**1.什麼是corner point？**

Corner point就是在圖形中，由兩段不同的edge結合、角度較小的點(如下圖)。



**2.為什麼要做corner detection？**

corner points 其實是影像處理裡面，所謂的interest points(feature points)的一種。而找出interest points有什麼意義呢？比如說在影像的拼接裡，原始的方式就是一個一個點來比較，可是這樣既花時間，又容易出錯(因為不同的影像可能會有不一樣的雜訊，或是rgb值不一致)。而藉由interest points，我們就可以比較這些點分布的狀況，來決定該怎麼拼接這兩張圖片。

feature points的使用非常廣泛，如圖像校準，動態偵測、視覺追蹤、影像的拼接…etc.，都常使用feature points來判斷。

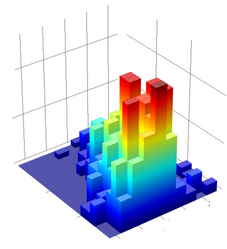
因此，如何正確的找出feature point，也是影像處理中十分重要的一環。

**3.Corner detector分類：**

目前所看到的paper大致可分為3+1類：

1. Intensity-based：

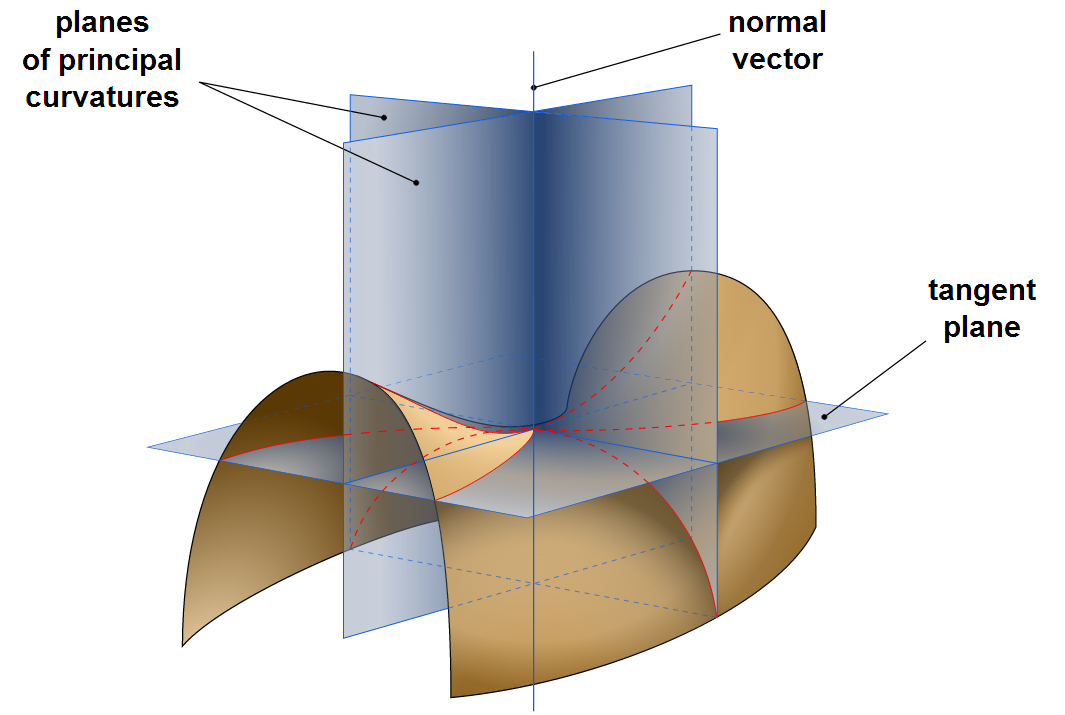
因為圖片的分布可以視為一個三維的圖形(如下圖)，所以我們可以利用一些幾何的性質來找出我們要的corner point。



這種類型的corner detector大多都是利用影像的gradient來進行運算，也就是藉由與周圍的點相減，得出的差來比較，由gradient的圖形來看(如下圖)，可以發現越是邊界的地方gradient值會越高，

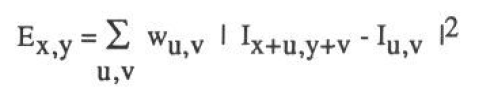


首先要介紹的是十分有名，而且最常被拿來比較的Harris & Stephens corner detector **[2]**(以下簡稱harris detector)，harris detector的核心概念就是：藉由gray-level的curvature來判斷，一般來說一個pixel附近的曲率如果是local maximum的地方有可能就是一個圖型的edge或corner。曲率的概念如下圖：

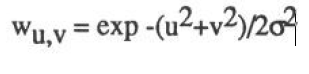


根據在不同principle axis的曲率計算後的結果，可以判斷出這個點到底是edge或corner，可是我們只知道x方向跟y方向的gradient，怎麼樣才能夠知道這個點的curvature呢？

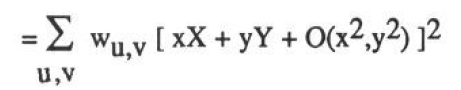
首先，先利用欲檢測的點，與其他周圍的點相減求出gradient(另外還會根據不同的點來賦予權重)，



