

## MATLAB 自動影像校準(Image Registration)技術

By Garima Sharma and Andy Thé, MathWorks 公司

影像校準(image registration)是從兩個或多個數據來進行影像對齊的過程，它牽涉到影像的整合以建立整合視圖、改善訊號的雜訊比(signal-to-noise ratio)，並從多張圖片中萃取(extract) 那些無法從單一影像中獲得的資訊。影像校準用於遙測(remote sensing)、醫學影像、製圖(cartography)，以及其他需要從影像中獲得正確資訊的應用，如-從衛星圖片發現淹水面積、MRI 掃描檢測腫瘤或工廠檢測等等。

該如何確定有效的影像校準方法依情況而異，可以是一個複雜和費時的過程，需要仔細選擇一個點轉換模式來提供影像間的參考點，及建立一個比較資訊的方法來界定正確對齊影像所需要的適當參數。

有兩個著名的方法可進行自動影像校準流程：以特徵(feature-based)為基礎和以影像強度(intensity-based)為基礎的校準演算法。在這篇文章中，我們將使用一個偵測發燒的例子來說明，以影像強度為基礎、利用 MATLAB imregister 函式及 MathWorks 的影像處理工具箱(Image Processing Toolbox)相關功能來建立的自動影像校準流程。這個工作流程是一個快速有效的方式，可以從不同的相機整合影像。

### 影像校準詞彙

**參考(固定)圖像：**影像在目標方向，指定為 2D 或 3D 影像灰階度

**目標(移動)圖像：**影像被轉換和參考圖像對齊，指定為 2D 或 3D 影像灰階度

**以影像強度為基礎的校準：**以影相對強度特徵值(intensity patterns)為基礎來對齊影像

**以特徵為基礎的校準：**使用特徵偵測(feature detection)、萃取(extraction)和匹配(matching)來對齊影像

## 發燒偵測實例：目標與挑戰

當嚴重急性呼吸系統綜合症（SARS）2003 年爆發時，台灣的桃園國際機場開始掃描檢查乘客發燒症狀以遏制這種致命病毒的蔓延，由於無法逐一檢查每位乘客，臨床醫生用紅外線熱成像、非侵入式技術透過分析溫度數據的紅外線攝影像來偵測發燒與否。

雖然這種方法有效，但真正實踐起來卻很有挑戰性。紅外線攝影機對環境條件非常敏感，必須考量所有可能影響溫度讀數的元素，才能正確地對攝影機進行調校，包括室內環境溫度、相對濕度、表面反射，以及攝影機和被檢體對象間的距離。有效的溫度掃描也取決對身體能產生可靠溫度資訊的部位識別要有一致性，在我們的例子中，例如眼睛周圍。

我們使用 MATLAB<sup>®</sup>、FLIR 紅外線（IR）攝影機以及 webcam 來建立一個溫度掃描機雛形(prototype)。紅外線（IR）攝影機可以測量臉部溫度 100 毫 K (millikelvins)的增量，而 webcam 則提供更多臉部特徵的細部資訊。透過校準這兩個不同來源的影像，我們將能夠偵測 webcam 攝影影像的眼睛周圍位置(圖 1)，並由紅外線攝影機得到的影像測量眼睛周圍的溫度。



圖 1：Webcam 影像

## 擷取影像和紅外線攝影機調校(Calibrating)

使用影像擷取工具箱™( Image Acquisition Toolbox™)，我們把從 webcam 和紅外線攝影機抓取到的影像輸入到 MATLAB 的工作區中，紅外線攝影機使用的是 GigE Vision® 介面，webcam 則使用標準的 interface 介面。

