XỬ LÝ ẢNH SỐ VÀ VIDEO SỐ

PHOTOGRAMMETRY

NHÓM RETINA

1. Bùi Đoàn Hữu Nhân
2. Nguyễn Tấn Thìn
3. Phạm Nguyên Minh Thy
4. Đinh Hữu Phúc Trung
5. Nguyễn Phạm Phúc Việt

NỘI DUNG

1. Introduction

- **Phép quang trắc** (*Photogrammetry*) là khoa học nghiên cứu việc tính toán, đo đạc các thuộc tính hình học như kích thước, hình dáng, vị trí của các vật thể bằng việc đánh giá các ảnh chụp từ các thiết bị điện tử như điện thoại, máy ảnh kĩ thuật số, drone, ...

- Photogrammetry là sự kết hợp giữa photo – ánh sáng và metry – trong metric liên quan tới đo đạc.

2. Motivation

- Về mặt khoa học, photogrammetry là phương pháp hiệu quả để giải quyết bài toán tái tạo cấu trúc 3D từ các ảnh 2D, đây là một trong những vấn đề cơ bản của thị giác máy tính.

- Bên cạnh đó, kết quả của photogrammetry có thể được sử dụng cho các tác vụ khác của thị giác máy tính như xác định vật thể, tái tạo bề mặt chi tiết cho vật thể, xác định ngữ cảnh cho thiết bị tự hành, công nghệ thực tế ảo, thực tế tăng cường, …

2. Motivation

Về mặt ứng dụng, photogrammetry được dùng trong nhiều lĩnh vực khác nhau, chẳng hạn:

- Trong địa chất, photogrammetry dùng để vẽ bản đồ địa hình của các địa điểm rộng lớn từ ảnh chụp trên cao sử dụng drone.
- Trong kiến trúc hay du lịch, photogrammetry dùng tái tạo, mô phỏng các công trình như tòa nhà, khung cảnh thành phố, di tích, … với bối cảnh thực tế
- Ngoài ra, Photogrammetry được dùng để chuyển những vật thể lớn thành mô hình 3D phục vụ nhiều mục đích khác nhau

2. Motivation

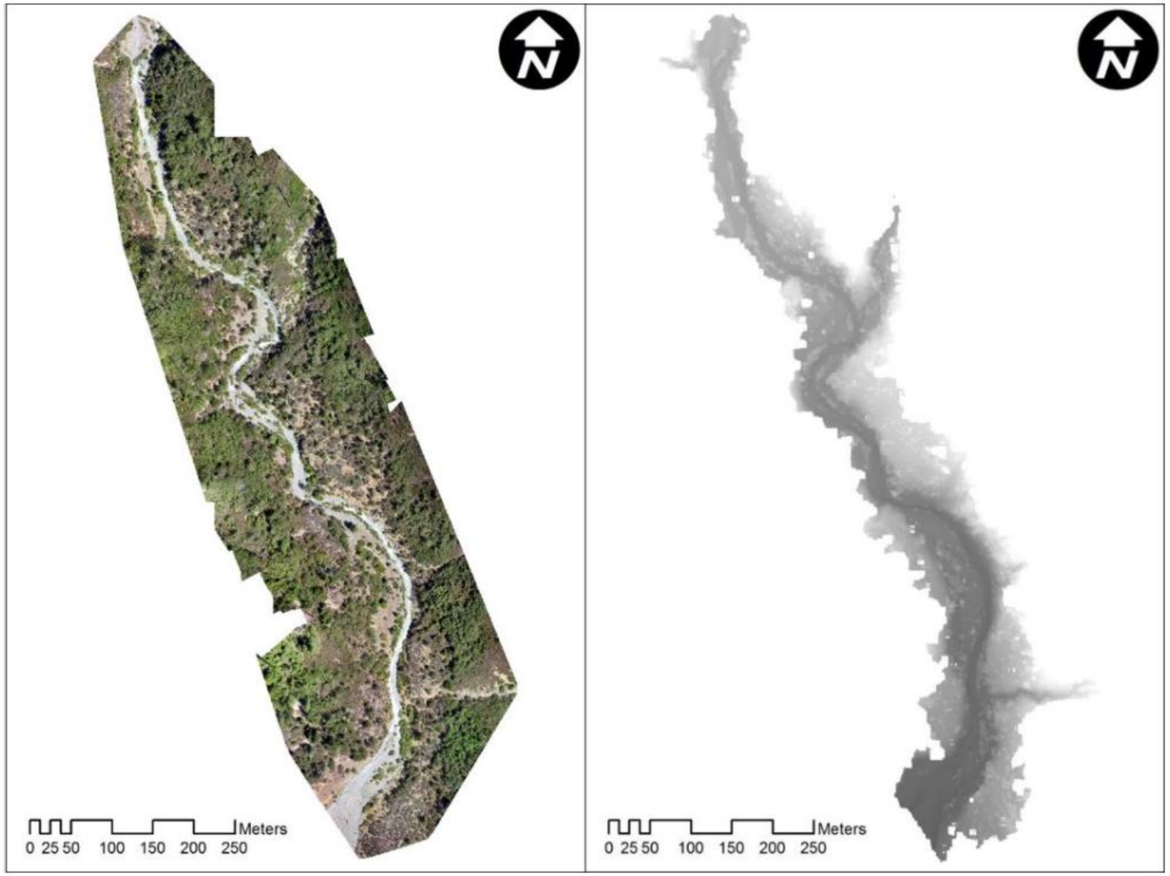
- Mô phỏng bối cảnh, kiến trúc

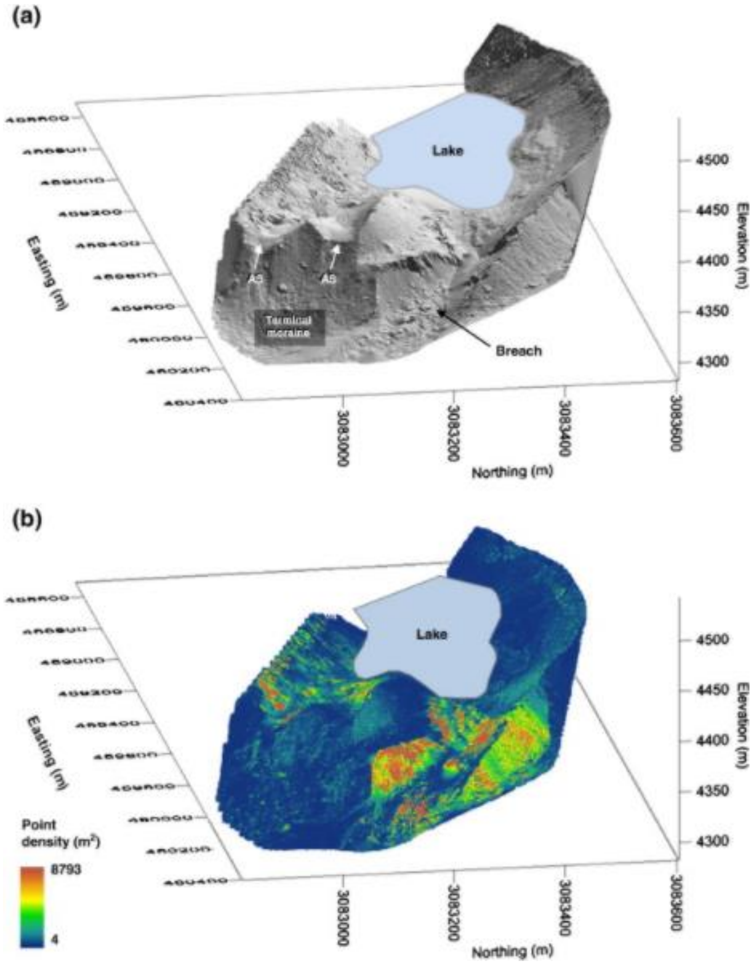
2. Motivation

- Mô phỏng bối cảnh, kiến trúc

2. Motivation

➢ Xây dựng bản đồ địa hình



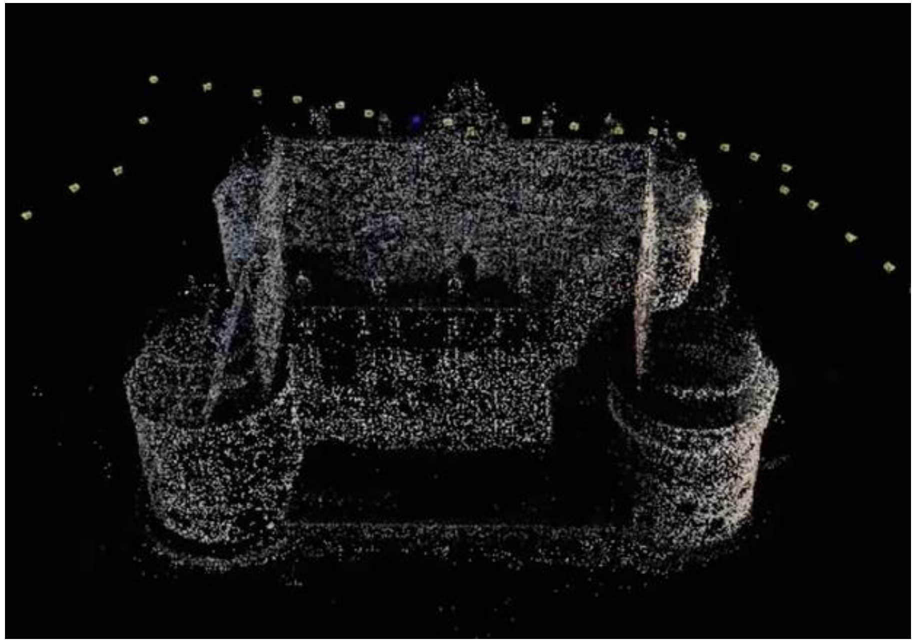


3. Input, output

Bài toán tái tạo cấu trúc 3D từ các ảnh chụp

- **Input:** Chuỗi các ảnh RGB chụp object/scene ở nhiều góc nhìn, vị trí khác nhau. Các ảnh phải có độ chồng lấp 60-70%. Các ảnh chụp từ camera đã biết thông số nội tại.
- **Output:** Đám mây điểm biểu diễn cấu trúc 3D của object/scene và các tham số (vị trí, hướng) của camera chụp vật thể.

3. Input, output



[Source: The Struture from Motion Pipeline]

4. Framework

Thuật toán Structure from Motion (SfM) là phương pháp cổ điển dùng cho mục đích tái tạo vật thể 3D. Các thuật toán SfM hiện tại được chia thành 3 loại **Incremental SfM**, **Global SfM** và **Hybrid SfM**. Cả 3 thuật toán này đều có chung các bước:

- **Bước 1:** Epipolar Geometry

  Trích xuất đặc trưng tương ứng (feature correspondences) và tính toán chuyển động tương đối giữa các cặp 2 hoặc nhiều camera dựa trên các điểm đặc trưng tương ứng giữa các ảnh.

- **Bước 2:** Camera Registration

  Từ quan hệ chuyển động tương đối giữa các cặp ảnh ở bước trên, ta quy tất cả camera về một hệ quy chiếu chung.

