ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN
PHOTOGRAMMETRY

MÔN XỬ LÝ ẢNH SỐ VÀ VIDEO SỐ

NHÓM RETINA

Giáo viên hướng dẫn: TS. Lý Quốc Ngọc

Tháng 1 - 2021

Mục Lục

Chương 1. Giới Thiệu

1.1. Giới thiệu Photogrammetry

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

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.

Mục tiêu chính của Photogrammetry là xây dựng mối quan hệ giữa thông tin thuộc tính hình học (geometrical data) trên mặt phẳng ảnh (image plane) với thông tin cấu trúc 3D của vật thể trong không gian. Từ đó thu thập được thông tin đại lượng vật lí của vật thể.

1.2. Động lực về mặt khoa học và ứng dụng

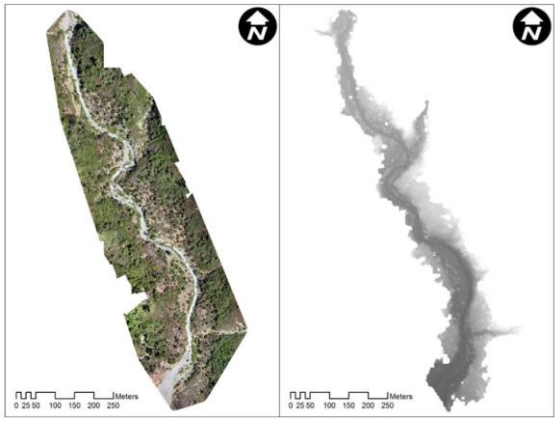
Về mặt khoa học, photogrammetry là phương pháp hiệu quả để giải quyết bài toán tái tạo cấu trúc 3D từ các ảnh 2D, đây là một trong những vấn đề cơ bản của thị giác máy tính. Bên cạnh đó, kết quả của photogrammetry có thể được sử dụng cho các tác vụ khác của thị giác máy tính như xác định vật thể, tái 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, …

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

- 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. Các bản đồ này có thế dùng để phân tích theo dõi quá trình thay đổi của địa hình.
- 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.
- Trong kỹ thuật thiết kế, sản xuất, photogrammetry có thể dùng trong khâu kiểm soát chất lượng, kiểm tra sản phẩm lỗi.
- Trong địa chất, cũng như bởi các nhà khảo cổ để nhanh chóng tạo ra các sơ đồ của các địa điểm rộng lớn hay phức tạp và hay bởi các nhà khí tượng học như một cách để xác định tốc độ gió thực của một cơn bão mà ở đó các dữ liệu thời tiết khách quan không thể lấy được.
- Ngoài ra, Photogrammetry được dùng để chuyển những vật thể lớn như tòa nhà, tàu thủy, máy bay, công trình xây dựng, hoặc vùng đất như cánh đồng thành mô hình 3D phục vụ nhiều mục đích khác nhau.

Hình 1. Kết quả mô hình mô phỏng thành phố Rome. Source: Structure from Motion Revisited (2016)

Hình 2. Kết quả mô hình mô phỏng lâu đài. Source: https://www.3dflow.net

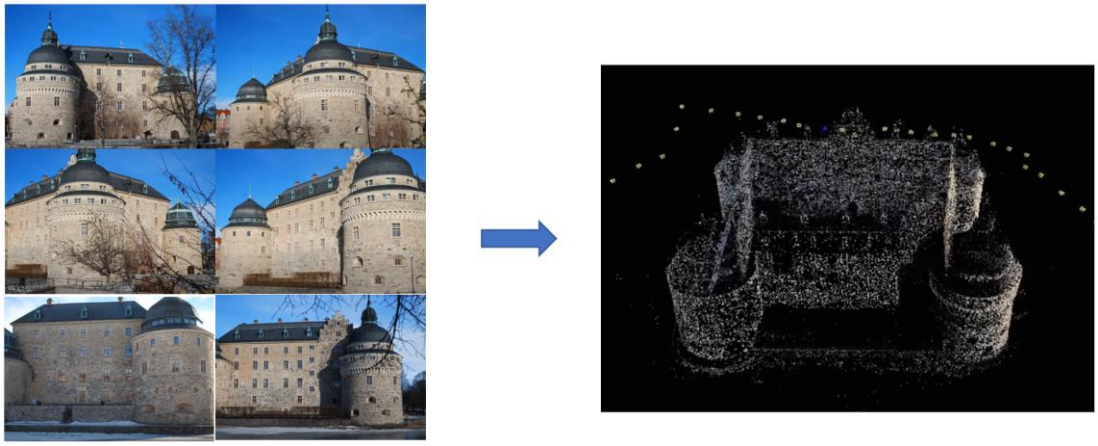


Hình 3. Bản đồ địa hình từ trên cao của Eel River. Source: http://gsp.humboldt.edu

1.3.Phát biểu bài toán

1.3.1. Input, output bài toán

Input là 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 là đám mây điểm biểu diễn cấu trúc 3D của object/scene và các tham số (vị trí, hướng) của camera chụp vật thể.



Hình 4. Input và Output của bài toán. Source: Video The Struture from Motion Pipeline

1.3.2. Framework xử lý chung

Trong đồ án này nhóm chọn tìm hiểu về thuật toán Structure from Motion (SfM). Các thuật toán Structure from Motion 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.