為了調校紅外線攝影機，我們調整了被攝體的距離、濕度、輻射係數（物理輻射能量的表面相對值，即物質吸收與輻射能量能力的指標，取決於溫度、發射角度及波長），以及其它特性。在影像擷取的過程中，大氣溫度為 295.15K、牆壁和受試者輻射係數為 0.98。

```
%% IR Camera object creation and configuration:
irCam = imaq.VideoDevice('gige',1,'Mono16');
irCam.ReturnedDataType = 'native';
irCam.DeviceProperties.IRFormat = 'TemperatureLinear100mK';
irCam.DeviceProperties.ObjectDistance = 0.91;
irCam.DeviceProperties.AtmosphericTemperature = 295.15;
irCam.DeviceProperties.ObjectEmissivity = 0.98;

%% Webcam object creation:
webCam = imaq.VideoDevice('winvideo');

%% Acquire snapshots
imageIR = step(irCam);
imageRGB = step(webCam);
```

## 影像視覺化

利用影像處理工具箱(Image Processing Toolbox)內的函式 imshow() 來看紅外線攝影機攝得的影像，因為我們已經擷取了 16-bit 的資料，其中實際讀到的溫度讀數測定為 100 毫 K(mK)的增量，在影像顯示到螢幕之前，我們依據數據進行了對比調整(contrast adjustment)(圖 2)。



圖 2，紅外線影像的對比調整

我們使用 `imshowpair()` 函數在同一視窗中顯示兩個影像

```
%% Read in the image pair
Fixed = imread('IR_image.png');
Moving = imread('Webcam_image.png');
Moving = rgb2gray(Moving);

%% View the 2 images
imshow(Moving);
% Use imshow to view the IR image
imshow(Fixed);

%% View the images side by side in a montage
imshowpair(Fixed,Moving,'montage');
```

這個函數提供了幾個視覺化的選項，包括 “falsecolor” -使用不同的顏色波段 (color band)來建立合成的 RGB 影像，和 'blend'-用於視覺化 alpha 混合影像 (blended images) (圖 3)。

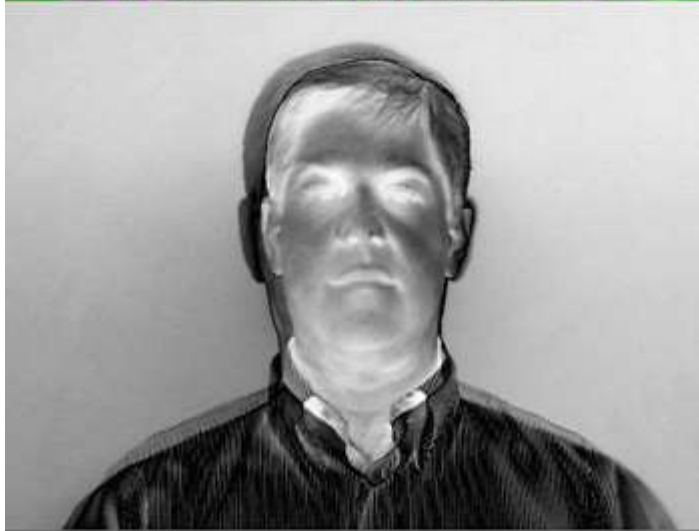


圖 3: 從上到下依次為 : 'falsecolor' , 'diff' , 和兩個影像 “blend” (混合) 的視覺效果。

## 影像校準

接下來開始影像校準過程，首先指定紅外線攝影機的影像為固定影像(fixed imag)，而 webcam 的影像當做移動影像(moving image)，固定影像是靜態參考，我們的目標是把移動影像和與固定影像對齊，由於以影像強度(intensity-based)為基礎的影像校準演算法需要灰階，所以我們把 webcam 攝得的影像，利用 'rgb2gray()' 函數將彩色影像轉換成灰階影像。

然後我們使用影像處理工具箱的 imregister() 函數來對齊影像，除了一對影像之外，以影像強度為基礎的自動影像校準還需要一個相似度準則(metric)、最佳化器(optimizer)和一個轉換型式，我們使用 imregconfig() 搭配 'multimodal' 選項，得到 'metric' 和 'optimizer' 值。然後我們再把得到的值代入 imregister() 函數中做為影像校準的起點。

```
%% Configure parameters in imregconfig
[optimizer,metric] = imregconfig('Multimodal');
```

