【Lecture 6 Scribe】

R93922020 楊惠菁
R93922050 羅婉嫣
B90902099 何佩璇

- **Introduction**

  Image Stitch includes two steps. At first it implies alignment to all the input images. And then it concatenates all the aligned images and blends the overlap region between two images to generate the final result: “panorama”. We call the first step “geometrical registration” because it will change images’ shapes. And the second step only changes images’ color so we call it “photometric registration.”

- **Applications of image stitching**
  - **Video stabilization**

    Given one video where the frames are jiggled, it will do some further processes to stabilize these jiggled frames. And in these processes, we need to do geometrical registration to align each frame and then concatenate these aligned frames together.
  - **Video summarization**

    Given one video, it will delete some redundant frame in the temporal domain and only extract some key frame. And then it stitches these extracted key frames to generate one full-of-scene image. So it summarizes the temporal information in the video to display into one big image in a spatial domain.
  - **Video compression**

    Given one video where each frames have similar backgrounds and only their foreground are different, it can extract the foreground information and make use of the backgrounds’ similarity to do some compression on the background field.
  - **Video matting**

    Given one video, at first it removes the foreground field and leaves one big hole in the frames, and then it estimates the background information to reconstruct the big hole.
  - **Panorama creation**

    Given many images taken in the same large scene such as playground, it uses Image Stitch to stitch all these input images to generate one full-scene image. Through this output image, we can see a whole scene and it will seem that we still stay there.

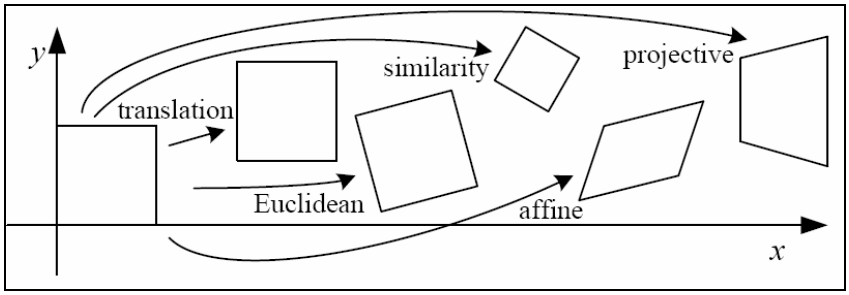
- **Why panorama?**

Humans' full-of-view is about $200\times135^{\circ}$, but the full-of-view of the common cameras is only about $50\times35^{\circ}$ which is much narrower than the former. In use of panorama mosaic, we will have a big picture having full-of-view: $360\times180^{\circ}$. So the panorama can help us to capture the almost the same scene we have truly experienced.

- **2D motion models**

  There are four kinds of 2D motion models:

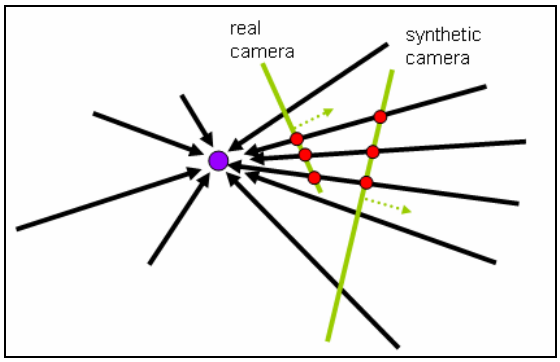
  - translation: $\boldsymbol{x'=x+t}$ *where* $\boldsymbol{x}=(x,y)$
  - rotation: $\boldsymbol{x'=Rx+t}$ *where* $\boldsymbol{x}=(x,y)$
  - similarity: $\boldsymbol{x'=sRx+t}$ *where* $\boldsymbol{x}=(x,y)$
  - affine: $\boldsymbol{x'=Ax+t}$ *where* $\boldsymbol{x}=(x,y)$
  - perspective: $\underline{\boldsymbol{x}}'\cong\boldsymbol{H}\underline{\boldsymbol{x}}$ *where* $\underline{\boldsymbol{x}}=(x,y,1)$



