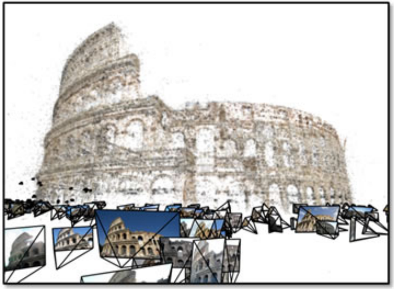
CS325 Artificial Intelligence
Computer Vision III – Structure from Motion (Ch. 24)

Dr. Cengiz Günay, Emory Univ.



Spring 2013

Structure from Motion

What??

Structure from Motion

What??

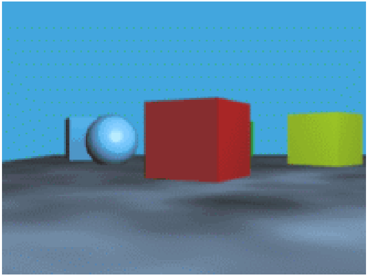
Structure: 3D information

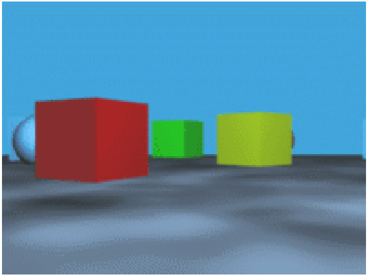
Motion: Camera motion

What??

Structure: 3D information

Motion: Camera motion

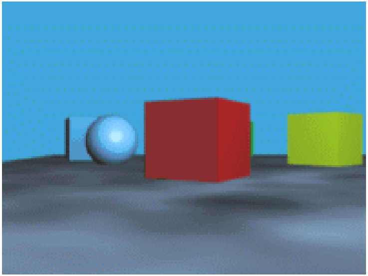


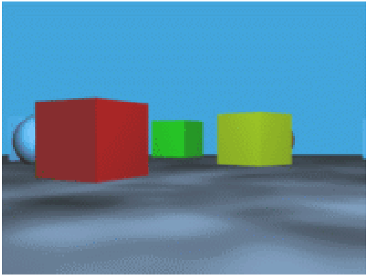


What??

Structure: 3D information

Motion: Camera motion





Looks familiar?

Entry/Exit Surveys

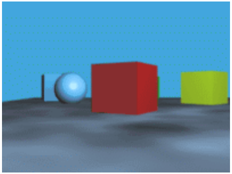
Exit survey: Computer Vision II – 3D Vision

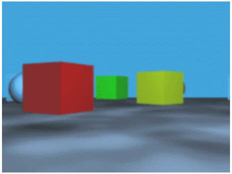
- Why don't we need to know the original object's size when we have stereo vision?
- What's the operating principle of the XBOX Kinect (R) motion tracker system?

Entry survey: Computer Vision III – Structure from Motion (0.25 pts)

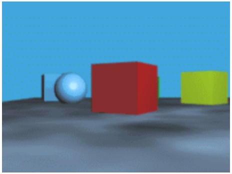
- Can you think of a way to apply the 3D vision alignment algorithms from last class for extracting structure from motion (SfM)?
- What would be a good application area for SfM?

