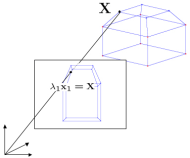
Symmetry-based 3-D Reconstruction from Perspective Images

Allen Y. Yang <yang@eecs.berkeley.edu>
with Yi Ma, Wei Hong, and Kun Huang

June 23, 2008. CVPR Tutorial

Notation

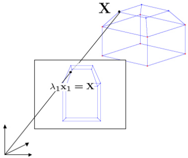
1. Perspective projection in **single views**: Loss of depth information



$$\lambda_1 \mathbf{x}_1 = X \in \mathbb{R}^3, \text{ where } \mathbf{x}_1 = [x, y, 1]^T.$$

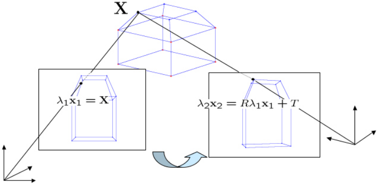
Notation

1. Perspective projection in **single views**: Loss of depth information



$$\lambda_1 \mathbf{x}_1 = X \in \mathbb{R}^3, \text{ where } \mathbf{x}_1 = [x, y, 1]^T.$$

2. 3-D reconstruction from **two views**



$$\mathbf{x}_2^T \widehat{T} R \mathbf{x}_1 = 0.$$

Multiple-View Geometry

Epipolar constraint is the **epitome** of multiple-view geometry/structure-from-motion:

$$\mathbf{x}_2^T \widehat{T} R \mathbf{x}_1 = 0.$$

- Rigid-object structure: for $\mathbf{X}$.
- Camera calibration: for $(\mathbf{x}_1, \mathbf{x}_2)$.
- Camera motion: for (R, T).

Image courtesy of Jana Kosecka.

Multiple Views Are Not Always Necessary

Figure: Perception of depth in single view.

3-D Reconstruction based on Symmetry

Human vision can perceive depth information from single view images

- Because of symmetric (planar) structures that are abundant in urban environments.

Symmetry Induces Visual Illusion

(a) Ames' Room

(b) Escher's Waterfall

Illusions in Art

When symmetry is wrongfully (and skillfully) applied, human vision leads to ambiguous 3-D reconstruction even contradicting the common sense learned in the brain.

Outlines

1. Three symmetric transformations: **Reflection**, **rotation**, and **translation**.

Outlines

1. Three symmetric transformations: **Reflection**, **rotation**, and **translation**.
2. Theory of symmetry group to describe the transformation between symmetric structures.

Outlines

1. Three symmetric transformations: **Reflection**, **rotation**, and **translation**.
2. Theory of symmetry group to describe the transformation between symmetric structures.
3. Extend multiple-view geometry to recovering pose/structure of symmetric objects.

Outlines

1. Three symmetric transformations: **Reflection**, **rotation**, and **translation**.
2. Theory of symmetry group to describe the transformation between symmetric structures.
3. Extend multiple-view geometry to recovering pose/structure of symmetric objects.
4. Demonstrate applications.

Outlines

1. Three symmetric transformations: **Reflection**, **rotation**, and **translation**.
2. Theory of symmetry group to describe the transformation between symmetric structures.
3. Extend multiple-view geometry to recovering pose/structure of symmetric objects.
4. Demonstrate applications.

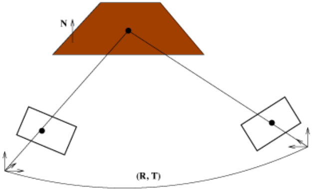
References:

- On symmetry and multiple view geometry. IJCV, 2004.
- Symmetry-based 3-D reconstruction from perspective images. CVIU, 2005.
- Large-baseline matching and reconstruction from symmetry cells. ICRA, 2004.
- Reconstruction of 3-D curves from perspective images without discrete features. ECCV, 2004.

Lecture notes available at: `www.eecs.berkeley.edu/~yang`

Homography: A Review

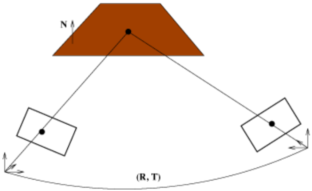
1. Two views of a planar structure satisfies a **homography** relation



$$\lambda_2 \mathbf{x}_2 = \lambda_1 H \mathbf{x}_1 \text{ where } H = (R + \frac{1}{d} T N^T) \in \mathbb{R}^{3\times 3}.$$

Homography: A Review

1. Two views of a planar structure satisfies a **homography** relation



$$\lambda_2 \mathbf{x}_2 = \lambda_1 H \mathbf{x}_1 \text{ where } H = (R + \frac{1}{d} T N^T) \in \mathbb{R}^{3\times 3}.$$

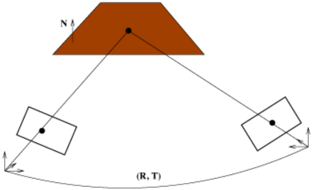
2. Convert bilinear relation in $(\mathbf{x}_1, \mathbf{x}_2)$ to linear in H

$$\widehat{\mathbf{x}_2} H \mathbf{x}_1 = 0 \Rightarrow (\mathbf{x}_1 \otimes \widehat{\mathbf{x}_2})^T H^s = 0,$$

where $H^s = [H_{11}, H_{21}, H_{31}, \cdots, H_{33}]^T \in \mathbb{R}^9$ and $\mathbf{x}_1 \otimes \widehat{\mathbf{x}_2} \in \mathbb{R}^{9\times 3}$.

Homography: A Review

1. Two views of a planar structure satisfies a **homography** relation



$$\lambda_2 \mathbf{x}_2 = \lambda_1 H \mathbf{x}_1 \text{ where } H = (R + \frac{1}{d} T N^T) \in \mathbb{R}^{3\times 3}.$$

2. Convert bilinear relation in $(\mathbf{x}_1, \mathbf{x}_2)$ to linear in H

$$\widehat{\mathbf{x}_2} H \mathbf{x}_1 = 0 \Rightarrow (\mathbf{x}_1 \otimes \widehat{\mathbf{x}_2})^T H^s = 0,$$

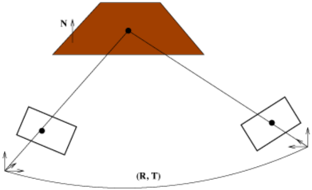
   where $H^s = [H_{11}, H_{21}, H_{31}, \cdots, H_{33}]^T \in \mathbb{R}^9$ and $\mathbf{x}_1 \otimes \widehat{\mathbf{x}_2} \in \mathbb{R}^{9\times 3}$.