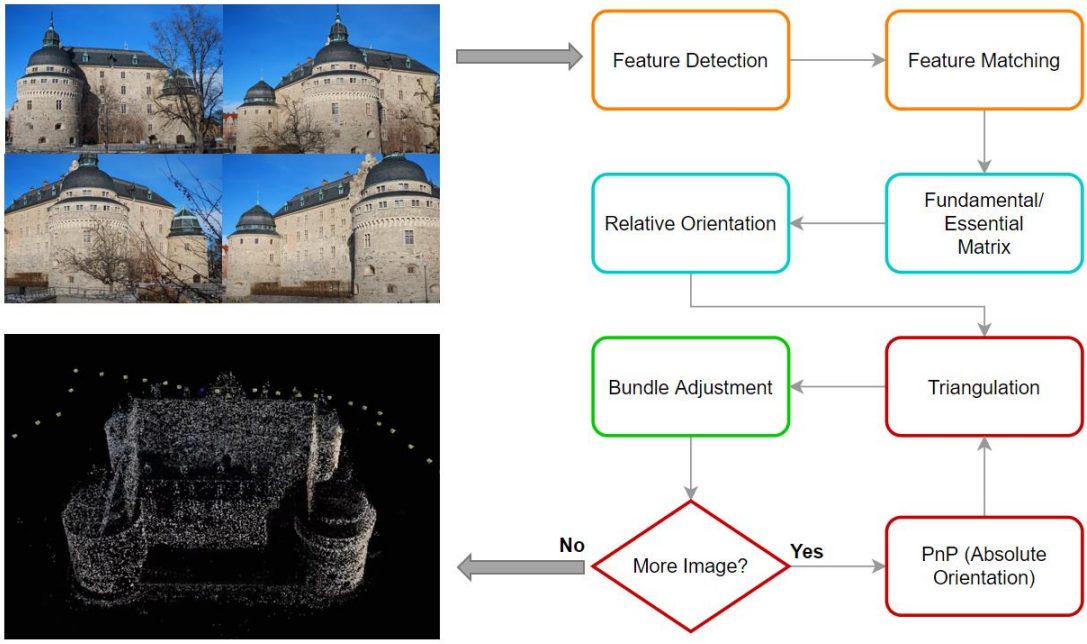
- Bước 2: Camera Registration

  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).

Tuy nhiên nhóm tập trung vào tìm hiểu Incremental SfM. Framework xử lý chung cho thuật toán Incremental SfM cũng dựa trên các bước ở trên, cụ thể các bước có trong Hình 5.



Hình 5. Framework xử lý của Incremental SfM.

Cụ thể về thuật toán Incremental SfM sẽ được trình bày ở phần sau.

Chương 2. Các Công Trình Liên Quan

Trong chương này nhóm em tập trung trình bày về phương pháp Incremental Structure from Motion (SfM) chung nhất với các bước cụ thể, sau đó sẽ nêu ra các ý tưởng, phương pháp cải tiến của 2 paper mà nhóm em chọn lọc được.

2.1. Phương pháp Incremental SfM chung

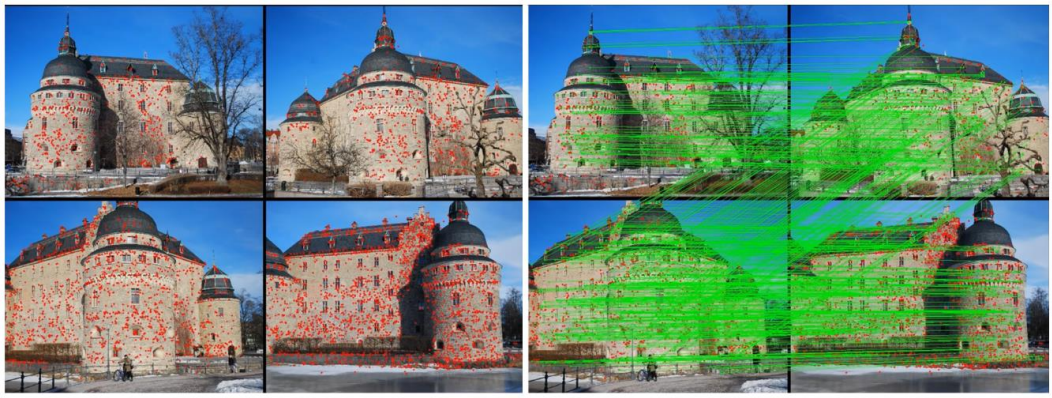
Ý tưởng chung của Incremental SfM là xuất phát từ 2 ảnh khởi tạo ban đầu, qua tính toán epipolar geometry (xác định chuyển động tương đối) ta thu được mô hình khởi tạo gồm một số điểm 3D và 2 camera làm điểm khởi đầu cho các bước tiếp. Sau đó lần lượt từng ảnh mới sẽ được thêm vào mô hình cùng với những điểm 3D mới tái tạo. Tuy nhiên ta không thể cứ thêm liên tục ảnh cho đến hết tập ảnh được. Bởi sau khi thêm một lượng ảnh nhất định lượng sai số tích lũy sẽ tăng lên, khi đó ta phải tiến hành bundle adjustment để tinh chỉnh lại mô hình gồm các điểm 3D và camera hiện tại, sau đó mới tiếp tục thêm ảnh mới vào mô hình. Sau cùng khi tất cả ảnh đã được thêm vào mô hình, ta tiến hành một bundle adjustment toàn cục để tinh chỉnh lại mô hình lần cuối.

Mục này sẽ trình bày các modules của Incremental SfM ở trên gồm: Feature Detection and Matching, Fundamental/Essential matrix, Triangulation, PnP, Bundle Adjustment.

2.1.1. Feature Detection and Matching

Đây là bước quan trọng trong SfM, tất cả các ảnh sẽ được trích xuất ra các điểm đặc trưng (features/keypoints), sử dụng SIFT hoặc SIFTGPU, ORB, BRIEF, …

Sau khi có các điểm đặc trưng thì tiến hành liên kết các điểm đặc trưng mà có cùng điểm 3D tương ứng. Bài toán tìm đặc trưng tương ứng (corresondence problem) này cũng có khá nhiều hướng giải, nhưng nhóm em không tập trung vào phần này nhiều. Kết quả của bước này là ta thu được các cặp ảnh với các cặp điểm đặc trưng tương ứng.



Hình 6. Trích xuất và kết nối đặc trưng giữa các ảnh. Source: The Struture from Motion Pipeline

2.1.2. Fundamental and Essensial Matrix

Hiện tại ta đang có các cặp ảnh với các cặp đặc trưng tương ứng (feature correspondences). Sfm sẽ chọn ra 2 ảnh ban đầu, tiêu chí chọn thường là chọn cặp ảnh nào có số lượng đặc trưng tương ứng nhiều nhất.

Ta tiến hành tính toán fundamental matrix và essential matrix từ các đặc trưng tương ứng của cặp ảnh này, mục tiêu là tìm được chuyển động tương đối giữa 2 ảnh để có thể quy tọa độ camera này về tọa độ camera còn lại.

Fundamental matrix là một ma trận 3x3 biểu diễn ràng buộc hình học giữa 2 điểm đặc trưng tương ứng (Hình 7).

