VFX Scribe

93944012
93944017
93922031

Image-based modeling

依據使用 images 的多寡可以大致分成下面兩種方式

一，Models from multiple (sparse) images

- Façade

二，Models from single images

- Tour into pictures
- Single view metrology
- Other approaches

Façade

- Use a sparse set of images
- Calibrated camera (intrinsic only)
- Designed specifically for modeling architecture
- Use a set of blocks to approximate architecture

- 3 steps: geometry reconstruction, texture mapping and model refinement
- 3 steps
    - Geometry reconstruction (photogrammetric modeling method)

- Texture mapping (view-dependent texrture mpping)
- Model refinement (model-based stereo)

Idea

It’s a new approach for modeling and rendering existing architectural scenes from a sparse set of still photographs. Our modeling approach, which combines both geometry-based and imagebased techniques, has two components.

- photogrammetricmodeling method
- *model-based* stereo algorithm,

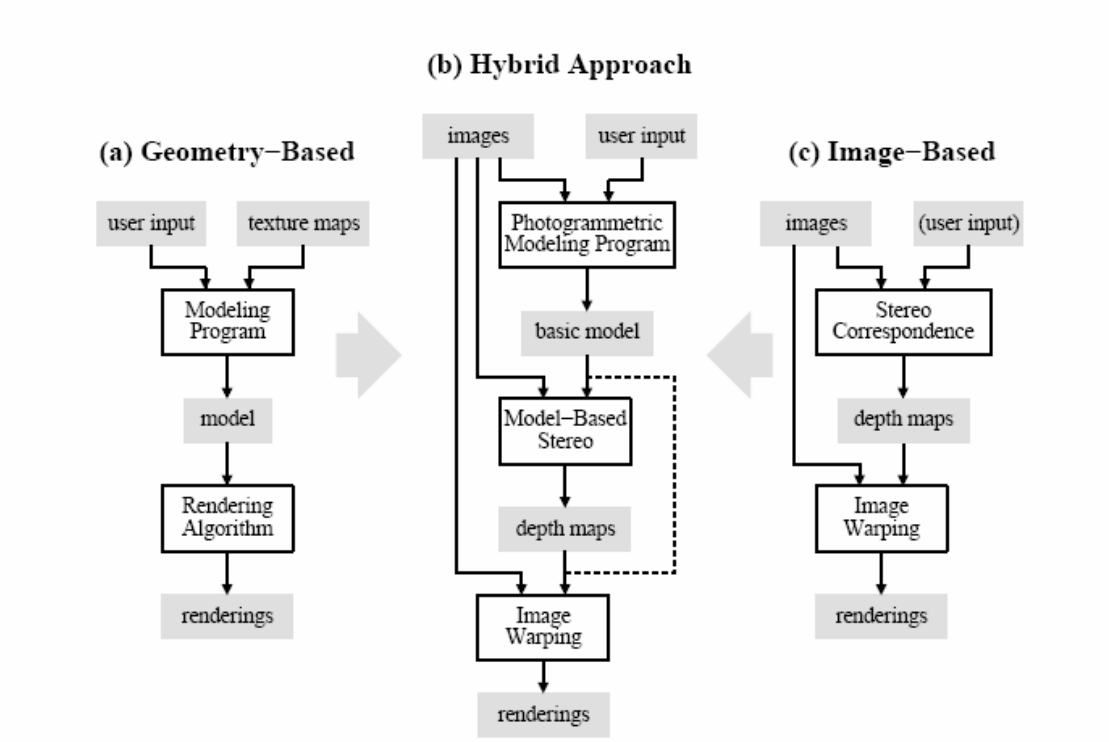
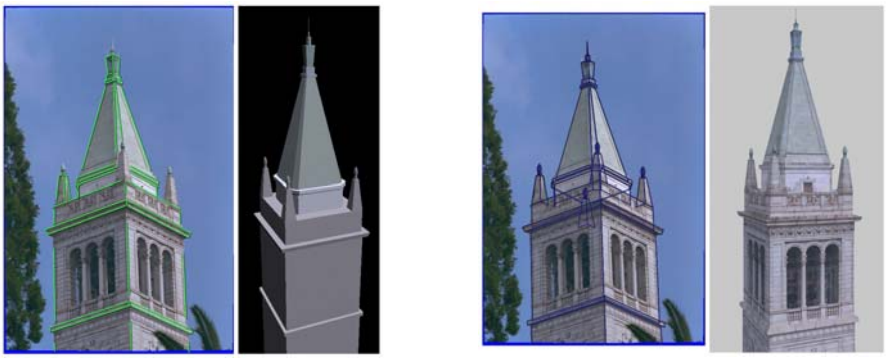


Figure 1: *Schematic of how our hybrid approach combines geometry-based and image-based approaches to modeling and rendering architecture from photographs.*

簡單的說就是從照片裡面所得到的資訊來推得建築物的大致結構，並由此結構根

據攝影機的視角來將圖片貼上，而得到類似真實立體的效果。



接著我們簡單的說明主要的三個步驟：

Step1

- **Geometry reconstruction (photogrammetric modeling method)**

使用者利用一些基本的”積木”先堆出大約的 model，然後將 model 與照片上面的線段一一對應，由於我們所建出的 model 只是大約形狀並不正確，所以我們必須根據數張圖來對 model 做一些調整。

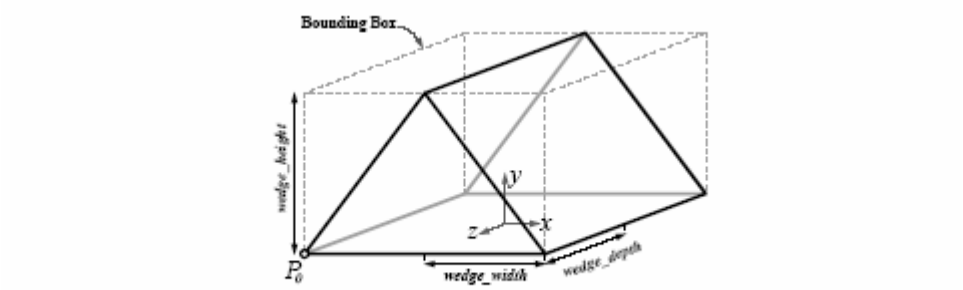


Figure 3: *A wedge block with its parameters and bounding box.*

上面是一塊”積木”的例子，我們可以發現我們需要解決相當多的變數，爲了減少變數的個數，我們利用了一個事實，那就是***很多”積木”之間彼此存在的特定的關係***，因此我們可以得到許多 constraint 並且減少了變數的個數。如下