3. Given N corresponding pair of points, estimate H^s from a null space

$$\left[\mathbf{x}_1^{(1)} \otimes \widehat{\mathbf{x}_2}^{(1)}, \cdots, \mathbf{x}_1^{(N)} \otimes \widehat{\mathbf{x}_2}^{(N)} \right]^T H^s = 0.$$

Homography: A Review

1. Two views of a planar structure satisfies a **homography** relation



$$\lambda_2 \mathbf{x}_2 = \lambda_1 H \mathbf{x}_1 \text{ where } H = (R + \frac{1}{d} T N^T) \in \mathbb{R}^{3\times 3}.$$

2. Convert bilinear relation in $(\mathbf{x}_1, \mathbf{x}_2)$ to linear in H

$$\widehat{\mathbf{x}_2} H \mathbf{x}_1 = 0 \Rightarrow (\mathbf{x}_1 \otimes \widehat{\mathbf{x}_2})^T H^s = 0,$$

where $H^s = [H_{11}, H_{21}, H_{31}, \cdots, H_{33}]^T \in \mathbb{R}^9$ and $\mathbf{x}_1 \otimes \widehat{\mathbf{x}_2} \in \mathbb{R}^{9\times 3}$.

3. Given N corresponding pair of points, estimate H^s from a null space

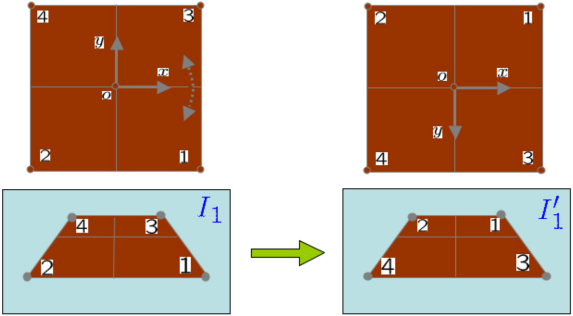
$$\left[\mathbf{x}_1^{(1)} \otimes \widehat{\mathbf{x}_2}^{(1)}, \cdots, \mathbf{x}_1^{(N)} \otimes \widehat{\mathbf{x}_2}^{(N)} \right]^T H^s = 0.$$

4. **Four-point algorithm** decomposes H to recover (R, T) up to two solutions.

Please run MATLAB script: `fourpoint.m`

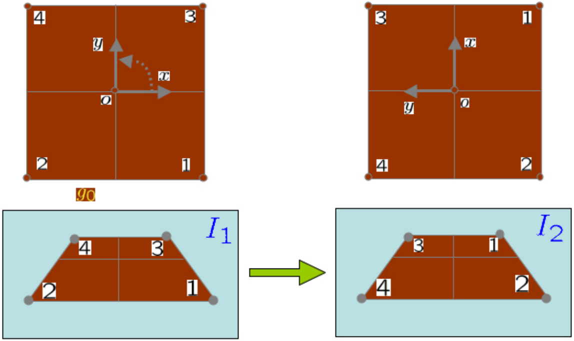
Equivalent Views of Symmetric Structures

- Reflection



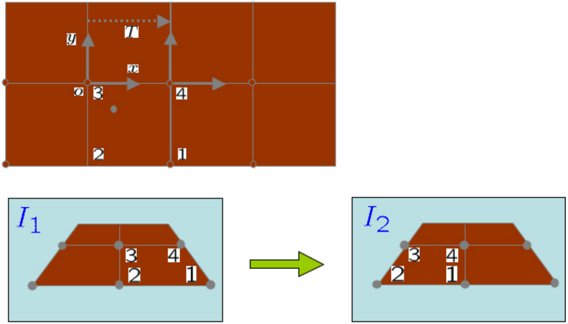
Equivalent Views of Symmetric Structures

- Rotation



Equivalent Views of Symmetric Structures

- Translation



Symmetry Group

1. Three types of isometric symmetries in Euclidean space: **Rotation**, **Reflection**, and **Translation** [Fedorov 1885, Weyl 1952]

Symmetry Group

1. Three types of isometric symmetries in Euclidean space: **Rotation**, **Reflection**, and **Translation** [Fedorov 1885, Weyl 1952]
2. Definition: Symmetric structure and its group action
   A set of geometric primitives $S \subset \mathbb{R}^3$ is called a **symmetric structure** if there exists a nontrivial group action G of rigid-body motions and reflections under which S is invariant
   $$g \in G : g(S) = S.$$

Symmetry Group

1. Three types of isometric symmetries in Euclidean space: **Rotation**, **Reflection**, and **Translation** [Fedorov 1885, Weyl 1952]
2. Definition: Symmetric structure and its group action
   A set of geometric primitives $S \subset \mathbb{R}^3$ is called a **symmetric structure** if there exists a nontrivial group action G of rigid-body motions and reflections under which S is invariant
   $$g \in G : g(S) = S.$$

Example (Symmetry group of a square)

Let $S = \left\{ \mathbf{p}_1 \doteq \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \mathbf{p}_2 \doteq \begin{bmatrix} -1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \mathbf{p}_3 \doteq \begin{bmatrix} -1 \\ -1 \\ 0 \\ 1 \end{bmatrix}, \mathbf{p}_4 \doteq \begin{bmatrix} 1 \\ -1 \\ 0 \\ 1 \end{bmatrix} \right\}$ in homogeneous coordinates.

Symmetry Group

1. Three types of isometric symmetries in Euclidean space: **Rotation**, **Reflection**, and **Translation** [Fedorov 1885, Weyl 1952]
2. Definition: Symmetric structure and its group action
   A set of geometric primitives $S \subset \mathbb{R}^3$ is called a **symmetric structure** if there exists a nontrivial group action G of rigid-body motions and reflections under which S is invariant

$$g \in G : g(S) = S.$$

Example (Symmetry group of a square)

Let $S = \left\{ \mathbf{p}_1 \doteq \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \mathbf{p}_2 \doteq \begin{bmatrix} -1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \mathbf{p}_3 \doteq \begin{bmatrix} -1 \\ -1 \\ 0 \\ 1 \end{bmatrix}, \mathbf{p}_4 \doteq \begin{bmatrix} 1 \\ -1 \\ 0 \\ 1 \end{bmatrix} \right\}$ in homogeneous coordinates.