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \tag{1}$$

Với 2 điểm $\mathbf{x}'$, $\mathbf{x}''$ có cùng điểm 3D tương ứng $\mathbf{X}$ thì tọa độ $\mathbf{x}' = (x', y', 1)^T$ và $\mathbf{x}'' = (x'', y'', 1)^T$ (biểu diễn dạng tọa độ đồng nhất) sẽ thỏa mãn công thức (2)

$$\mathbf{x}'^T \mathbf{F} \mathbf{x}'' = 0 \tag{2}$$

Từ công thức này ta thấy cứ một cặp điểm đặc trưng tương ứng sẽ tạo ra một ràng buộc cho ma trận fundamental

$$\begin{bmatrix} x_1' & y_1' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_1'' \\ y_1'' \\ 1 \end{bmatrix} = 0 \tag{3}$$

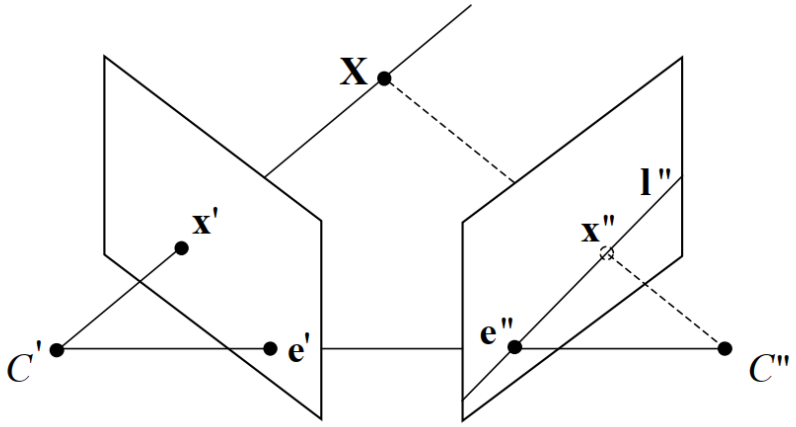
Với n cặp đặc trưng tương ứng ta thu được hệ phương trình tuyến tính mà ẩn là các thành phần của ma trận fundamental

$$\underbrace{\begin{bmatrix} x_1'x_1'' & x_1'y_1'' & x_1' & y_1'x_1'' & y_1'y_1'' & y_1' & x_1'' & y_1'' & 1 \\ & & & \vdots & & & & & \\ x_n'x_n'' & x_n'y_n'' & x_n' & y_n'x_n'' & y_n'y_n'' & y_n' & x_n'' & y_n'' & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix}}_{\mathbf{f}} = \mathbf{0} \tag{4}$$

hay

$$\mathbf{A}\mathbf{f} = \mathbf{0} \tag{5}$$

Sử dụng phương pháp SVD ta sẽ giải ra được $\mathbf{f}$ từ đó thu được ma trận fundamental $\mathbf{F}$.



Hình 7. Epipolar Geometry cho 2 camera. x', x'' và X cùng nằm trên mặt phẳng gọi là epipolar plane.

Từ ma trận fundamental $\mathbf{F}$ ta dễ dàng tính được ma trận essential $\mathbf{E}$ bởi theo điều kiện bài toán ban đầu thì camera của chúng ta đã biết trước thông số nội tại (calibrated camera), tức là ta đã có ma trận nội tại $\mathbf{K}', \mathbf{K}''$ của 2 camera

$$\mathbf{E} = \mathbf{K}'^{\mathrm{T}}\mathbf{F}\mathbf{K}'' \tag{6}$$

Phân tích ma trận $\mathbf{E}$ dùng SVD

$$\mathbf{E} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}} = \mathbf{U}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\mathbf{V}^{\mathrm{T}} \tag{7}$$

$$\mathbf{E} = \mathbf{U}\underbrace{\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\mathbf{Z}}\underbrace{\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{W}}\mathbf{V}^{\mathrm{T}} = \mathbf{U}\mathbf{Z}\mathbf{W}\mathbf{V}^{\mathrm{T}} \tag{8}$$

với $\mathbf{U}, \mathbf{V}$ là các ma trận trực giao, $\mathbf{D}$ là ma trận đường chéo.

Từ đó ta tính được ma trận đối xứng xiên (skew-symmetric matrix) $\mathbf{S}_t$ của vector dịch chuyển $\mathbf{t}$ và ma trận xoay $\mathbf{R}$. Về mặt toán học, có nhiều giá trị có thể có cho $\mathbf{S}_t$ và $\mathbf{R}$

$$\begin{aligned} \mathbf{S}_t^{(1)} &= \mathbf{U}\mathbf{Z}\mathbf{U}^{\mathrm{T}} \\ \mathbf{S}_t^{(2)} &= \mathbf{U}\mathbf{Z}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}} \\ \mathbf{R}^{(1)} &= \mathbf{U}\mathbf{W}\mathbf{V}^{\mathrm{T}} \\ \mathbf{R}^{(2)} &= \mathbf{U}\mathbf{W}^{\mathrm{T}}\mathbf{V}^{\mathrm{T}} \end{aligned} \tag{9}$$

Về mặt vật lý thì ta chỉ chọn một cặp giá trị $\mathbf{S}_t$ và $\mathbf{R}$ thỏa mãn, khi đó các điểm 3D sẽ nằm trước camera.

Từ $\mathbf{S}_t$ ta suy ra $\mathbf{t}$ được bởi công thức

$$\mathbf{S_t} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \text{ với } \mathbf{t} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix} \quad (10)$$

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

$$\begin{aligned} \mathbf{P'} &= \mathbf{K'}\begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \\ \mathbf{P''} &= \mathbf{K''}\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \end{aligned} \quad (11)$$

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

2.1.3. Triangulation

Triangulation dùng để tính toán tọa độ điểm 3D dựa vào điểm ảnh tương ứng (điểm đặc trưng) quan sát được trên các camera và ma trận chiếu của các camera.