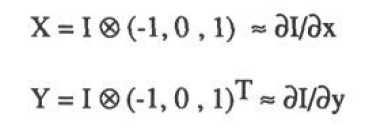
mask使用gaussian kernal



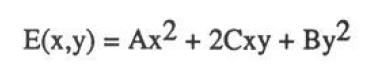
Ex,y可以使用泰勒展開求出下式



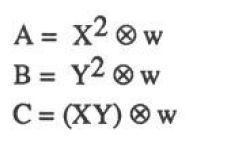
其中



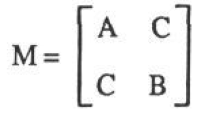
展開以後可以得到



A,B,C分別為：

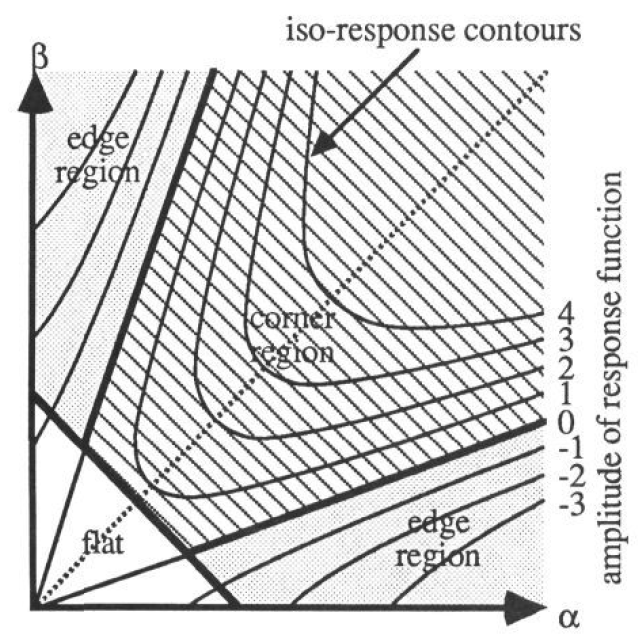


因為是quadratic form因此可以把A,B,C寫成下列矩陣

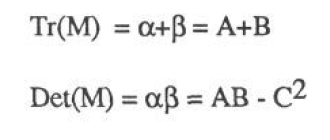


這個矩陣所找出來的eigenvalue以及eigenvector，分別就是在這個點周圍的curvature以及principle axis，利用不同方向的

Curvature 大小，可以得知這個點到底是什麼點，

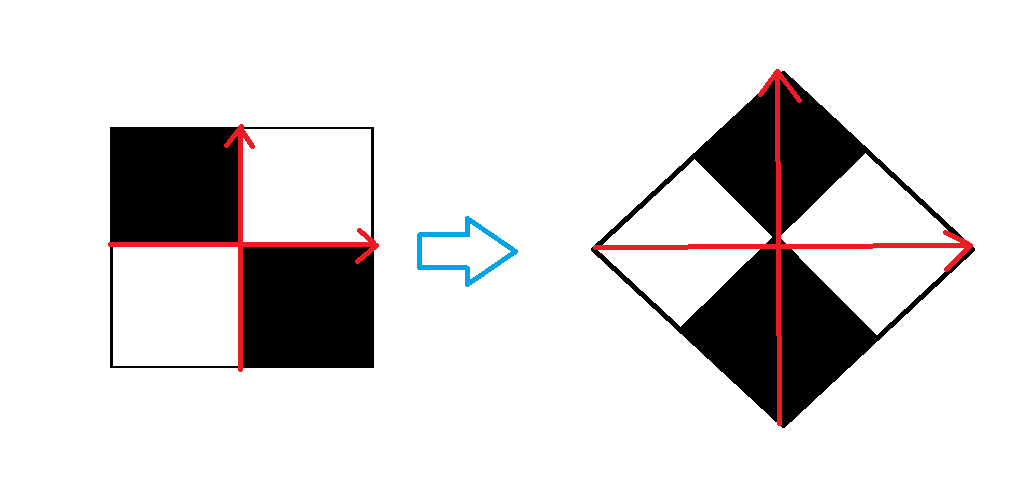


而harris定義R來做為判斷的依據





harris的做法的優點是他可以直接從圖形的principle axis方向去看曲率，就像是將圖形翻轉過來(如下圖)，



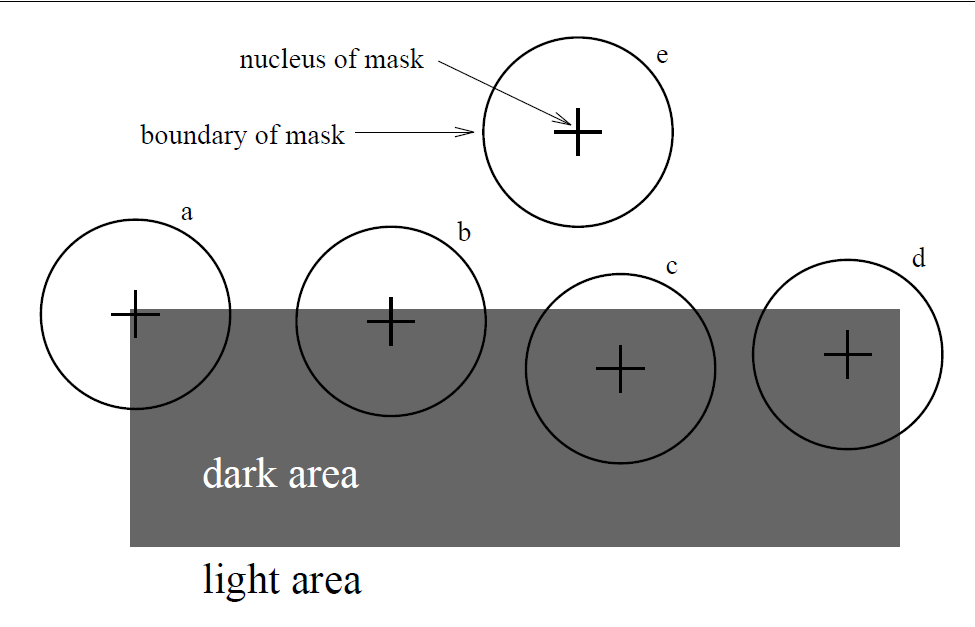
