Structure From Motion

Chul Min Yeum
Assistant Professor
Civil and Environmental Engineering
University of Waterloo, Canada

CIVE 497 – CIVE 700: Smart Structure Technology

Last updated: 2021-03-19

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING

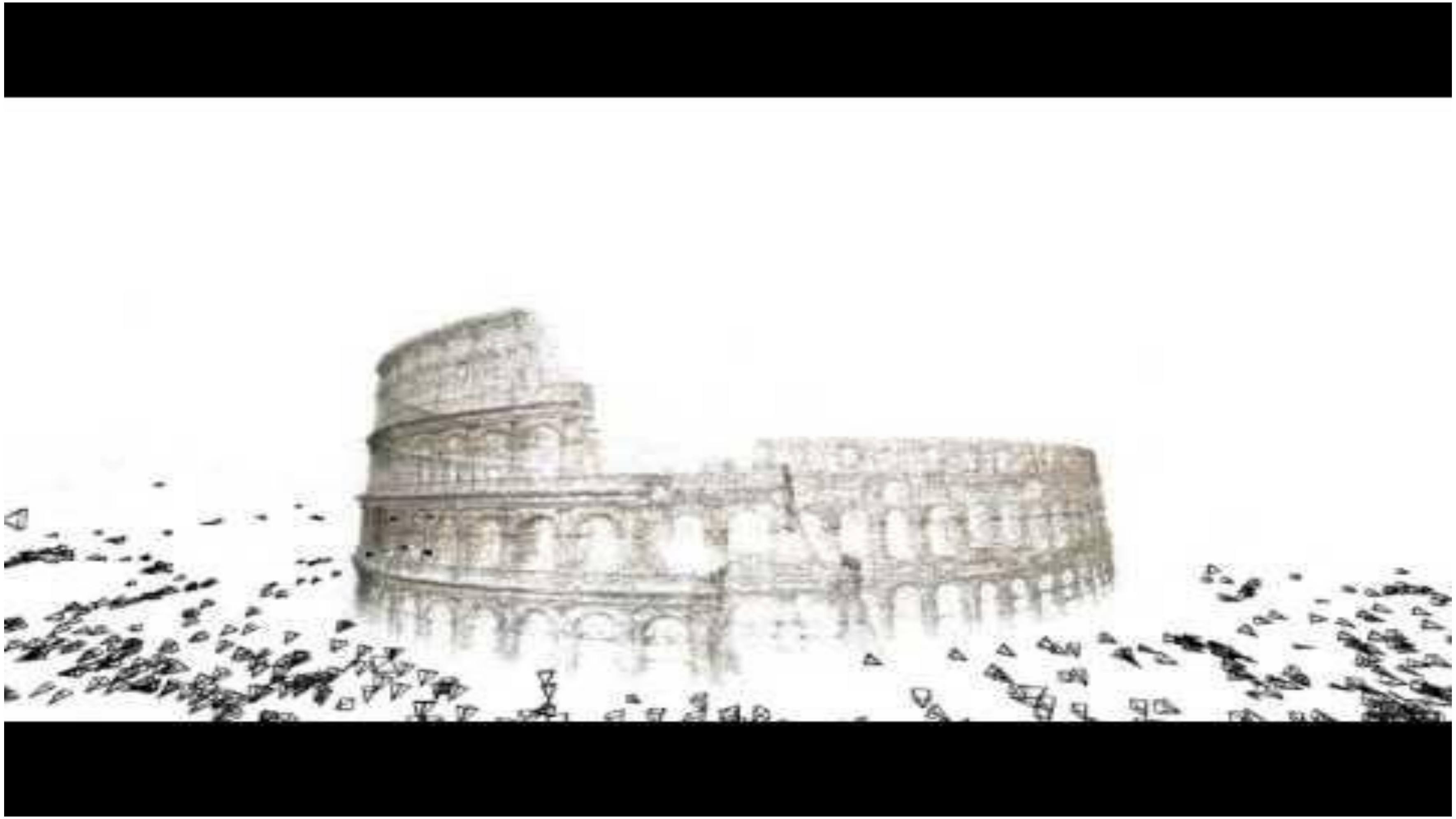
What is Structure from Motion (SfM) ?