Với các dữ kiện

$\mathbf{x}_i = \begin{bmatrix} x_i & y_i & 1 \end{bmatrix}^T$ là tọa độ điểm ảnh (tọa độ so với mặt phẳng ảnh của camera) tương ứng với **X** quan sát được trên camera thứ i

$\mathbf{P}_1 = \mathbf{K}_1\begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}$ là ma trận chiếu (3x4) của camera số 1 (ma trận này đặc biệt vì hệ tọa độ camera số 1 là hệ quy chiếu chung)

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

Cần xác định $\mathbf{X} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T$ là tọa độ điểm 3D cần tìm (tọa độ so với hệ quy chiếu camera số 1)

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 điểm **X** lên camera thứ i, điểm này sẽ bị lệch so với điểm $\mathbf{x}_i$ quan sát được vì có sai số, nhiễu trong quá trình đo. Mục tiêu là tìm **X** sao cho sai số này nhỏ nhất ở tất cả camera quan sát được **X**

Việc cực tiểu hóa sai số tương đương việc tìm sao **X** cho **AX** nhỏ nhất với

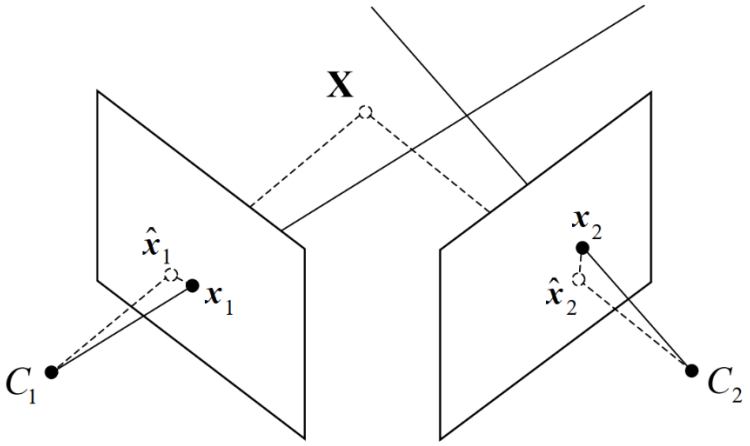
$$\mathbf{A} = \begin{bmatrix} x_1\mathbf{p}_3^{iT} - \mathbf{p}_1^{iT} \\ y_1\mathbf{p}_3^{iT} - \mathbf{p}_2^{iT} \\ \dots \\ x_N\mathbf{p}_3^{iT} - \mathbf{p}_1^{iT} \\ y_N\mathbf{p}_3^{iT} - \mathbf{p}_2^{iT} \end{bmatrix} \quad (12)$$

Dùng SVD phân tích **A**

$$\mathbf{A} = \mathbf{UDV}^{\mathrm{T}} \quad (13)$$

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

Chọn **X** là vector cột cuối cùng của **V** ứng với singular value nhỏ nhất của **A** . Khi đó **AX** sẽ nhỏ nhất.



Hình 8. Minh họa triangulation với 2 ảnh.

2.1.4. Projective-n-Points

Phương pháp **Projective-n-Points** trong bài toán Sfm dùng để tính toán vị trí của camera thứ i so với hệ tọa độ chung là camera 1 khi đã biết thông số bên trong của camera. Khi một ảnh mới được them 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. Cụ thể hơn là khi ta có được tọa độ của n điểm trong hệ tọa độ chung chứa camera thứ i và thông số bên trong của camera, phương pháp PnP có thể tính toán để cho ra kết quả là thông số bên ngoài của camera. Thông số bên trong của camera đã biết được biểu diễn dưới dạng 1 ma trận. Thông số bên ngoài của camera là ma trận xoay và vector hướng của camera so với hệ tọa độ chung.

Phương pháp P-3-P là cơ chế đơn giản hóa và được áp dụng trên phương pháp PnP.

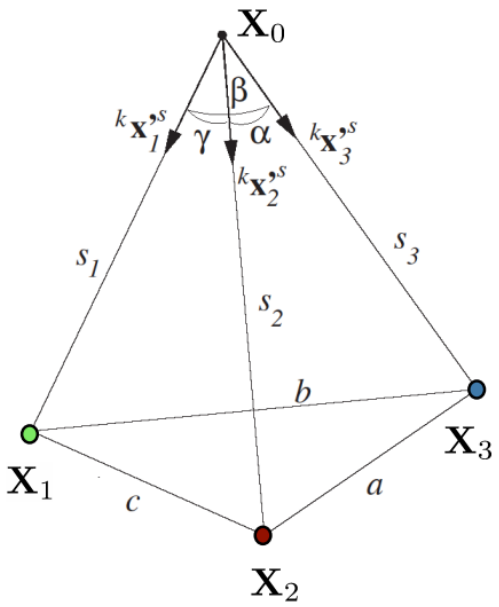
Cách giải quyết của P-3-P cụ thể là:

- Input: Tọa độ 3 điểm X_i với $1 \leq i \leq 3$ trong hệ tọa độ thực tế, tọa độ x_i trong ảnh tương ứng được ghi nhận bởi camera và các thông số bên trong của camera.
- Output: Tính toán X_0 là C_i – tâm chiếu và R là ma trận xoay của camera thứ i cần xác định vị trí, hướng tương đối so với hệ tọa độ chung.

Quá trình gồm 2 bước thực hiện chính:

Bước 1: Tính toán khoảng cách từ tâm chiếu đến 3 điểm X_i trong thực tế.

Bước 2: Tính toán hướng, ma trận xoay.



Hình 9. Source: Photogrammetry Computer Vision, Cyrill Stachniss

Trong đó:

X_1, X_2, X_3: tọa độ 3D của 3 điểm trong hệ tọa độ thực tế.

X_0: C_i – tâm chiếu của camera thứ i

${}^k x_i^s$: vector hướng từ tâm chiếu camera đến điểm X_i. Vì đây là cablirated camera nên vector này đã được xác định.

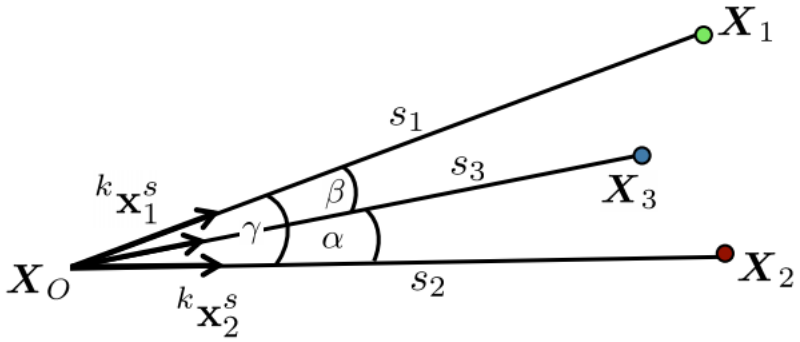
s_i: Khoảng cách từ tâm chiếu camera đến X_i.

