

● Lab Objective

在這個 Lab 中，我們透過實作幾個簡單的影像處理的 function 來熟悉 MATLAB 以及一些影像處理的技巧。

● Design Implementation

1. Result

A. RGB \rightarrow YUV

Original



Y



U



V



Original



Y



U

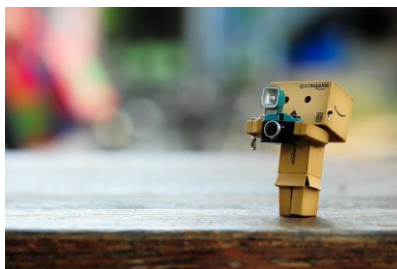


V



B. Flipping

Horizontal



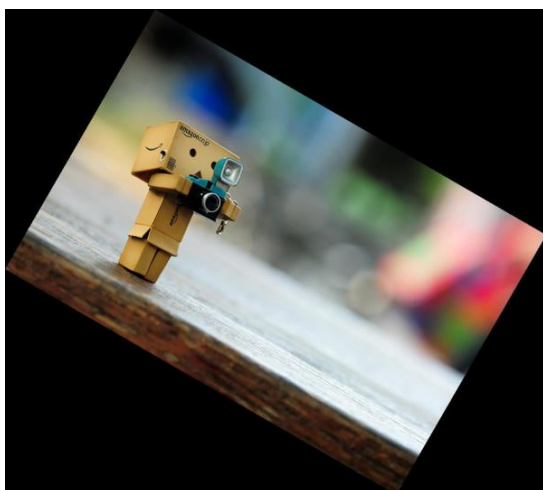
Vertical



Horizontal + Vertical



C. Rotation



D. Shear



E. Resize

1024 x 681



614 x 408



1920 x 1920

