

主題：影像處理

- 應用範例：TED 演講：“See invisible motion, hear silent sounds”

講者以及他的團隊透過影像處理的相關技術，來放大樣本影片中的顏色變化與細微動作，並發展出許多應用，像是：

1. 藉由放大臉部的顏色變化，來判斷出脈搏。由此可以不用透過實際接觸來測得脈搏。
2. 藉由放大胸部的起伏變化來判斷出呼吸頻率。
3. 藉由放大手腕的血管變化來判斷出脈搏。
4. 放大一般物體隨著聲音的細微震動，來藉此還原出聲音。甚至可以做到用耳機在影像的震動來還原出音樂。(音樂辨識軟體可以辨識出是哪條曲子的程度。)

在另一個相關演講：“New video technology that reveals an object's hidden properties”中，還有一個應用：藉由放大一般物體隨著聲音的細微震動，來模擬並建立物體的數位 3D 模型

在另一個相關演講：“Detecting Pulse from Head Motions in Videos”中，也有用放大頭部晃動的方式來判斷出脈搏。

底下是可以深入了解背後的相关演算法與數學原理的文章連結：

1. <http://people.csail.mit.edu/mrub/PhDThesis/>
2. [http://people.csail.mit.edu/mrub/papers/VisualMic\\_SIGGRAPH2014.pdf](http://people.csail.mit.edu/mrub/papers/VisualMic_SIGGRAPH2014.pdf)
3. <http://people.csail.mit.edu/nwadhwa/phase-video/phase-video.pdf>

當中涉及的相關技術：

Phase-based Optical Flow

<http://www.cimat.mx/~omar/vision2/investigacion/localPhase.pdf>

Complex Steerable Pyramids

<http://www.cns.nyu.edu/pub/eero/simoncelli95b.pdf>

- 入門參考書

Alasdair McAndrew, An Introduction to Digital Image Processing with Matlab

透過這本書可以理解到關於影像處理的基本內容，像是數位影像簡介、去噪、分割、壓縮等等，並同時操作 Matlab 來實作例子。

- 其他學習資源

網路上還有許多的相關學習資源，像是

1. <https://sisu.ut.ee/imageprocessing/avaleht> (網站和影片)
2. <https://www.youtube.com/watch?v=CVV0TvNK6pk> (youtube 影片)

- 光流 (optical flow)

參考資料：

1. <http://alex-phd.blogspot.tw/2014/03/optical-flowopencv.html>
2. [http://robots.stanford.edu/cs223b05/notes/CS%20223-B%20T1%20staven\\_s\\_opencv\\_optical\\_flow.pdf](http://robots.stanford.edu/cs223b05/notes/CS%20223-B%20T1%20staven_s_opencv_optical_flow.pdf)
3. [https://en.wikipedia.org/wiki/Optical\\_flow](https://en.wikipedia.org/wiki/Optical_flow)

光流可以用來描述物體在平面影像序列中的運動。透過追蹤同一點在影像序列中的位置變化，來得知物體在影像中的運動速度。

此外，投影效果可以利用不同物體在影像中的速度差異，來判斷出物體的實際遠近差異和實際運動方向的差異。舉例來說，如果有同樣的實際運動速度，則近的物體在影像中的速度會比較快。而如果有同樣的實際運動速率，則平行於影像平面的運動在影像中會有較快的速率。

追蹤同一點的基本想法是，在某個影格中同一點附近的行為，會在下一個影格中被保持。假設黑白影像中，我們想要追蹤影格矩陣  $I_1$  中  $(u_x, u_y)$  所代表的光點。考慮他附近的點所形成的小矩陣：

$$\begin{pmatrix} I_1(u_x - 1, u_y + 1) & I_1(u_x, u_y + 1) & I_1(u_x + 1, u_y + 1) \\ I_1(u_x - 1, u_y) & I_1(u_x, u_y) & I_1(u_x + 1, u_y) \\ I_1(u_x - 1, u_y - 1) & I_1(u_x, u_y - 1) & I_1(u_x + 1, u_y - 1) \end{pmatrix}$$

則在下一個影格  $I_2$  中，光點移動到的  $(u_x + \delta_x, u_y + \delta_y)$  附近的小矩陣：

$$\begin{pmatrix} I_2(u_x + \delta_x - 1, u_y + \delta_y + 1) & I_2(u_x + \delta_x, u_y + \delta_y + 1) & I_2(u_x + \delta_x + 1, u_y + \delta_y + 1) \\ I_2(u_x + \delta_x - 1, u_y + \delta_y) & I_2(u_x + \delta_x, u_y + \delta_y) & I_2(u_x + \delta_x + 1, u_y + \delta_y) \\ I_2(u_x + \delta_x - 1, u_y + \delta_y - 1) & I_2(u_x + \delta_x, u_y + \delta_y - 1) & I_2(u_x + \delta_x + 1, u_y + \delta_y - 1) \end{pmatrix}$$

應該要和之前的小矩陣“很像”。所以，我們將追蹤光點描述成一個最佳化問題：

$$\min \epsilon(\delta_x, \delta_y) := \sum_{x,y \text{ near } u_x, u_y} (I_1(x, y) - I_2(x + \delta_x, y + \delta_y))^2 \text{ for small } (\delta_x, \delta_y)$$

另一種看法是，根據同一光點的描述得到：

$$I(x + \delta_x, y + \delta_y, t + \delta_t) = I(x, y, t)$$

再由泰勒逼近得到

$$I_x V_x + I_y V_y = -I_t$$

此即光流方程式。計算光流的眾多方法中，Lucas-Kanade 光流法，

[https://en.wikipedia.org/wiki/Lucas%E2%80%93Kanade\\_method](https://en.wikipedia.org/wiki/Lucas%E2%80%93Kanade_method)

即是利用光流方程式和鄰域中速度相同的假設，透過解線性方程來求得光流(實際上，由於該線性方程通常是 over-determined，所以會變成  $\min \|Av - b\|^2$  這類最小平方法的最佳化問題。)