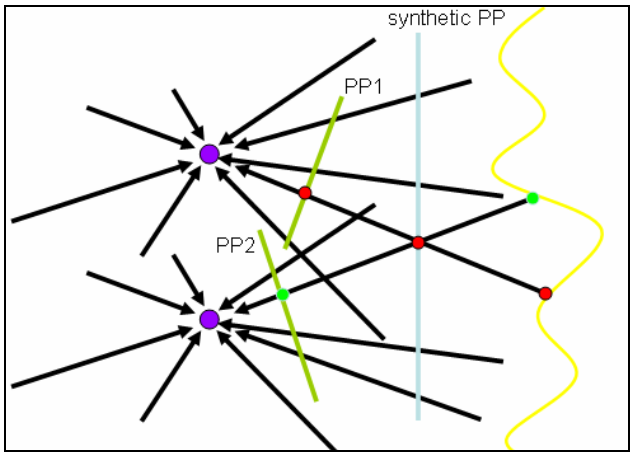
In the follow, there are some comparison among them

Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\begin{bmatrix} I & t \end{bmatrix}_{2\times3}$	2	orientation + ···	
rigid (Euclidean)	$\begin{bmatrix} R & t \end{bmatrix}_{2\times3}$	3	lengths + ···	
similarity	$\begin{bmatrix} sR & t \end{bmatrix}_{2\times3}$	4	angles + ···	
affine	$\begin{bmatrix} A \end{bmatrix}_{2\times3}$	6	parallelism + ···	
projective	$\begin{bmatrix} \tilde{H} \end{bmatrix}_{3\times3}$	8	straight lines	

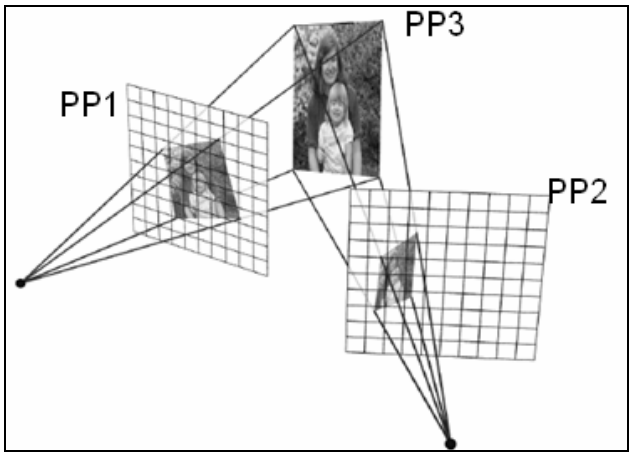
If the camera shares the same optical center, we can use the 2D transformations above to synthesize any camera view like the following:



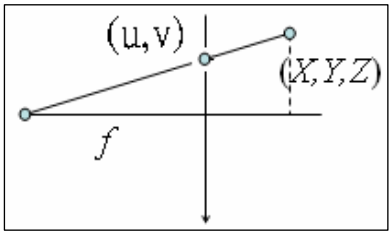
But when the cameras do not share the same optical center, the 2D transformation will cause some problems. In the following figure, we can see that there are two points in the scene, green and red, projecting to the same point in the synthetic projecting plane. So the 2D transformation won't work in the situation.



If we use the 3D transformation, this problem can be solved.



- **3D motion models**



The goal is to project the 3D point *P = (X, Y, Z)* onto the mosaic projection plane to get the 2D coordinate *p= (u, v)*. If we know the extrinsic and intrinsic parameter of camera, we can easily to do such thing by the formula *p=K*R*P* where *R* is carries the extrinsic parameter information and *K* carries the intrinsic parameter information like the following. There are two kind of projection plane's coordinates used in 3D motion model. The first one is rotational and the other is cylindrical. Just as their names, the first method projects input images onto a mosaic plane in the rotational coordinate and the second method project them onto a plane in the cylindrical coordinate.

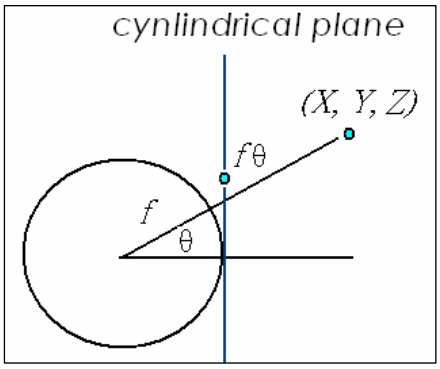
- **Estimate the registration parameters**

  There are two kind of method to estimate the registration parameters. The first one is “direct method”. It makes use of all the pixels’ information to do pixel-to-pixel matching to estimate the parameters. The second method is “feature based method”. This method only uses the feature points in the image to estimate the parameters.