Công thức tính tọa độ điểm X_i trong hệ tọa độ camera ${}^k X_i^s$:

$$ {}^k X_i^s = s_i \, {}^k x_i^s = R(X_i - X_O) \tag{14}$$

Tính s_i với ${}^k x_i^s$ với $i = 1,2,3$

Bước 1: Tính góc α, β, γ



Hình 10. Source: Photogrammetry Computer Vision, Cyrill Stachniss

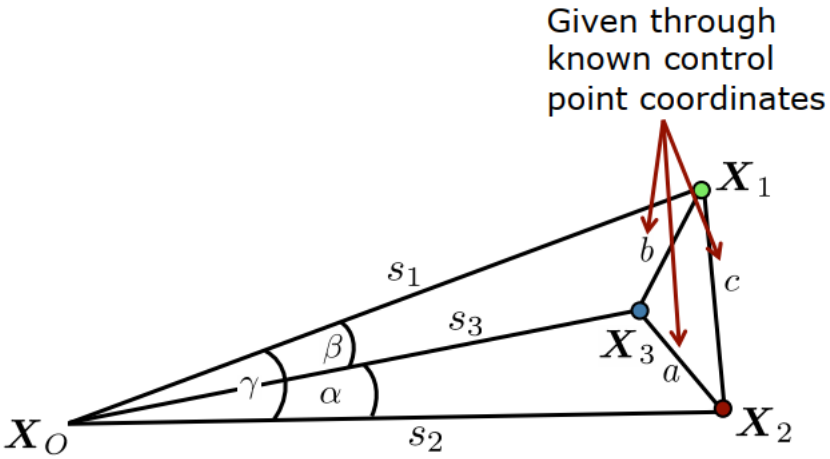
Ta có công thức tính góc giữa 2 vector $\overrightarrow{n1}, \overrightarrow{n2}$:

$$\cos(\overrightarrow{n1},\overrightarrow{n2}) = \frac{\overrightarrow{n1}.\overrightarrow{n2}}{\left|\overrightarrow{n1}\right|\left|\overrightarrow{n2}\right|} \tag{15}$$

Từ đó tính α, β, γ:

$$\begin{aligned} \alpha &= \arccos({}^{k}x_2^{s}, {}^{k}x_3^{s}) \\ \beta &= \arccos({}^{k}x_3^{s}, {}^{k}x_1^{s}) \\ \gamma &= \arccos({}^{k}x_1^{s}, {}^{k}x_2^{s}) \end{aligned} \tag{16}$$

Bước 2: Tính a, b, c



Hình 11. Source: Photogrammetry Computer Vision, Cyrill Stachniss

$$\begin{aligned} a &= \left\| X_3 - X_2 \right\| \\ b &= \left\| X_1 - X_3 \right\| \\ c &= \left\| X_2 - X_1 \right\| \end{aligned} \tag{17}$$

Bước 3: Tính s_1, s_2, s_3

Dùng công thức luật cos cho 3 tam giác, thu được

$$\begin{aligned} a^2 &= s_2^{\,2} + s_3^{\,2} - 2s_2 s_3 \cos\alpha \\ b^2 &= s_1^{\,2} + s_3^{\,2} - 2s_1 s_2 \cos\beta \\ c^2 &= s_1^{\,2} + s_2^{\,2} - 2s_1 s_2 \cos\gamma \end{aligned} \tag{18}$$

Đặt $u = \frac{s_2}{s_1}, v = \frac{s_3}{s_1}$ rồi thế vào 3 phương trình trên, thu được

$$s_1^{\,2} = \frac{a^2}{u^2 + v^2 - 2uv\cos\alpha} = \frac{b^2}{1 + v^2 - 2v\cos\beta} = \frac{c^2}{1 + u^2 - 2u\cos\gamma} \tag{19}$$

Từ đó biến đổi được thành phương trình bậc 4 của biến v

$$A_4 v^4 + A_3 v^3 + A_2 v^2 + A_1 v + A_0 = 0 \tag{20}$$

với

$$A_4 = (\frac{a^2-c^2}{b^2}-1)^2 - \frac{4c^2}{b^2}\cos^2\alpha$$

$$A_3 = a[\frac{a^2-c^2}{b^2}(1-\frac{a^2-c^2}{b^2})\cos\beta - (1-\frac{a^2+c^2}{b^2})\cos\alpha\cos\gamma + 2\frac{c^2}{b^2}\cos^2\alpha\cos\beta]$$

$$A_2 = 2[(\frac{a^2-c^2}{b^2})^2 - 1 + 2(\frac{a^2-c^2}{b^2})^2\cos^2\beta + 2(\frac{b^2-c^2}{b^2})^2\cos^2\alpha - 4(\frac{a^2+c^2}{b^2})\cos\alpha\cos\beta\cos\gamma + 2(\frac{b^2-a^2}{b^2})\cos^2\gamma]$$

$$A_1 = 4[-(\frac{a^2-c^2}{b^2})(1+\frac{a^2-c^2}{b^2})\cos\beta + \frac{2a^2}{b^2}\cos^2\gamma\cos\beta - (1-(\frac{a^2+c^2}{b^2}))\cos\alpha\cos\gamma]$$

$$A_0 = (1+\frac{a^2-c^2}{b^2})^2 - \frac{4a^2}{b^2}\cos^2\gamma$$

Vậy là tính toán được s_1, s_2, s_3.

Tính được ${}^kX_i^s$ với $0 \le i \le 3$ từ công thức (14).

Từ 3 điểm trong hệ tọa độ chung và 3 điểm trong hệ tọa độ camera, tính được rigidbody transformation và thu được R và X_O.