影像校準開始，我們使用 imregister() 函數，預先設定(default)轉換型式 'translation'，然後使用 imshowpair() 來觀看結果。受試者在兩個影像中的輪廓都有些錯位(圖 4)，頭部周圍的影像和肩膀之間的差距，表示有縮放(scale)和旋轉(rotation)的問題。

```
%% Default registration
registered = imregister(Moving,Fixed,'translation',optimizer,metric);
figure;
imshowpair(registered,Fixed);
title('Default registration');
```





圖 4，預先設定的影像校準結果

請注意，要從以最佳化為基準的影像校準方法中獲得好的結果，經常是要對最佳化器(optimizer)和相似度準則(metric values)進行多次的調整；另外還要注意的是，雖然 `imshowpair()` 本函數預先設定的模式在我們的示範影像中運作的很好，但它可能不適用於所有的配對影像。最好嘗試去運用 `imshowpair()` 內各種不同的視覺化形式，例如 'falsecolor', 'diff', 'blend' 和 'montage'，才能針對特定的配對影像中找出最好的辨識結果。

考慮到縮放和旋轉扭曲，我們把 `imregister()` 中的轉換形式，從 'translation' 轉換成 'similarity'。

```
%% Final registration
registered = imregister(Moving, Fixed, 'Similarity', optimizer, metric);
figure;
imshowpair(registered, Fixed); title('Final Registration');
```

現在我們有一個相當準確的校準影像，眼睛部位都緊密對齊了（圖 5）。



圖 5: 最後的校準結果

畫面中呈現出綠色和紫紅色區域是因為影像來自不同的來源，不過它們並不代表校準失真。

### 偵測眼睛和讀取溫度

為了偵測眼睛，我們使用電腦視覺系統工具箱(Computer Vision System Toolbox™)內的階層式(cascade)物件偵測器，本物件偵測器使用 Viola-Jones 的演算法，也就是使用 Haar 類(Haar-like)的特徵和多階的 Gentle Adaboost 分類器(classifier)來偵測眼睛；然後在眼睛附近畫一個四方圖來強調影像內要校準的部位。



```

%% Detect the eyes in the RGB image
eyesDet = vision.CascadeObjectDetector('EyePairSmall');
bbox = step(eyesDet, Moving);
drawBox = vision.ShapeInserter('BorderColor','Black');
image = step(drawBox, registered, int32(bbox));
hold on; rectangle('Position',bbox,'EdgeColor',[1 1 0]);
subsIR = int32(bbox(:,1:2)+bbox(:,3:4)/2);

%% Compute temperature near the eyes
value = mean2(imcrop(registered,bbox));
foreheadTemperature = value/10 - 272; % In Celsius
foreheadTemperature = (foreheadTemperature*9/5) + 32; % Convert to
Fahrenheit

%% Embed temperature on IR image and display
ti = vision.TextInserter('Color',[255 0 0]);
ti.Location = int32(bbox(:,1:2)+bbox(:,3:4)/2);
ti.Text = sprintf('%3d F', int8(foreheadTemperature));
contAdj =
vision.ContrastAdjuster('CustomProductInputDataType',numerictype([],32,8));
imageContrastAdjusted = step(contAdj, Fixed);
textAdded = step(ti, imageContrastAdjusted);
text(320, 180, '98 \circ F ', 'Color',[1 1 0])

```

由於我們已經校準了影像，我們可以使用 webcam 影像中的眼睛方框，去抽樣紅外線影像靠近眼睛的溫度值，再把溫度的量測值顯示到校準圖像眼睛附近前我們先將溫度量測單位從毫 K 轉成華氏，如此一來我們可以看到受測者並沒有發燒（圖 6）。



圖 6: 校準後的影像與讀到的溫度值