所以當圖片進行翻轉以後，仍然可以找出相同的corner point，

這種做法的缺點是實作上需要花上不少的時間。

老師跟貝老師的論文**[3]**可以改善雜訊的問題，而且加上了判斷的部分，可以更精確的找出corner(但也更花時間)。

另外一個有名的作法是SUSAN，是使用一個圓形的mask，計算

周圍pixels跟核心的點的數值來判斷一個點是否為Corner。

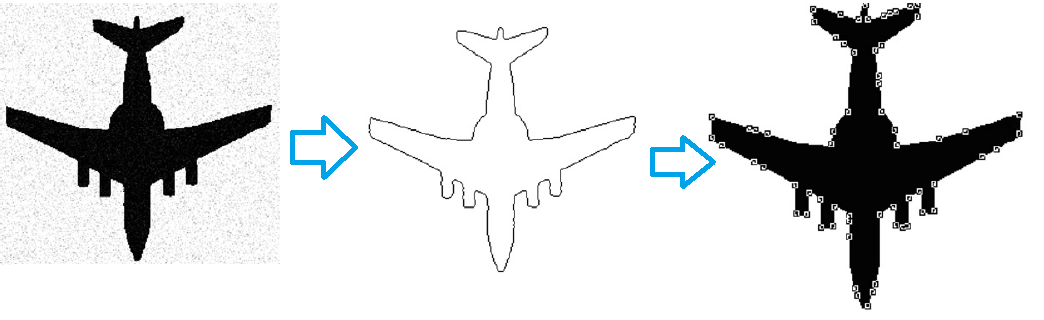


由圖形可以看出，核心點在不同的位置，結果就會有不同的差別，藉由這種方式來判斷corner。

1. Boundary-based：

Boundary-based的演算法是利用edge detector先將edge找出，

再利用找出的邊界來判斷圖形的corner point，如下圖：



可以明顯的看出來，這種做法與Intensity-based最大的差異就是由3D的概念變為2D的概念。

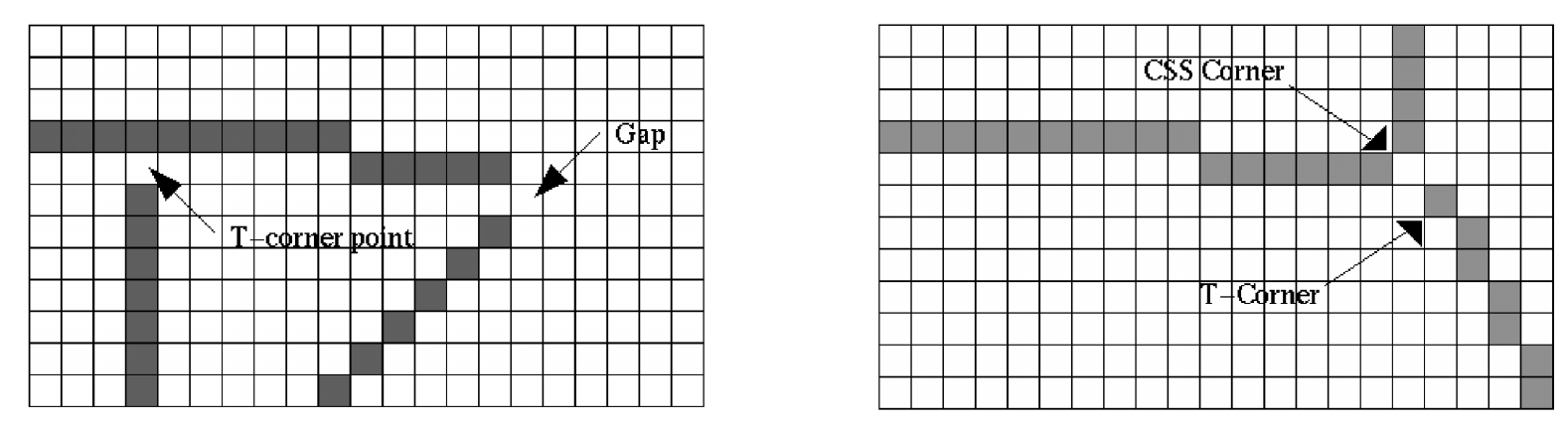
這種做法的優點是經由良好的edge detector，可以濾掉不必要的資訊(如雜訊)，經由boundary的特性來判斷此點是否是corner，像是利用線段的曲率(ex: CSS**[4][5])**，由於變為2D的圖形，在2D- Cartesian coordinate space 中，一個座標為 γ(t) = (*x*(*t*),*y*(*t*))的線段，其曲率定義為