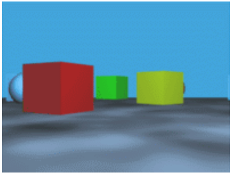
Triangulate from Camera Positions

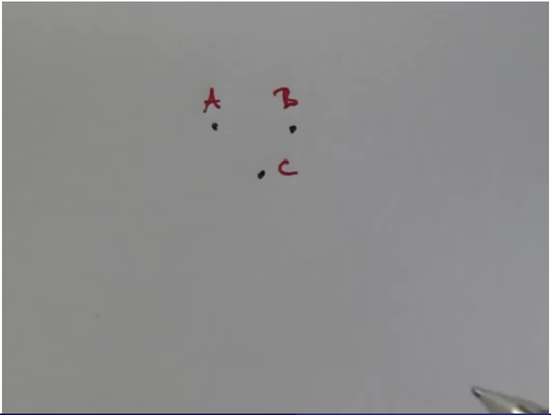




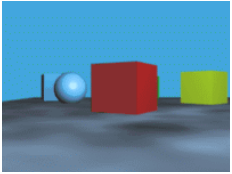
Triangulate from Camera Positions

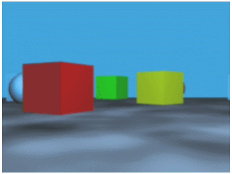


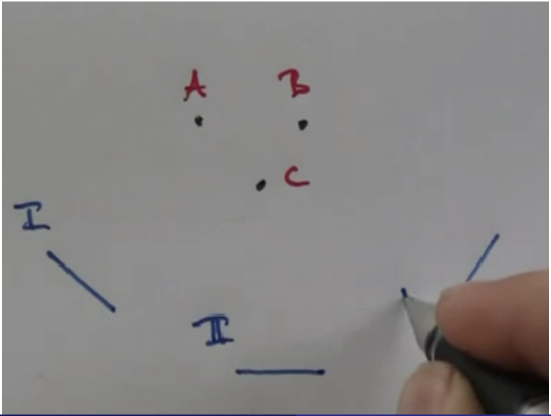




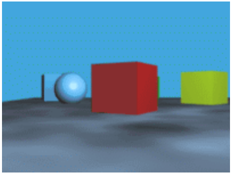
Triangulate from Camera Positions

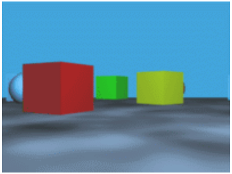


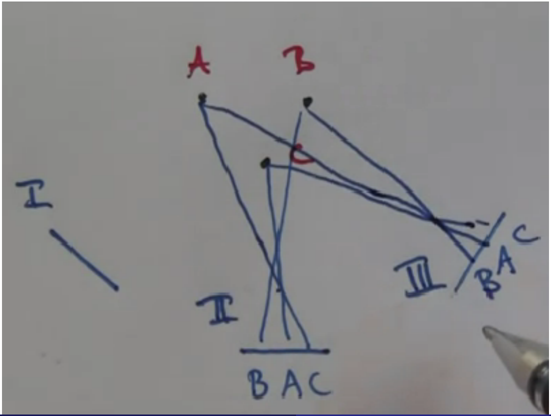




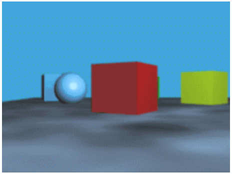
Triangulate from Camera Positions

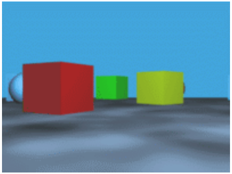


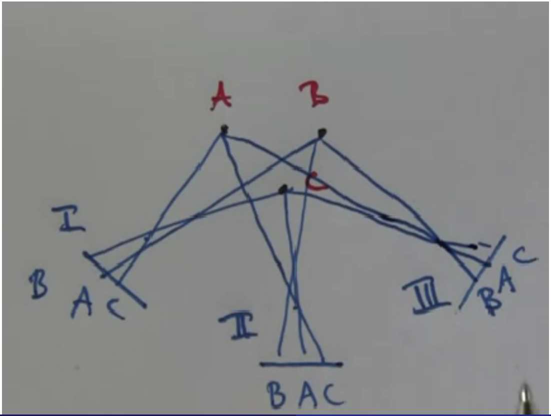




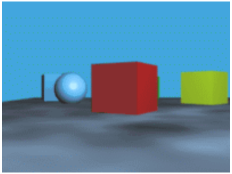
Triangulate from Camera Positions

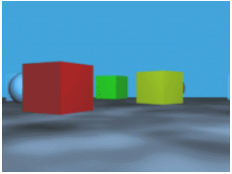


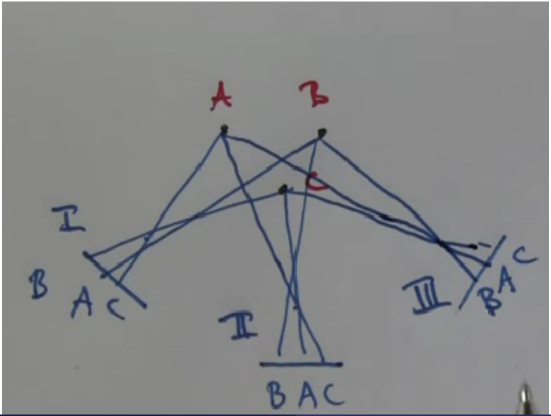




Triangulate from Camera Positions



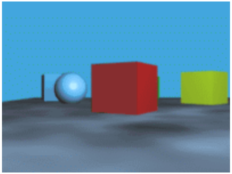


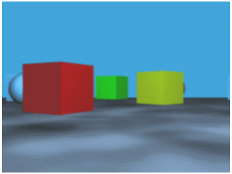


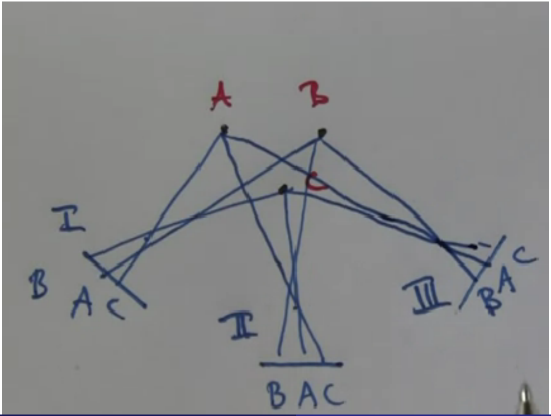
Can we find locations of A, B, C?