Từ P-3-P mở rộng cho PnP, ta cần nhiều điểm hơn để có thể đảm bảo thu được kết quả tối ưu nhất. Sử dụng phương pháp bình phương tối thiểu để điều chỉnh kết quả cho tới khi đạt được kết quả tối ưu cho R và X_O (sử dụng điểm thứ 4 để kiểm tra kết quả của 3 điểm dùng cho P-3-P).

2.1.5. Bundle Adjustment

Sau khi các ảnh mới và các điểm 3D mới lần lượt được thêm vào mô hình thì độ lỗi tích lũy cũng tăng theo. Vì thế mà sau khi thêm một lượng ảnh nhất định thì ta tiến hành bundle adjustment. Đây là thuật toán tối ưu phi tuyến để tinh chỉnh lại thông số camera (vị trí, hướng) và tọa độ các điểm 3D nhằm tối thiểu hóa sai số phép chiếu (reprojection error)

$$E(\mathbf{X},\mathbf{R},\mathbf{T}) = \sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij} L\left(\left\|\mathbf{x}_{ij} - P(\mathbf{X}_j,\mathbf{R}_i,\mathbf{t}_i)\right\|_2^2\right) \tag{21}$$

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

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

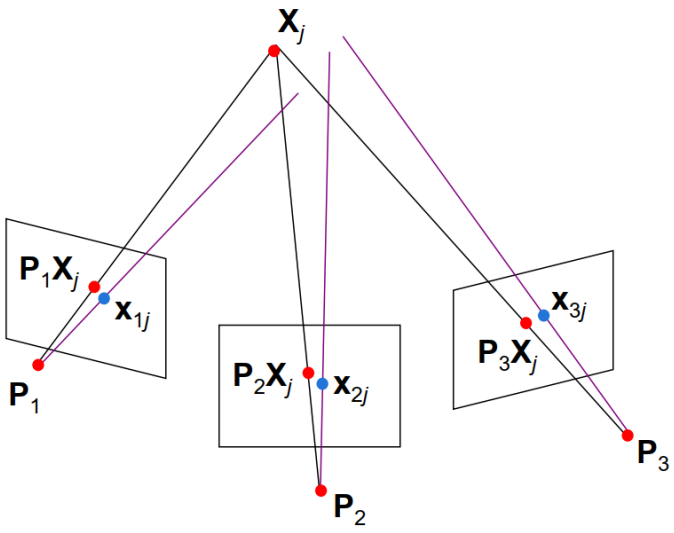
$\mathbf{x}_{ij}$ là điểm ảnh (điểm đặc trưng) của $\mathbf{X}_j$ quan sát được bởi camera thứ i

$\mathbf{R}_i, \mathbf{t}_i$ 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 (loss function)

W_{ij} bằng 1 hoặc 0 biểu thị camera thứ i có nhìn thấy điểm $\mathbf{X}_j$ không



Hình 12. Minh họa bundle adjustment. Các điểm xanh là các điểm ảnh thực tế thu nhận được. Các điểm đỏ trên mặt phẳng ảnh là các điểm chiếu từ X xuống mặt phẳng ảnh thông qua ma trận chiếu

Thuật toán tối ưu phi tuyến thường dùng là Gauss-Newton hoặc Levenberg–Marquardt. Thuật toán cần một giá trị khởi tạo giá trị **P** tốt (**P** là biến cần tối ưu), sau đó xấp xỉ tuyến tính sai số tại giá trị $\mathbf{P} + \Delta\mathbf{P}$ và tìm $\Delta\mathbf{P}$ sao cho sai số tại $\mathbf{P} + \Delta\mathbf{P}$ là nhỏ nhất. Tiến hành cập nhật $\mathbf{P} \leftarrow \mathbf{P} + \Delta\mathbf{P}$. Cứ thế lặp lại quá trình cho đến khi hội tụ ta thu được giá trị **P** cần tìm.

Thuật toán Gauss-Newton

Phần này nhóm em trình bày thuật toán Gauss-Newton cho một biến để hiểu rõ hơn về ý tưởng của thuật toán tối ưu phi tuyến.

Giả thiết có

$\mathbf{x}_i$ là các giá trị ghi nhận với trạng thái **p**

$f_i(\mathbf{p})$ là hàm ánh xạ giữa **p** với giá trị dự đoán $\hat{\mathbf{x}}_i$

Cần ước lượng trạng thái **p** mà mô tả $\mathbf{x}_i$ tốt nhất, tức là sai số giữa $\hat{\mathbf{x}}_i$ và $\mathbf{x}_i$ là nhỏ nhất.

Đầu tiên khởi tạo **p** một giá trị (giá trị này tốt thì thuật toán sẽ hội tụ nhanh). Tổng sai số tại trạng thái **p**

$$E(\mathbf{p}) = \sum_i \|\mathbf{r}_i\|^2 = \sum_i \| f_i(\mathbf{p}) - \mathbf{x}_i \|^2 \qquad (22)$$

với là $\mathbf{r}_i = \mathbf{r}_i(\mathbf{p})$ sai số trên từng giá trị ghi nhận

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

$$E(\mathbf{p} + \Delta\mathbf{p}) = \sum_i \| f_i(\mathbf{p} + \Delta\mathbf{p}) - \mathbf{x}_i \|^2 \approx \sum_i \| f_i(\mathbf{p}) + \mathbf{J}_i(\mathbf{p})\Delta\mathbf{p} - \mathbf{x}_i \|^2 \qquad (23)$$

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

Biến đổi toán học

$$
\begin{aligned}
E(\mathbf{p}+\Delta\mathbf{p}) &\approx \sum_i \left\| f_i(\mathbf{p}) - \mathbf{x}_i + \mathbf{J}_i(\mathbf{p})\Delta\mathbf{p} \right\|^2 \\
&= \sum_i \left\| \mathbf{r}_i + \mathbf{J}_i(\mathbf{p})\Delta\mathbf{p} \right\|^2 = \sum_i \left(\left\| \mathbf{r}_i \right\|^2 + 2\mathbf{r}_i^T \mathbf{J}_i \Delta\mathbf{p} + \Delta\mathbf{p}^T \mathbf{J}_i{}^T \mathbf{J}_i \Delta\mathbf{p} \right) \\
&= \underbrace{\sum_i \left\| \mathbf{r}_i \right\|^2}_{c} + 2\underbrace{\left(\sum_i \mathbf{r}_i^T \mathbf{J}_i \right)}_{\mathbf{b}^T} \Delta\mathbf{p} + \Delta\mathbf{p}^T \underbrace{\left(\sum_i \mathbf{J}_i{}^T \mathbf{J}_i \right)}_{\mathbf{H}} \Delta\mathbf{p} \\
&= c + 2\mathbf{b}^T \Delta\mathbf{p} + \Delta\mathbf{p}^T \mathbf{H} \Delta\mathbf{p}
\end{aligned}
\tag{24}
$$