Consider all possible symmetry actions $g = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4}$:

1. Rotation: $e = I_{4\times 4}$, $a_1 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $a_2 = a_1^2$, $a_3 = a_1^3$ such that

$$e(\mathbf{p}_i) = \mathbf{p}_i; \quad a_1(\mathbf{p}_1) = \mathbf{p}_2, a_1(\mathbf{p}_2) = \mathbf{p}_3, a_1(\mathbf{p}_3) = \mathbf{p}_4, \cdots$$

2. Reflection: e, $b = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ such that

$$b(\mathbf{p}_1) = \mathbf{p}_2, b(\mathbf{p}_2) = \mathbf{p}_1, b(\mathbf{p}_3) = \mathbf{p}_4, b(\mathbf{p}_4) = \mathbf{p}_3.$$

G is a group, called **dihedral group** D_4.

Example (Symmetry group of a rectangle)

Let $S = \left\{ \mathbf{p}_1 \doteq \begin{bmatrix} 2 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \mathbf{p}_2 \doteq \begin{bmatrix} -2 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \mathbf{p}_3 \doteq \begin{bmatrix} -2 \\ -1 \\ 0 \\ 1 \end{bmatrix}, \mathbf{p}_4 \doteq \begin{bmatrix} 2 \\ -1 \\ 0 \\ 1 \end{bmatrix} \right\}$ in homogeneous coordinates.

Example (Symmetry group of a rectangle)

Let $S = \left\{ \mathbf{p}_1 \doteq \begin{bmatrix} 2 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \mathbf{p}_2 \doteq \begin{bmatrix} -2 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \mathbf{p}_3 \doteq \begin{bmatrix} -2 \\ -1 \\ 0 \\ 1 \end{bmatrix}, \mathbf{p}_4 \doteq \begin{bmatrix} 2 \\ -1 \\ 0 \\ 1 \end{bmatrix} \right\}$ in homogeneous coordinates.

Consider all elements $g \in G$:

1. Group actions: $e = I_{4\times 4}$, $g_x = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $g_y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $g_z = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.
2. G is a group: $G = \{e, g_x, g_y, g_z\}$

$$g_x^2 = g_y^2 = g_z^2 = e,\ g_x g_y = g_z,\ g_y g_z = g_x,\ g_z g_x = g_y.$$

Example (Symmetry group of a rectangle)

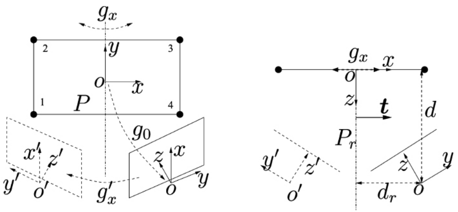
Let $S = \left\{ \mathbf{p}_1 \doteq \begin{bmatrix} 2 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \mathbf{p}_2 \doteq \begin{bmatrix} -2 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \mathbf{p}_3 \doteq \begin{bmatrix} -2 \\ -1 \\ 0 \\ 1 \end{bmatrix}, \mathbf{p}_4 \doteq \begin{bmatrix} 2 \\ -1 \\ 0 \\ 1 \end{bmatrix} \right\}$ in homogeneous coordinates.
Consider all elements $g \in G$:

1. Group actions: $e = I_{4\times 4}$, $g_x = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $g_y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $g_z = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.
2. G is a group: $G = \{e, g_x, g_y, g_z\}$

$$g_x^2 = g_y^2 = g_z^2 = e,\ g_x g_y = g_z,\ g_y g_z = g_x,\ g_z g_x = g_y.$$

Note that these group actions above directly apply to symmetric structures in 3-D. Next, we explain symmetric group in image induced by perspective projection.

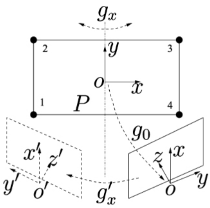
Induced Symmetry Group under Perspective Projection

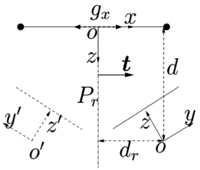


1. **Perspective projection**: Denote homography H_0 that projects $\mathbf{X}_0$ to the image $\mathbf{X}$

$$\lambda \mathbf{x} = \mathbf{X} = H_0 \mathbf{X}_0, \text{ where } H_0 = [R_0(1), R_0(2), T_0] \in \mathbb{R}^{3\times 3}.$$

Induced Symmetry Group under Perspective Projection





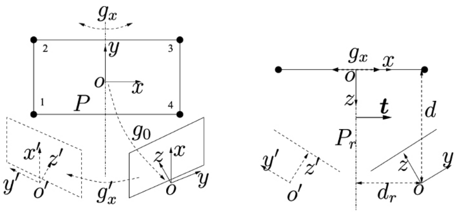
1. **Perspective projection**: Denote homography H_0 that projects $\mathbf{X}_0$ to the image $\mathbf{X}$

$$\lambda \mathbf{x} = \mathbf{X} = H_0 \mathbf{X}_0, \text{ where } H_0 = [R_0(1), R_0(2), T_0] \in \mathbb{R}^{3\times 3}.$$

2. Action g in space induces **hidden** perspective view $\mathbf{X}'$ and homography H'

$$H_0(g(\mathbf{X}_0)) = H_0 g H_0^{-1}(H_0 \mathbf{X}_0) \doteq H' \mathbf{X}.$$

Induced Symmetry Group under Perspective Projection



1. **Perspective projection**: Denote homography H_0 that projects $\mathbf{X}_0$ to the image $\mathbf{X}$

$$\lambda \mathbf{x} = \mathbf{X} = H_0 \mathbf{X}_0, \text{ where } H_0 = [R_0(1), R_0(2), T_0] \in \mathbb{R}^{3\times 3}.$$

2. Action g in space induces **hidden** perspective view $\mathbf{X}'$ and homography H'

$$H_0(g(\mathbf{X}_0)) = H_0 g H_0^{-1}(H_0 \mathbf{X}_0) \doteq H' \mathbf{X}.$$

3. **Homography group**: Let $G = \{e, g_1, \cdots, g_n\}$, define

$$G' \doteq H_0 G H_0^{-1} = \{e, H_0 g_1 H_0^{-1}, \cdots, H_0 g_n H_0^{-1}\}.$$

Homography Group

1. G' is a group induced by projection H_0.

Example (Homography group for a rectangle)

Recall for rectangle, $G = \{e, g_x, g_y, g_z\}$. Hence

$$G' = \{I, H_0 g_x H_0^{-1}, H_0 g_y H_0^{-1}, H_0 g_z H_0^{-1}\} \doteq \{I, H'_x, H'_y, H'_z\}.$$

Check that G' indeed is a group

$$(H'_x)^2 = (H'_y)^2 = (H'_z)^2 = I; H'_x H'_y = H'_z; H'_y H'_z = H'_y; H'_z H'_x = H'_y.$$

Homography Group

1. G' is a group induced by projection H_0.

Example (Homography group for a rectangle)