1. Always
2. Sometimes
3. Never

Triangulate from Camera Positions



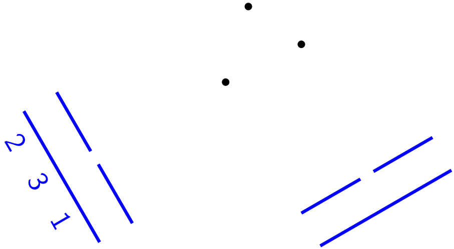




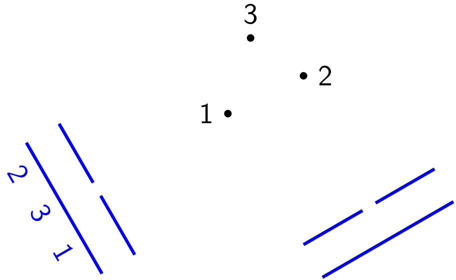
Can we find locations of A, B, C?

1. Always
2. **Sometimes**
3. Never

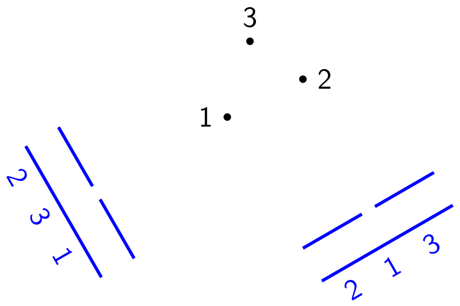
Example with Two Cameras



Example with Two Cameras



Example with Two Cameras



Demos:

[Tomasi & Kanade 92]

SfM Examples: 3D Reconstruction From Snapshots

Lots of examples on the Wikipedia page:

- A Fountain
- Duomo of Pisa
- An alley
- Dots and texture

SfM is Also Called "Camera Tracking"

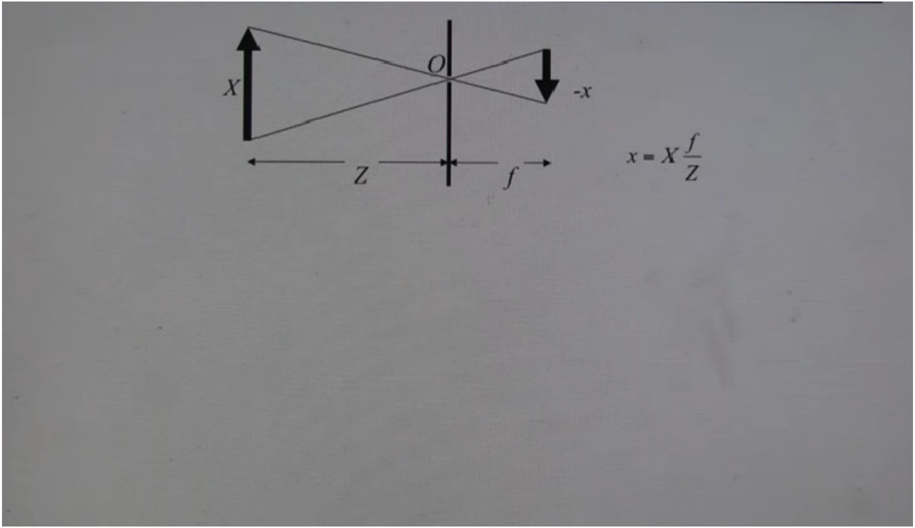
Nowadays, it is even available in open-source programs:

- Blender 3D modeling software:
  see video of its camera tracking plugin
- More on the Wikipedia page

So How Does SfM Work?

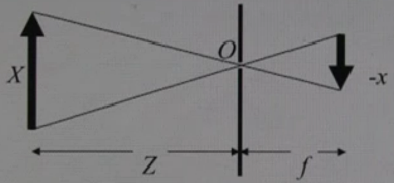
So How Does SfM Work?

Here's the math:



So How Does SfM Work?