- **Bước 3:** Bundle Adjustment

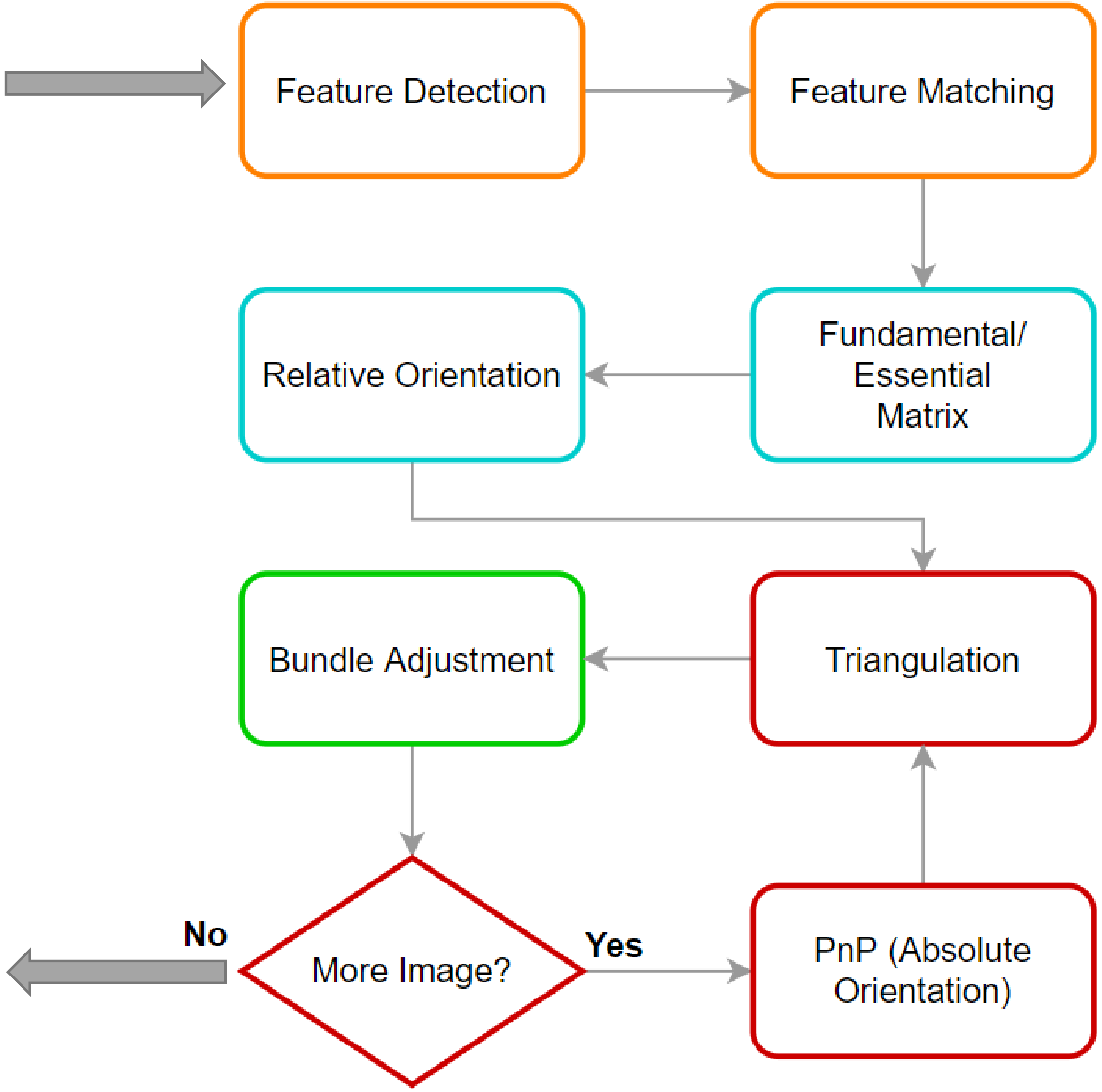
  Thực hiện quá trình tối ưu phi tuyến để tinh chỉnh tọa độ các điểm 3D và camera để tối thiểu hóa hàm sai số phép chiếu (reprojection error).

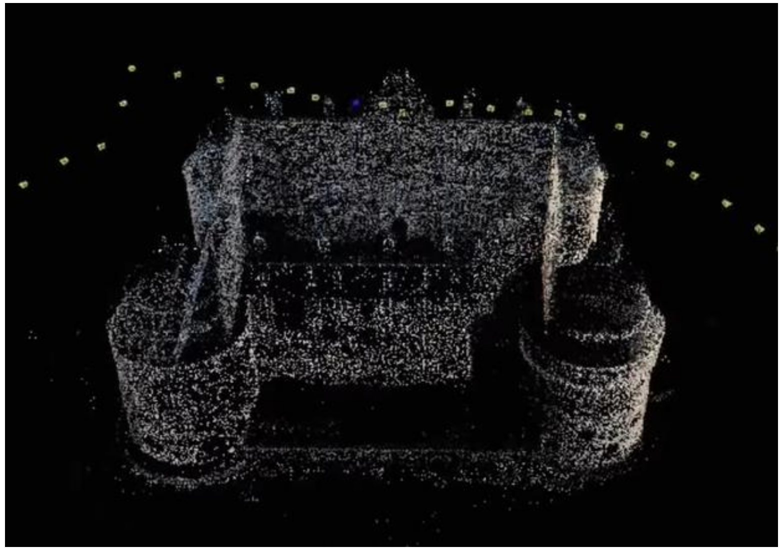
5. Related works

- Phần này trình bày phương pháp **Incremental Structure from Motion (SfM)** chung
- Sau cùng là bảng tổng hợp 2 bài báo cái tiến phương pháp **Structure from Motion (SfM)** thông thường là
  - Structure-From-Motion Revisited (CVPR-2016)
  - Fast incremental structure from motion based on parallel bundle (2020)

SfM - Pipeline

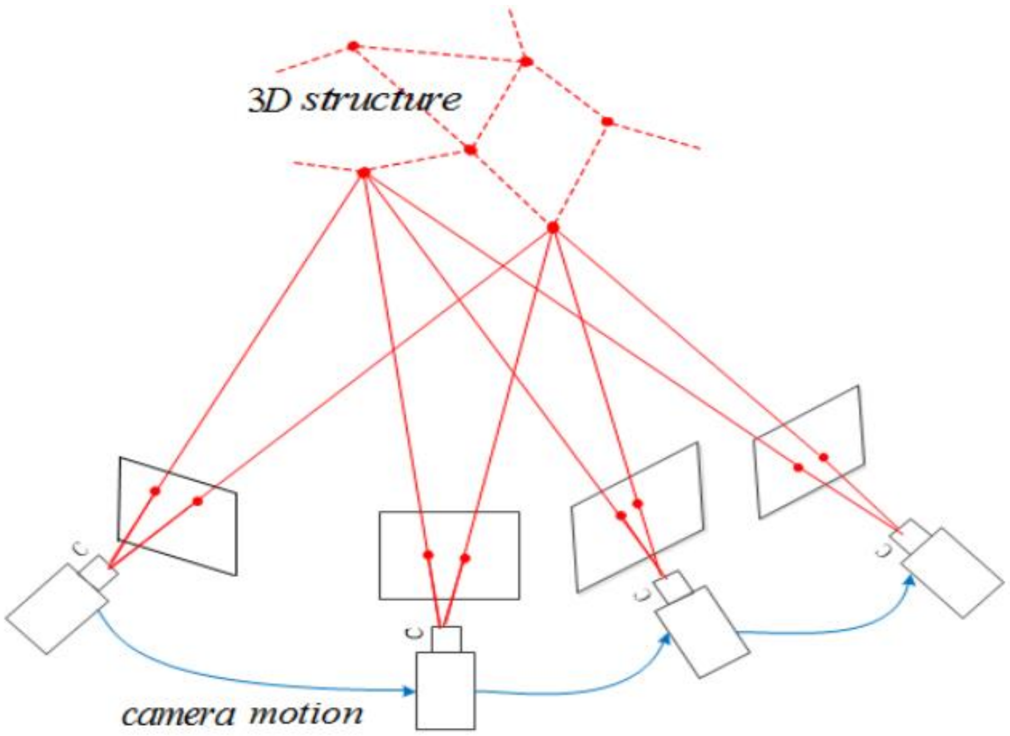
Incremental Structure from Motion





SfM - Pipeline

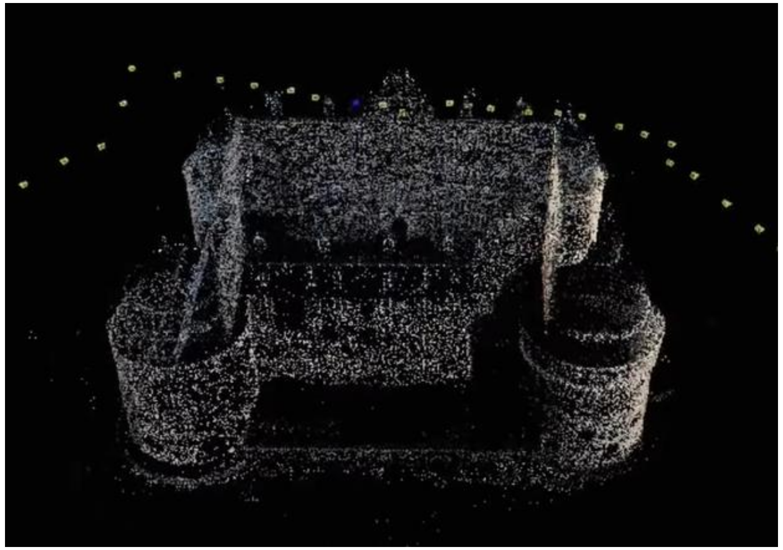
- **Incremental Structure from Motion** bắt đầu từ 2 ảnh và lần lượt nhận thêm từng ảnh mới