Pictures

Scene structure & camera locations
and parameters

http://www.cs.cornell.edu/projects/bigsfm/

Example: BigSfM - Reconstructing the World from Internet Photos



http://www.cs.cornell.edu/projects/bigsfm/

Example: RESCUAV in Globalmedic

https://globalmedic.ca/rescuav/

Projection Matrix

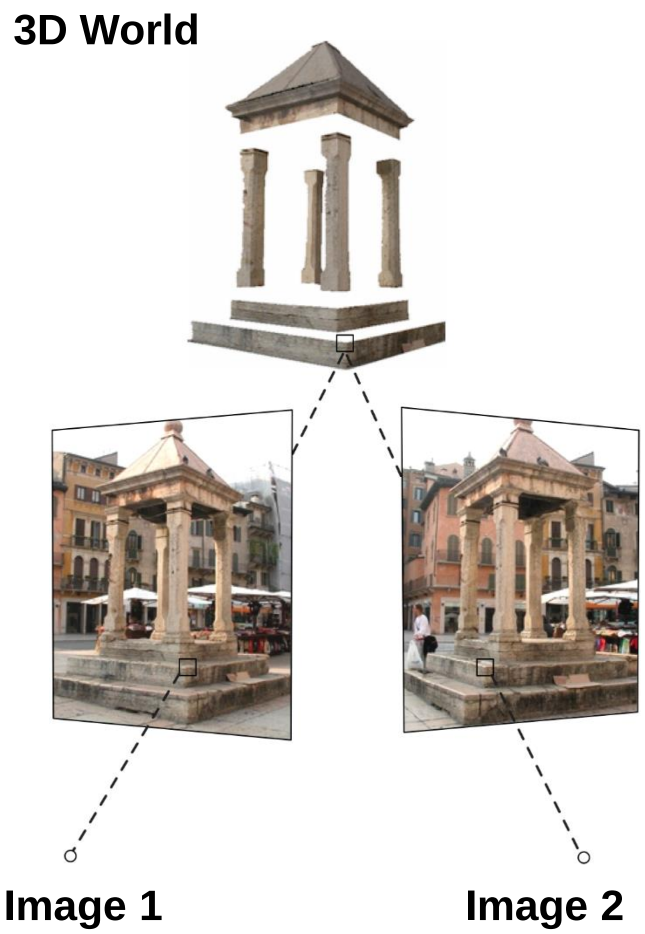
i : Image number

$$\mathbf{x} = \mathbf{P_i X}$$

2D point **3D point**

- **If we knew a projective matrix in each image, we can compute the image point corresponding to the world point**
- **If we knew more than two image points indicating same world point, we can compute the location of the world point. (triangulation)**

FAQ in Camera (Multi-view) Geometry



Q1. Can we compute a 3D location of a single image point?

Q2. Why do we need to know a camera model?

Q3. Can we compute a 2D point on a image if we know 3D points?

Q4. Can we measure a real-distance from two images?

Q5. What is the role of GPS data?

Triangulation Methods (3D Position from 2D Points on Images)

$$\mathbf{x} = \mathbf{P}\mathbf{X}$$

$$\mathbf{x}' = \mathbf{P}'\mathbf{X}$$

$$\begin{bmatrix} \mathbf{P}_1^{\mathbf{T}} - \mathbf{u}\mathbf{P}_3^{\mathbf{T}} \\ \mathbf{P}_2^{\mathbf{T}} - \mathbf{v}\mathbf{P}_3^{\mathbf{T}} \\ \mathbf{P}_1'^{\mathbf{T}} - \mathbf{u}'\mathbf{P}_3'^{\mathbf{T}} \\ \mathbf{P}_2'^{\mathbf{T}} - \mathbf{v}'\mathbf{P}_3'^{\mathbf{T}} \end{bmatrix} \mathbf{X} = \mathbf{A}\mathbf{X} = 0$$

Find a null vector of A when Ax=0

Suppose that $x = PX = \begin{bmatrix} p_1^T \\ p_2^T \\ p_3^T \end{bmatrix} X = \begin{bmatrix} p_1^T X \\ p_2^T X \\ p_3^T X \end{bmatrix} = \begin{bmatrix} uw \\ vw \\ w \end{bmatrix}$

Where p_i^T is the i th row of the matrix P.

u, v : pixel location $\begin{bmatrix} uw \\ vw \\ w \end{bmatrix}$ is a point in HC.