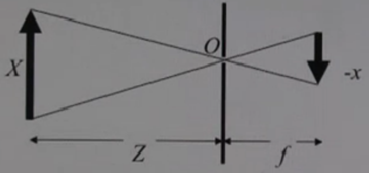
Here's the math:



$$x = X\frac{f}{Z}$$

$$\begin{pmatrix} p_{x,j} \\ p_{y,j} \end{pmatrix} = f\,\frac{\begin{pmatrix} \cos\phi_i & \sin\phi_i & 0 \\ -\sin\phi_i & \cos\phi_i & 0 \end{pmatrix}\begin{pmatrix} \cos\varphi_i & 0 & \sin\varphi_i \\ 0 & 1 & 0 \\ -\sin\varphi_i & 0 & \cos\varphi_i \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi_i & \sin\psi_i \\ 0 & -\sin\psi_i & \cos\psi_i \end{pmatrix}\begin{pmatrix} P_{x,j} \\ P_{y,j} \\ P_{z,j} \end{pmatrix} + \begin{pmatrix} b_{x,i} \\ b_{y,i} \end{pmatrix}}{\begin{pmatrix} 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \cos\varphi_i & 0 & \sin\varphi_i \\ 0 & 1 & 0 \\ -\sin\varphi_i & 0 & \cos\varphi_i \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi_i & \sin\psi_i \\ 0 & -\sin\psi_i & \cos\psi_i \end{pmatrix}\begin{pmatrix} P_{x,j} \\ P_{y,j} \\ P_{z,j} \end{pmatrix} + b_{z,i}}$$

Here's the math:



$$x = X\frac{f}{Z}$$

$$\sum_{i,j}\left(\begin{pmatrix}p_{x,j}\\ p_{y,j}\end{pmatrix} - f\,\frac{\begin{pmatrix}\cos\phi_i & \sin\phi_i & 0\\ -\sin\phi_i & \cos\phi_i & 0\end{pmatrix}\begin{pmatrix}\cos\varphi_i & 0 & \sin\varphi_i\\ 0 & 1 & 0\\ -\sin\varphi_i & 0 & \cos\varphi_i\end{pmatrix}\begin{pmatrix}1 & 0 & 0\\ 0 & \cos\psi_i & \sin\psi_i\\ 0 & -\sin\psi_i & \cos\psi_i\end{pmatrix}\begin{pmatrix}P_{x,j}\\ P_{y,j}\\ P_{z,j}\end{pmatrix}+\begin{pmatrix}b_{x,i}\\ b_{y,i}\end{pmatrix}}{\begin{pmatrix}0 & 0 & 1\end{pmatrix}\begin{pmatrix}\cos\varphi_i & 0 & \sin\varphi_i\\ 0 & 1 & 0\\ -\sin\varphi_i & 0 & \cos\varphi_i\end{pmatrix}\begin{pmatrix}1 & 0 & 0\\ 0 & \cos\psi_i & \sin\psi_i\\ 0 & -\sin\psi_i & \cos\psi_i\end{pmatrix}\begin{pmatrix}P_{x,j}\\ P_{y,j}\\ P_{z,j}\end{pmatrix}+b_{z,i}}\right)^2 \to \min$$

SfM Math

Non-linear least-squares optimization problem:

- Gradient descent
- Conjugate gradient
- Gauss Newton methods (e.g., Levenberg-Marquardt)
- Singular Value Decomposition (e.g., PCA)

How Many of Those Parameters Can We Recover?

Let's assume we have

- m camera poses
- n points to recover

How Many of Those Parameters Can We Recover?

Let's assume we have

m camera poses

n points to recover

Each camera pose has 6 parameters:

- 3 for x, y, z
- 3 for pointing angle α, β, ϕ

Each point has 3 parameters:

- 3 for x, y, z

How Many of Those Parameters Can We Recover?

Let's assume we have

m camera poses

n points to recover

Each camera pose has 6 parameters:

- 3 for x, y, z
- 3 for pointing angle α, β, ϕ

Each point has 3 parameters:

- 3 for x, y, z

Total:

- **Unknown parameters: $6m + 3n$.**
- **Constraints from 2D images: $2nm$**

How Many of Those Parameters Can We Recover?

Let's assume we have

m camera poses

n points to recover

Each camera pose has 6 parameters:

- 3 for x, y, z
- 3 for pointing angle α, β, ϕ

Each point has 3 parameters:

- 3 for x, y, z

Total:

- **Unknown parameters:** $6m + 3n$.
- **Constraints from 2D images:** $2nm$

To recover all points, must satisfy:

How Many of Those Parameters Can We Recover?