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

$$\Delta\mathbf{p} = -\mathbf{H}^{-1}\mathbf{b} \tag{25}$$

Tiến hành cập nhật ta thu được trạng thái **p** mới

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p} \tag{26}$$

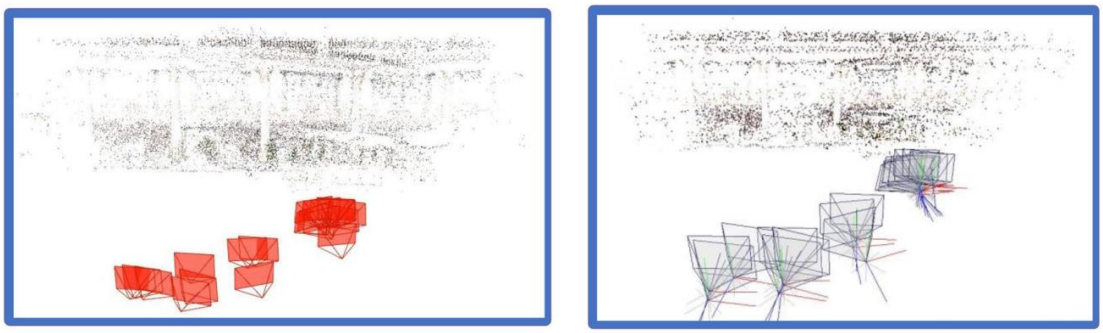
Thuật toán lại lại các bước cho đến khi hội tụ.

2.2. Tổng kết một số công trình

STT	Tên bài báo	Năm	Ý tưởng cải tiến	Dataset	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

Hình 13. Tập dataset TAB: có chứa nhiều vật thể có cấu trúc đồng dạng, lặp lại. Source: Fast incremental structure from motion based on parallel bundle

Kết quả trực quan khi chạy 2 mô hình của 2 công trình trên tập dataset TAB



(a) COLMAP **(b)** Fast-ISFM

Hình 14. Kết quả của 2 mô hình. Kết quả của Fast-ISFM có phần nhỉnh hơn, quỹ đạo camera thu được gần với quỹ đạo camera thực tế hơn . Source: Fast incremental structure from motion based on parallel bundle

Chương 3. Cài Đặt Và Thử Nghiệm

3.1. Tìm hiểu source code

Nhóm em sử dụng code của phần mềm COLMAP để tìm hiểu. Vì đây là một phần mềm khá lớn và phức tạp nên nhóm em chỉ tìm hiểu những phần quan trọng trong source code chứa trên Github của phần mềm.

Code của phần mềm chứa trong thư mục colmap/src/, trong thư mục này ta quan tâm một số thư mục sau: base, estimator, exe, feature, optim:

Base

- camera, camera_models (.h/.cc): định nghĩa các thông số của mô hình pinhole camera như id, tiêu cự, hệ số bóp méo (distortion), …
- triangulation (.h/.cc): cài đặt thuật toán triangulation.
- essential_matrix (.h/.cc): các hàm xử lý với ma trận essential, rút trích các thông số từ ma trận essential.

Estimator

- essential_matrix (.h/.cc): cài đặt thuật toán ước lượng ma trận essential.
- fundamental_matrix (.h/.cc): cài đặt thuật toán ước lượng ma trận fundamental.
- absolute_pose (.h/.cc): cài đặt thuật toán PnP ước lượng vị trí, hướng của camera.

Exe

- colmap.cc: chứa hàm main thực thi chương trình.

Feature

- sift, extraction, matching (.h/.cc): thuật toán SIFT để trích xuất đặc trưng, thuật toán kết nối các điểm đặc trưng.

Optim

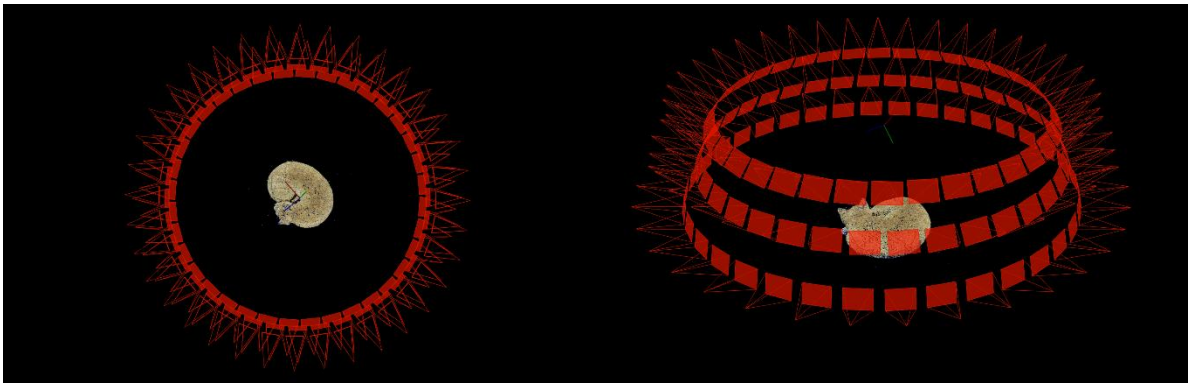
- bundle_adjustment (.h/.cc): cài đặt thuật toán bundle adjustment để tối ưu tọa độ điểm 3D và thông số camera.

3.2. Kết quả thử nghiệm

Nhóm em sử dụng phần mềm COLMAP để chạy thử nghiệm trên 3 tập dataset khác nhau.