- **Overview of making a cylindrical panorama**

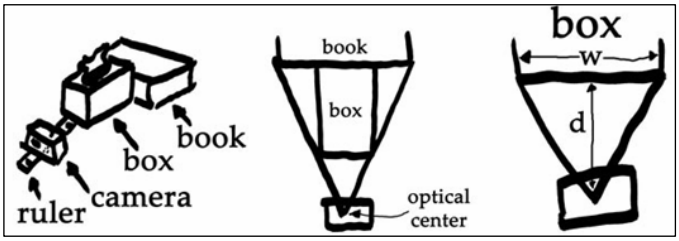
  1. Take pictures on a tripod (or handheld)
  2. Warp to cylindrical coordinate
  3. Compute pairwise alignments using the hierarchical Lucas-Kanade algorithm
  4. Fix up the end-to-end alignment
  5. Blending
  6. Crop the result and import into a viewer

- **Cylindrical Projection**



With the camera’s focal length, we can calculate the projection function showed in the following figure.

But in fact we don’t always know the focal length of the input images. There is a simple method to estimate it. We put a box and book in front of the camera and calculate the real focal length from the relation of *d* and *w*.



- 拍照的時候，如果鏡頭有瑕疵，容易產生 distortion，有兩種 distortion，一種是 pin cushion，另一種是 barrel。鏡片邊緣造成的 distortion 最嚴重。可以利用某個假定的 distortion model 做 restoration。
- 輸入兩張影像，如果不做 pre-warping 直接接合的話，會發現有些地方無論怎麼接都接不起來。如果先做 pre-warping 的話，就可以接合的比較好（在投影片的例子中地板的部分可以完全接在一起。）
- 轉到圓柱座標系的好處是：相機在原本座標系中的旋轉，就是在圓柱座標系

裡面的位移。因此可以用 LucasKanade Algorithm 估計位移向量 uv，有了 uv 就可以將還場接合。

- **LucasKarnade Algorithm**
  讀進兩張 image，先將其中一張做 t 的位移。接著對影像中每個點計算 x 方向的微分，及 y 方向的微分，這樣可以算出 Hessian Matix。利用 Hessian 解 At = b，可算出新的位移量 t。
- **Pyramid LucasKarnade Algorithm**
  利用建 pyramid 的方式，可以估得比較準確的位移量。先從 down sampled 的 image 開始粗估，可以得到位移量 t，然後將這個 t（要先做 scale）當作 initial guess 代到下一層解析度比較高的影像。這樣一層一層的做修正結果會比較好。
- 將還場接合後，接下來要做 blending 的動作，blending 可以讓接合的還場看起來效果更好。比如說，如果拍攝的時候曝光度不一，接起來的時候顏色會很不均勻，這時可以藉由 blending 讓它 smooth。
- blending 的基本作法如下：在兩張影像重疊的部分，計算每個 pixel 新的顏色值，這個值是將兩張影像在該 pixel 的顏色值做內差得來，而越靠近影像中心，該影像貢獻的顏色值的 weighting 越大。
- **還場接合**
  把所有的影像根據算出來的位移量 uv 接合在一起。接著做 blending，最後把參差不齊的部分裁掉。可是這樣的作法有時候會有問題。每次估計出來的 uv 值可能有誤差，而這個誤差在接合的過程中會一直累積下去，因此接到最後一張的時候有可能跟首張合不起來。解決辦法有很多，例如：可以將第一張影像 duplicate，將它跟最後一張接合在一起。然後對第一張之後的每張影像修正 y 方向的位移(y1 – yn)/(n -1)。或是計算 global warping y’ = y + ax。
- 投影片中介紹兩種還場的 viewer。第一種是柱狀還場的 viewer，這個例子是由好幾圈柱狀還場合起來的。
- cube viewer：將還場貼在一個 cube 上，轉動 cube 的時候，上面的 texture 會跟著轉。好處是可以看到頭頂跟地板。
- 除了 direct method 外，這幾年比較 popular 的是 feature-based method，因爲自從 SIFT 發表以後，不但可以有效率找到對應點（幾乎 realtime，且找到的對應點比 LucasKanade 準確）而且可以完整的描述特徵點的資訊。因此，在這個方法中，只使用 feature point 來估計參數，不像 direct method 需要用到所有的點來估計。
- 利用 RANSAC，可以找出一個比較 robust 的 model。因爲資料點通常會比求解方程式多很多，而且很多都是 outlier，所以必須利用 RANSAC，估計出一個比較適合的 model。給定 N 個資料點，假設大部分的點都是由同一個 model 產生出來的，這個 model 的參數爲 Θ，RANSAC 的目的就是要估計出一個比較 robust 的 Θ。

