)=E(I-1)
30 CONTINUE
103 WRITE(3,101)(A(I),D(I),I=1,91)
100 DO 102 I=1,91
102 D(I)=D(I)/6.2832
101 FORMAT(2F12,8)
    WRITE(7,101)(A(I),D(I),I=1,91)
    CALL CLOSE(2)
    CALL CLOSE(3)
    CALL CLOSE(7)
    TYPE *, '..... WANT TO TRY AGAIN ? (Y/N)'
    ACCEPT 3000,AGAIN
    IF(AGAIN.EQ.'Y')GOTO 999

```

```
      END  
C.....  
      SUBROUTINE MAX(X,A,B)  
      DIMENSION X(1000)  
      A=X(1)  
      B=A  
      DO          1      I=2,1000  
                IF(A.LT.X(I))A=X(I)  
                IF(B.GT.X(I))B=X(I)  
1      CONTINUE  
      RETURN  
      END  
>
```

DIR


```

C ..... FUNCTION : DISTF(X,
FUNCTION DISTF(X, A, DIST, N)
REAL A(N), DIST(N)
C ... EVALUATE DISTF(X) BY INTERPOLATING DISTF(X) ALONG DIST(A(1),
C ... DIST(A(N))
C A SHOULD BE IN ASCENDING ORDER
IF(X-A(1))3, 3, 4
3 DISTF=0
RETURN
4 IF(X-A(N))5, 6, 6
6 DISTF=1
RETURN
5 DO 1 I=2, N
IF(A(I-1).LT.X .AND. X.LE.A(I))GOTO 2
1 CONTINUE
2 DELTA=(X-A(I-1))/(A(I)-A(I-1))
DISTF=(1.-DELTA)*DIST(I-1)+DELTA*DIST(I)
RETURN
END

```

```

C ..... MAIN PROGRAM FOR CRAMER-VON MISES STATISTICS FOR THE
C PURPOSE OF 3-D PARALLELEPIPED PATTERN CLASSIFICATION
C
C REAL A(91), DIST(91), DATAM(9, 80), FEAT(80)
C ..... A AND DIST ARE DISTRIBUTION TERMS : DIST(A(I))
C
C BYTE INFNM(20), AGAIN
CALL ASSIGN(3, 'FEATUR.DAT')
C ..... READ 9 FEATURES FOR 80 OR LESS PICTURES ...
C
C TYPE *, '... INPUT NUMBER OF PICTURES : NP '
ACCEPT 3001, NP
3001 FORMAT(I2)
TYPE *, '.... READ FEATUR DATA : DATAM'
READ(3, 3003) ((DATAM(I, J), I=1, 9), J=1, NP)
3003 FORMAT(9(1X, F9.5))
C
C .... READ THEORETICAL DISTRIBUTION DATA A(I), DIST(I)
C
3002 FORMAT(2(F12.8))
DO 1 I=1, 9
TYPE *, '.... PLEASE INPUT THE THEORETICAL FILE NAME '
ACCEPT 3000, (INFNM(I), I=1, 19)
3000 FORMAT(20A1)
CALL ASSIGN(2, INFNM)
READ(2, 3002) ((A(K), DIST(K)), K=1, 91)
CALL CLOSE(2)
DO 2 J=1, NP
FEAT(J)=DATAM(I, J)
CALL ORD(FEAT, NP)
C ..... INPUT CRITICAL VALUES
C
C TYPE *, '... INPUT CRITICAL VALUES ....'
ACCEPT 3002, CRITIC
C
C //// RUN CRAMER-VON MISES TEST ////
C
C ANP=NP
WSQ=0
DO 3 J=1, NP
3 WSQ=WSQ+ (DISTF(FEAT(J), A, DIST, 91)
- (2*J-1.) / (2. *ANP)) **2
WSQ=WSQ+1. / (12. *ANP)

```

```

IF (STAT19.GT.CRIT10)GOTO 4
TYPE *, '... ACCEPT FEATURE :', I
GOTO 1
4
TYPE *, '**** REJECT FEATURE :', I
1
CONTINUE
STOP
END
C..... SUBROUTINE : ORD (V, NOV) .....
SUBROUTINE ORD (V, NOV)
C
C ARRANGE V IN ASCENDING ORDER
C
REAL V(100)
L=NOV
1
L=L/2
IF (L) 2, 2, 3
2
RETURN
3
M=NOV-L
DO 6 I=1, M
4
J=I
JJ=J+L
IF (V(J)-V(JJ)) 6, 6, 5
5
V1=V(J)
V(J)=V(JJ)
V(JJ)=V1
J=J-L
IF (J) 6, 6, 4
6
CONTINUE
GOTO 1
END

```