Dataset 1

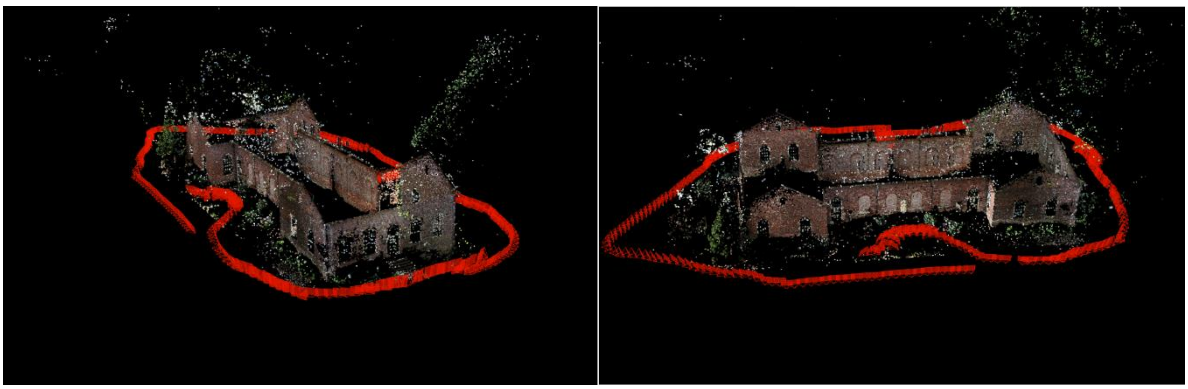
Hình 15. Một số ảnh input của dataset 1



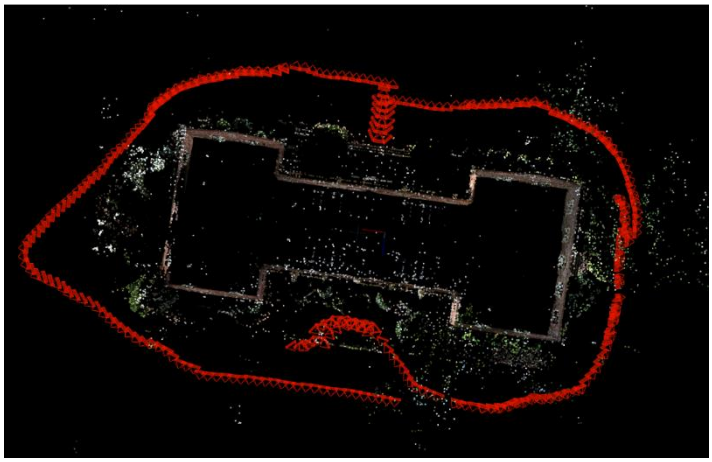
Hình 16. Kết quả nhìn từ trên cao và từ góc xiên của dataset 1

Dataset 2

Hình 17. Một số ảnh input của dataset 2



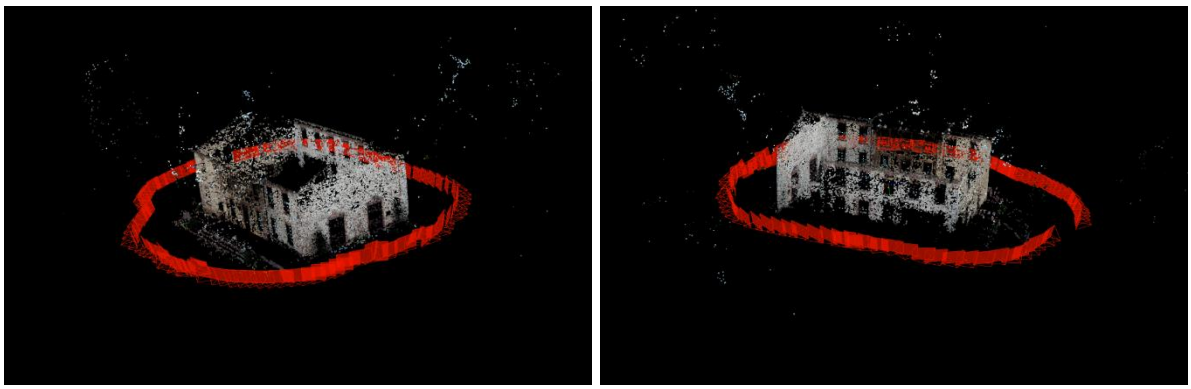
Hình 18. Kết quả nhìn từ góc xiên của dataset 2



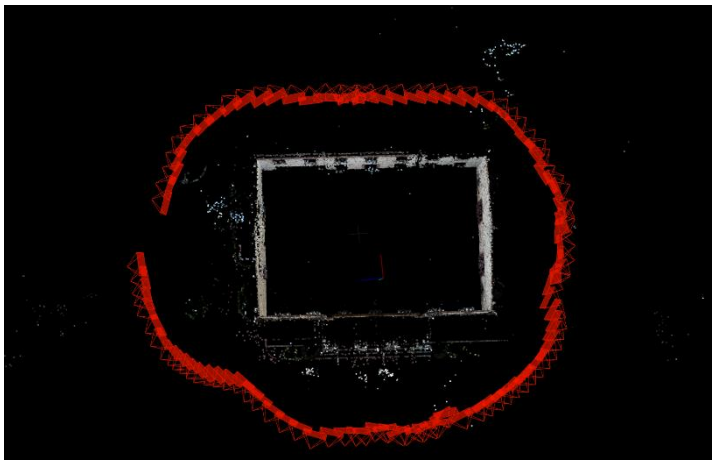
Hình 19. Kết quả nhìn từ trên cao của dataset 2

Dataset 3

Hình 20. Một số ảnh input của dataset 3



Hình 21. Kết quả nhìn từ góc xiên của dataset 3



Hình 22. Kết quả nhìn từ trên cao của dataset 3

Tài liệu tham khảo

[1] Schönberger, J.L., Frahm, J.-M.: Structure-from-motion revisited. CVPR (2016).

[2] Qi Hu1, Jianxin Luo1, Guyu Hu1, Weiwei Duan1, Hui Zhou2: 3D Point Cloud Generation Using Incremental Structure-from-Motion (2018).

[3] Mingwei Cao, Liping Zheng, Wei Jia, Xiaoping Liu: Fast incremental structure from motion based on parallel bundle adjustment (2020).

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

[5] Wolfgang FörstnerBernhard P. Wrobel, Photogrammetric Computer Vision (2016).

[6] Cyrill Stachniss, Photogrammetry Computer Vision (2020).

[7] Cyrill Stachniss, Photogrammetry II (2020).