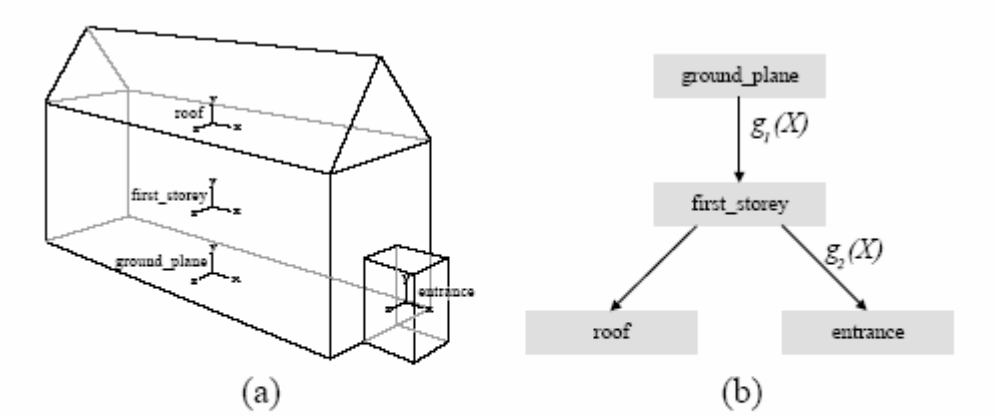


Figure 4: **(a)** *A geometric model of a simple building.* **(b)** *The model's hierarchical representation. The nodes in the tree represent parametric primitives (called* blocks*) while the links contain the spatial relationships between the blocks.*

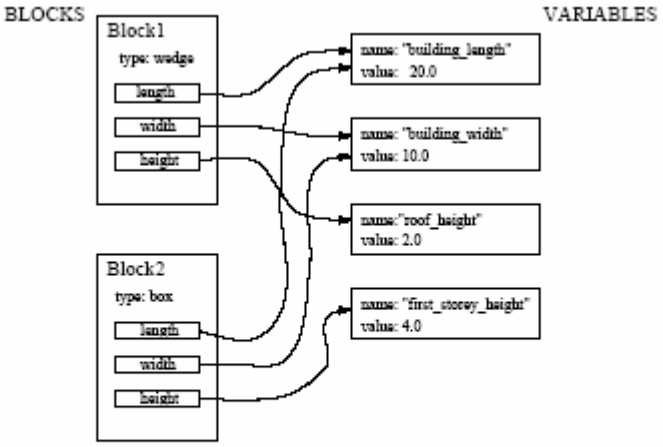


Figure 5: *Representation of block parameters as symbol references. A single variable can be referenced by the model in multiple places, allowing constraints of symmetry to be embedded in the model.*

至於如何藉由照片的資訊來重建，間單說明如下：

The straight line can be defined by a pair of vectors <v , d> where v represents the direction of the line and d represents a point on the line

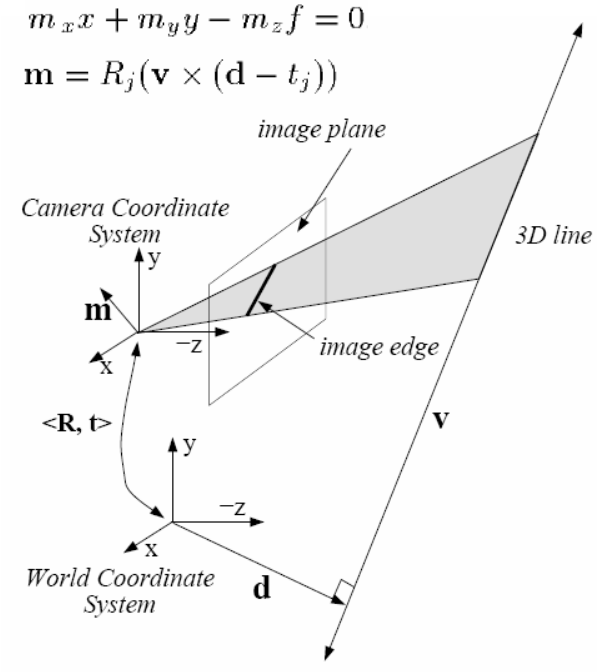
The position of the camera with respect to world coordinates is given in terms of a rotation matrix Rj and a translation vector t

m, cameras's normal

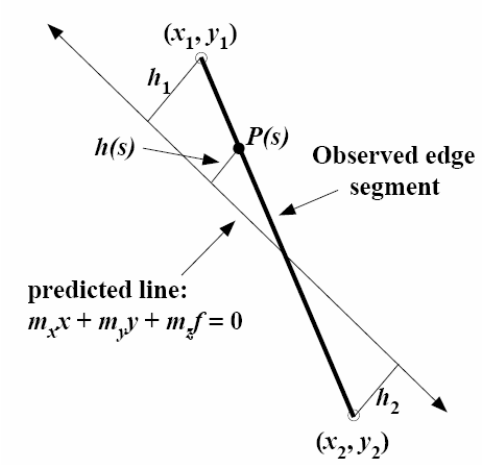
由下圖我們可以知真實 3D line 與照片上面的線段的相對應關係

$$m_x x + m_y y - m_z f = 0$$

$$\mathbf{m} = R_j(\mathbf{v} \times (\mathbf{d} - t_j))$$



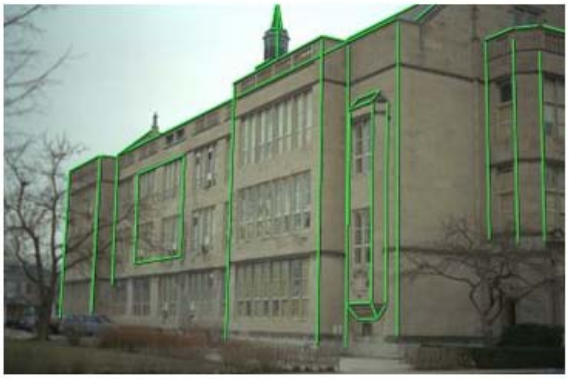
知道相對關係之後，經由預測的線段和實際觀察到的線段我們可以藉由 minimal 他們之間的誤差來得出最佳解。