\kappa = \frac{|x'y''-y'x''|}{(x'^2+y'^2)^{3/2}},

很直觀的，一個連續線段曲率最大的地方，非常有可能是它的角點，所以我們就可以利用這個概念來找出我們所想要的角點。

在[4]這篇paper裡，就是利用這個概念來做，首先為了補足edge

detector的問題，會先將找出來的邊界變細，再將不連續之處補起來。例如下圖



接著針對找出來的邊界：



對其跟gaussian function做convolution



可以得到



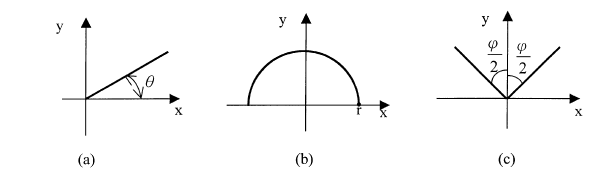
再利用曲率的概念



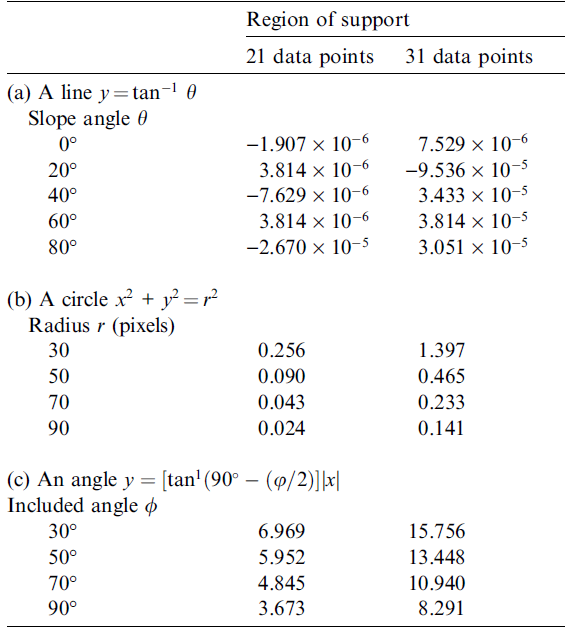
可以得出曲率，這篇論文利用不斷變化σ值，判斷corner的變化來找出真正的corner point。

另外還有利用坐標的covariance matrix 來做判斷**[6][7]**，

這個概念是利用eigenvalue或是eigenvector來判斷，若是一個橢圓形的線段，利用所有座標來計算出covariance matrix後，eigenvalue的值就是其長軸跟短軸的值，如果是一條直線，則其中一個eigenvalue 會非常趨近於0，用下列表格來看就會十分清楚，對於圖形(a)(b)(c)：



其較小的eigenvalue λs(也就是短軸長度)：



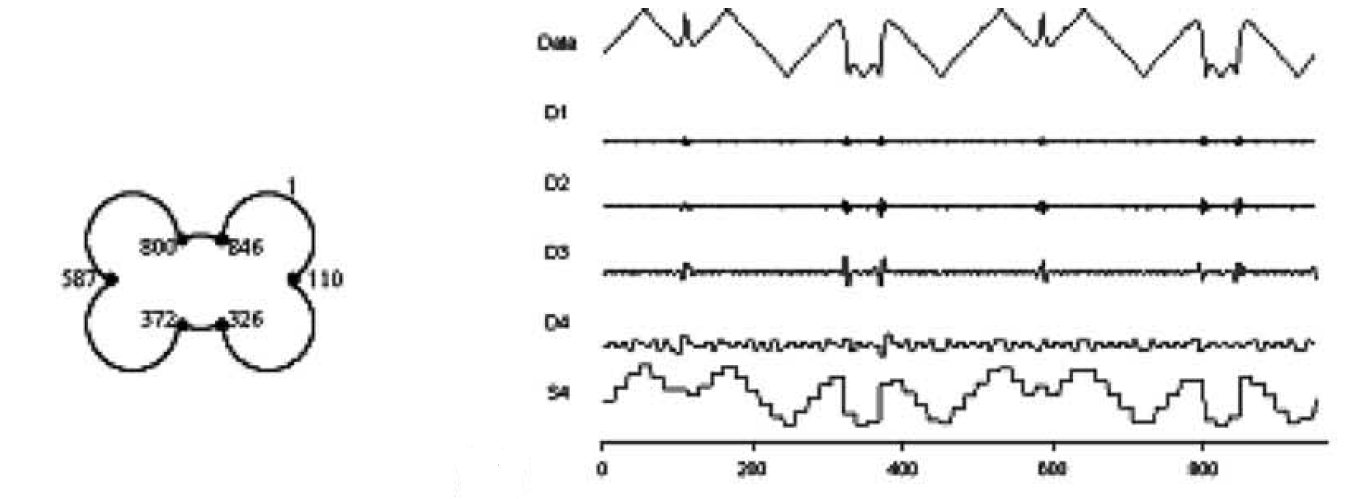
可以看到若是直線，λs極小，另外較大的eigenvalue應該極大，而關於圓形部分，參考的點越少，其λs越小，這其實跟弧長與曲率的概念有關，綜合以上，我們利用λs的特性來判斷是否為corner point。

雖然做完boundary再來找corner會讓判斷變得較為單純(變為 one-dimension的集合)，但是缺點就是必須要有一個良好的edge detector來做到好的boundary，否則在尋找上就會有困難，因為目前看到的boundary-based corner detector 都預設edge是一個一個pixel的排列下去，如果碰到線段太厚、或是有斷點的情形出現，就必須重新後製，而且當圖型過於複雜時，要判斷的線段集合就會非常多，反而讓判斷變的複雜起來(因為要先設定線段集合)

因此boundary-based corner detector應該較適用於比較單純的圖形。

1. Wavelet 的運用：

目前看到的部分是先做edge detect，再把線段的一些特性當成是Input signal(如由線段的covariance matrices的eigenvector求出的角度**[8]**)，再利用小波去將訊號高頻的地方找出來，即為corner，如下圖：



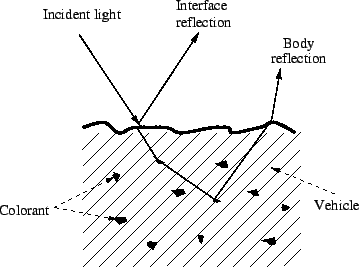
跟上面boundary-based的方法的相同缺點就是必須有個良好的edge detector ，而且因為是edge detector找出來的圖形，當圖形是由多重線段組成時(如五芒星)，要如何去判斷圖形的集合呢？這就變成要分割集合，或是重複加入之前設為其他集合的點，要考慮的狀況就會變得十分的複雜。