Let's assume we have

m camera poses

n points to recover

Each camera pose has 6 parameters:

- 3 for x, y, z
- 3 for pointing angle α, β, ϕ

Each point has 3 parameters:

- 3 for x, y, z

Total:

- **Unknown parameters:** $6m + 3n$.
- **Constraints from 2D images:** $2nm$

To recover all points, must satisfy: $6m + 3n \leq 2nm$

How Many of Those Parameters Can We Recover?

Let's assume we have

m camera poses

n points to recover

Each camera pose has 6 parameters:

- 3 for x, y, z
- 3 for pointing angle α, β, ϕ

Each point has 3 parameters:

- 3 for x, y, z

Total:

- **Unknown parameters:** $6m + 3n$.
- **Constraints from 2D images:** $2nm$

To recover all points, must satisfy: $6m + 3n \leq 2nm$

Which parameters can't we recover at all?

How Many of Those Parameters Can We Recover?

Let's assume we have

m camera poses

n points to recover

Each camera pose has 6 parameters:

- 3 for x, y, z
- 3 for pointing angle α, β, ϕ

Each point has 3 parameters:

- 3 for x, y, z

Total:

- **Unknown parameters:** $6m + 3n$.
- **Constraints from 2D images:** $2nm$

To recover all points, must satisfy: $6m + 3n \leq 2nm$

Which parameters can't we recover at all?

- Absolute frame of reference (x, y, z)

How Many of Those Parameters Can We Recover?

Let's assume we have

m camera poses

n points to recover

Each camera pose has 6 parameters:

- 3 for x, y, z
- 3 for pointing angle α, β, ϕ

Each point has 3 parameters:

- 3 for x, y, z

Total:

- **Unknown parameters:** $6m + 3n$.
- **Constraints from 2D images:** $2nm$

To recover all points, must satisfy: $6m + 3n \leq 2nm$

Which parameters can't we recover at all?

- Absolute frame of reference (x, y, z)
- Absolute orientation angle (α, β, ϕ)

How Many of Those Parameters Can We Recover?

Let's assume we have

m camera poses

n points to recover

Each camera pose has 6 parameters:

- 3 for x, y, z
- 3 for pointing angle α, β, ϕ

Each point has 3 parameters:

- 3 for x, y, z

Total:

- **Unknown parameters:** $6m + 3n$.
- **Constraints from 2D images:** $2nm$

To recover all points, must satisfy: $6m + 3n \leq 2nm$

Which parameters can't we recover at all?

- Absolute frame of reference (x, y, z)
- Absolute orientation angle (α, β, ϕ)
- Scale

How About Motion of Subjects?

How About Motion of Subjects?

Subjects assumed to be **static**.

How About Motion of Subjects?

Subjects assumed to be **static**.

Can we also recover structure of moving subjects? (*Ask your neighbor*)

1. Always
2. Sometimes
3. Never

How About Motion of Subjects?

Subjects assumed to be **static**.

Can we also recover structure of moving subjects? (*Ask your neighbor*)

1. Always
2. **Sometimes**
3. Never

Remember?

How About Motion of Subjects?

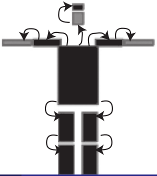
Subjects assumed to be **static**.

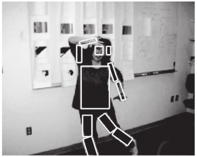
Can we also recover structure of moving subjects? (*Ask your neighbor*)

1. Always
2. **Sometimes**
3. Never

Remember?

Need to have models of objects:





How Does the Brain Do It?

SfM converts

- From camera images:
- To object locations:

How Does the Brain Do It?

SfM converts

- From camera images: **Egocentric** or viewer-centered representation
- To object locations:

How Does the Brain Do It?

SfM converts

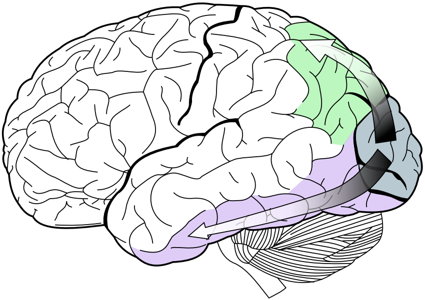
- From camera images: **Egocentric** or viewer-centered representation
- To object locations: **Allocentric** or object-centered representation

How Does the Brain Do It?

SfM converts

- From camera images: **Egocentric** or viewer-centered representation
- To object locations: **Allocentric** or object-centered representation

The brain has **two separate visual pathways** for these:



Ventral is allocentric and dorsal is egocentic. Read more here.