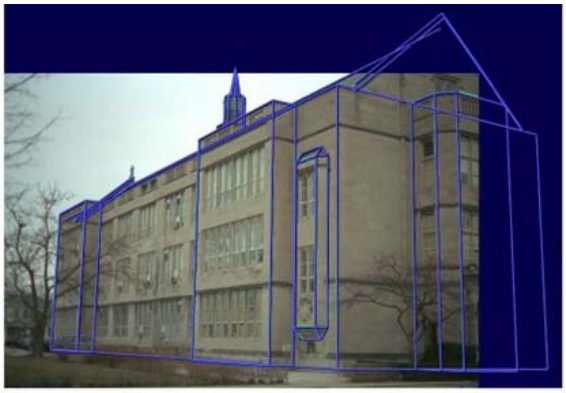


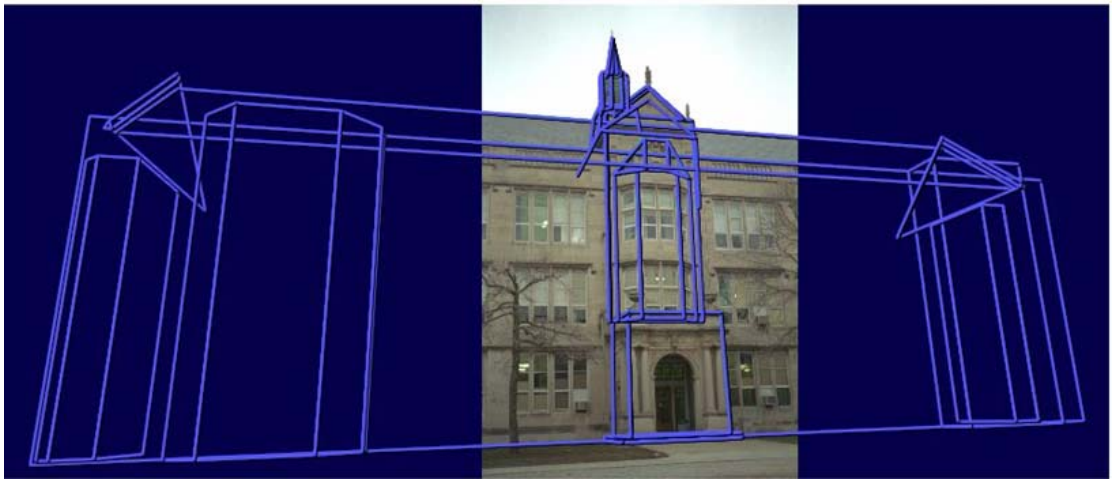
$$Err_i = \int_0^l h^2(s)ds$$

$$\text{minimize } \mathcal{O} = \sum Err_i$$

Result：





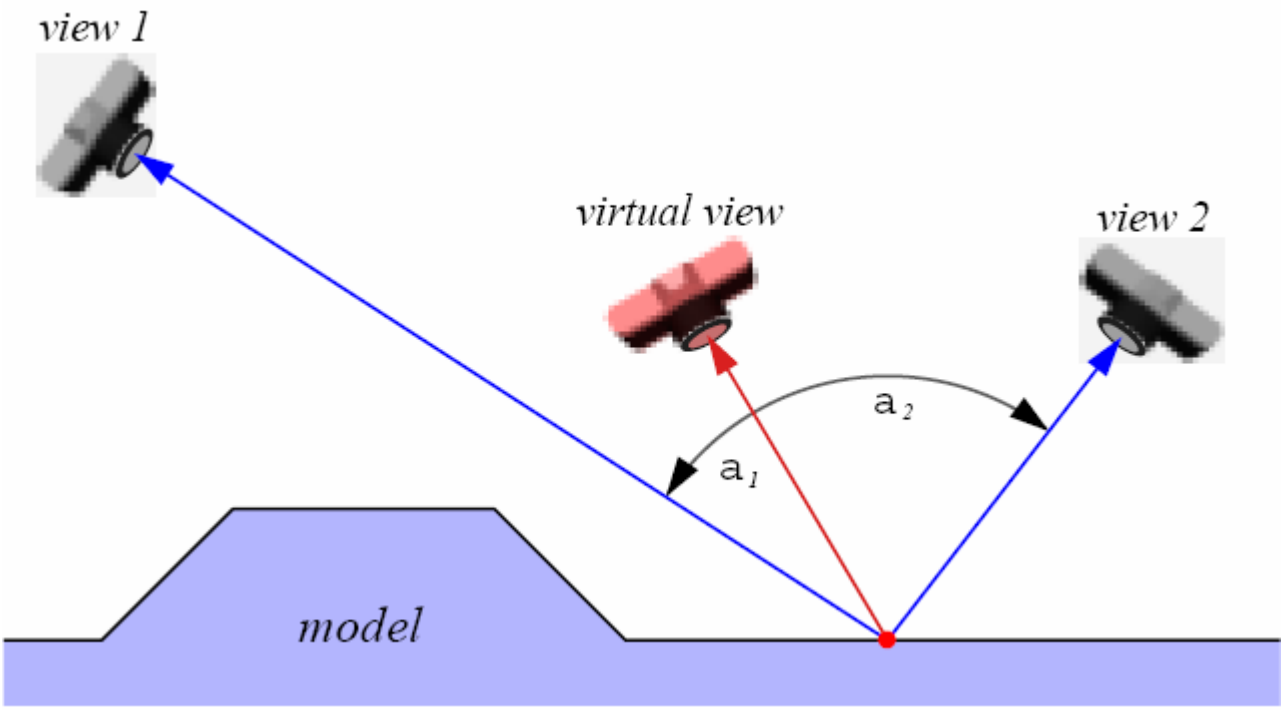


Step2

- **Texture mapping (view-dependent texrture mpping)**

一個 texture mapping 的例子：

view-dependent texture mpping　　顧名思義就是根據視角的不同來以不同的 texture 來 mapping，值得注意的是他是利用角度來做內差。