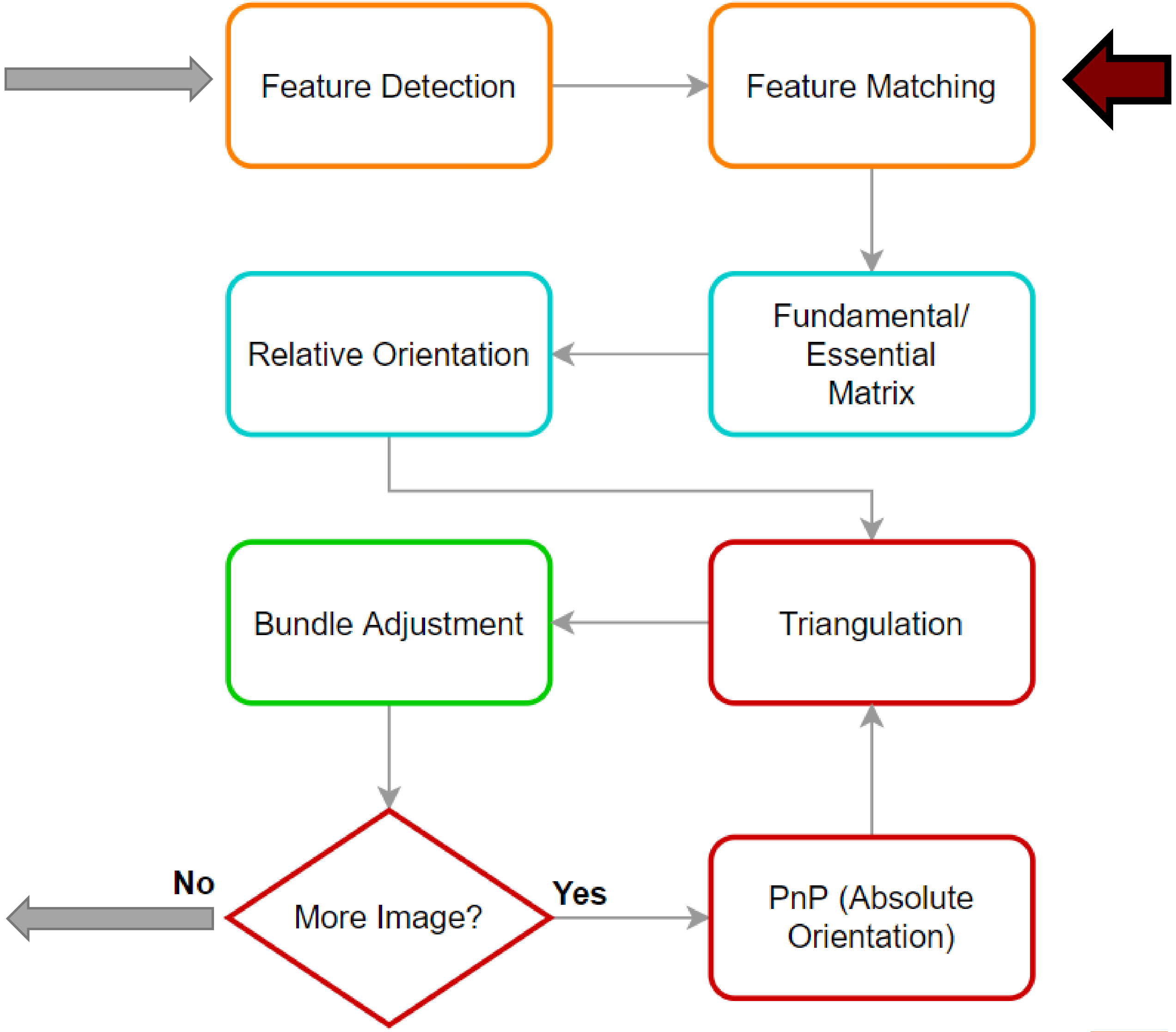


Source: 3D Point Cloud Generation Using Incremental Structure-from-Motion

SfM - Pipeline

Incremental Structure from Motion

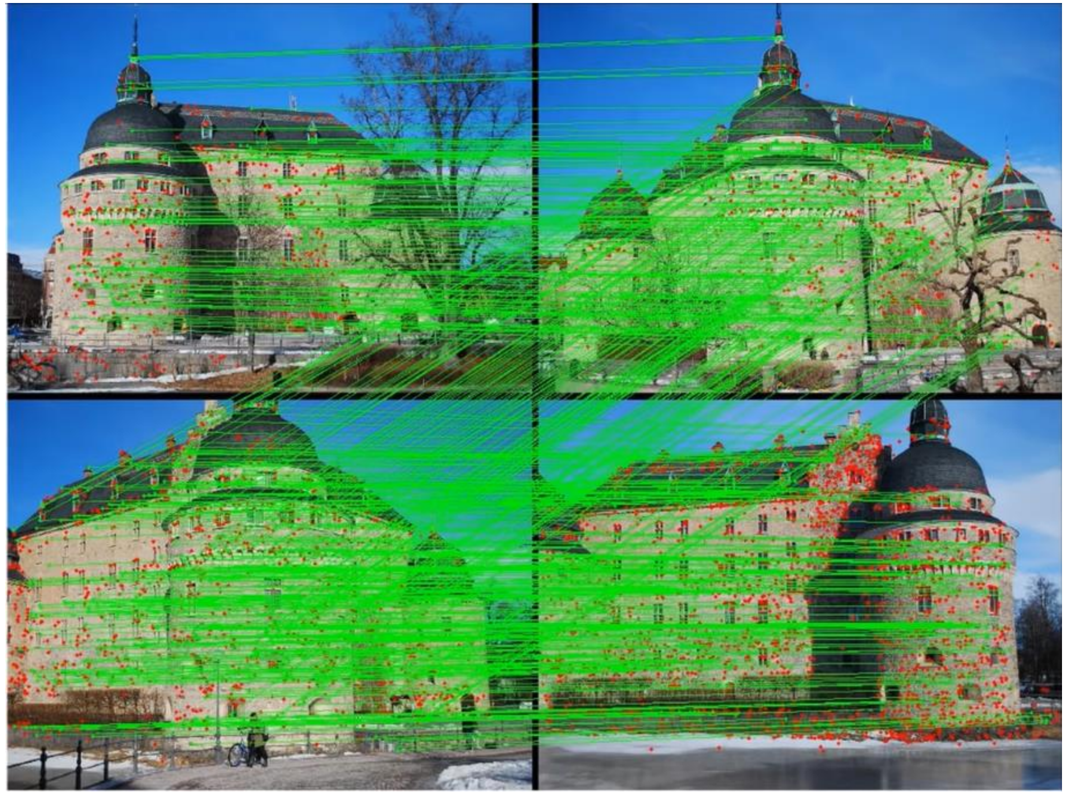




[Source: The Struture from Motion Pipeline]

SfM – Feature Detection & Matching

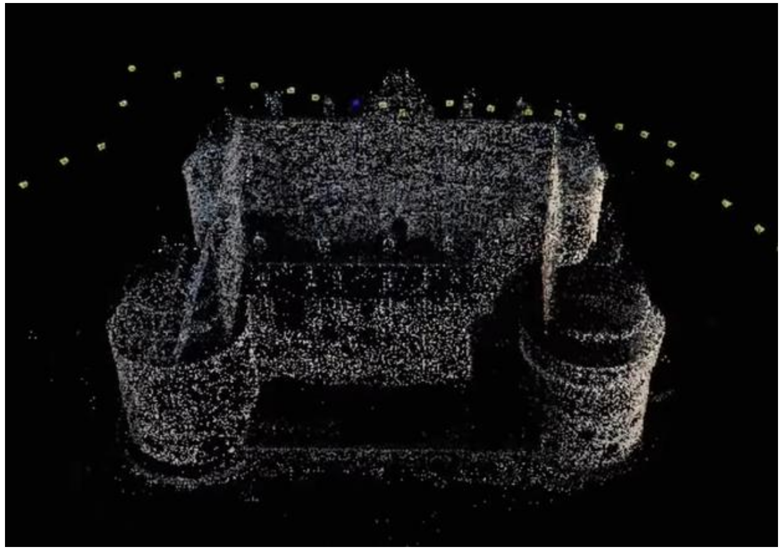
- Phương pháp thường dùng để trích xuất đặc trưng là SIFTm ngoài ra còn có SIFTGPU, ORB,…
- Liên kết các điểm đặc trưng tương ứng (feature correspondences) ở các cặp ảnh

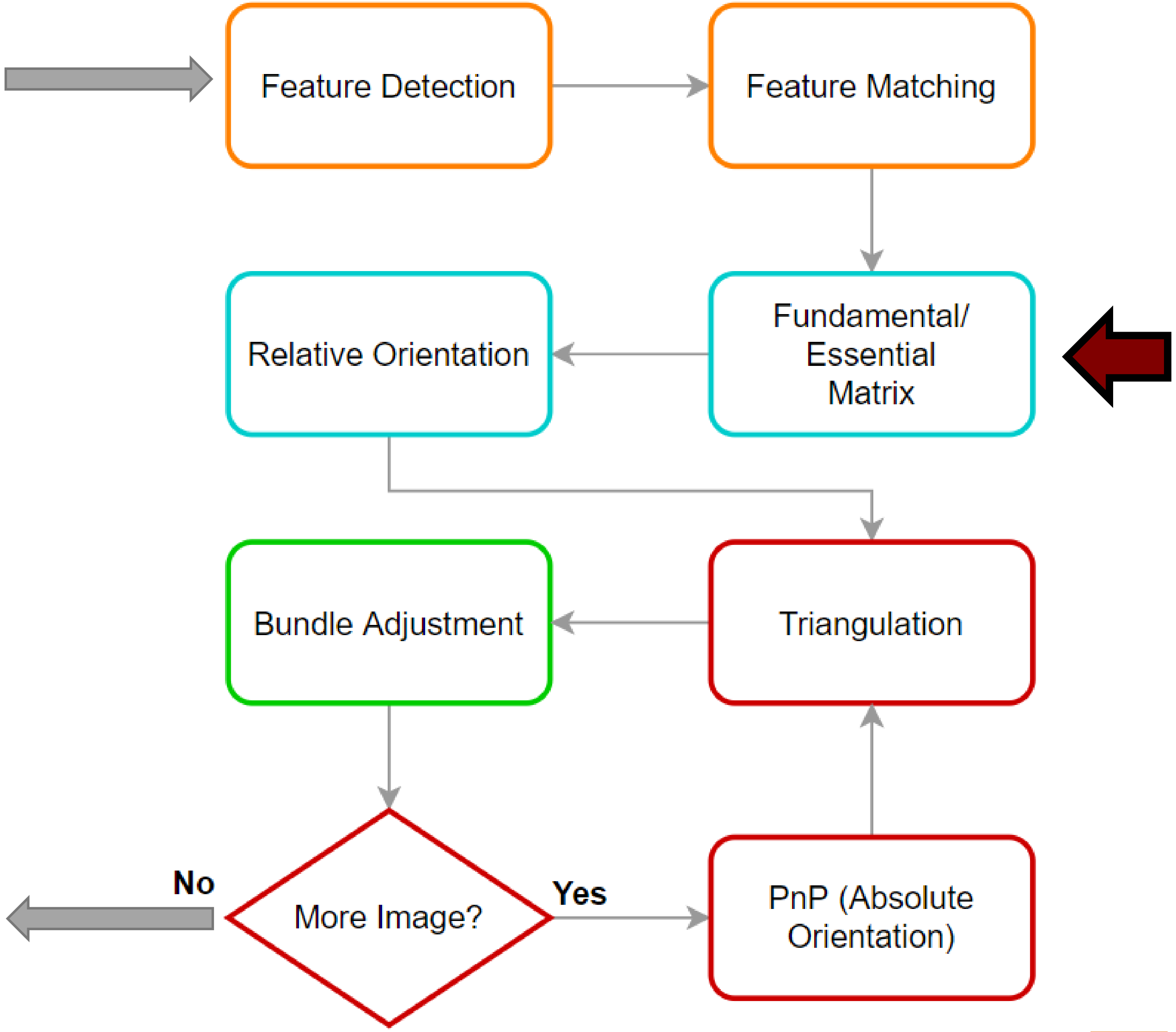


Source: The Struture from Motion Pipeline

SfM - Pipeline

Incremental Structure from Motion



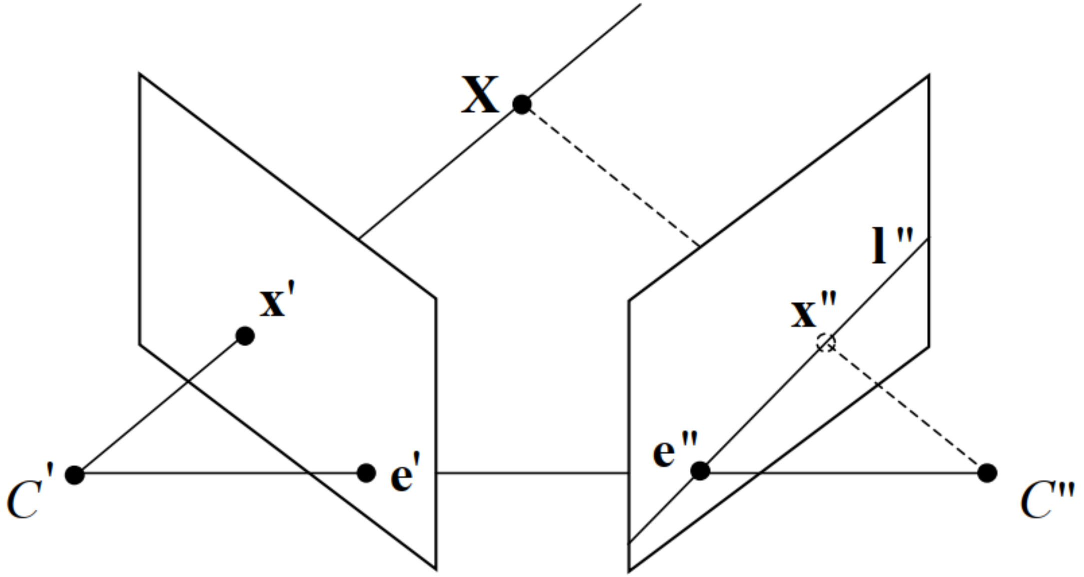


[Source: The Struture from Motion Pipeline]

SfM – Fundamental Matrix

- Fundamental matrix là ma trận 3x3

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$$



- Fundamental matrix biểu diễn ràng buộc hình học giữa 2 điểm đặc trưng tương ứng

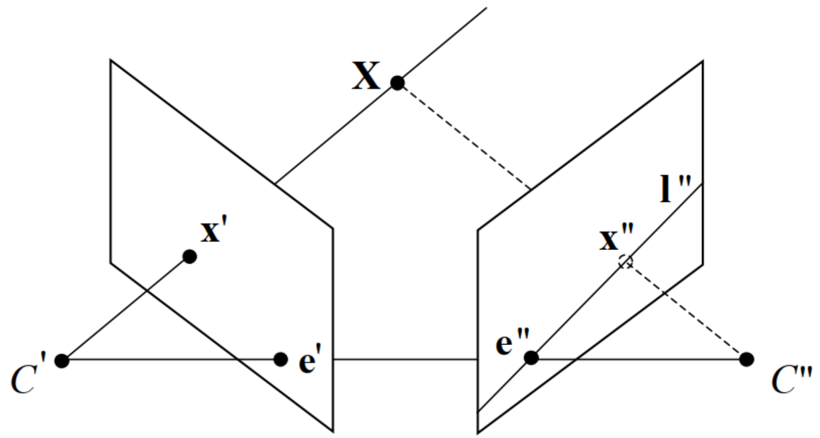
$$\mathbf{x}' = (x', y', 1)^T$$
$$\mathbf{x}'' = (x'', y'', 1)^T$$
$$\mathbf{x}'^T \mathbf{F} \mathbf{x}'' = 0$$

$$\Rightarrow \quad \begin{bmatrix} x_1' & y_1' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_1'' \\ y_1'' \\ 1 \end{bmatrix} = 0$$

SfM – Fundamental Matrix

- Với n cặp đặc trưng tương ứng ta thu được hệ phương trình tuyến tính

$$\underbrace{\begin{bmatrix} x'_1x''_1 & x'_1y''_1 & x'_1 & y'_1x''_1 & y'_1y''_1 & y'_1 & x''_1 & y''_1 & 1 \\ & & & & \vdots & & & & \\ x'_nx''_n & x'_ny''_n & x'_n & y'_nx''_n & y'_ny''_n & y'_n & x''_n & y''_n & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix}}_{\mathbf{f}} = \mathbf{0}$$



hay $\mathbf{Af} = \mathbf{0}$

- Sử dụng **SVD** để giải **f** từ đó thu được ma trận fundamental **F**.