- 以下是 RANSAC 的演算法：重複跑 k 次，每次都隨機取出 n 個點，利用這 n 的點估計出 Θ，接著把其餘 N-n 個點帶入新的 model 裡面，算出在這個 model 下有多少個點是 inlier。重複 k 次以後，可以讓 inlier 個數最多的 Θ，就是求得的解。現在的問題是 k 和 n 要如何決定？
- 如何決定 k：每個點是真正 inlier 的機率是 p，P 是演算法跑 k 次以後成功的機率，那麼 $P = 1 - (1 - p^n)^k$
  經過實驗以後發現如果希望 P = 0.99，如果 n 一樣，p 越大，所需的 k 就越小。如果 p 一樣，n 越大，所需的 k 就越大。通常 p 未知，所以在做 RANSAC 的時候，n 選小一點比較好。
- 投影片的範例中，要找一條直線 fit 這些 sample 點。因爲只要兩個點就可決定一條直線，因此取 n = 2。隨機找兩個點。通過這兩點的直線，就是估計出來的 model。計算其他 N-n 個點到這條線的距離。若距離小於某個 threshold 則判爲 inlier，否則爲 outlier。在這個 model 下，只有三個點是 inlier。再重新取兩個點，重新估計另一個 model，在這個 model 下也是只有三個點是 inlier。重複以上步驟，這次找到的 model 就相當好，有 15 個點是 inlier，因此可以將這個 Θ 當作最後的解。
- 傳統的還場影像都是 1D 的接合：也就是拍攝的時候只有一個方向的旋轉，因此這些影像可以根據旋轉的角度排序，程式再根據這些排序把影像一張一張接起來。但是如果拍攝時有兩個方向的旋轉，就很難做這樣的排序，因此程式很難根據特定的順序接合。
- Recognizing Panorama 之動機：
  若 motion model 爲 1D Rotation，則尚可利用 ordering 來做 image matching。然而，若 motion model 爲 2D Rotation，便無法利用 ordering 來做 image matching。

1D Rotation。

2D Rotation。

- Recognizing Panorama[1]之四個步驟：
  (1)SIFT Feature Matching
  (2)Image Matching

[1] Recognising Panoramas , M. Brown, D. G. Lowe, ICCV 2003

(3)Bundle Adjustment

(4)Multi-band Blending

- Recognizing Panorama 第一個步驟 - SIFT Feature Matching
  假設相機參數可用 3 個旋轉角度 $\theta = [\theta_1 \theta_2 \theta_3]$ 和焦距 f 表之。

  若第 i 張影像之點 $\tilde{u}_i$ 與第 j 張影像之點 $\tilde{u}_j$ 對應到空間中同一個點，

  則 $\tilde{u}_i = H_{ij}\tilde{u}_j$ ，其中 $H_{ij} = K_i R_i R_j^T K_j^{-1}$ ， $K_i = \begin{bmatrix} f_i & 0 & 0 \\ 0 & f_i & 0 \\ 0 & 0 & 1 \end{bmatrix}$

  $$\mathbf{R}_i = e^{[\boldsymbol{\theta}_i]_\times}, \ [\boldsymbol{\theta}_i]_\times = \begin{bmatrix} 0 & -\theta_{i3} & \theta_{i2} \\ \theta_{i3} & 0 & -\theta_{i1} \\ -\theta_{i2} & \theta_{i1} & 0 \end{bmatrix}$$