Recall for rectangle, $G = \{e, g_x, g_y, g_z\}$. Hence

$$G' = \{I, H_0 g_x H_0^{-1}, H_0 g_y H_0^{-1}, H_0 g_z H_0^{-1}\} \doteq \{I, H'_x, H'_y, H'_z\}.$$

Check that G' indeed is a group

$$(H'_x)^2 = (H'_y)^2 = (H'_z)^2 = I; H'_x H'_y = H'_z; H'_y H'_z = H'_y; H'_z H'_x = H'_y.$$

2. Homography between first camera and "hidden" camera

$$\widehat{\mathbf{x}'} H' \mathbf{x} = 0.$$

Homography Group

1. G' is a group induced by projection H_0.

Example (Homography group for a rectangle)

Recall for rectangle, $G = \{e, g_x, g_y, g_z\}$. Hence

$$G' = \{I, H_0 g_x H_0^{-1}, H_0 g_y H_0^{-1}, H_0 g_z H_0^{-1}\} \doteq \{I, H'_x, H'_y, H'_z\}.$$

Check that G' indeed is a group

$$(H'_x)^2 = (H'_y)^2 = (H'_z)^2 = I; H'_x H'_y = H'_z; H'_y H'_z = H'_y; H'_z H'_x = H'_y.$$

2. Homography between first camera and "hidden" camera

$$\widehat{\mathbf{x}'} H' \mathbf{x} = 0.$$

3. Using four-point algorithm, decompose H' into

$$H' \to \{R', \frac{1}{d} T', N\}.$$

Homography Group

1. G' is a group induced by projection H_0.

Example (Homography group for a rectangle)

Recall for rectangle, $G = \{e, g_x, g_y, g_z\}$. Hence

$$G' = \{I, H_0 g_x H_0^{-1}, H_0 g_y H_0^{-1}, H_0 g_z H_0^{-1}\} \doteq \{I, H'_x, H'_y, H'_z\}.$$

Check that G' indeed is a group

$$(H'_x)^2 = (H'_y)^2 = (H'_z)^2 = I; H'_x H'_y = H'_z; H'_y H'_z = H'_y; H'_z H'_x = H'_y.$$

2. Homography between first camera and "hidden" camera

$$\widehat{\mathbf{x}'} H' \mathbf{x} = 0.$$

3. Using four-point algorithm, decompose H' into

$$H' \rightarrow \{R', \frac{1}{d} T', N\}.$$

4. **Goal**: Recover R_0, T_0, and the symmetric structure S up to a scale factor.

Symmetry-based 3-D Reconstruction

1. Decomposition of $g' = g_0 g g_0^{-1}$

$$\begin{bmatrix} R' & T' \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_0 & T_0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_0 & T_0 \\ 0 & 1 \end{bmatrix}^{-1}$$

$$\Rightarrow \quad \begin{cases} R' = R_0 R R_0^T, \\ T' = (I - R_0 R R_0^T) T_0 + R_0 T. \end{cases}$$

Symmetry-based 3-D Reconstruction

1. Decomposition of $g' = g_0 g g_0^{-1}$

$$\begin{bmatrix} R' & T' \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_0 & T_0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_0 & T_0 \\ 0 & 1 \end{bmatrix}^{-1}$$

$$\Rightarrow \quad \begin{cases} R' = R_0 R R_0^T, \\ T' = (I - R_0 R R_0^T) T_0 + R_0 T. \end{cases}$$

2. **Solution**
   - Assume symmetric structure (S, g) in space! (Next section we discuss how to verify)

Symmetry-based 3-D Reconstruction

1. Decomposition of $g' = g_0 g g_0^{-1}$

$$\begin{bmatrix} R' & T' \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_0 & T_0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_0 & T_0 \\ 0 & 1 \end{bmatrix}^{-1}$$

$$\Rightarrow \quad \begin{cases} R' = R_0 R R_0^T, \\ T' = (I - R_0 R R_0^T) T_0 + R_0 T. \end{cases}$$

2. **Solution**
   - Assume symmetric structure (S, g) in space! (Next section we discuss how to verify)
   - Create hidden view g' **from real camera view** and recover R' and T'.

Symmetry-based 3-D Reconstruction

1. Decomposition of $g' = g_0 g g_0^{-1}$

$$\begin{bmatrix} R' & T' \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_0 & T_0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_0 & T_0 \\ 0 & 1 \end{bmatrix}^{-1}$$

$$\Rightarrow \quad \begin{cases} R' = R_0 R R_0^T, \\ T' = (I - R_0 R R_0^T) T_0 + R_0 T. \end{cases}$$

2. **Solution**
   - Assume symmetric structure (S, g) in space! (Next section we discuss how to verify)
   - Create hidden view g' **from real camera view** and recover R' and T'.
   - Imposing **positive depth** and **symmetry** constraints, four-point algorithm gives **unique solution**!

Symmetry-based 3-D Reconstruction

1. Decomposition of $g' = g_0 g g_0^{-1}$

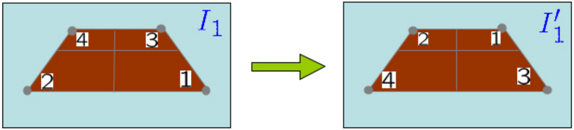
$$\begin{bmatrix} R' & T' \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_0 & T_0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_0 & T_0 \\ 0 & 1 \end{bmatrix}^{-1}$$

$$\Rightarrow \quad \begin{cases} R' = R_0 R R_0^T, \\ T' = (I - R_0 R R_0^T) T_0 + R_0 T. \end{cases}$$

2. **Solution**
   - Assume symmetric structure (S, g) in space! (Next section we discuss how to verify)
   - Create hidden view g' **from real camera view** and recover R' and T'.
   - Imposing **positive depth** and **symmetry** constraints, four-point algorithm gives **unique solution**!
   - Solve for R_0 and T_0 based on symmetry assumption.