1. Color ：

這也是一種有趣的判斷方式，color invariant feauture的判斷方式其實199X年就有提出了，這邊只介紹我所看過的最新paper，也就是[9]，利用Dichromatic reflection model ，公式如下：



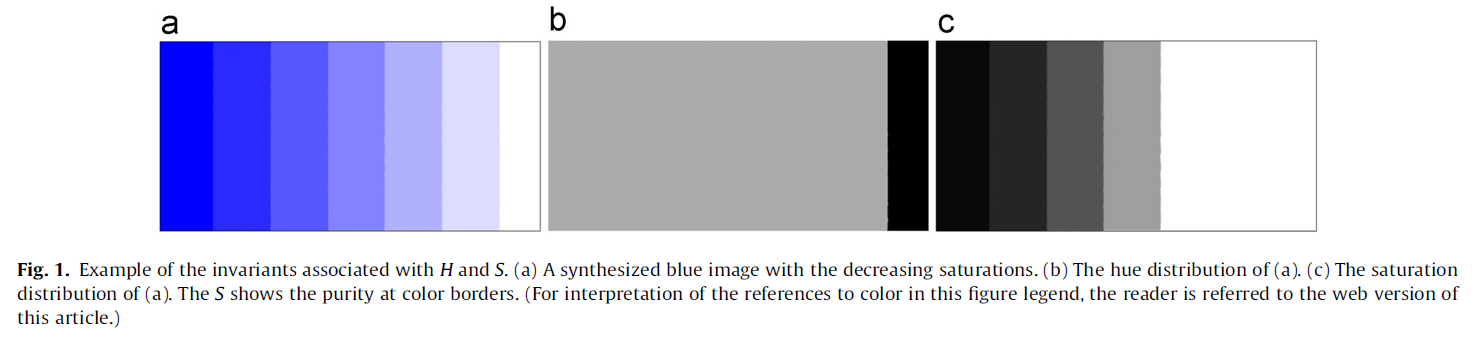
其概念是，光源投射到物體上後，會從表面以及物體內部反射到我們的眼中，所以可分為body 跟 interface兩個部分的反射光。(如下圖)



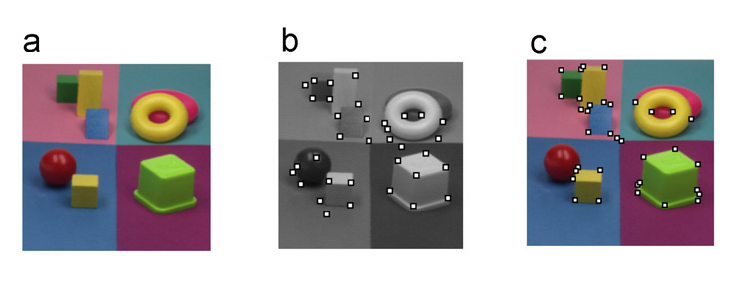
在假設單一且平均的光源下(如太陽光)，可推導出hue 與saturation的 color invariant ，藉由用這個數值去做corner detect(利用harris corner detector)，可以得出我們所要找的corner points。

而最關鍵的response，E值要怎麼找呢？可以利用rgb值，經過gaussian color model得到我們想要的數值。

這種做法的好處是可以免除掉一些光澤及陰影的影響，如果像原本灰階的方式，就會在陰影處找到corner，



比如說像上圖，a是原本的圖片，b是hue的分布圖，c是saturation的分布圖，可以看到跟原本轉成灰階的結果有些差別。利用這個概念，就可以得出與灰階的corner detector不同的corner points.



上圖(a)是原圖，(b)是harris detector 的結果，(c)是這篇paper演算法的結果，可以看出來在光澤與陰影的地方，並不像harris會誤判(這篇paper把陰影及光澤的部分設為不該搜尋到的corner point)

這種做法的缺點是，處理的資料是rgb，所以比原本的數值還要多，還有原本的假設是單一且平均的光源，如果光源不平均，或是非單一光源，會不會影響到結果？這都是有可能的問題。

**.如何判斷一個corner detector 的好壞？**

最主要可以用兩點來判斷一個corner detector 的好壞：

1. **效率(efficiency)：**

corner 往往被用來當作一個圖片的feature，通常會利用這些

feature來做影像處理判斷的依據，如果把太多資源花在尋求

feature的步驟上，就會影響到後面影像處理的速度。要改善速

度，就要讓corner detector的實作簡單，或是使用其他方式來增進效能(ex: machine learning)

