# Method

### 1. Rotate

1. 建立一張全黑圖(放旋轉後的結果)，以及**順時針**旋轉30度的旋轉矩陣

2. 遍歷全黑圖的所有點，找到他對中心順時針旋轉30度後，對應到原圖的座標位置(src\_x, src\_y)

3. 看位置是否超出界線，如果無就要對那個點做interpolation，因為旋轉完座標不是整數。

一張含有 文字, 螢幕擷取畫面 的圖片

AI 產生的內容可能不正確。

**Interpolation:**

Way 1(Nearest Neighbor Interpolation):

直接無條件捨去(src\_x, src\_y)找到小於他們最靠近的整數點

一張含有 文字, 字型, 螢幕擷取畫面, 數字 的圖片

AI 產生的內容可能不正確。

Way 2(Bilinear Interpolation):

一張含有 圖表, 螢幕擷取畫面, 行 的圖片

AI 產生的內容可能不正確。一張含有 文字, 螢幕擷取畫面, 字型, 軟體 的圖片

AI 產生的內容可能不正確。

先用y到附近左右兩點的距離比例，算出圖中AB兩點的值，存在tmp1與tmp2裡

再用這兩個值用x到兩點的距離比例，得到(src\_x, src\_y)真正的值

Way 3(Bicubic interpolation)

Bicubic interpolation的方法就是先找周圍最近的16個點，對每個水平的四個點算出大約的三次函數曲線，在算出該水平對應點的值，最後拿算出來的四個點再找一個三次函數，算出最後結果，code有點長就不貼了。算三次曲線的方法，是用以下推出來的公式，最後clip在(0~255)



### 2. Image wrapping and homography

一張含有 文字, 字型, 白色, 行 的圖片

AI 產生的內容可能不正確。 一張含有 文字, 字型, 數字, 白色 的圖片

AI 產生的內容可能不正確。

找到四組對應的點，要先找出左圖中的H矩陣，找到電視上的座標(x , y)對應原圖(x’, y’)的H矩陣。

用右圖的大矩陣，找出四組點，並使用NumPy裡的linear Algebra function解出H(h11~h32)

最後遍歷電視裡的點，找到對應原圖的src\_x與src\_y，再用前面講的一模一樣的三種interploation得出那個pixel的值。

# 2. Result

一張含有 螢幕擷取畫面, 室內, 視窗, 設計 的圖片

AI 產生的內容可能不正確。一張含有 室內, 視窗, 設計 的圖片

AI 產生的內容可能不正確。一張含有 室內, 螢幕擷取畫面, 視窗, 設計 的圖片

AI 產生的內容可能不正確。

由左而右分別為 Nearest neighbor、bilinear與bicubic interpolation

一張含有 室內, 牆, 油畫, 相框 的圖片

AI 產生的內容可能不正確。一張含有 室內, 牆, 油畫, 相框 的圖片

AI 產生的內容可能不正確。一張含有 室內, 牆, 電視, 油畫 的圖片

AI 產生的內容可能不正確。

由左而右分別是Nearest neighbor、bilinear與bicubic interpolation

# Feedback

因為空間不夠，把結果圖片縮小，所以可能看不太出來，但是看原圖可以明顯看出差異。

使用nearest neighbor interpolation會明顯有馬賽克的感覺，因為我們插值的方法就單純是取靠近的pixel的value值，每次插值只考慮附近一個點，所以旋轉後的圖可能會有幾個pixel對應到原圖同樣的地方，造成資訊變少、畫質變差，有馬賽克的感覺。

使用bilinear interpolation有明顯變好的感覺，因為考慮了附近四個pixels的值，相較nearest neighbor interpolation畫質有變好許多。

使用bicubic interpolation又比bilinear interpolation更好了，許多物品邊界處理有更好的效果，變得更加的銳利，還有後面山脈的細節處理更好，肉眼看畫質明顯更好許多，因為這個方法足足考慮了周圍16個點，用了5個三次函數後才得到每個pixel應該賦予的值，運算複雜度與執行時間也需要更長。