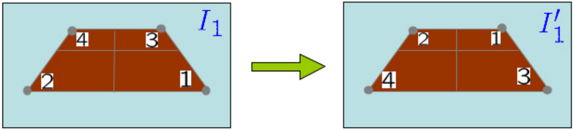
Reflective Symmetry

1. Create hidden view



Reflective Symmetry

1. Create hidden view

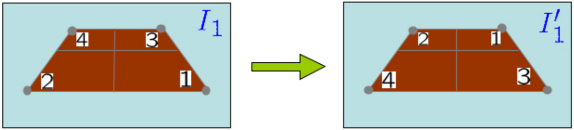


2. **Decomposition** of $H' \rightarrow \{R', T', N\}$
   Denote $\mathbf{t} = T'/\|T'\|$, then $R' = I - 2\mathbf{t}\mathbf{t}^T$

$$\text{Epipolar}: (\mathbf{x}')^T \widehat{T'} \mathbf{x} = 0.$$

Reflective Symmetry

1. Create hidden view



2. **Decomposition** of $H' \to \{R', T', N\}$
   Denote $\mathbf{t} = T'/\|T'\|$, then $R' = I - 2\mathbf{t}\mathbf{t}^T$

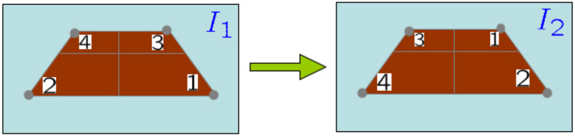
$$\text{Epipolar}: (\mathbf{x}')^T \widehat{T'}\mathbf{x} = 0.$$

3. **Pose**
   Denote $\mathbf{v}_1$ to be the eigenvector of R' corresponding to $\lambda_1 = -1$. Then

$$R_0 = \left[\pm\mathbf{v}_1, \pm\widehat{N}\mathbf{v}_1, N\right].$$

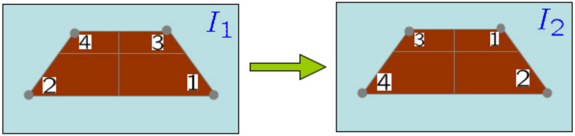
Rotational Symmetry

1. Create hidden view



Rotational Symmetry

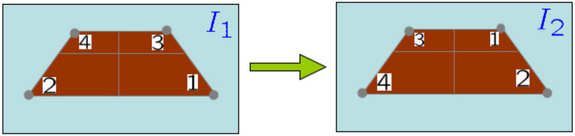
1. Create hidden view



2. **Decomposition** of $H' \to \{R', T', N\}$
   Let $\omega \in \mathbb{R}^3$ be the rotational axis of $R' = e^{\hat{\omega}\theta}$. Then $\omega \perp T'$: **orbital motion**.

Rotational Symmetry

1. Create hidden view



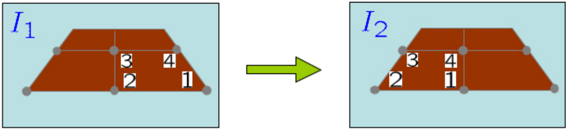
2. **Decomposition** of $H' \to \{R', T', N\}$
   Let $\omega \in \mathbb{R}^3$ be the rotational axis of $R' = e^{\hat{\omega}\theta}$. Then $\omega \perp T'$: **orbital motion**.
3. **Pose**
   The eigenvalues of R' are $1, e^{j\theta}$, and $e^{-j\theta}$, corresponding to eigenvectors $\mathbf{v}_1, \mathbf{v}_2$, and $\mathbf{v}_3$. Then

$$R_0 = [-\mathrm{Im}(\mathbf{v}_2)\cos(\alpha) - \mathrm{Re}(\mathbf{v}_2)\sin(\alpha), \mathrm{Re}(\mathbf{v}_2)\cos(\alpha) - \mathrm{Im}(\mathbf{v}_2)\sin(\alpha), \pm\mathbf{v}_1]$$

   where $\alpha \in \mathbb{R}$ is an arbitrary angle.

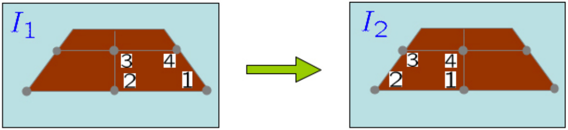
Translational Symmetry

1. Create hidden view



Translational Symmetry

1. Create hidden view

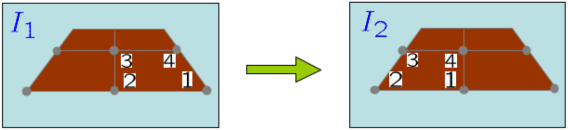


2. **Decomposition** of $H' \longrightarrow \{R', T', N\}$

$$R' = R_0 I R_0^T = I, T' = R_0 T.$$

Translational Symmetry

1. Create hidden view



2. **Decomposition** of $H' \longrightarrow \{R', T', N\}$

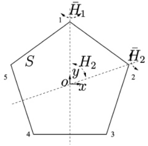
$$R' = R_0 I R_0^T = I, \; T' = R_0 T.$$

3. **Pose**
If we choose plane normal as z-axis, and symmetry translation T' as x-axis, then

$$R_0 = \left[T', \widehat{N} T', N \right].$$

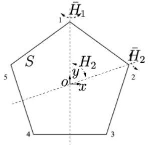
Simulation: 3-D reconstruction of a regular pentagon

1. Let $X_1 = [0, 1, 0]^T$, $X_2 = R_{xy}(\frac{2\pi}{5})X_1$,
   $X_3 = R_{xy}^2(\frac{2\pi}{5})X_1$,
   $X_4 = R_{xy}^3(\frac{2\pi}{5})X_1$,
   $X_5 = R_{xy}^4(\frac{2\pi}{5})X_1$.

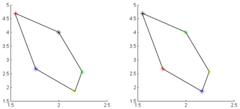


Simulation: 3-D reconstruction of a regular pentagon

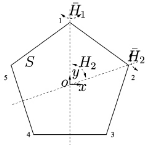
1. Let $X_1 = [0, 1, 0]^T$, $X_2 = R_{xy}(\frac{2\pi}{5})X_1$,
   $X_3 = R_{xy}^2(\frac{2\pi}{5})X_1$,
   $X_4 = R_{xy}^3(\frac{2\pi}{5})X_1$,
   $X_5 = R_{xy}^4(\frac{2\pi}{5})X_1$.



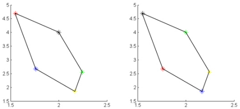
2. Let $R_0 = \begin{bmatrix} \cos(\frac{\pi}{10}) & 0 & -\sin(\frac{\pi}{10}) \\ 0 & 1 & 0 \\ \sin(\frac{\pi}{10}) & 0 & \cos(\frac{\pi}{10}) \end{bmatrix}$, $T_0 = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$.
   $g = R_{xy}(\frac{2\pi}{5})$.



Simulation: 3-D reconstruction of a regular pentagon



1. Let $X_1 = [0, 1, 0]^T$, $X_2 = R_{xy}(\frac{2\pi}{5})X_1$,
   $X_3 = R_{xy}^2(\frac{2\pi}{5})X_1$,
   $X_4 = R_{xy}^3(\frac{2\pi}{5})X_1$,
   $X_5 = R_{xy}^4(\frac{2\pi}{5})X_1$.