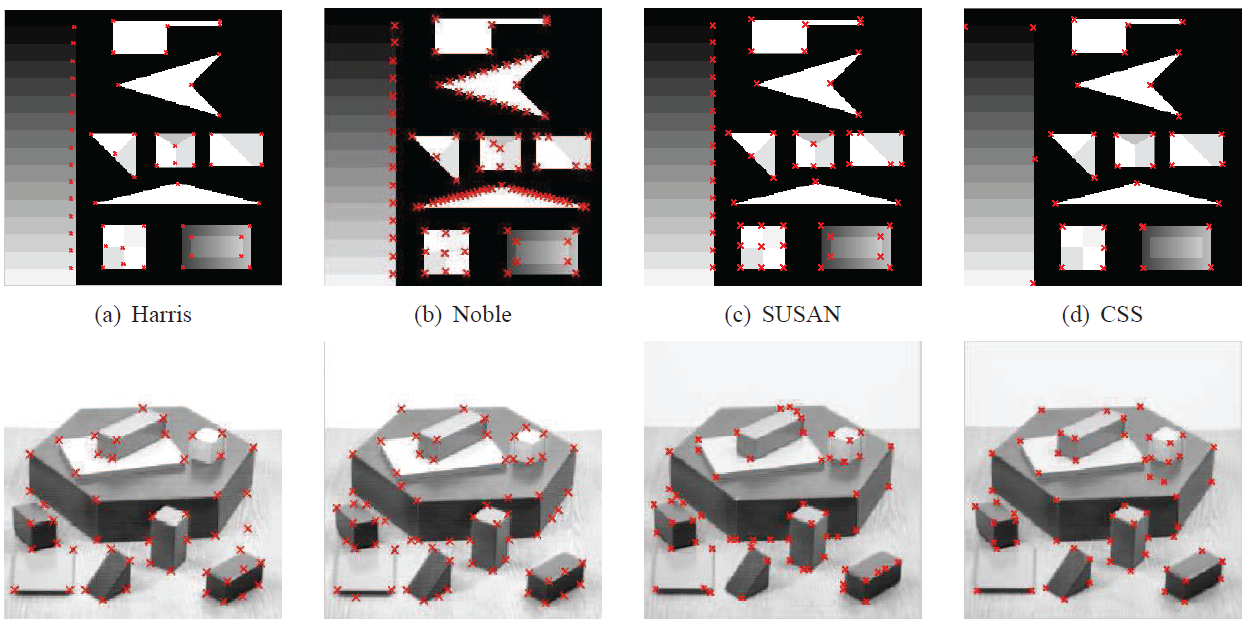
1. **重複性(repeatability)：**

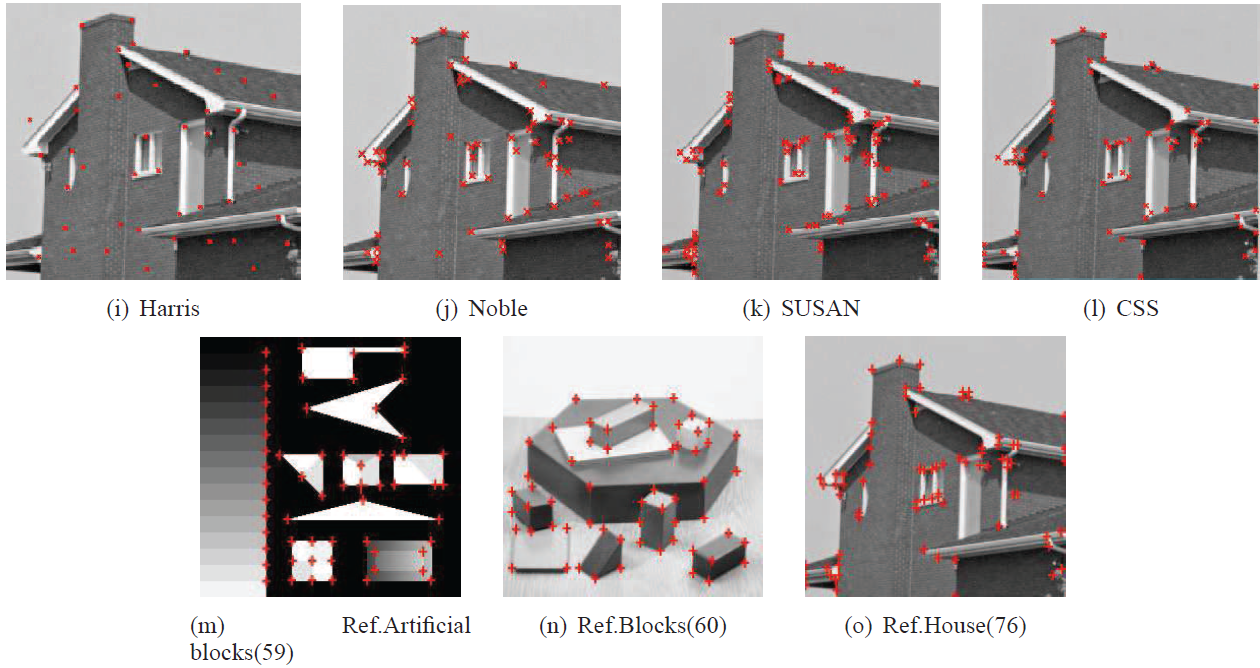
Repeatability代表著一個corner detector 在各種情況下的效率。