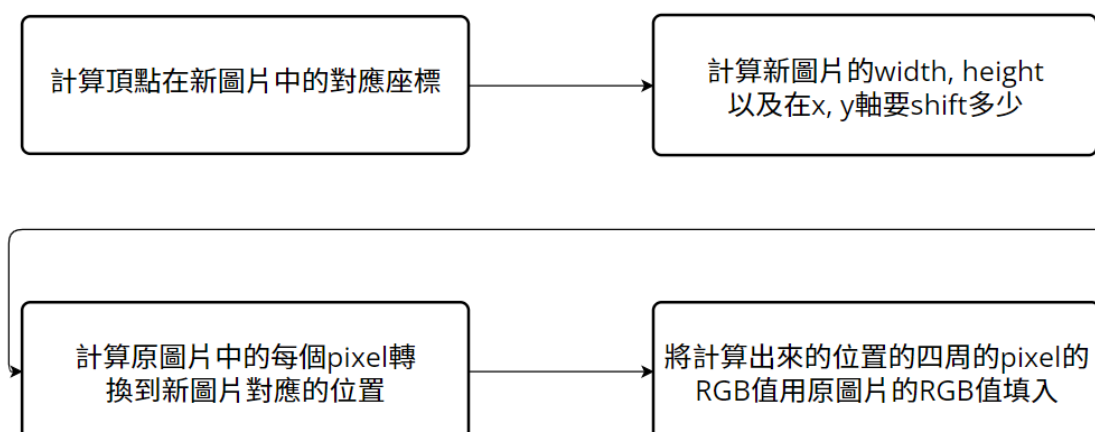


1152 x 1152

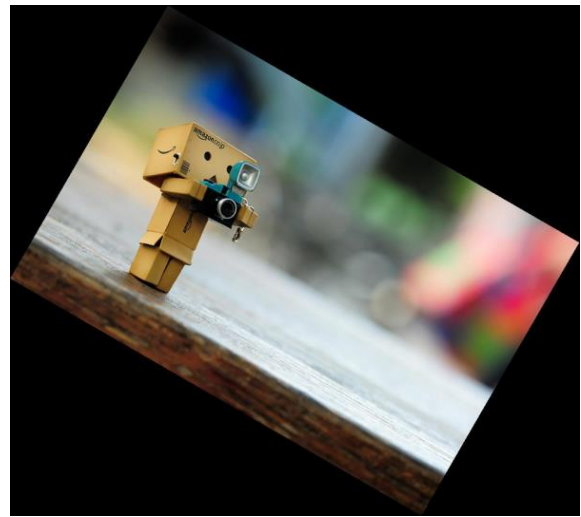


2. Forward Warping

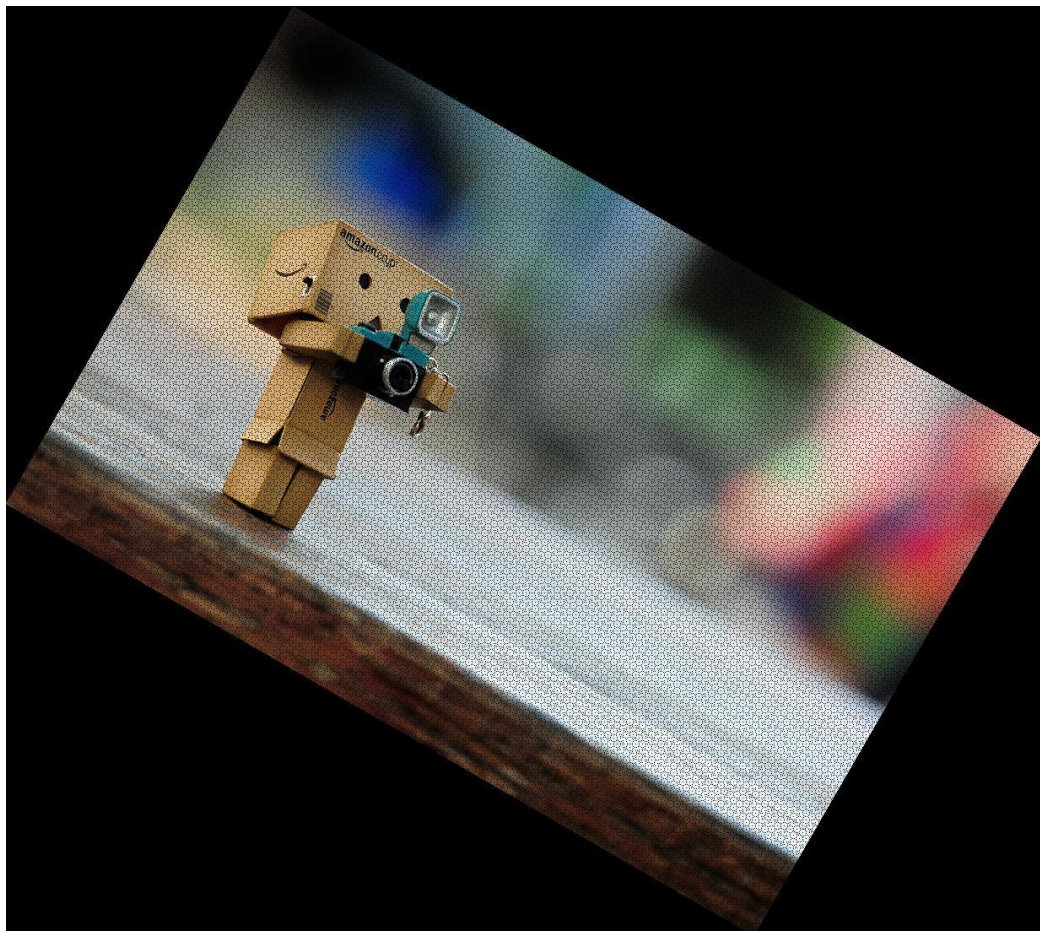
我所採用的流程圖如下，詳細的 code 我寫在 rotation_forward.m 中。Forward Warping 與 Backward Warping 的最大差別在於我們要將原圖片的座標轉換到新圖片上。其餘流程都跟 Backward Warping 類似，另一個差別在於我們沒辦法做 Bilinear interpolation 因為我們沒有新圖片的任何資訊，我們採用另一種方式叫 **splatting**，這部分我採用最簡的的方法：將對應座標四周的 pixel 都填如原先位置的 RGB 值。