  使用 SIFT 從影像取出 feature。對每張影像的每個 feature，利用 k-d Tree 做 k Nearest Neighbor Matching，從所有其他影像的所有 feature 找出 k 個最相像的 feature。

- Recognizing Panorama 第二個步驟 - Image Matching
  對一張影像 I_k ，找出與其 feature 相符數最多的前六名影像 I_n (其中 $1 \le n \le 6$)，這六張影像 I_n 可視爲 I_k 之可能 match。接著，利用 RANSAC 找出符合 I_k 與 I_n 之間 homography 的 inlier feature。接著，驗證是否爲好的 image matching，若 $n_i > 5.9 + 0.22 n_f$ （ n_i 代表 feature 中 inlier 個數、 n_f 代表所有 feature 數）[2]，可視爲好的 matching。

- Recognizing Panorama 第三個步驟 - Bundle Adjustment
  建置環場影像時，常會遇到第一張影像和最後一張影像在垂直方向有位移 (Drift) ，利用 Bundle Adjustment，來重新估算相機參數，以最小化重新投影造成的誤差。

- Recognizing Panorama 第四個步驟 - Multi-band Blending
  構成環場影像的各張影像可能因爲曝光等原因而有色差，利用 blending 來取得較平滑的影像。比起使用 linear blending，使用 multi-band blending 細節較清晰，在此篇論文中，作者使用 2 band。將影像分成兩個 band：低頻和高頻。低頻影像用較大的 weighting(使用較大 σ 之 Gaussian function)，高頻影像用較小的 weighting 方式(使用較小 σ 之 Gaussian function) 。兩個 band 各自做完 blending 後，再合起來。

---

[2]在假定 p_1=0.7，p_0=01，p_{min}=0.97 之情況下。

p_1 爲已知 I_k 與 I_n 是正確 image matching 下，feature 是 inlier 之機率

p_0 爲已知 I_k 與 I_n 是錯誤 image matching 下，feature 是 inlier 之機率

p_{min} 爲已知 feature 是 inlier，爲正確 image matching 之可接受最小機率。

- Recognizing Panorama Algorithm review：

Algorithm: Panoramic Recognition

Input: n unordered images

I. Extract SIFT features from all n images

II. Find k nearest-neighbours for each feature using a k-d tree

III. For each image:
  (i) Select m candidate matching images (with the maximum number of feature matches to this image)
  (ii) Find geometrically conisistent feature matches using RANSAC to solve for the homography between pairs of images
  (iii) Verify image matches using probabilistic model

IV. Find connected components of image matches

V. For each connected component:
  (i) Perform bundle adjustment to solve for the rotation $\theta_1, \theta_2, \theta_3$ and focal length f of all cameras
  (ii) Render panorama using multi-band blending

Output: Panoramic image(s)

- Direct method 與 feature-based method 之比較：
  Direct method 之優點：利用所有資訊，所以可以非常精確
  Direct method 之缺點：
  (1)必須人工指定是哪兩張圖片有重疊（相鄰之圖片）
  (2)iterative method 需要指定 initial guess
  (3)假設 brightness constancy
  feature method 之優點：
  (1)無須人工指定是哪兩張圖片有重疊
  (2)無需指定 initial guess
  (3)因使用 feature，比起使用所有資訊的 direct method 而言，理論上應較快。
  (4)因使用 feature，降低 outlier 的影響力，其結果較可靠。
- 爲何早期 direct method 盛行？
  因爲當時沒有好的方法來找 feature，直到 2003 年 SIFT 出現。
- 環場影像(panorama)在數位視覺效果的應用：
  (1)Background plates：用來 pre-visualization、background rendering

(2)Image-based lighting：把一個虛擬物體放置到真實場景的影片中，我們需要對物體作 rendering，使其融入場景。物體之成像與其反射特性、入射光等等有關。所以我們必須取得場景中各角度光的資訊，來對物體作 rendering。傳統的作法，將一顆玻璃球放入場景中來取得所有方向的光的顏色、強度等(搭配 High Dynamic Range 技術) 。在電影特洛依(Troy)中拍攝影像合成環場影像，利用機器手臂在環場影像中旋轉，約費時 20 秒，即可取得場景中的 real lighting 資訊，將此資訊輸入 rendering 軟體，即可對虛擬 3D 物體作 rendering。