$uw = p_1^T X$ $\quad$ $vw = p_2^T X$

$u p_3^T X = p_1^T X$ $\quad$ $v p_3^T X = p_2^T X$

$(p_1^T - u p_3^T) X = 0$ $\quad$ $(p_2^T - v p_3^T) X = 0$

$$\begin{bmatrix} p_1^T - u p_3^T \\ p_2^T - v p_3^T \\ \boxed{p_1'^T - u' p_3'^T} \\ \boxed{p_2'^T - u' p_3'^T} \end{bmatrix} \begin{bmatrix} X \end{bmatrix} = 0$$

from the second image.

X: 4x1 vector

Example: Triangulation Methods

```
% synthetic projection matrix creation
P1 = [eye(3,3) zeros(3,1)];
P2 = eye(3,3)*[rotx(10)*roty(20)*rotz(30) [5;5;1]];

% synthetic 100 numbers of 3D points (X)
X = rand(4, 1);

% images points corresponding to each X
x1 = P1*X; x1 = x1(1:2)./x1(3);
x2 = P2*X; x2 = x2(1:2)./x2(3);

% triangulartion
A = zeros(4,4);
A(1,:) = x1(1)*P1(3,:) - P1(1,:);
A(2,:) = x1(2)*P1(3,:) - P1(2,:);
A(3,:) = x2(1)*P2(3,:) - P2(1,:);
A(4,:) = x2(2)*P2(3,:) - P2(2,:);

X_comp = null(A); % computed X using null

% or you can use the last column vector of V
[U,D,V] = svd(A);
% X_comp = V(:,4); % computed X using null
% check if the last diagonal element of D becomes 0.

% compare the following two results: They are the same.
X(1:3)/X(4)
X_comp(1:3)/X_comp(4)
```

```
ans = 3×1
    0.0665
    0.1398
    1.1851

ans = 3×1
    0.0665
    0.1398
    1.1851
```

```
% synthetic projection matrix creation
P1 = [eye(3,3) zeros(3,1)];
P2 = eye(3,3)*[rotx(10)*roty(20)*rotz(30) [5;5;1]];

% synthetic 100 numbers of 3D points (X)
X = rand(4, 1);

% images points corresponding to each X
x1 = P1*X; x1 = x1(1:2)./x1(3);
x2 = P2*X; x2 = x2(1:2)./x2(3);

% triangulartion
A = zeros(4,4);
A(1,:) = x1(1)*P1(3,:) - P1(1,:);
A(2,:) = x1(2)*P1(3,:) - P1(2,:);
A(3,:) = x2(1)*P2(3,:) - P2(1,:);
A(4,:) = (x2(2)+0.001)*P2(3,:) - P2(2,:); % let's add noise on X2

% computed X using null
null(A)

% it says no null vector because it becomes a full rank matrix due
% added.

% Then, you need to use SVD
[U,D,V] = svd(A);
D
% If you see the last diagonal element is very small value, this
% last column vector of V becoms the best approximated null vecto

X_comp = V(:,end);

% A times an approximated null vector
A*X_comp

% compare the following two results: They are very similar
X(1:3)/X(4)
X_comp(1:3)/X_comp(4)
```