實作結果如下，左邊是 Backward Warping 右邊是 Forward Warping，可以發現結果相似。



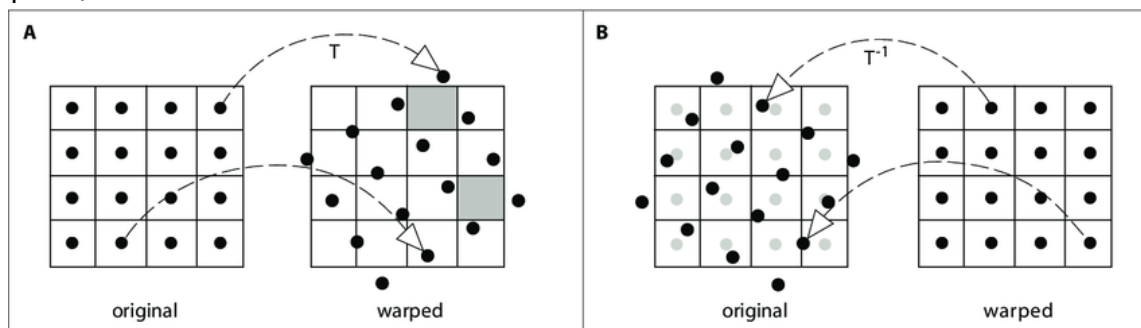
若我們沒有做 **splatting**，結果如下。可以看到有一個一個小黑點(hole)出現在圖片中，這些黑點的出現原因可以看下一題的討論。簡單來說就是我們經過轉換後可能會有某些 pixel 沒有座標點掉在裡面。



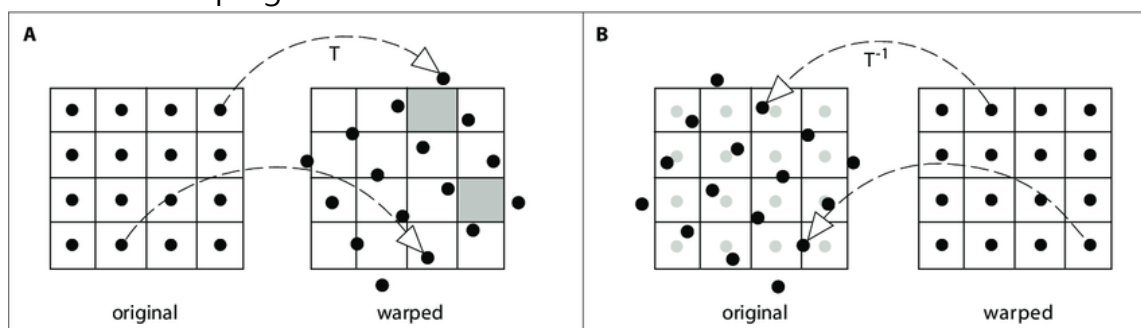
3. Backward Warping v.s Forward Warping

Forward Warping 與 Backward Warping 的差別在於座標的轉換。Forward Warping 是將原先的 image 的座標轉換到新的 image 的座標，接著將原先座標的 RGB 值填到對應的轉換座標。Backward Warping 則是將新的 image 的座標經過反矩陣轉換回原先 image 的座標，再透過一些方式(ex. Bilinear Interpolation)計算出對應的 RGB 值。

Forward Warping 的好處在於我們不需要去計算反矩陣，降低計算複雜度也避免反矩陣可能不存在的問題，其缺點則在於會出現有 **hole** 的情況，如下圖所示(A 圖)，可以看到我們轉換過去之後會出現有某幾個 pixel 沒有點掉再裡面的狀況，這就會導致這個 pixel 是黑色的。要解決這個問題，我們可以透過一些方式去填補這些洞(ex. splatting, 將每個轉換過去的 pixel 根據一些 weighting function(例如：距離)，去將這個 pixel 存的 RGB 值分給周圍的 pixel)。