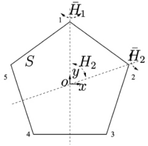
2. Let $R_0 = \begin{bmatrix} \cos(\frac{\pi}{10}) & 0 & -\sin(\frac{\pi}{10}) \\ 0 & 1 & 0 \\ \sin(\frac{\pi}{10}) & 0 & \cos(\frac{\pi}{10}) \end{bmatrix}$, $T_0 = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$.
   $g = R_{xy}(\frac{2\pi}{5})$.

3. From **four-point algorithm**: $H' = \begin{bmatrix} -3.4913 & -0.9045 & 11.6960 \\ 0.9323 & 0.3090 & 0.2083 \\ -1.4593 & -0.2939 & 4.8003 \end{bmatrix}$.
   Only one decomposition satisfies both positive depth **and** symmetry constraints:

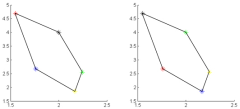
$$R' = \begin{bmatrix} 0.3750 & -0.9045 & -0.2031 \\ 0.9045 & 0.3090 & 0.2939 \\ -0.2031 & -0.2939 & 0.9340 \end{bmatrix}, \quad t' = \begin{bmatrix} 0.9510 \\ -0.0068 \\ 0.3090 \end{bmatrix}, \quad N = \begin{bmatrix} -0.3090 \\ 0.0000 \\ 0.9511 \end{bmatrix}.$$

Berkeley
University of California

Simulation: 3-D reconstruction of a regular pentagon



1. Let $X_1 = [0, 1, 0]^T$, $X_2 = R_{xy}(\frac{2\pi}{5})X_1$,
   $X_3 = R_{xy}^2(\frac{2\pi}{5})X_1$,
   $X_4 = R_{xy}^3(\frac{2\pi}{5})X_1$,
   $X_5 = R_{xy}^4(\frac{2\pi}{5})X_1$.



2. Let $R_0 = \begin{bmatrix} \cos(\frac{\pi}{10}) & 0 & -\sin(\frac{\pi}{10}) \\ 0 & 1 & 0 \\ \sin(\frac{\pi}{10}) & 0 & \cos(\frac{\pi}{10}) \end{bmatrix}$, $T_0 = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$.
   $g = R_{xy}(\frac{2\pi}{5})$.

3. From **four-point algorithm**: $H' = \begin{bmatrix} -3.4913 & -0.9045 & 11.6960 \\ 0.9323 & 0.3090 & 0.2083 \\ -1.4593 & -0.2939 & 4.8003 \end{bmatrix}$.

   Only one decomposition satisfies both positive depth **and** symmetry constraints:

$$R' = \begin{bmatrix} 0.3750 & -0.9045 & -0.2031 \\ 0.9045 & 0.3090 & 0.2939 \\ -0.2031 & -0.2939 & 0.9340 \end{bmatrix}, \quad t' = \begin{bmatrix} 0.9510 \\ -0.0068 \\ 0.3090 \end{bmatrix}, \quad N = \begin{bmatrix} -0.3090 \\ 0.0000 \\ 0.9511 \end{bmatrix}.$$

4. **Estimated pose** of the pentagon

$$R_0 = \begin{bmatrix} 0.9511 & 0.0000 & -0.3090 \\ 0 & 1.0000 & 0.0000 \\ 0.3090 & -0.0000 & 0.9511 \end{bmatrix}, \quad t_0 = \begin{bmatrix} 0.4046 \\ 0.6069 \\ 0.2023 \end{bmatrix}.$$

Please run MATLAB script: `rotational_symmetry.m`

Symmetry-based Image Segmentation

1. Symmetry-based reconstruction only useful when symmetric structures are present.

Symmetry-based Image Segmentation

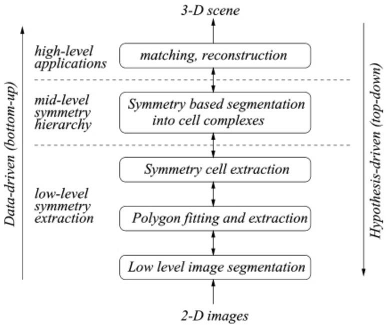
1. Symmetry-based reconstruction only useful when symmetric structures are present.
2. This section discusses detection of **symmetry cells** from natural images.

Symmetry-based Image Segmentation

1. Symmetry-based reconstruction only useful when symmetric structures are present.
2. This section discusses detection of **symmetry cells** from natural images.
3. Yet we will show the geometry in turn helps to validate the existence of symmetry cells.

Symmetry-based Image Segmentation

1. Symmetry-based reconstruction only useful when symmetric structures are present.
2. This section discusses detection of **symmetry cells** from natural images.
3. Yet we will show the geometry in turn helps to validate the existence of symmetry cells.

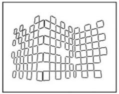


Low-Level Symmetry Cell Extraction

1. Homogeneous color patterns: Color-based mean shift algorithm with conservative segmentation parameters.

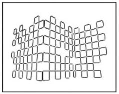
Low-Level Symmetry Cell Extraction

1. Homogeneous color patterns: Color-based mean shift algorithm with conservative segmentation parameters.
2. Polygon extraction: [Wuescher & Boyer 1991]
   The contour of a polygon is defined as piecewise line segments with zero constant curvature and corners with local curvature peaks.



Low-Level Symmetry Cell Extraction

1. Homogeneous color patterns: Color-based mean shift algorithm with conservative segmentation parameters.
2. Polygon extraction: [Wuescher & Boyer 1991]
   The contour of a polygon is defined as piecewise line segments with zero constant curvature and corners with local curvature peaks.



3. Local symmetry test

To verify if a polygon region can be interpreted as the image of an 3-D object with a symmetry group G, we verify whether all elements in G lead to a consistent structure and pose in space.

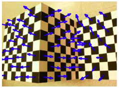


Figure: Rectangle symmetry cells are verified against g_x, g_y, and g_z.

Mid-Level Symmetry Hierarchy

Global symmetry test

Symmetry cells that are consistent with other adjacent cells more likely correspond to symmetric structures in space:

Cells that pass the local test are clustered by their orientation (normal vectors) and distance.


Mid-Level Symmetry Hierarchy

Global symmetry test

Symmetry cells that are consistent with other adjacent cells more likely correspond to symmetric structures in space:
Cells that pass the local test are clustered by their orientation (normal vectors) and distance.

Coplanarity

For cells in one cluster with same reflective and rotational symmetries, coplanarity is verified by estimation of translational symmetry.