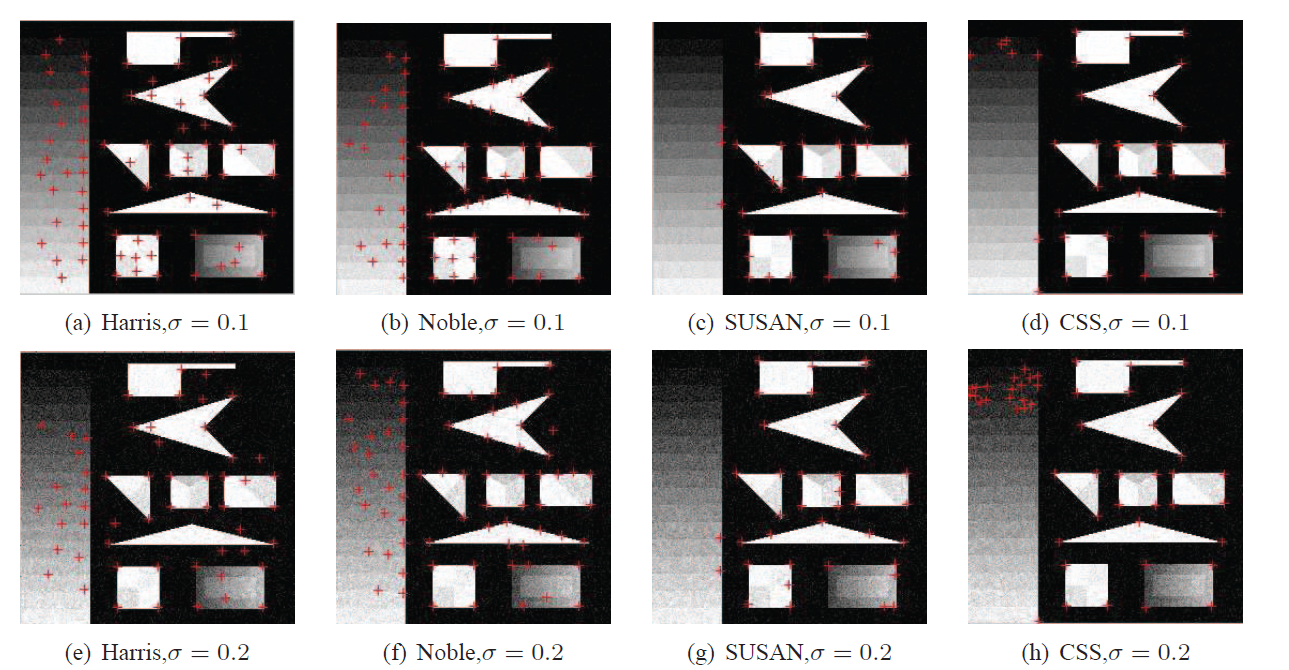
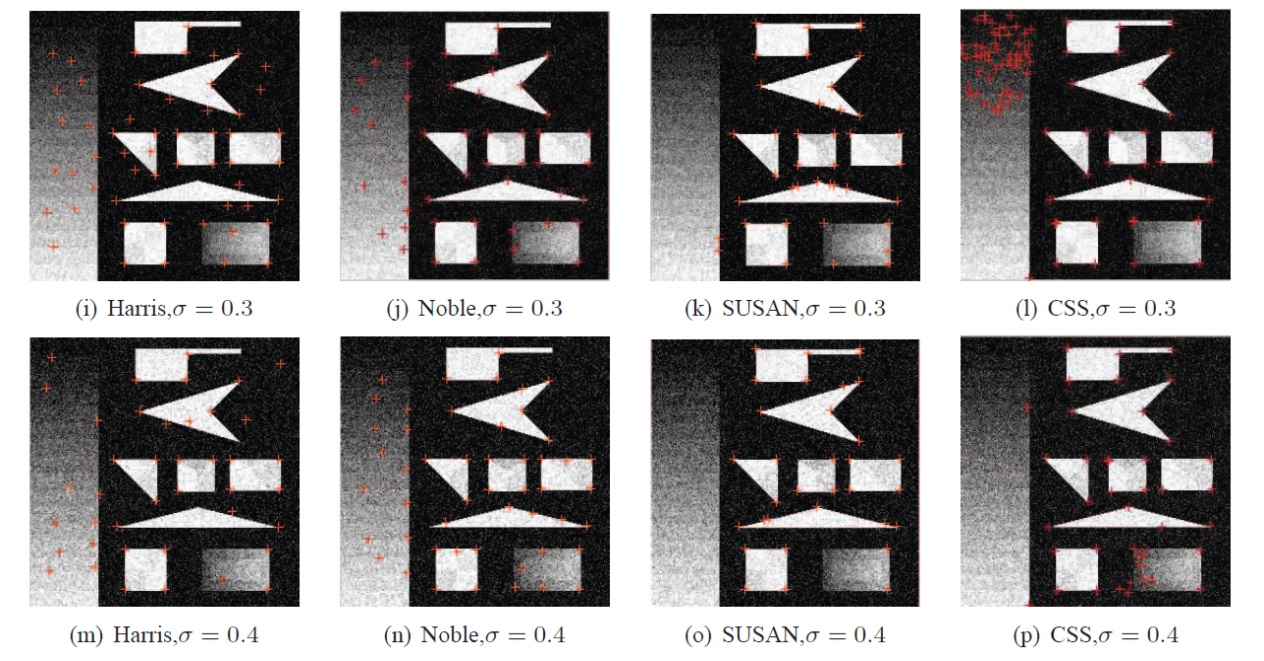
通常在判斷一張標準的測試圖以後(ex: 積木圖、房子圖、單純的幾何圖)