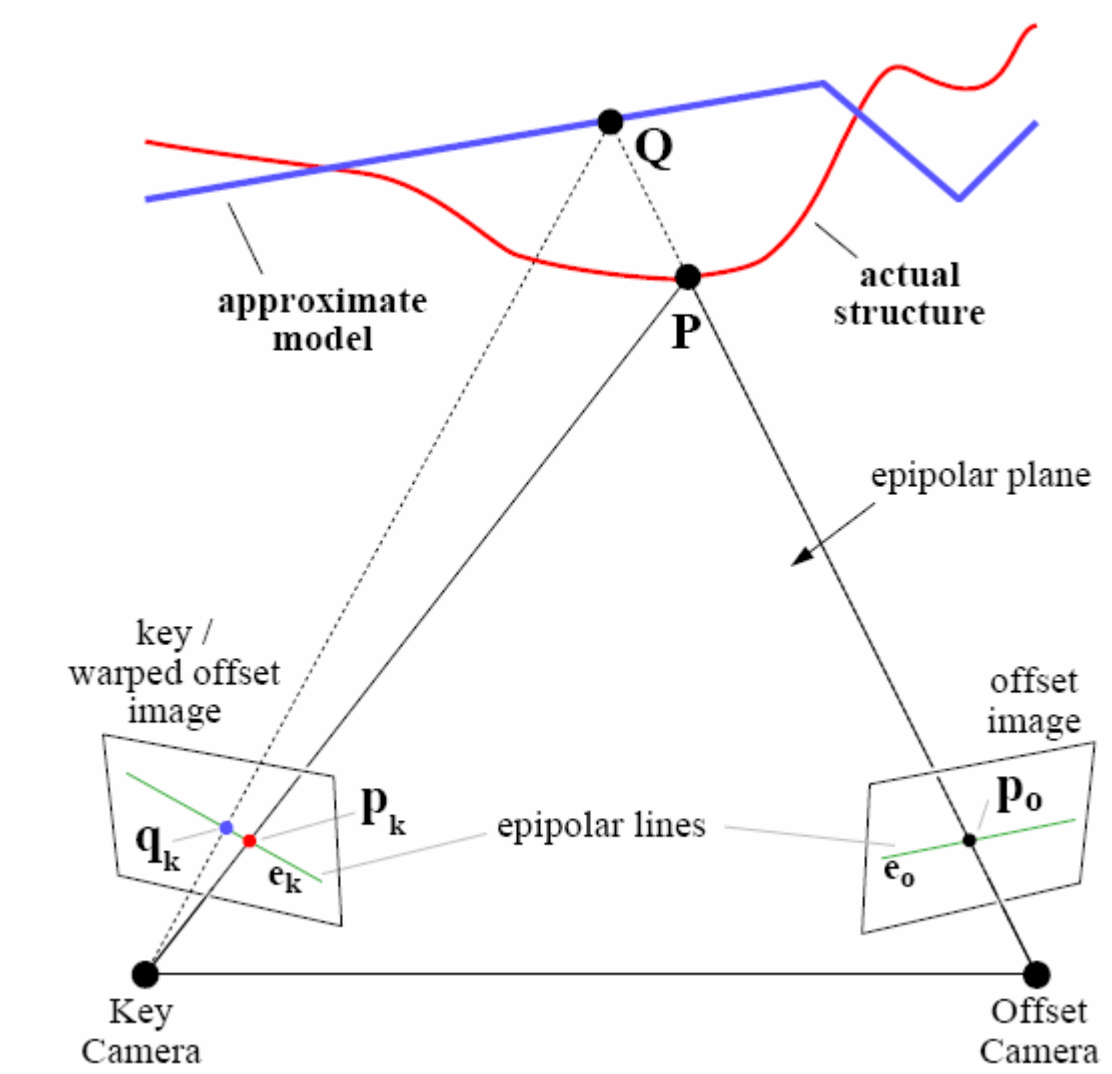


下圖我們可以發現若是不依據視角不同而做調整會有何結果，我們可以發現 non view-dependent texture mapping (左圖)　和 view-dependent texture mapping (右圖) 的效果是差很多的。

Step3

- **Model refinement (model-based stereo)**

利用 **Epipolar geometry** 來求得深度的不同，建出更細緻更具 3D 立體感覺的的 model。



重建的順序以及可能發生錯誤的部分：

Stereo reconstruction pipeline

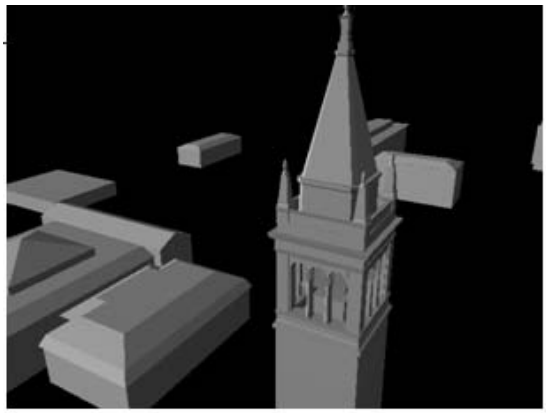
- Calibrate cameras
- Rectify images
- Compute disparity