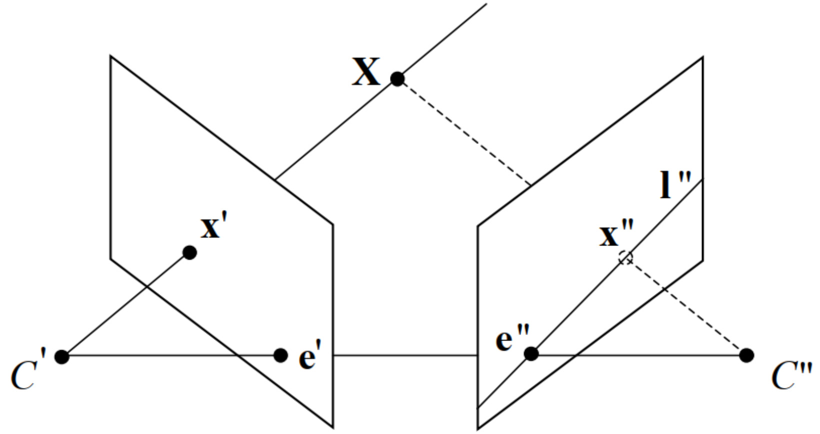
SfM – Fundamental Matrix

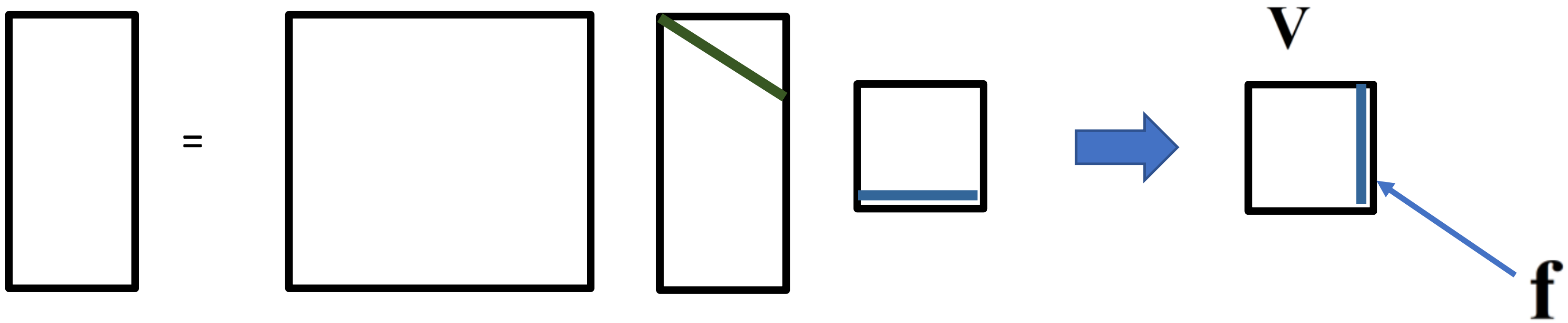
- Dùng SVD để giải

$$\mathbf{Af} = \mathbf{0}$$

- f là singular vector ứng với singular value của A bằng 0

$$\mathbf{A} = \mathbf{UDV}^{\mathrm{T}}$$





$N \times 9$ $\quad N \times N$ $\quad N \times 9$ $\quad 9 \times 9$

- Giải ra **f** từ đó thu được ma trận fundamental **F**.

Source: 2008-SFM-chapters

SfM – Essential Matrix

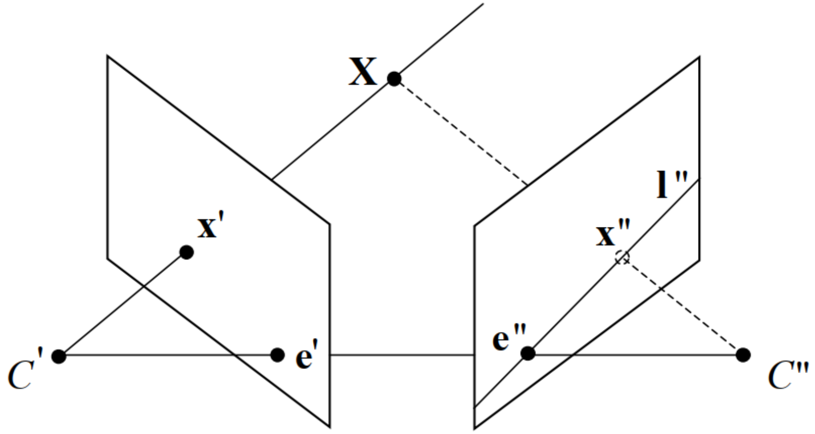
- Tính được essential matrix vì đã biết ma trận nội tại $\mathbf{K}', \mathbf{K}''$

$$\mathbf{E} = \mathbf{K}'^{\mathrm{T}}\mathbf{F}\mathbf{K}''$$

- Tính SVD cho ma trận E

$$\mathbf{E} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}} = \mathbf{U}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\mathbf{V}^{\mathrm{T}}$$

$$\mathbf{E} = \mathbf{U}\underbrace{\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\mathbf{Z}}\underbrace{\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{W}}\mathbf{V}^{\mathrm{T}} = \mathbf{U}\mathbf{Z}\mathbf{W}\mathbf{V}^{\mathrm{T}}$$



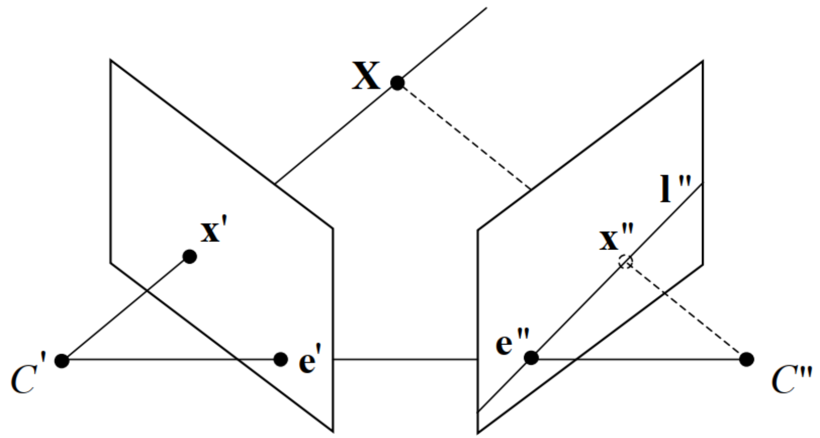
Với **U**, **V** là các ma trận trực giao, D là ma trận đường chéo

Source: 2008-SFM-chapters

SfM – Essential Matrix

- Từ đó tính được ma trận đối xứng xiên S_t của phép dịch chuyển **t** và ma trận xoay **R**

$$\mathbf{S}_\mathbf{t}^{(1)} = \mathbf{UZU}^\mathrm{T}$$
$$\mathbf{S}_\mathbf{t}^{(2)} = \mathbf{UZ}^\mathrm{T}\mathbf{U}^\mathrm{T}$$
$$\mathbf{R}^{(1)} = \mathbf{UWV}^\mathrm{T}$$
$$\mathbf{R}^{(2)} = \mathbf{UW}^\mathrm{T}\mathbf{V}^\mathrm{T}$$



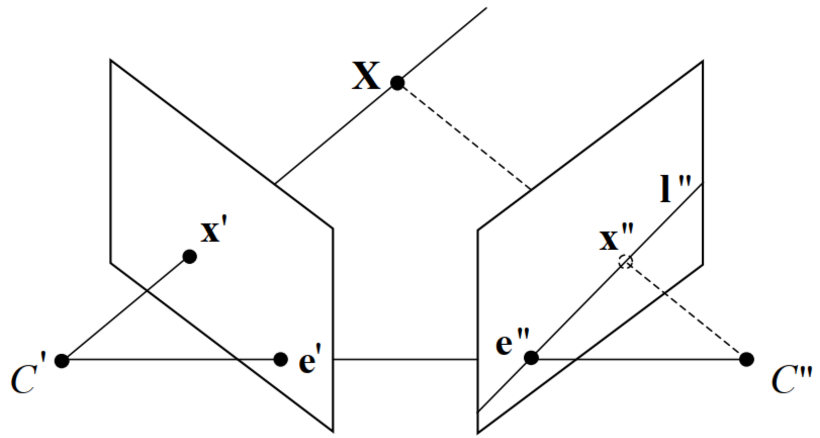
- Về mặt toán học có thể cho ra nhiều kết quả nhưng chỉ một kết quả hợp lí về mặt vật lí

- Từ S_t suy ra t bởi công thức. $$\mathbf{S}_\mathbf{t} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \text{ với } \mathbf{t} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}$$

SfM – Essential Matrix

- Từ **t** và **R** thu được ma trận chiếu của camera 1 và 2.

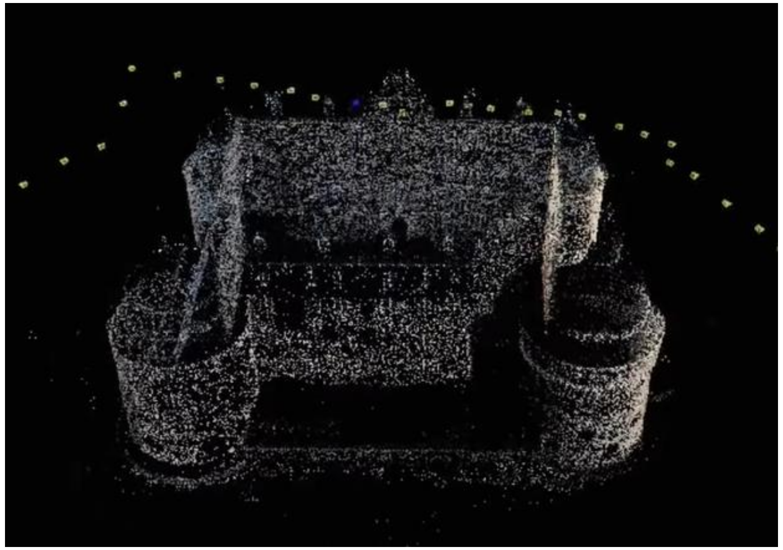
$$\mathbf{P}' = \mathbf{K}'\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}$$
$$\mathbf{P}'' = \mathbf{K}''\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$$