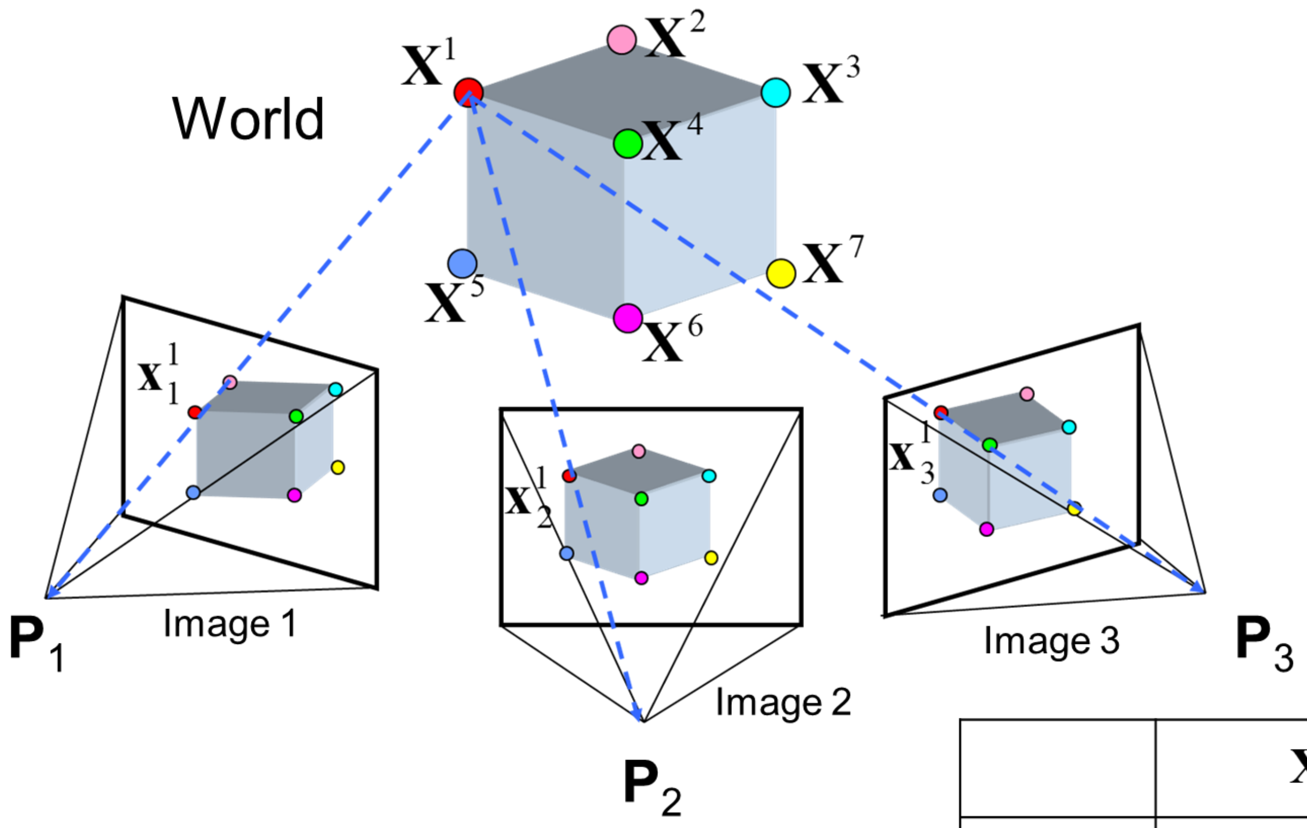
```
ans =

  4×0 empty double matrix
```

There is no a null vector

```
D = 4x4
    5.7748         0         0         0
         0    1.4448         0         0
         0         0    0.7589         0
         0         0         0    0.0003

ans = 4x1
10^-3 ×
   -0.1125
    0.2428
    0.1616
   -0.1176

ans = 3x1
    1.0301
    0.0277
    0.8462

ans = 3x1
    1.0287
    0.0281
    0.8453
```

Close to zero

Multiview Geometry (More than Two Images)



Input
Observed 2D image position

Output
Unknown Camera Parameters (with some guess)
Unknown Point 3D coordinate (with some guess)