Mid-Level Symmetry Hierarchy

Global symmetry test

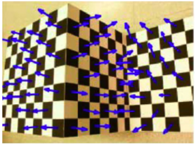
Symmetry cells that are consistent with other adjacent cells more likely correspond to symmetric structures in space:
Cells that pass the local test are clustered by their orientation (normal vectors) and distance.

Coplanarity

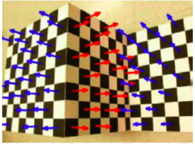
For cells in one cluster with same reflective and rotational symmetries, coplanarity is verified by estimation of translational symmetry.

Symmetry Bundle Adjustment

Coplanar symmetry cells shall have a common normal N and distance d from the camera center.



$\Longrightarrow$



Geometric Segmentation

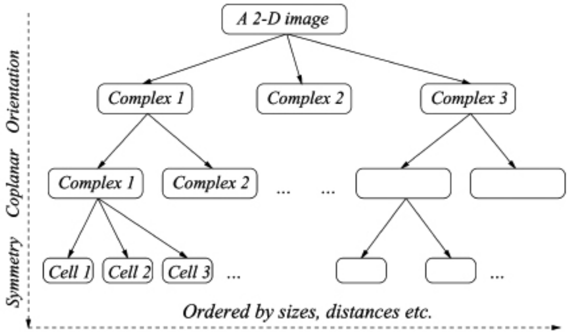
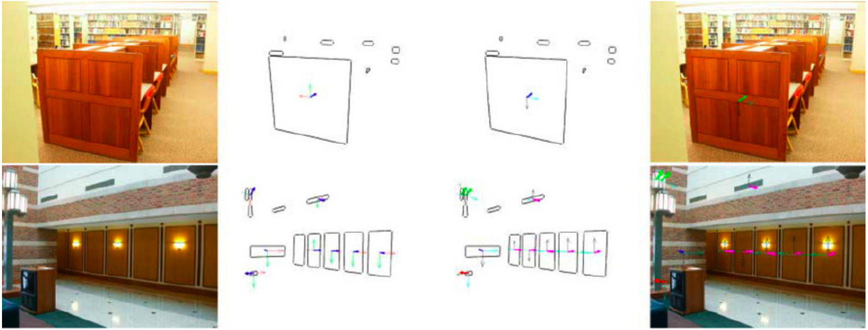
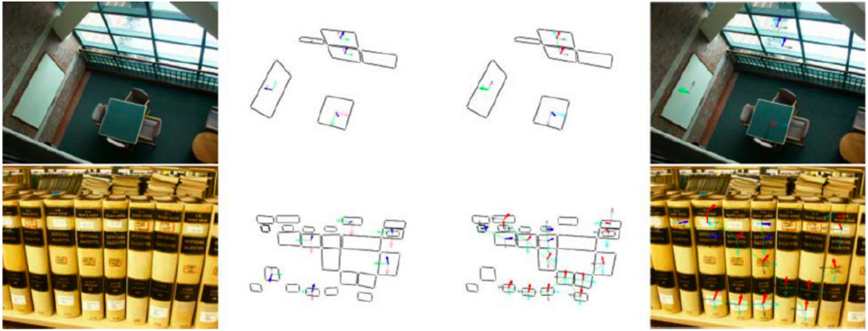


Figure: A hierarchical segmentation of an image by the geometry of symmetry.

Experiments

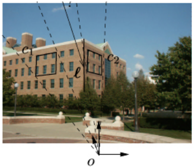




Geometry from Multiple Images

This section discusses **building the correct geometry** between multiple symmetry cells

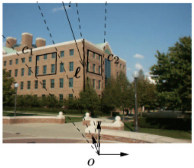
1. Alignment of two cells in a single image.



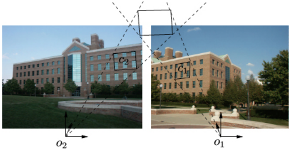
Geometry from Multiple Images

This section discusses **building the correct geometry** between multiple symmetry cells

1. Alignment of two cells in a single image.



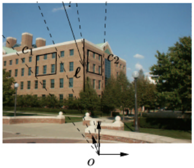
2. Alignment of one cell in two images.



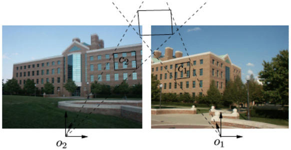
Geometry from Multiple Images

This section discusses **building the correct geometry** between multiple symmetry cells

1. Alignment of two cells in a single image.

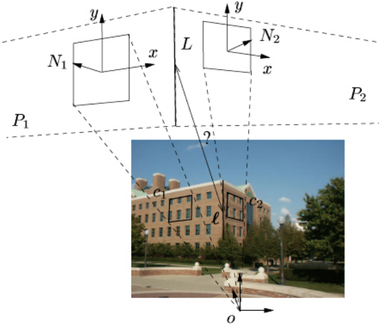


2. Alignment of one cell in two images.



3. Alignment of multiple cells in multiple images.

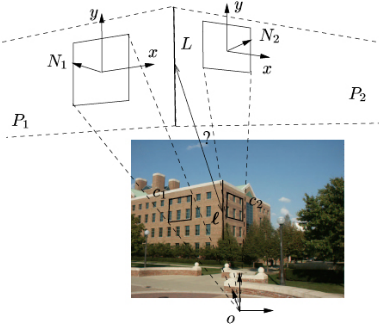
Two Cells in Single Image



$$d_2 = \alpha d_1$$

1. Two cells are recovered up to unknown distances d_1, d_2.

Two Cells in Single Image

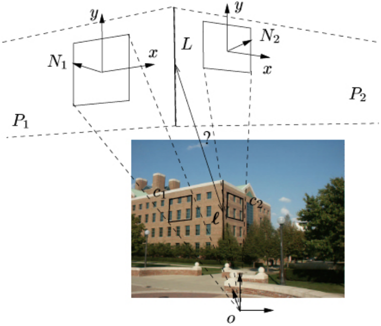


$$d_2 = \alpha d_1$$

1. Two cells are recovered up to unknown distances d_1, d_2.
2. Pick a common point at the intersection of two planes $\mathbf{x}$:

$$\lambda(\mathbf{x}) = \frac{d}{N^T \mathbf{x}}.$$

Two Cells in Single Image

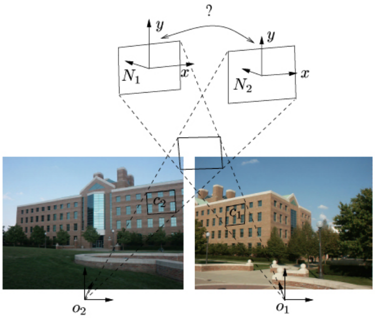


$$d_2 = \alpha d_1$$

1. Two cells are recovered up to unknown distances d_1, d_2.
2. Pick a common point at the intersection of two planes $\mathbf{x}$:
$$\lambda(\mathbf{x}) = \frac{d}{N^T\mathbf{x}}.$$
3. Since $\mathbf{x}$ is at intersection, $\lambda_1 \equiv \lambda_2$:
$$\alpha = \frac{d_2}{d_1} = \frac{N_2^T\mathbf{x}}{N_1^T\mathbf{x}}.$$