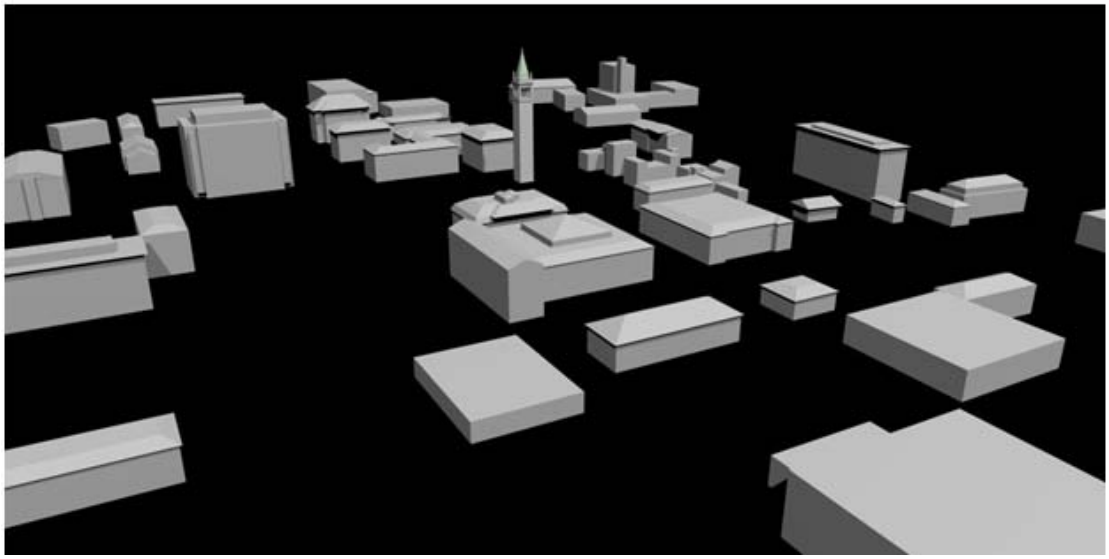
- Estimate depth

What will cause errors?

- Camera calibration errors
- Poor image resolution
- Occlusions
- Violations of brightness constancy (specular reflections)
- Large motions
- Low-contrast image regions

Final Result:

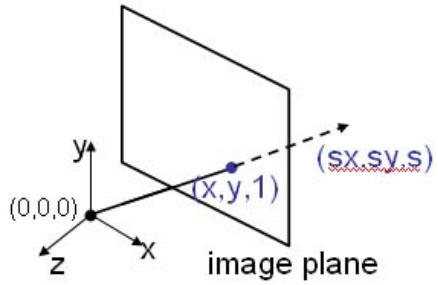




Models from single images

The projective space :

- **A point in a image is a ray in projective space**.

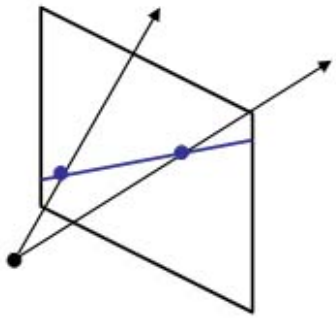


在(sx, sy, s)上, s 代入任何實數都會投影到 normalized image plane 上的 (x, y, 1). 因此在 image 上的一點, 對應到 projective space 是一條 ray.

(x, y, 1) ~= (sx, sy, s).

(normalized image plane 是如同上所示, z = 1 的 image plane)

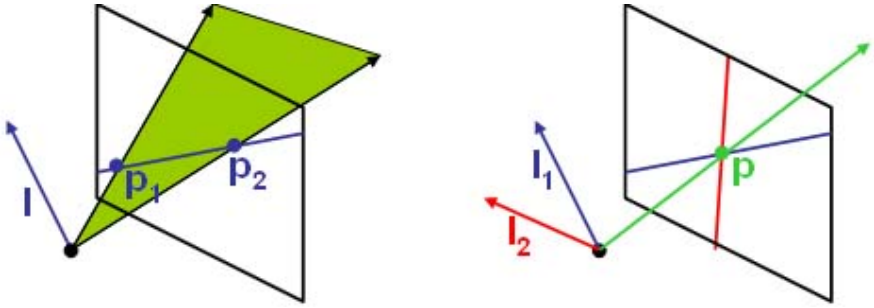
- **A line in a image is a plane in projective space**.



可以想見, 一個在 image plane 上的線是很多點的集合, 一個點在 projective space 上是對應到一條 ray, 因此在 image plane 上的線

在 projective space 將形成一個平面. 一個 3D 的平面可以用他的 normal 來表示. Ex. ax + by + cz = 0 這個平面可用(a, b, c)這個 vector 來表示.

- **Point and line duality**

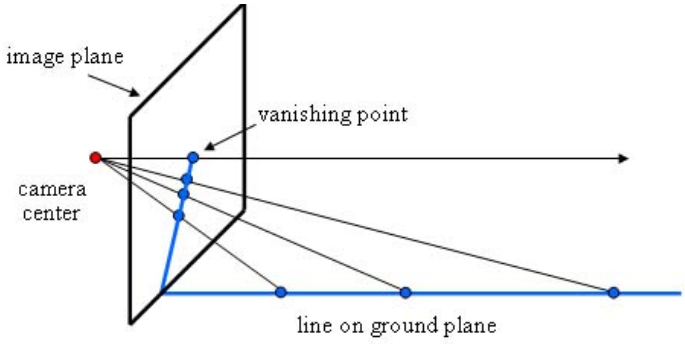


首先我們知道 image plane 上面有兩點 p1, p2, 那他們所決定 image plane 上面的線, 在 projective space 中是一個平面, 代表的 normal vector 就是 I. 由 vector space 上面的運算可知 I = p1 X p2.

又兩個 image image I1, I2 (如右圖)會相交成一條 ray, 在 image plane 上面就是一個點. 由上圖知道, vector I1, I2, p 會成以下的關係 :