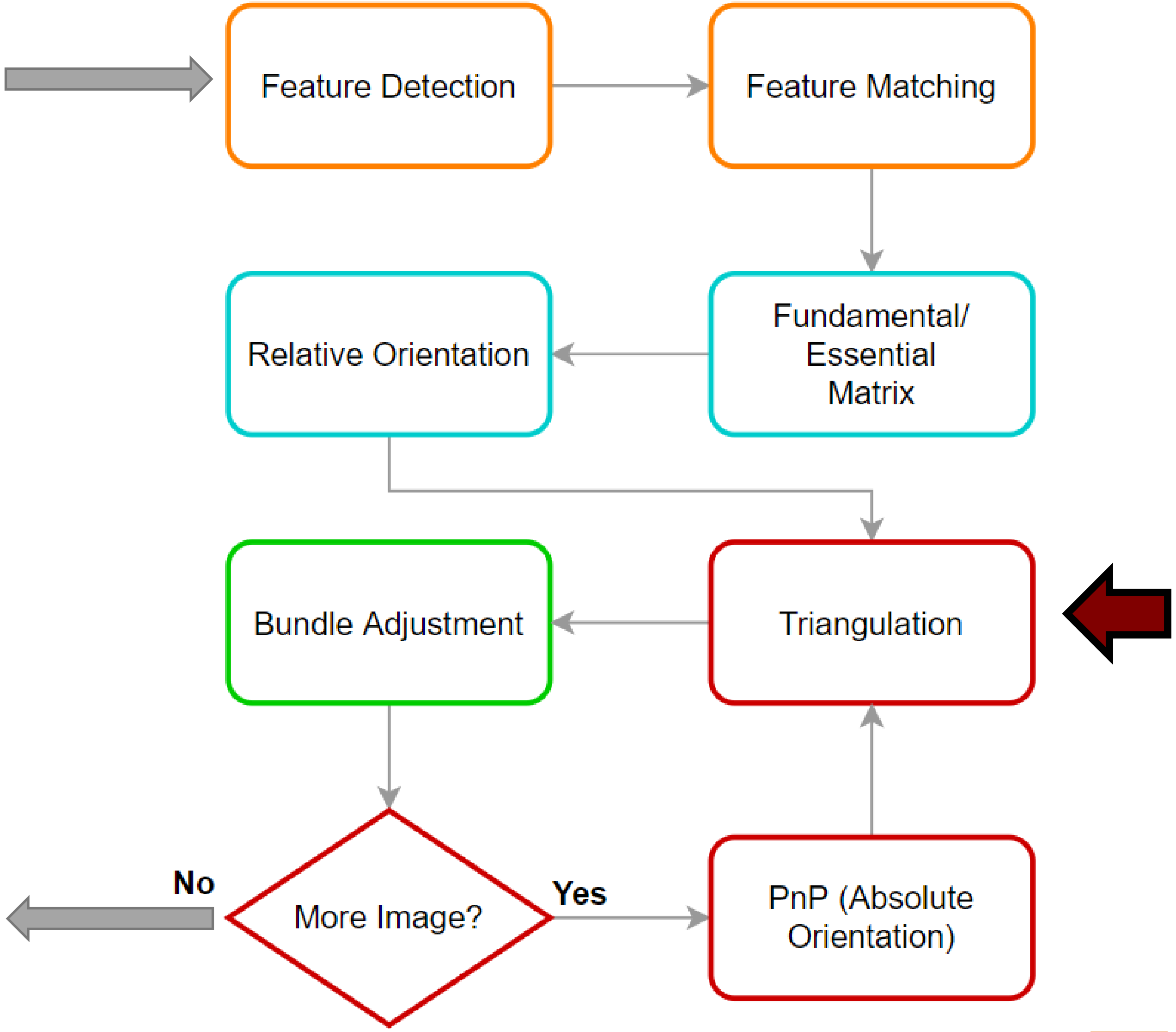


- Hệ tọa độ camera 1 sẽ là hệ tọa độ chung cho các camera nên $\mathbf{R} = \mathbf{I}, \mathbf{t} = \mathbf{0}$.

SfM - Pipeline

Incremental Structure from Motion





SfM – Triangulation

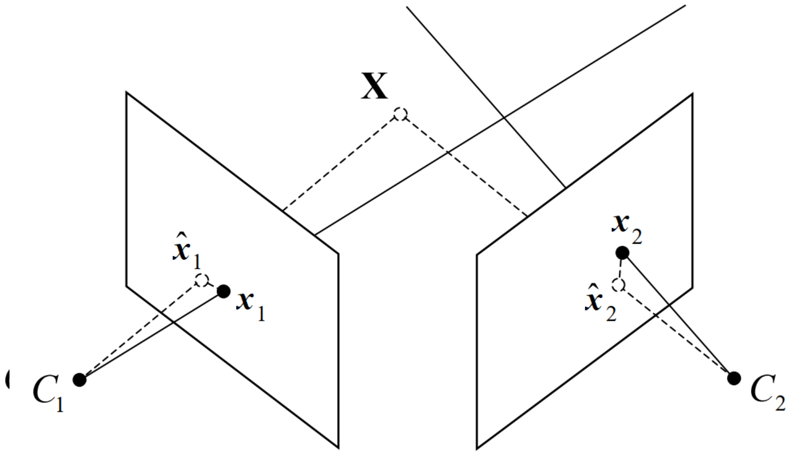
- Với

$\mathbf{x}_i = \begin{bmatrix} x_i & y_i & 1 \end{bmatrix}^T$ là tọa độ điểm ảnh tương ứng với **X** quan sát được từ camera i

$\mathbf{P}_1 = \mathbf{K}_1 \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}$ là ma trận chiếu (3x4) của camera số 1

$$\mathbf{P}_i = \mathbf{K}_i \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \end{bmatrix} = \begin{bmatrix} p_{11}^i & p_{12}^i & p_{13}^i & p_{14}^i \\ p_{21}^i & p_{22}^i & p_{23}^i & p_{24}^i \\ p_{31}^i & p_{32}^i & p_{33}^i & p_{34}^i \end{bmatrix} = \begin{bmatrix} \mathbf{p}_1^{iT} \\ \mathbf{p}_2^{iT} \\ \mathbf{p}_3^{iT} \end{bmatrix}$$ là ma trận chiếu (3x4) của camera i

- Cần xác định $\mathbf{X} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T$

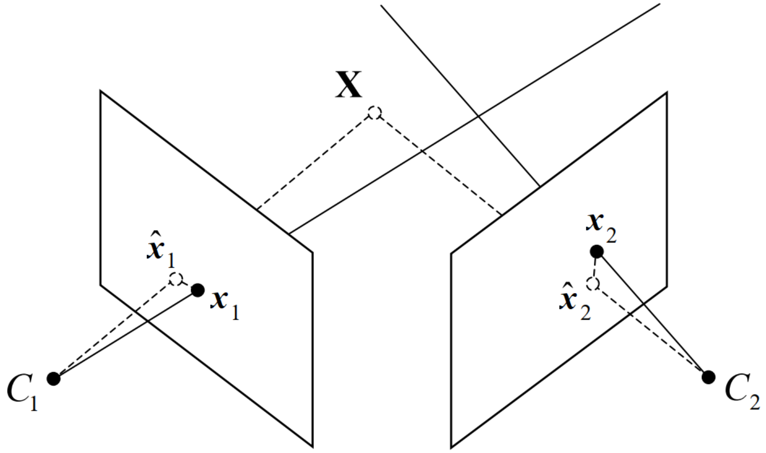


SfM – Triangulation

- Đặt $\hat{\mathbf{x}}_i = \mathbf{P}_i\mathbf{X} = \begin{bmatrix} \hat{x}_i & \hat{y}_i & 1 \end{bmatrix}^T$ là điểm ảnh chiếu từ $\mathbf{X}$ lên camera i, điểm này sẽ lệch so với $\mathbf{x}_i$ do có sai số

- Tổng sai số giữa $\hat{\mathbf{x}}_i$ và $\mathbf{x}_i$ là $\mathbf{AX}$ với

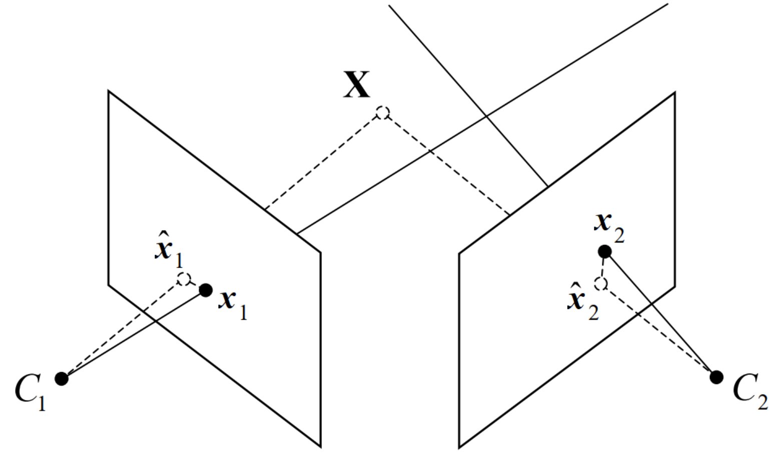
$$\mathbf{A} = \begin{bmatrix} x_1\mathbf{p}_3^{i\,T} - \mathbf{p}_1^{i\,T} \\ y_1\mathbf{p}_3^{i\,T} - \mathbf{p}_2^{i\,T} \\ \dots \\ x_N\mathbf{p}_3^{i\,T} - \mathbf{p}_1^{i\,T} \\ y_N\mathbf{p}_3^{i\,T} - \mathbf{p}_2^{i\,T} \end{bmatrix}$$



- Tìm **X** sao cho tổng sai số là nhỏ nhất

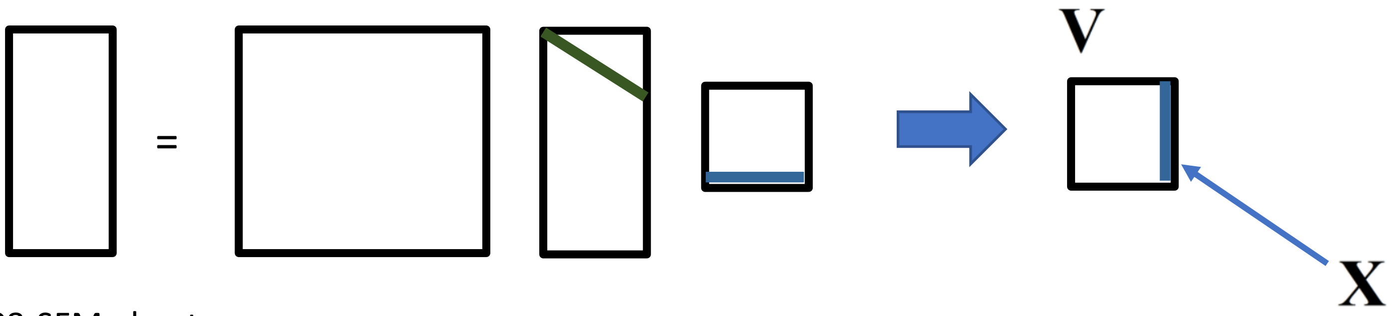
SfM – Triangulation

- Với $\mathbf{A} = \begin{bmatrix} x_1\mathbf{p}_3^{iT} - \mathbf{p}_1^{iT} \\ y_1\mathbf{p}_3^{iT} - \mathbf{p}_2^{iT} \\ \dots \\ x_N\mathbf{p}_3^{iT} - \mathbf{p}_1^{iT} \\ y_N\mathbf{p}_3^{iT} - \mathbf{p}_2^{iT} \end{bmatrix}$



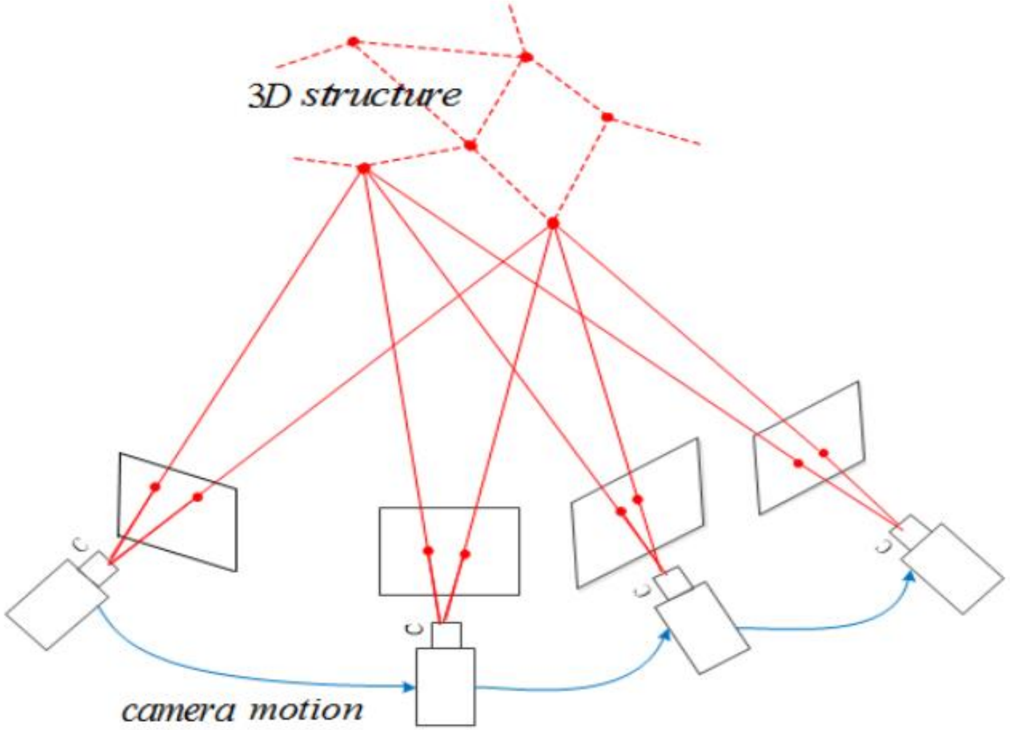
- Tìm **X** sao cho **AX** là nhỏ nhất
- Dùng SVD phân tích **A**, giải ra **X** là singular vector ứng với singular value nhỏ nhất của **A**