Example: Photo-Editing

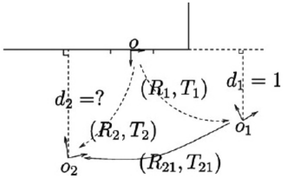
One Cell in Two Images



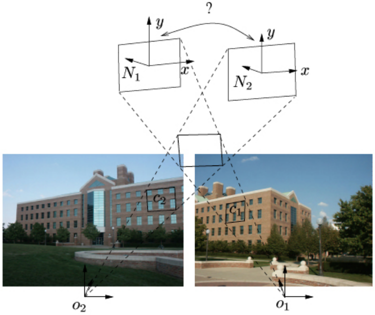
$$\begin{bmatrix} R_{21} & T_{21} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_2 & T_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_1 & T_1 \\ 0 & 1 \end{bmatrix}^{-1} \in \mathbb{R}^{4\times 4},$$

where $T_1 = t_1$ and $T_2 = \alpha t_2$.

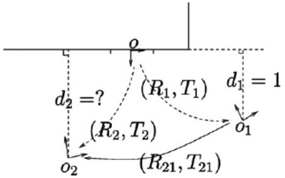
1. The cell is recovered independently with unknown distances d_1, d_2.



One Cell in Two Images



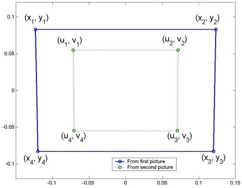
1. The cell is recovered independently with unknown distances d_1, d_2.



2. **Solution to scale** α: Reconstruction of S with (R_1, t_1) and (R_2, t_2) only differ up to the scale factor α.

$$\begin{bmatrix} R_{21} & T_{21} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_2 & T_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_1 & T_1 \\ 0 & 1 \end{bmatrix}^{-1} \in \mathbb{R}^{4\times 4},$$

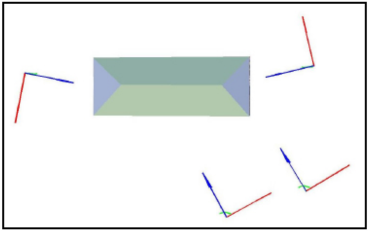
where $T_1 = t_1$ and $T_2 = \alpha t_2$.

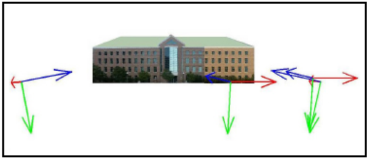


Example: Semi-automatic 3-D reconstruction









Multiple Cells in Multiple Images

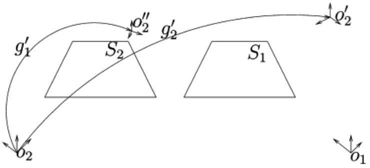
1. **One ambiguity** for multiple-view of a rectangle $\begin{bmatrix} R_{21} & T_{21} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_2 & T_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_1 & T_1 \\ 0 & 1 \end{bmatrix}^{-1}$

   This ambiguity cannot be eliminated if a single cell is present.

Multiple Cells in Multiple Images

1. **One ambiguity** for multiple-view of a rectangle $\begin{bmatrix} R_{21} & T_{21} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_2 & T_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_1 & T_1 \\ 0 & 1 \end{bmatrix}^{-1}$

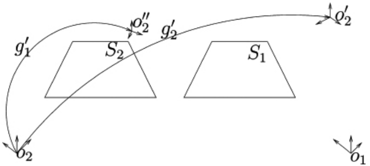
   This ambiguity cannot be eliminated if a single cell is present.
2. **Complex-to-complex matching:** Ambiguity is eliminated using multiple cells in two views



Multiple Cells in Multiple Images

1. **One ambiguity** for multiple-view of a rectangle $\begin{bmatrix} R_{21} & T_{21} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_2 & T_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_1 & T_1 \\ 0 & 1 \end{bmatrix}^{-1}$

   This ambiguity cannot be eliminated if a single cell is present.
2. **Complex-to-complex matching:** Ambiguity is eliminated using multiple cells in two views



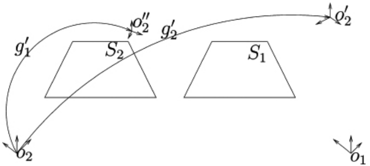
Introduce mismatches across multiple cells S_1, S_2!

Multiple Cells in Multiple Images

1. **One ambiguity** for multiple-view of a rectangle $\begin{bmatrix} R_{21} & T_{21} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_2 & T_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_1 & T_1 \\ 0 & 1 \end{bmatrix}^{-1}$

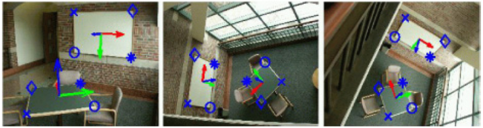
   This ambiguity cannot be eliminated if a single cell is present.

2. **Complex-to-complex matching:** Ambiguity is eliminated using multiple cells in two views

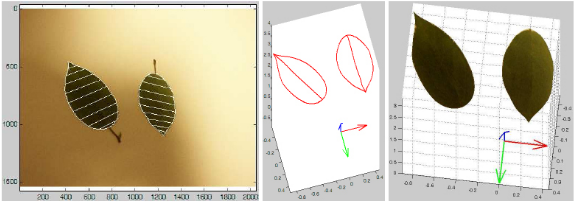


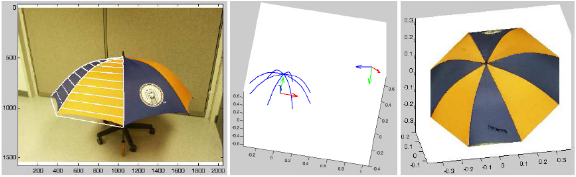
Introduce mismatches across multiple cells S_1, S_2!

3. **Pictorial matching:** Eliminate mismatch using **shape similarity** and **interior texture** information.



Extension: Symmetric Curves and Surfaces





Reference: Wei Hong et al.. Reconstruction of 3-D curves from perspective images without discrete features. ECCV, 2004.

Thank you!

Allen Y. Yang, <yang@eecs.berkeley.edu>