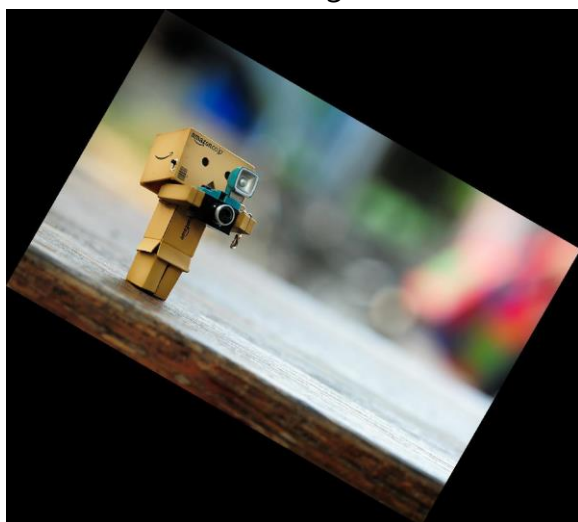
相反的，若我們採用 **backward warping**(B 圖)，轉換回來的座標亦有可能不掉在某個 pixel 中，但我們可以透過四周的 pixel 去做 **interpolation** 來找出視覺上可以接受的 RGB 值。然而，Backward Warping 的缺點也很明顯，就是反矩陣不一定存在，計算上也更加複雜。



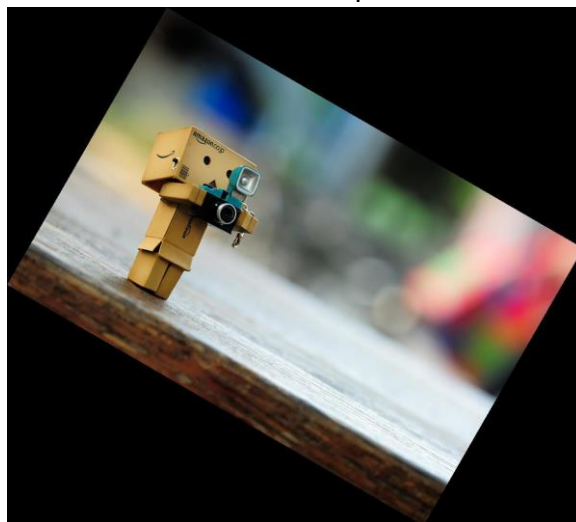
4. Nearest-Neighbor for Backward Warping in rotation

這部分我們嘗試將 Backward Warping 中用的 bilinear interpolation 換成 nearest-neighbor method，並比較兩者在結果上有何不同。結果如下：

Nearest-neighbor



Bilinear Interpolation



大體上來看兩者差異不大，但如果我們放大來看(下圖)可以發現，用 **bilinear interpolation** 的圖片中，物體的邊界較為平滑，而使用 **nearest neighbor** 的圖片的邊界則有毛邊，變化較大。我認為這個原因就在於 Bilinear Interpolation 就類似在做加權平均，等價於 1D 訊號的 moving average filter，會讓訊號更加平滑，而 nearest neighbor 則類似於 floor 和 ceiling 這兩個 function，他們都是階梯函數，邊界區分明顯。

Bilinear Interpolation



Nearest Neighbor



● Conclusion

這個 Lab 讓我熟悉 MATLAB 的語法，體驗用程式的方式去實現一些數學上的操作。Backward Warping 以及 Forward Warping 則是全新的知識，理解兩種不同的方法之間的 tradeoff(複雜度 v.s 人工生成資料)。希望這些知識能對我後面的 Lab 有很大的幫助。

● References

- 教授與助教的講義
- Prof. Chung Yung-Yu' s handout
[Microsoft PowerPoint - lec02_morphing.ppt \[相容模式\] \(ntu.edu.tw\)](#)