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}}$$



SfM – Projective n-Point

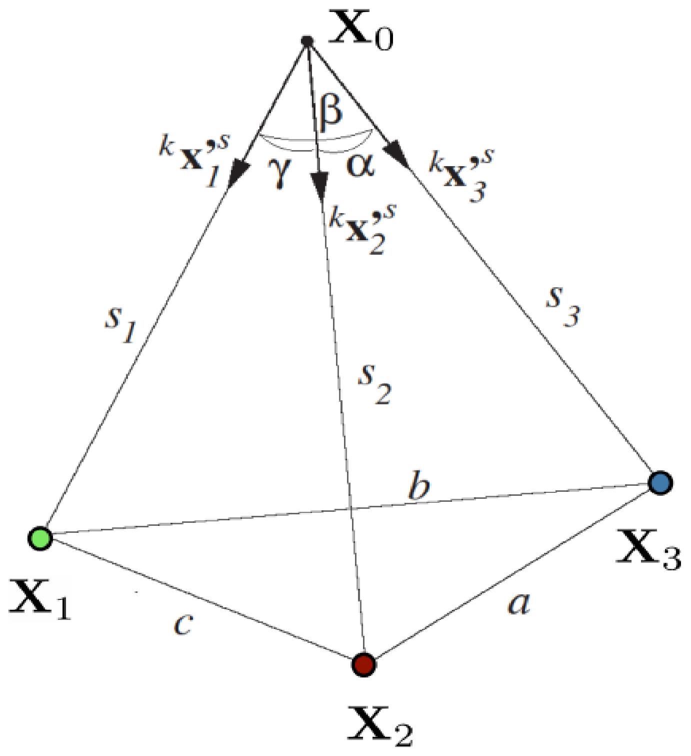
- Khi một ảnh mới được thêm vào tập ảnh hiện tại, PnP sẽ giải quyết bài toán xác định vị trí, góc nhìn của ảnh (camera) mới này trong hệ tọa độ chung khi đã biết thông số bên trong của camera.
- PnP sử dụng quan hệ tương ứng 2D-3D của các điểm đặc trưng trong ảnh và đám mây điểm hiện tại để tính toán tọa độ tâm chiếu camera $\mathbf{X_0}$ và ma trận xoay **R**



Source: 3D Point Cloud Generation Using Incremental Structure-from-Motion

SfM – Projective n-Point

- Phương pháp P-3-P là cơ chế đơn giản hóa cho PnP.
- Input: Tọa độ 3 điểm X_i trong hệ tọa độ thực tế và vector hướng ${}^{k}x_i^{s}$ từ tâm chiếu camera đến X_i .
- Output: tọa độ tâm chiếu camera X_O và ma trận xoay R.



Trong đó:

X_0: C_i - tâm chiếu của camera thứ i.

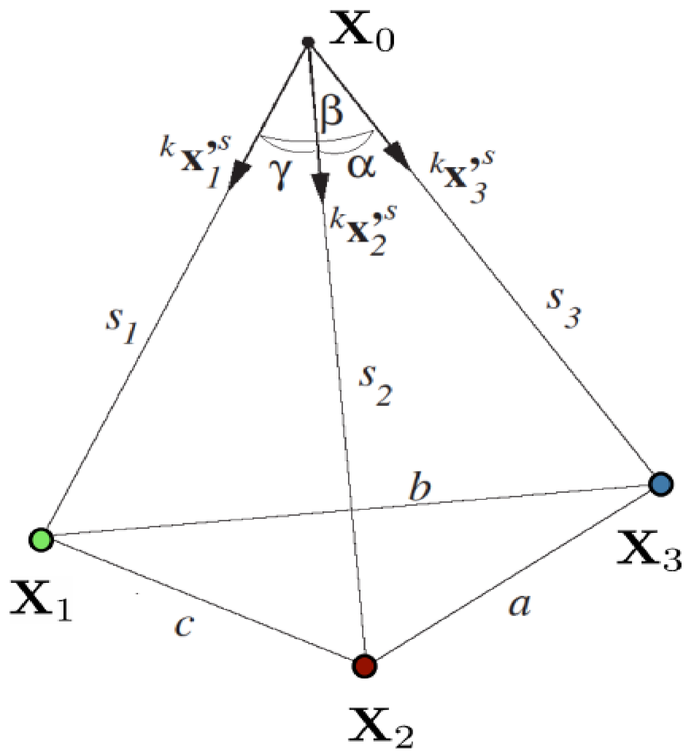
s_i : Khoảng cách từ tâm chiếu camera đến X_i.

SfM – Projective n-Point

- Tìm tọa độ 3 điểm X_i trong hệ tọa độ camera - ${}^{k}X_i^{s}$.

$$ {}^{k}X_i^{s} = s_i\, {}^{k}x_i^{s} = R(X_i - X_O) $$

  - Tính α, β, γ.
  - Tính a, b, c.
  - Tính s_1, s_2, s_3.
  - Tính ${}^{k}X_i^{s}$.
- Tính toán tọa độ tâm chiếu camera X_O và ma trận xoay R.



SfM – Projective n-Point

Tính α, β, γ

- Công thức tính góc giữa 2 vector

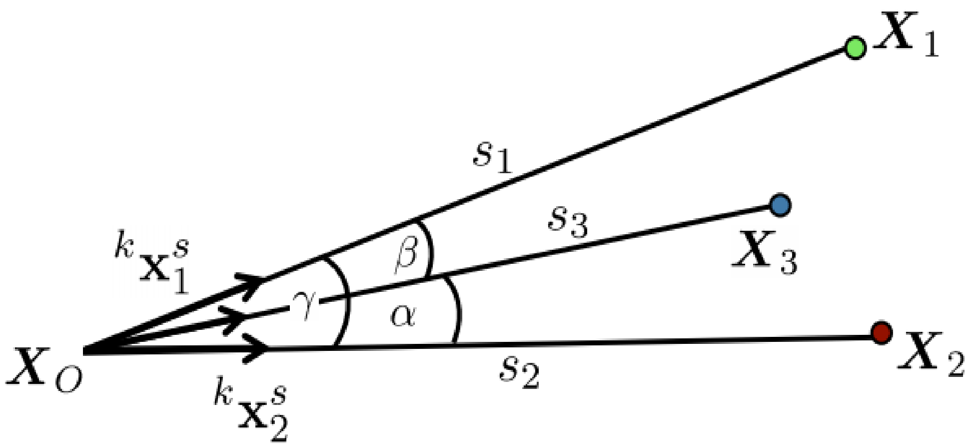
$$\cos(\overrightarrow{n1}, \overrightarrow{n2}) = \frac{\overrightarrow{n1}.\overrightarrow{n2}}{\left|\overrightarrow{n1}\right|\left|\overrightarrow{n2}\right|}$$

- Tính được α, β, γ

$$\alpha = \arccos({}^{k}x_2{}^{s}, {}^{k}x_3{}^{s})$$

$$\beta = \arccos({}^{k}x_3{}^{s}, {}^{k}x_1{}^{s})$$

$$\gamma = \arccos({}^{k}x_1{}^{s}, {}^{k}x_2{}^{s})$$



Source: Photogrammetric Computer Vision course

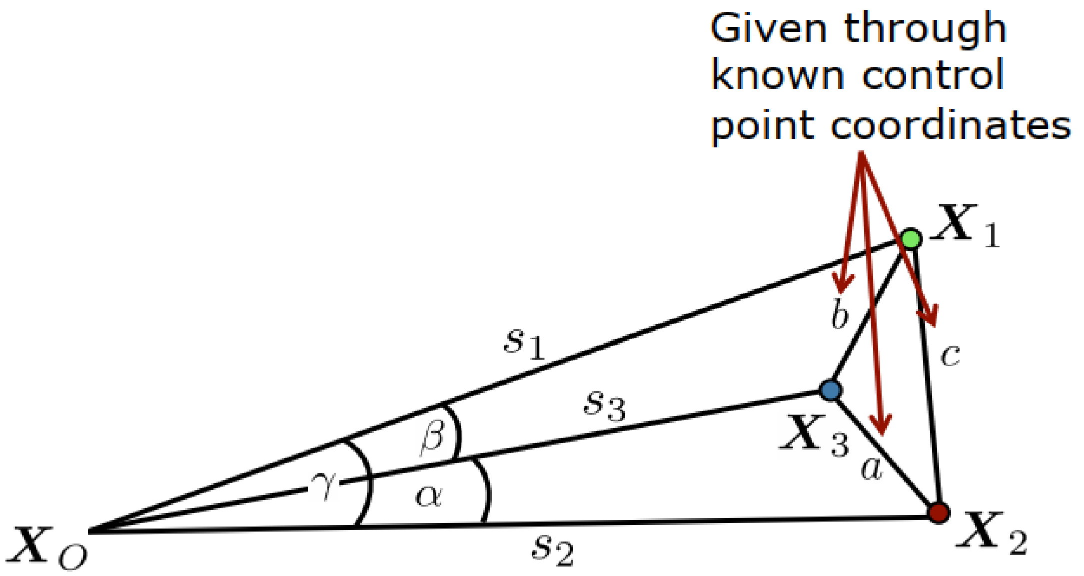
SfM – Projective n-Point

Tính a, b, c

$$a = \|X_3 - X_2\|$$

$$b = \|X_1 - X_3\|$$

$$c = \|X_2 - X_1\|$$



SfM – Projective n-Point

Tính s_1, s_2, s_3

- Công thức luật cos tại 3 tam giác.

$$a^2 = s_2^2 + s_3^2 - 2s_2s_3\cos\alpha$$
$$b^2 = s_1^2 + s_3^2 - 2s_1s_2\cos\beta$$
$$c^2 = s_1^2 + s_2^2 - 2s_1s_2\cos\gamma$$

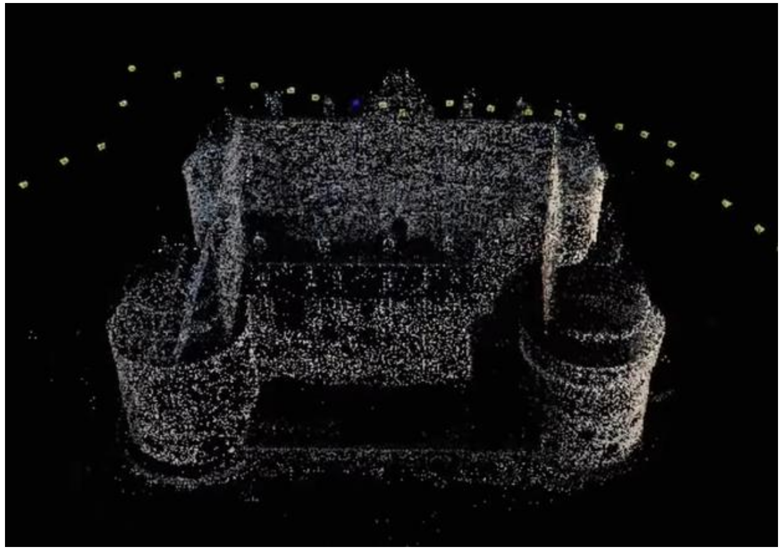
- Đặt $u = \frac{s_1}{s_2}, \mathrm{v} = \frac{s_3}{s_1}$.
- Thu được phương trình bậc 4 biến v.
- Giải được s_1, s_2, s_3.
- Tính được ${}^{k}X_i^{s}$.