	$\mathbf{X}^1$	$\mathbf{X}^2$	....	$\mathbf{X}^M$
Image 1	$\mathbf{x}_1^{\ 1} = \mathbf{P}_1\mathbf{X}^1$		....	$\mathbf{x}_1^{\ M} = \mathbf{P}_1\mathbf{X}^M$
Image 2		$\mathbf{x}_2^{\ 2} = \mathbf{P}_2\mathbf{X}^2$	....	$\mathbf{x}_2^{\ M} = \mathbf{P}_2\mathbf{X}^M$
⋮	⋮	⋮	⋱	⋮
Image N	$\mathbf{x}_N^{\ 1} = \mathbf{P}_N\mathbf{X}^1$	$\mathbf{x}_N^{\ 2} = \mathbf{P}_N\mathbf{X}^2$	....	$\mathbf{x}_N^{\ M} = \mathbf{P}_N\mathbf{X}^M$

Incremental SFM

Pick a pair of images with lots of feature inliers (and preferably, good EXIF data)

- Initialize intrinsic parameters (focal length, principal point) from EXIF or use of calibrate cameras
- Estimate extrinsic parameters (**R** and **t**) using eight-point or five-point algorithm from fundamental or essential matrix
- Use triangulation to initialize model points {**X**}

While remaining images exist

- Find an image with many feature matches with images in the model
- Estimate extrinsic parameters for a new images
- Run RANSAC on feature matches to register new image to model
- Triangulate new points
- Perform bundle adjustment to re-optimize everything

Observation

$$
\begin{array}{ccc}
\tilde{\mathbf{x}}_1^1 & \tilde{\mathbf{x}}_1^2 & \\
\tilde{\mathbf{x}}_2^1 & \tilde{\mathbf{x}}_2^2 & \tilde{\mathbf{x}}_2^3 \\
\tilde{\mathbf{x}}_3^1 & & \tilde{\mathbf{x}}_3^3
\end{array}
$$

Features matching

=

Re-projection

$\mathbf{x}_1^1 = \mathbf{P}_1\mathbf{X}^1$		....	$\mathbf{x}_1^M = \mathbf{P}_1\mathbf{X}^M$
	$\mathbf{x}_2^2 = \mathbf{P}_2\mathbf{X}^2$	....	$\mathbf{x}_2^M = \mathbf{P}_2\mathbf{X}^M$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\mathbf{x}_N^1 = \mathbf{P}_N\mathbf{X}^1$	$\mathbf{x}_N^2 = \mathbf{P}_N\mathbf{X}^2$	....	$\mathbf{x}_N^M = \mathbf{P}_N\mathbf{X}^M$

$$\boxed{\min \sum_i \sum_j \left(\tilde{\mathbf{x}}_i^{\,j} - \mathbf{P}_i \mathbf{X}^j \right)^2}$$

Optimization problem

A valid solution must let the re-projection close to the observation.

What Makes This Problem Challenging?

- Not enough overlaps across the images
- Not enough features on the scene in the world
- Wrong or inaccurate matching
- $O(N^2)$ complexity (matching)

Problem Statement: Image-based Measurement

i : Image number

$$\mathbf{x} = \mathbf{P_i X}$$

Input Input Output

Obtain the projective matrices from SfM software

Slide Credits and References

- Lecture notes: JianXiong Xiao. “Multi-view 3D Reconstruction for Dummies”. Princeton Vision Group
- CVPR 2015 Tutorial: SfM Pipelines
- http://vision.princeton.edu/courses/SFMedu/
- http://cs.brown.edu/courses/cs143/
- http://people.csail.mit.edu/torralba/courses/6.869/6.869.computervision.htm
- http://www.cs.utexas.edu/~grauman/courses/fall2009/schedule.htm
- http://graphics.cs.cmu.edu/courses/15-463/2010_spring/463.html
- https://courses.engr.illinois.edu/cee598vsc/sp2015/lecturenotes/
- VisualSfM: http://ccwu.me/vsfm/doc.html
- Pix4D: https://support.pix4d.com/hc/en-us/sections/200591059-Manual#gsc.tab=0
- https://slazebni.cs.illinois.edu/spring19/lec17_sfm.pdf