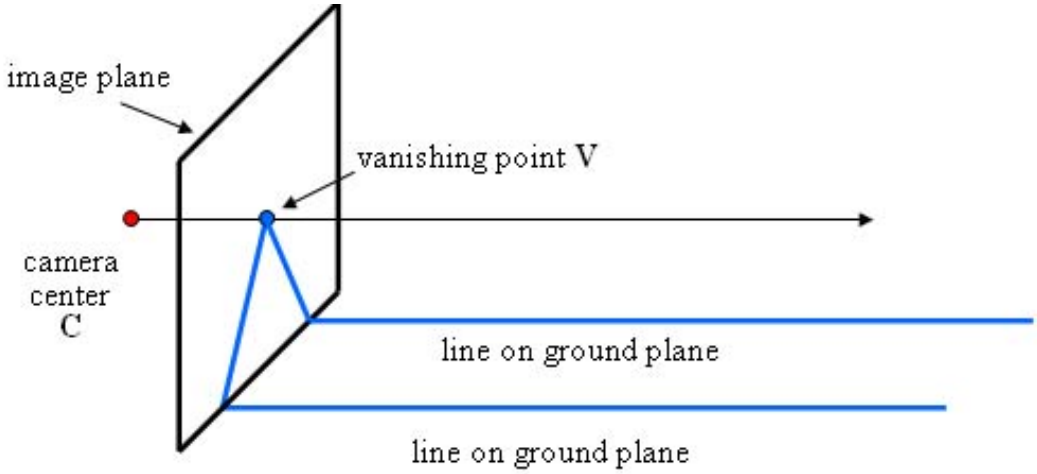
p = I1 X I2.
(X : cross product)

Vanishing points :



由上圖可知, Vanish point 是一條在 ground plane 上的線，在無線遠處投影到 image plane 上面的點.

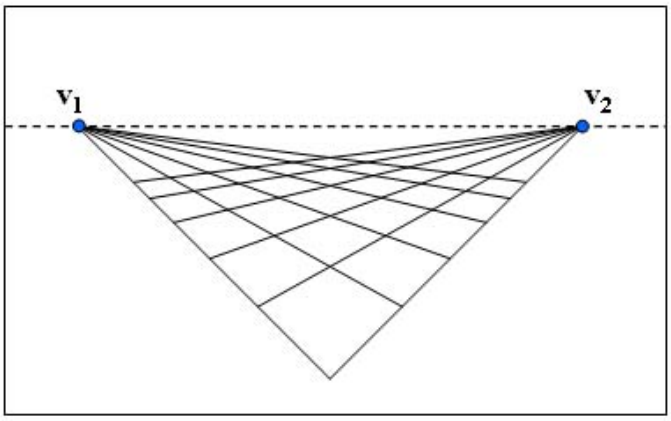
以下是看兩條平行線的例子



我們可發現以下特性 ：

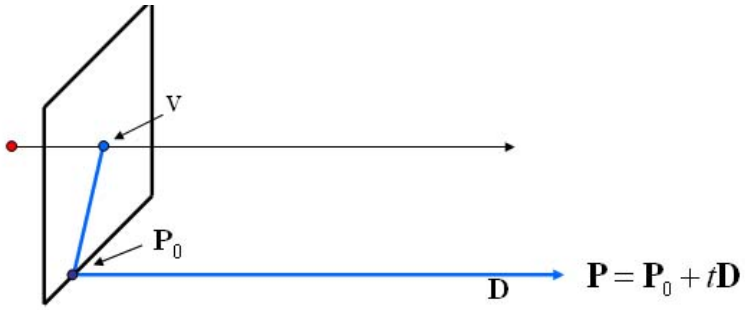
- 任何兩條平行線，他們形成的 vanishing point 是同一個點.
- Camera center (C) 跟 vanishing point 的連線，平行於那兩條平行線.
- 任何一張圖可能會含有兩個以上的消失點. 如果有兩個消失點，就代表有兩群不同方向的平行線.

如果所有任何可能方向的平行線，所形成的一堆 vanish points 會變得怎樣呢?



- 任何一組在同一平面上的平行線會產生一個 vanishing point.
- 所有的 vanishing points 會形成一條線，稱爲 vanishing line.
- 在不同平面上產生的 vanishing line 都不同.

以上所說明的特性可以用數學來證明 ：

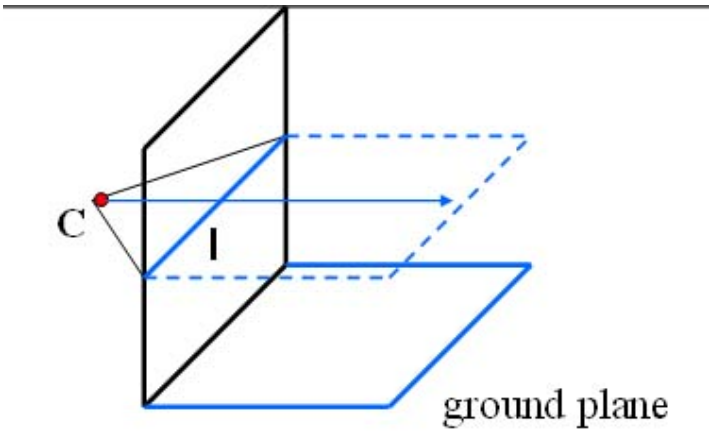


$$\mathbf{P}_t = \begin{bmatrix} P_X + tD_X \\ P_Y + tD_Y \\ P_Z + tD_Z \\ 1 \end{bmatrix} \cong \begin{bmatrix} P_X/t + D_X \\ P_Y/t + D_Y \\ P_Z/t + D_Z \\ 1/t \end{bmatrix} \qquad t \to \infty \qquad \mathbf{P}_\infty \cong \begin{bmatrix} D_X \\ D_Y \\ D_Z \\ 0 \end{bmatrix}$$

P 是在 ground plane 上的一點，加一個 vector D 可以形成一條線.

線上的每個點都可以表示成 Pt. 當 t =∞ (即在無限遠的時候)，那個無限遠的點跟 camera center 的連線會交於 vanishing point.

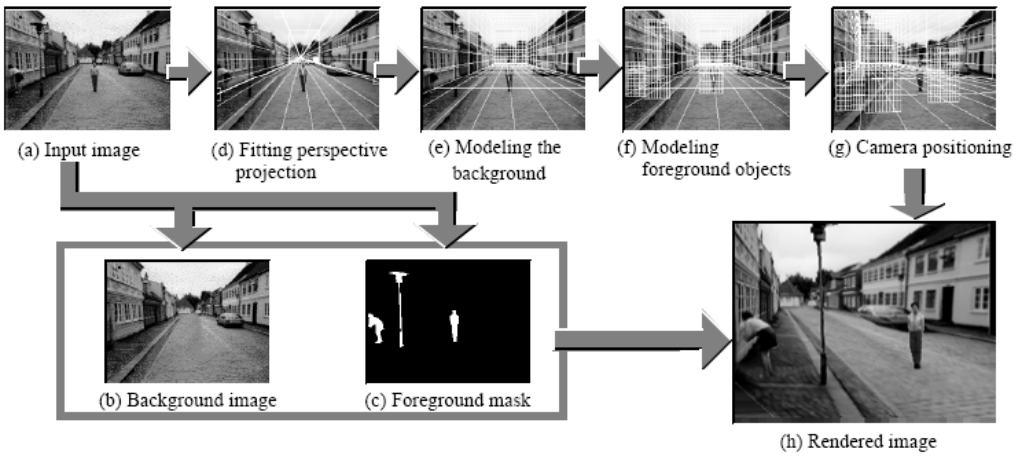
以下討論 vanishing line 的特性..