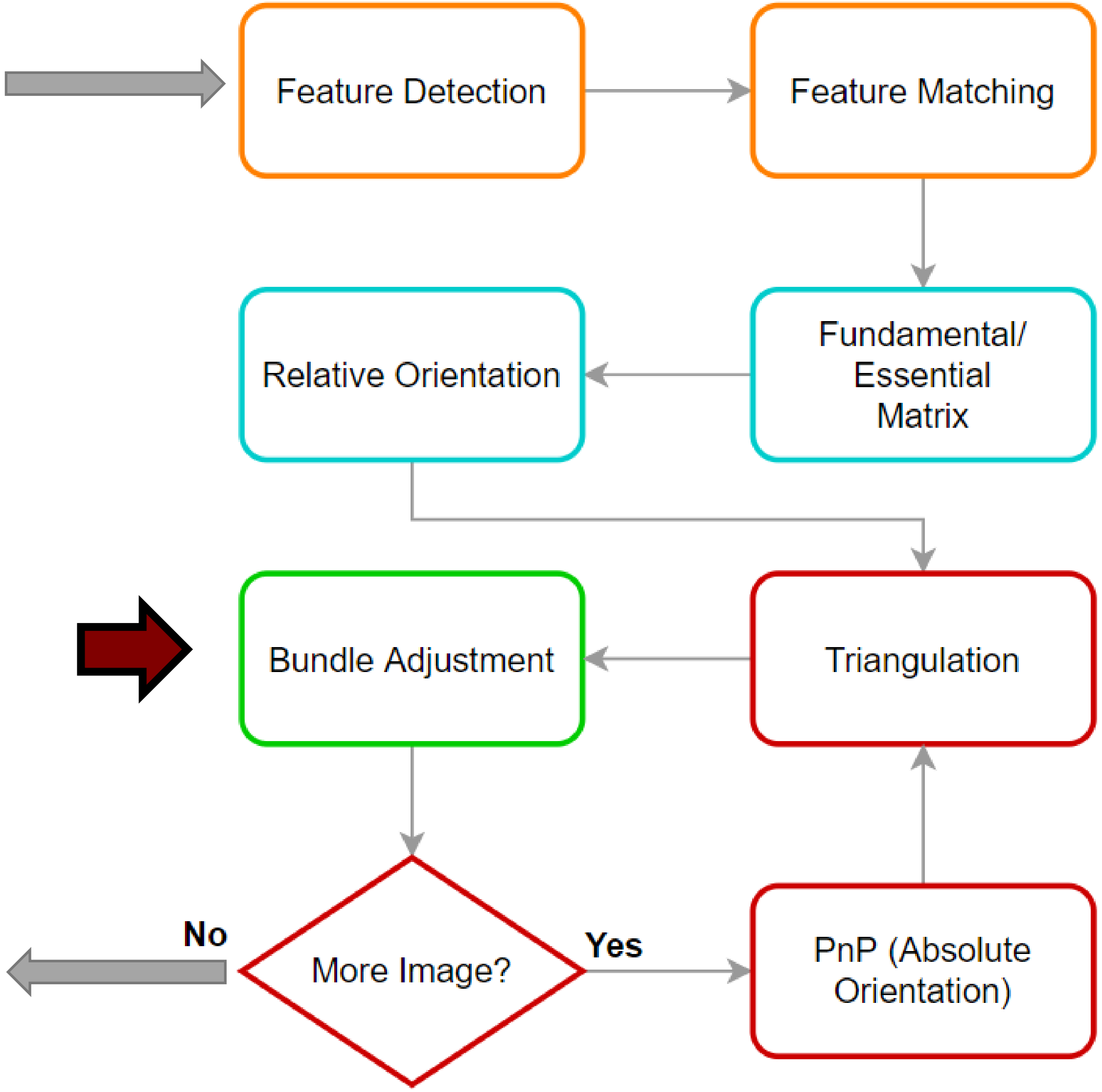
SfM – Projective n-Point

- Từ 3 điểm trong hệ tọa độ chung và 3 điểm trong hệ tọa độ camera, tính rigidbody transformation và thu được R và X_O.
- Từ P-3-P mở rộng cho PnP, ta cần nhiều điểm hơn để đảm bảo thu được kết quả tối ưu nhất khi dùng phương pháp bình phương tối tiểu để điều chỉnh kết quả cho R và X_O (sử dụng điểm thứ 4 để kiểm tra kết quả 3 điểm dùng trong P-3-P.

SfM - Pipeline

Incremental Structure from Motion





SfM – Bundle Adjustment

- Mục đích của BA là cực tiểu hóa hàm sai số phép chiếu

$$E(\mathbf{X},\mathbf{R},\mathbf{T}) = \sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij} L\left(\left\|\mathbf{x}_{ij} - P(\mathbf{X}_j,\mathbf{R}_i,\mathbf{t}_i)\right\|_2^2\right)$$

- Với m, n là số lượng camera và điểm 3D hiện tại

  $\mathbf{X}_j$ là tọa độ điểm 3D thứ j

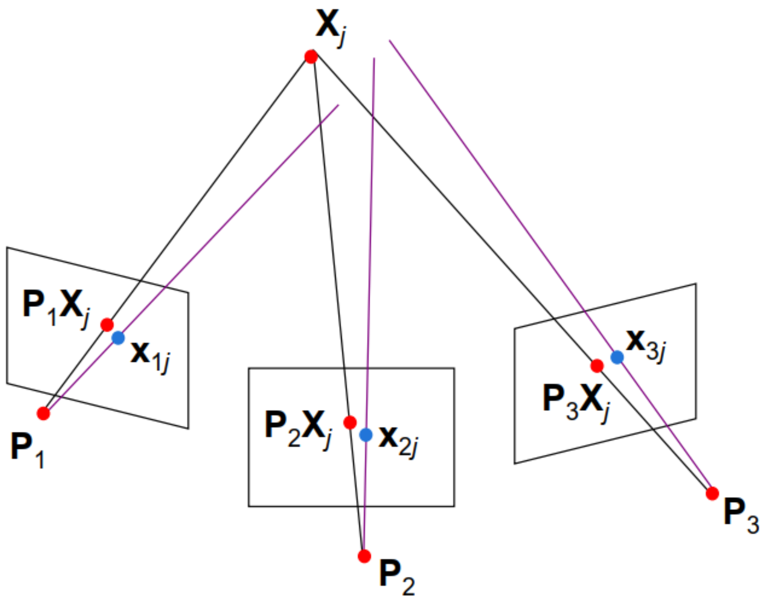
  $\mathbf{x}_{ij}$ là điểm ảnh của $\mathbf{X}_j$ quan sát được bởi camera thứ i

  $\mathbf{R}_i, \mathbf{t}_i$ là thông số camera thứ i

  $P(\mathbf{X}_j,\mathbf{R}_i,\mathbf{t}_i) = \mathbf{K}_i(\mathbf{R}_i\mathbf{X}_j + \mathbf{t}_i)$ là hàm chiếu từ $\mathbf{X}_j$ xuống camera thứ i

  $L(\cdot)$ là hàm mất mát

  W_{ij} có giá trị 0 hoặc 1, biểu thị camera I có nhìn thấy $\mathbf{X}_j$ không



Source: Structure from Motion-Vision Spring 2012

SfM – Bundle Adjustment

- BA dùng thuật toán tối ưu phi tuyến, thường là Levenberg–Marquardt hoặc Gauss–Newton
- Thuật toán cần một giá trị khởi tạo **P** tốt
- Sau đó xấp xỉ tuyến tính sai số tại giá trị $\mathbf{P} + \Delta\mathbf{P}$
- Tìm $\Delta\mathbf{P}$ sao cho sai số tại $\mathbf{P} + \Delta\mathbf{P}$ là nhỏ nhất
- Tiến hành cập nhật $\mathbf{P} \leftarrow \mathbf{P} + \Delta\mathbf{P}$
- Lặp lại quá trình cho đến khi hội tụ

SfM – Bundle Adjustment

- Sau đây là thuật toán tối ưu phi tuyến Gauss-Neuton cho một biến
- Có $\mathbf{x}_i$ là các giá trị ghi nhận với trạng thái **p**

  $f_i(\mathbf{p})$ là hàm ánh xạ **p** với giá trị dự đoán $\hat{\mathbf{x}}_i$
- Cần ước lượng p mà mô tả $\mathbf{x}_i$ tốt nhất, tức là sai số giữa $\mathbf{x}_i$ và $\hat{\mathbf{x}}_i$ là nhỏ nhất

SfM – Bundle Adjustment

- Khởi tạo giá trị **p**. Tổng sai số tại trạng thái **p**

$$E(\mathbf{p}) = \sum_i \|\mathbf{r}_i\|^2 = \sum_i \|f_i(\mathbf{p}) - \mathbf{x}_i\|^2$$

- Xấp xỉ tuyến tính tổng sai số tại trạng thái $\mathbf{p} + \Delta\mathbf{p}$ sử dụng khai triển Taylor

$$E(\mathbf{p}+\Delta\mathbf{p}) = \sum_i \|f_i(\mathbf{p}+\Delta\mathbf{p}) - \mathbf{x}_i\|^2 \approx \sum_i \|f_i(\mathbf{p}) + \mathbf{J}_i(\mathbf{p})\Delta\mathbf{p} - \mathbf{x}_i\|^2$$

với $\mathbf{J}_i(\mathbf{p})$ là ma trận Jacobian với trạng thái **p**

- Biến đổi toán học $E(\mathbf{p}+\Delta\mathbf{p}) \approx \sum_i \|f_i(\mathbf{p}) - \mathbf{x}_i + \mathbf{J}_i(\mathbf{p})\Delta\mathbf{p}\|^2$

$$= \sum_i \|\mathbf{r}_i + \mathbf{J}_i(\mathbf{p})\Delta\mathbf{p}\|^2 = \sum_i \left(\|\mathbf{r}_i\|^2 + 2\mathbf{r}_i^T\mathbf{J}_i\Delta\mathbf{p} + \Delta\mathbf{p}^T\mathbf{J}_i^T\mathbf{J}_i\Delta\mathbf{p}\right)$$

$$= \underbrace{\sum_i \|\mathbf{r}_i\|^2}_{c} + 2\underbrace{\left(\sum_i \mathbf{r}_i^T\mathbf{J}_i\right)}_{\mathbf{b}^T}\Delta\mathbf{p} + \Delta\mathbf{p}^T\underbrace{\left(\sum_i \mathbf{J}_i^T\mathbf{J}_i\right)}_{\mathbf{H}}\Delta\mathbf{p}$$

$$= c + 2\mathbf{b}^T\Delta\mathbf{p} + \Delta\mathbf{p}^T\mathbf{H}\Delta\mathbf{p}$$

SfM – Bundle Adjustment

- Ở đây $E(\mathbf{p}+\Delta\mathbf{p})$ có dạng bậc 2, ta giải tìm được $E(\mathbf{p}+\Delta\mathbf{p})$ đạt giá trị nhỏ nhất tại

$$\Delta\mathbf{p} = -\mathbf{H}^{-1}\mathbf{b}$$

- Cập nhật trạng thái **p**

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$$

- Lặp lại quá trình cho đến khi hội tụ

SfM – Bundle Adjustment

- Ở đây $E(\mathbf{p}+\Delta\mathbf{p})$ có dạng bậc 2, ta giải tìm được $E(\mathbf{p}+\Delta\mathbf{p})$ đạt giá trị nhỏ nhất tại