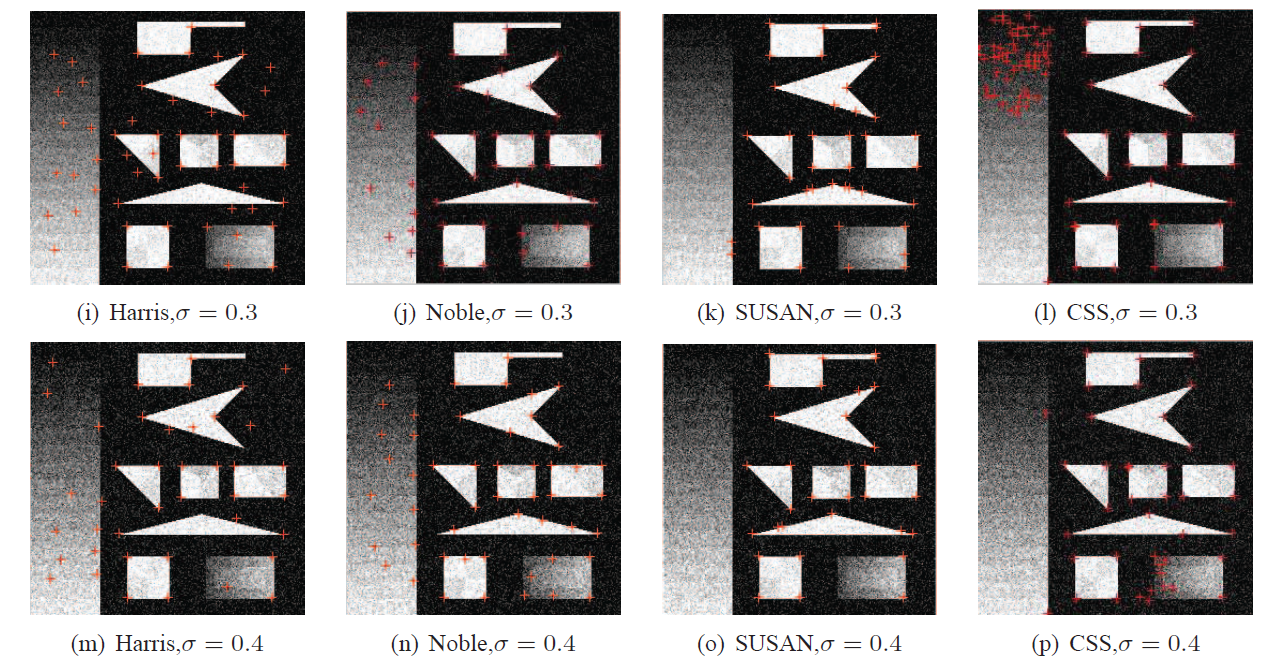
還會將這張圖分別做增加雜訊、幾何處理、放大縮小圖片、從不同視角的重新拍攝的動作，然後對這些圖片重新做一次corner detect，然後利用一些數學工具來判斷跟原本圖片結果的差異(ex: ACU**[1]**、CCN**[1]**、LE)

**以下是一些corner detections 的比較：**









可以看到，SUSAN在漸層的部分的抗雜訊能力似乎較為強健。不過在invariant部分，其實是harris會比較佳，而碰到這種簡單的圖形，boundary-based的CSS就占上風了。

**Summary：**

近年來比較廣泛利用的feature detector其實是SIFT、SURF，因為它的旋轉跟尺度變化不變性，能夠讓它廣泛的在各種狀況使用，但是SIFT的缺點就是所需要消耗的時間較多(相對某些演算法)。

不過，近年來其實有個比較有名的corner detector(我所看到的部分)，就是FAST演算法(features from accelerated segment test)，因為他的快速判斷能力，可以在需要real-time判斷的場合中使用，雖然準確性沒有像SIFT那麼高，但是速度十分的快速。不過FAST牽扯到machine learning的概念，所以我並不是很了解實際上的做法(目前沒有學到)，

可以參考**[10][11][12]**，不過由此可看出，藉由machine learning的方式，可以加快目前演算法的速度，也許也是可以發展的方向？

其實corner detector的好壞並不能用單純某方面去看，每個corner detector都擁有它的優點，就像上列的兩種演算法，SIFT無法在real-time的應用上被使用，而FAST也沒有像SIFT的演算法如此準確。所以到底要選擇哪種corner detector來使用，在不同的狀況下都會有不同的結果，孰優孰劣，必須要由實際應用的環境來判斷。

References：

**[1]** Mohanna F. , Mokhtarian F. 2001 “Performace Evaluation of Corner Detection

Algorithms under Similarity and Affine Transforms.”

**[2]** Harris, C., Stephens, M., 1988.” A combined corner and edge detector.”

**[3]** Soo-Chang Pei, Jian-Jiun Ding, 2007“IMPROVED HARRIS’ ALGORITHM FOR

CORNER AND EDGE DETEC-TIONS”

**[4]** Mokhtarian F. and Suomela Riku. IEEE TRANSACTIONS ON PATTERN ANALYSIS

AND MACHINE INTELLIGENCE VOL. 20, NO. 12,(1998) ” Robust Image Corner Detection Through Curvature Scale Space.”

**[5]** Xiaohong Zhang , Ming Lei , Dan Yang , Yuzhu Wang , Litao Ma . Pattern

“Recognition Letters 28 (2007) 545–554 .Multi-scale curvature product for robust image corner detection in curvature scale space.”

**[6]** Du-Ming Tsai , H.-T. Hou, H.-J. Su. Pattern Recognition Letters 20 (1999) 31-40

“Boundary-based corner detection using eigenvalues of covariance matrices”

**[7]** Xiaohong Zhang a,, HongxingWangb, AndrewW.B.Smith c, XuLing a, BrianC.Lovell

c, DanYang . Pattern Recognition 43(2010)1207–1223 . “Corner detection based on gradient correlation matrices of planar curves”

**[8]** Chi-Hao Yeh. Pattern Recognition Letters 24 (2003) 2797–2806 . “Wavelet-based

corner detection using eigenvectors of covariance matrices”

**[9]** Jun Chu , JunMiao, GuimeiZhang, LuWang . Optics & Laser Technology 45 (2013) 756–762 “Edge and corner detection by color invariants “

**[10]** Fusing Points and Lines for High Performance Tracking. Edward Rosten and Tom Drummond.2005

**[11]** Machine learning for high-speed corner detection

**[12]** Adaptive and Generic Corner Detection Based on the Accelerated Segment Test

Elmar Mair, Gregory D. Hager, Darius Burschka, Michael Suppa, and Gerhard Hirzinger. ECCV(2011)