- I 是 image plane 跟通過 C 並和 ground plane 平行的 plane 之交線.
- I 可以由 two sets of parallel lines on ground plane. 因爲 two sets of parallel lines 可以在 image plane 形成兩個 vanishing points. 就可以連出 I 這條線.
- 任何的點跟 C 一樣高, 將會投在 I 上. 比 C 高則投在 I 上面. (高度是相對於 ground plane 而言)
- I 提供了一個可以比較 image 中物體高度的方式.

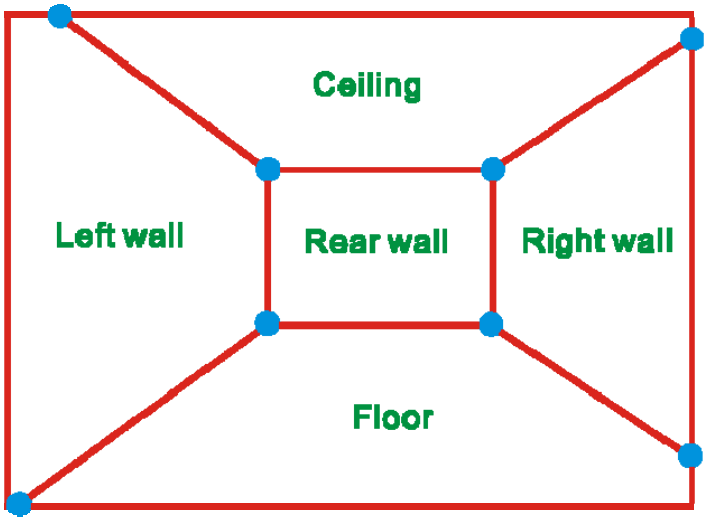
Tour into pictures

這篇 paper 提供一種方式, 使用者只要輸入一張 image, 就可以產生一個 3D 環境可以讓 User 在裡面切換視角瀏覽.



上圖就是演算法的大致流程.

- (a) 輸入一張 image.
- (b)(c) 用 foreground mask 標出前景的位置，並把前景先去掉產生出完整的 background.
- (d) 指定 vanishing point.
- (e) 拉出 rear wall，和相對應的 box volumn，整個架構如下圖所示



1. 兩兩相交的 plane 互相垂直.
2. rear wall 跟 image plane 平行.

3. Floor 是 y = 0 這個平面.
4. camera 的 view up vecto 是 Floor 的 normal vector.

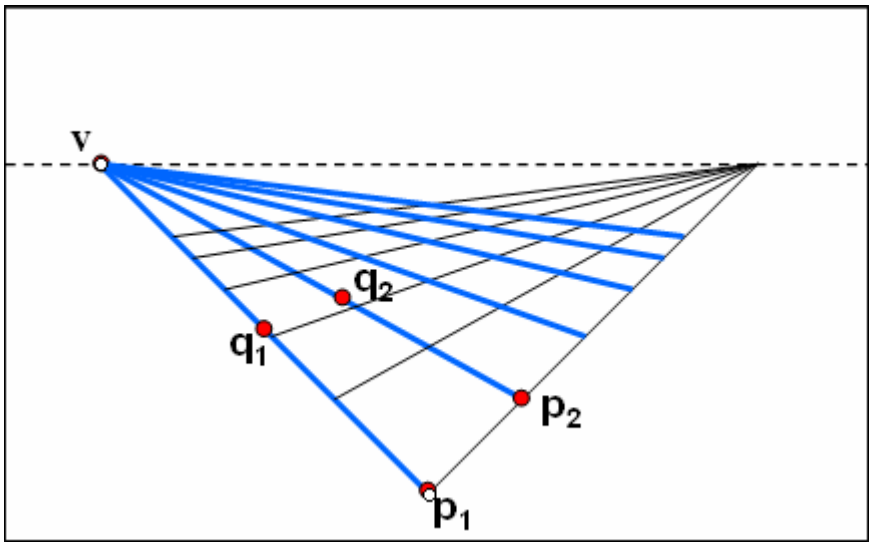
- (f) model 出前景的物件. (ex. 用個 bill board, 把前景貼上去)
- (g) 調整 camera position
- (h) render 出對應的 image.

此外這 paper 做了一個假設, 就是假設全部場景可以用一個 vanishing point 來表示.