$$\Delta\mathbf{p} = -\mathbf{H}^{-1}\mathbf{b}$$

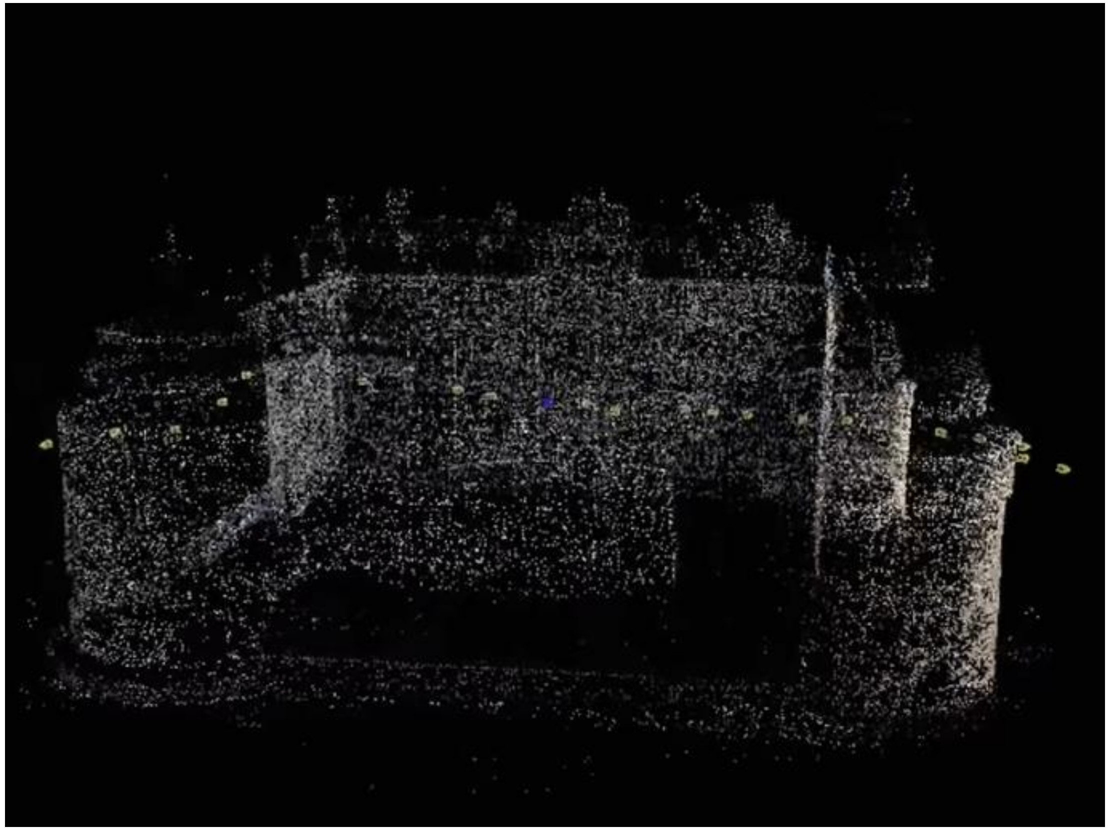
- Cập nhật trạng thái **p**

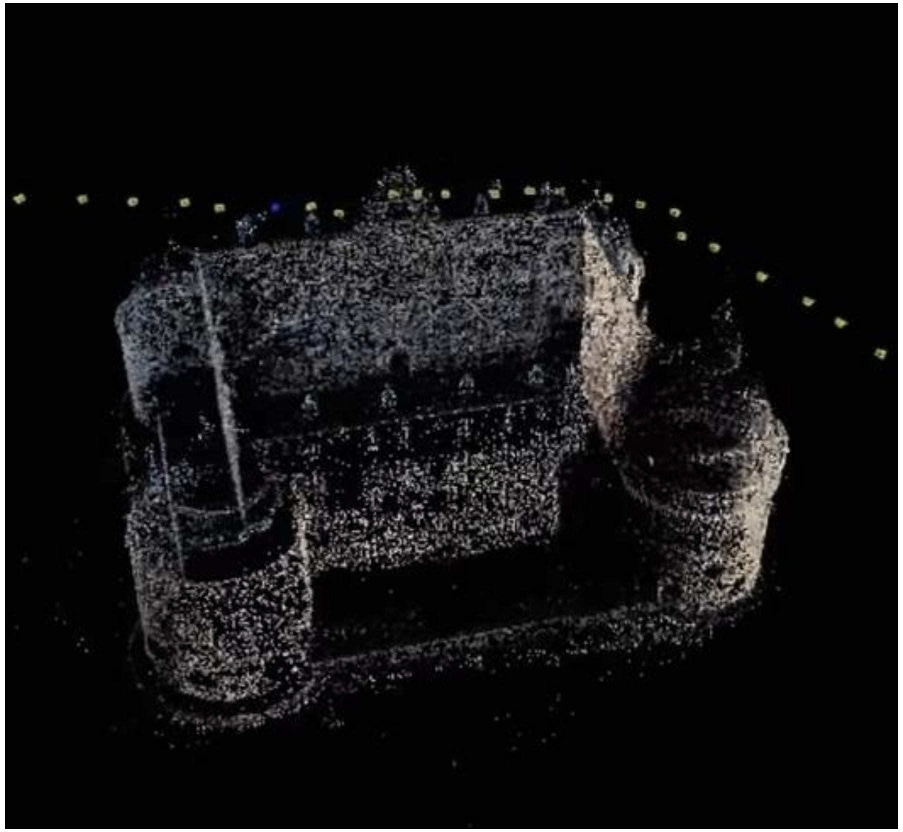
$$\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$$

- Lặp lại quá trình cho đến khi hội tụ

SfM – Result

Mô hình đám mây điểm và vị trí camera





Source: The Struture from Motion Pipeline

5. Related works

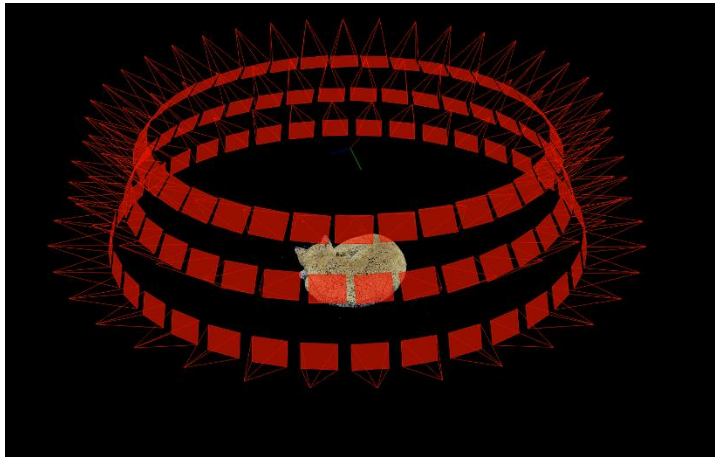
STT	Tên bài báo	Năm	Ý tưởng cải tiến	Data	Thời gian chạy	Ưu điểm	Nhược điểm
1	Structure from Motion Revisited (COLMAP)	2016	Cải tiến chiến thuật lựa chọn ảnh mới: lựa chọn ảnh nhìn thấy nhiều điểm 3D đã có và các điểm phải có phân bố tốt. Phương pháp triangulation làm việc tốt hơn với nhiễu Dùng hàm lỗi Cauchy trong bước bundle adjustment để giải quyết các điểm outliers	Ancient Building (TAB) dataset gồm 20 ảnh cùng độ phân giải 4160x3210 pixels	60.28s	Cải tiến độ chính xác hơn phương pháp SfM thông thường	Bước bundle adjustment vẫn lặp lại nhiều lần. Đây là bước tốn khá nhiều chi phí tính toán. Do đó, khi số lượng người sở hữu camera tăng lên thì dữ liệu ảnh cũng tăng theo, thì hệ thống chạy khá lâu Ảnh có đặc trưng lặp lại cấu trúc tương tự nhau ảnh hưởng đến độ chính xác trong bước tìm đặc trưng tương ứng
2	Fast incremental structure from motion based on parallel bundle (Fast-ISFM)	2020	Sử dụng SIFTGPU là phiên bản cài đặt SIFT có thể chạy song song để tăng tốc độ trích xuất đặc trưng. Phương pháp Parallel bundle adjustment chạy song song trên GPU đáp ứng các hệ thống SfM quy mô lớn	Ancient Building (TAB) dataset gồm 20 ảnh cùng độ phân giải 4160x3210 pixels	110.35s	Tăng tốc độ tính toán đáng kể đặc biệt ở bước Bundle Adjustment. Đồng thời vẫn giữ được độ chính xác khá cao	Đối với những ảnh có đặc trưng lặp lại cấu trúc tương tự nhau thì ảnh hưởng đến độ chính xác trong bước tìm đặc trưng tương ứng

6. Experiments

Dataset 1

Một số ảnh input

Kết quả nhìn từ trên cao

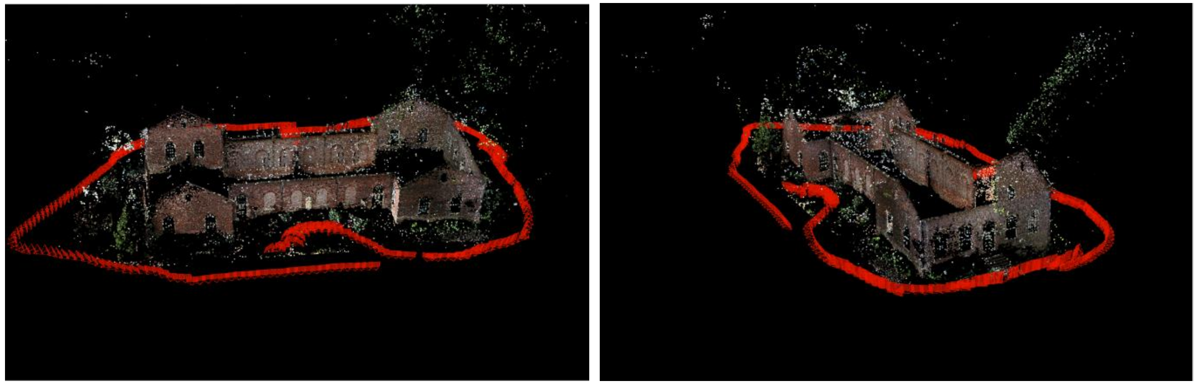


Kết quả nhìn từ góc xiên

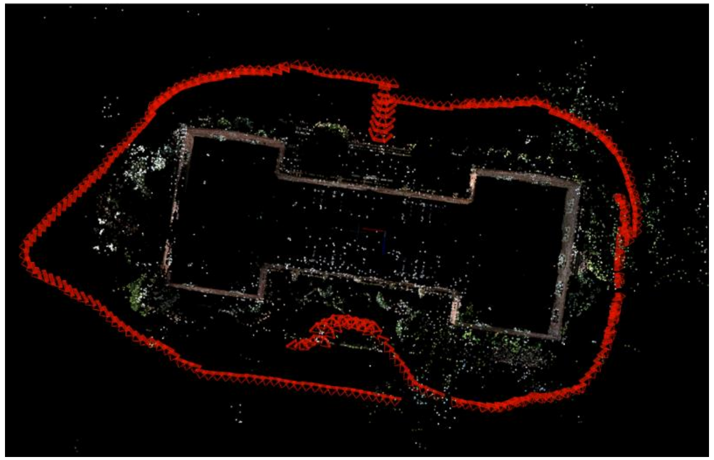
6. Experiments

Dataset 2

Một số ảnh input



Kết quả nhìn từ góc xiên

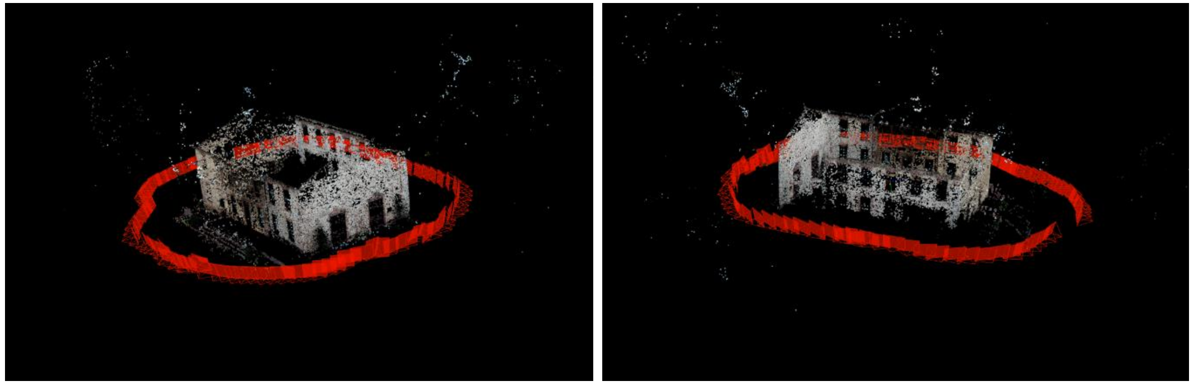


Kết quả nhìn từ trên cao

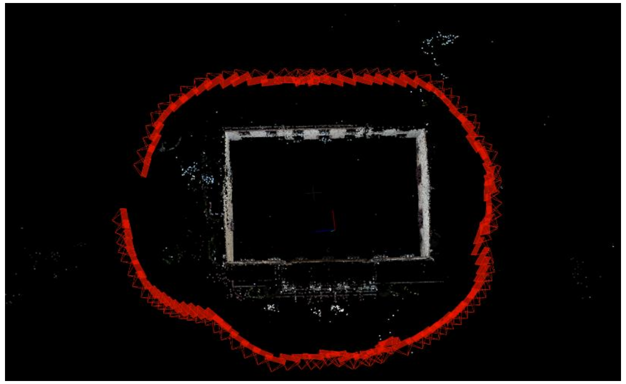
6. Experiments

Dataset 3

Một số ảnh input



Kết quả nhìn từ góc xiên



Kết quả nhìn từ trên cao

References

[1] Structure-From-Motion Revisited (2016)

[2] 3D Point Cloud Generation Using Incremental Structure-from-Motion (2018)

[3] Fast incremental structure from motion based on parallel bundle adjustment (2020)

[4] Computer Vision, Algorithms and Appications (2010)

[5] Photogrammetric Computer Vision (2016)

[6] Photogrammetry Computer Vision (2020)

[7] Photogrammetry II (2020)

Thank you!