Criminisi *et al.* ICCV 1999

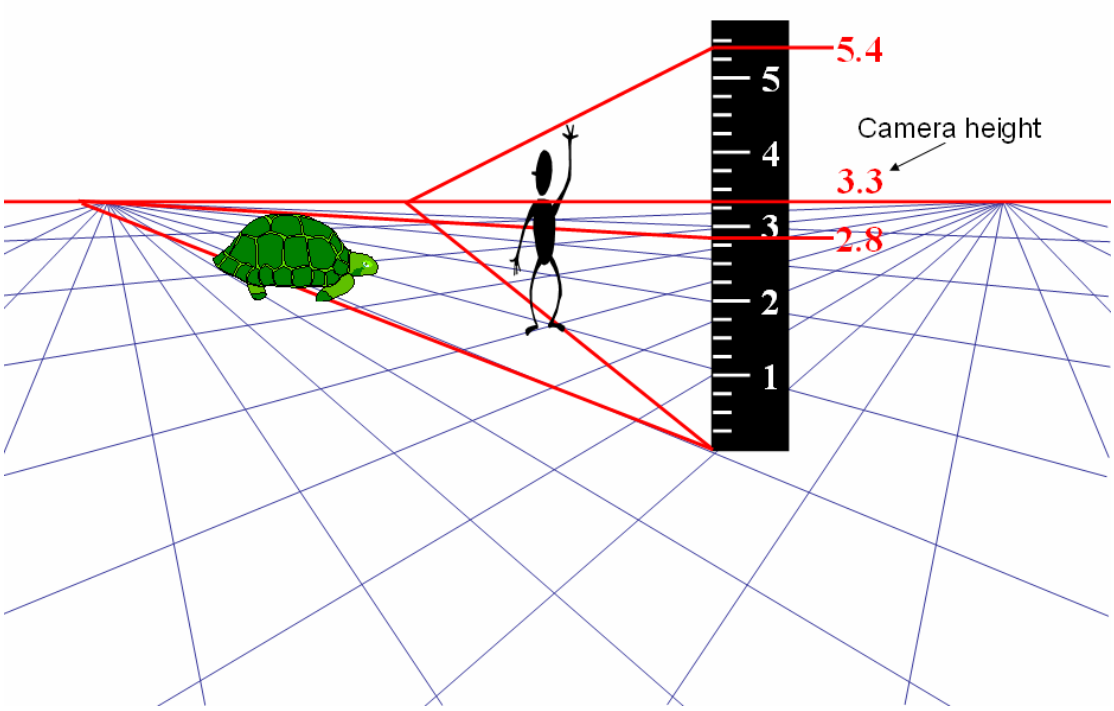
這篇 paper 主要是在研究如何在單一一張 image 中做度量，而度量的結果則可應用在 image modeling 上面。

Vanishing Point & Vanishing Line



如圖，空間中所有平行的直線，投影在 image 上面，必會相交於一點(圖中的 v 點)，此點即爲 Vanishing Point。由同一個平面上的兩組平行線會得到兩個 Vanishing Points，此兩個點形成的直線即爲 Vanishing Line。此平面上所有的平行線得到的 Vanishing Points 都會落在 Vanishing Line 上面。

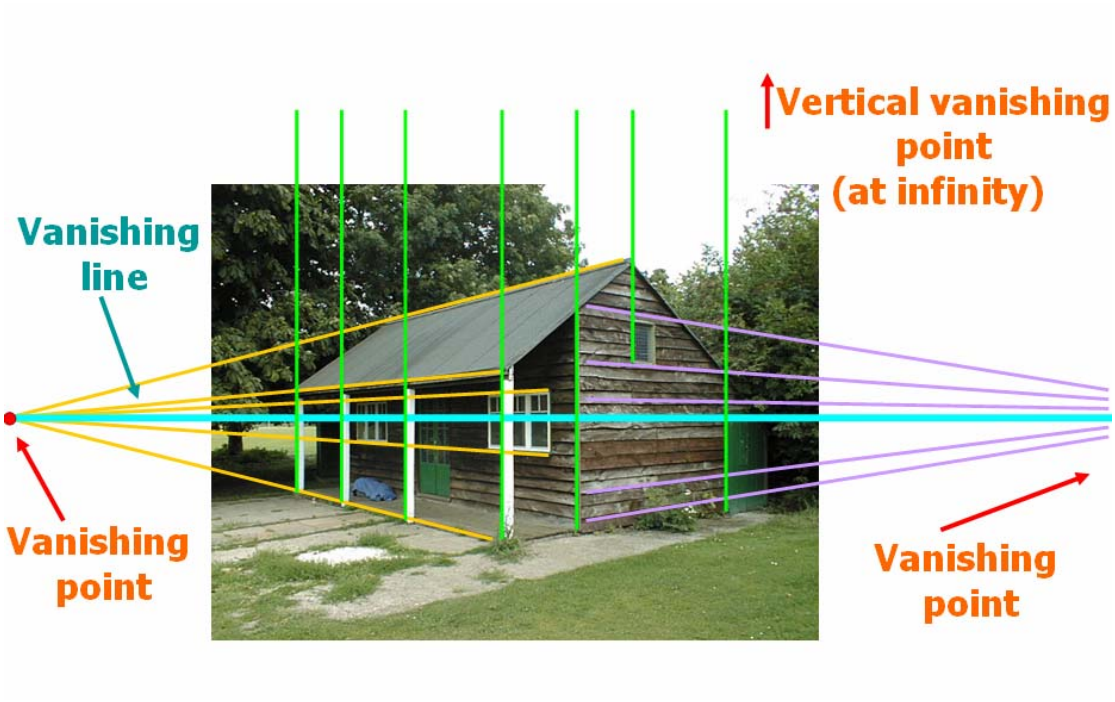
Measuring Height



此篇 paper 假設人的腳站在地平面上。另外 camera 離地面的高度會與 image 中落在 vanishing line 上的東西的高度一致，所以若已知 vanishing line，則可測量物體的高度。

如上圖，裡面的人假設腳站在地平面上，所以可由腳與尺底的連線和 vanishing line 的交點，找出對應到的 vanishing point，再由 vanishing point 與頭頂的連線反查對應到的高度，此高度即爲人的高度。

Modeling



如上圖，可利用圖片上平行的線段，來找出 vanishing line，借此求出 image 中部分線段的長度(如下圖)，經由這些長度資訊來 modeling。

Automatic Popup

這篇 paper 可由單張 image 產生 3D 影像，而且不需要任何使用者的介入。這裡假設所有 image 由 ground、vertical、sky 三個部分所構成，其中 vertical 是跟地表垂直的物體。

先將 image 做 segmentation(圖 b)，再來由事先就有的 training data 中，以及原本 image 的資訊，來將每個 segment 歸類在上面三種部分之中(圖 c)。因爲假設物體與地面垂直，所以可由 vertical 部分與 ground 部分的交線長出垂直線來 fit model。

雖然這樣做與實際的 model 並不完全契合，不過很多並不會造成視覺上的影響。另外，因爲這篇 paper 是全自動，所以雖然成功率不高，可是可以經由 input 大量資料，來取得成功的結果。

(a) input image

(b) superpixels

(d) labeling

(e) novel view