Multi-view geometry

Build 3D models from images

Carry out manipulation tasks

Camera-based 3D capture system

http://webdocs.cs.ualberta.ca/~vis/ibmr/

Training for Amazon manipulation challenge

And many other usages…
Multi-view Geometry

Relates
• 3D world points
• Camera calibr.
• 2D image points

The reconstruction of 3D models of objects from a collection of 2D images
Multi-view Geometry

Typical processing pipeline  (C. Hernandez MVS tutorial)
Multi-view Geometry
Calibration accuracy on these datasets appears to be on the order of a pixel (a pixel spans about 1/4mm on the object).
Dorsal and Ventral Pathways
Where/What or Action/Perception?

Dorsal (magno) Pathway
- to parietal lobe
- spatial vision – localization in space
- “WHERE”

Ventral (parvo) Pathway
- to temporal lobe
- object recognition
- “WHAT”

• Humans don’t internalize detailed 3D maps!
• Use external world as map. (Hayhoe, Pelz, Rensink, Goodale etc)
Course content

- Video processing: Image motion and tracking
- 2D projective geometry
- 3D projective geometry
- Cameras, their math model and calibration
- Two-camera “stereo” 3D reconstruction
- Multi-view geometry and general 3D reconstruct.
- Light, surface reflectance and math modeling
- Computer vision systems
Challenges: Geometric size ambiguity
Challenges
Light / photometric ambiguity
Challenges
Light / photometric ambiguity
Fundamental types of video processing
“Visual motion detection”

• Relating two adjacent frames: (small differences):
  \[
  \text{Im}(x + \delta x, \ y + \delta y, \ t + \delta t) = \text{Im}(x, \ y, \ t)
  \]
Fundamental types of video processing
“Visual Tracking” / Stabilization

• Globally relating all frames: (large differences):
Intensity images / video
non-trivial to get information

\[ \text{abs}(\text{Image 1} - \text{Image 2}) = ? \]

Note: Almost all pixels change!

Constancy: The physical scene is the same
How do we make use of this?
MTF – Modular Tracking Framework

Modular Tracking Framework
A Unified Approach to Registration based Tracking
Abhineet Singh and Martin Jagersand

- Open source
- C++ implementation
- ROS interface
- Matlab/Pyhton
- Cross platform
Application: Augmenter Reality
Beyond 3D
Non-rigid and articulated motion

- Humans ubiquitous in graphics applications
- A practical, realistic model requires
  - Skeleton
  - Geometry (manually modeled, laser scanned)
    - Physical simulation for clothes, muscle
  - Texture/appearance (from images)
  - Animation (mocap, simulation, artist)
Beyond 3D:
Non-rigid and articulated motion

PhD work of Neil Birkbeck, Best thesis prize winner
Uses of partial information from 2D-3D camera geometry

- Measurements in single images
- Visual constraints: verify alignments, detect impossible configurations (Escher paintings)
- Visual servoing
- Video tracking
- Rendering
- … many more

Why? Compact, accurate
Few relative alignments vs. complete global geometry
Single View Metrology

Vanishing point

Vertical vanishing point (at infinity)

Vanishing line

Geometric Cues
Estimated Height

- The distance $\| t_r - b_r \|$ is known
- Used to estimate the height of the man in the scene

Single View Metrology (559 citations)

A. Criminisi, I Reid, A Zisserman
Geometry for hand-eye coordination
Image-Based Visual Servoing (IBVS)
Intro to Image-based Visual Servoing (IBVS)
Vision-Based Control (Visual Servoing)

- Current Image Features
- Desired Image Features
Vision-Based Control (Visual Servoing)

- Current Image Features
- Desired Image Features
Vision-Based Control (Visual Servoing)

- □: Current Image Features
- ★: Desired Image Features
Vision-Based Control (Visual Servoing)

- : Current Image Features
- : Desired Image Features
Vision-Based Control (Visual Servoing)

- Current Image Features
- Desired Image Features
u,v Image-Space Error

- Current Image Features
- Desired Image Features

\[
E = \left[ \begin{array}{c} y_0 \end{array} \right]
\]

One point

\[
E = [\mathbf{y}_0 - \mathbf{y}^*]
\]

Pixel coord

\[
E = \begin{bmatrix} y_u \ y_v \end{bmatrix} - \begin{bmatrix} y_u \ y_v \end{bmatrix}^*
\]

Many points

\[
E = \begin{bmatrix} y_1 \cdots y_8 \end{bmatrix}^* - \begin{bmatrix} y_1 \cdots y_8 \end{bmatrix}_0
\]
Problem: Lots of coordinate frames to calibrate

Camera
  – Center of projection
  – Different models

Robot and scene
  – Base frame
  – End-effector frame
  – Object
Visual specifications

• Point to Point task “error”:

$$E = [y^* - y_0]$$

$$E = \begin{bmatrix} y_1^* \\ \vdots \\ y_{16}^* \end{bmatrix} - \begin{bmatrix} y_1 \\ \vdots \\ y_{16} \end{bmatrix}_0$$

Why 16 elements?
Visual specifications 2

• Point to Line

$$E_{pl}(y, l) = \begin{bmatrix} y_l \cdot l_l \\ y_r \cdot l_r \end{bmatrix}$$

$$l_l = [y_2 \times y_3]_l$$

Note: y’s in homogeneous coord.

$$y_l = \begin{bmatrix} u_l \\ v_l \\ k \end{bmatrix}$$

$$[y_3]$$

$$[y_2]$$

How to design visual specifications in a principled way?
The 2D projective plane

- 2D projective space models perspective imaging
- Each 3D ray is a point in $P^2$: homogeneous coords.
- Ideal points
- $P^2$ is $R^2$ plus a "line at infinity" $l_\infty$

Homogeneous coordinates:

$$\begin{pmatrix} X \\ Y \\ k \end{pmatrix} \equiv s \begin{pmatrix} X \\ Y \\ k \end{pmatrix} \quad s \neq 0$$

Inhomogeneous equivalent:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{k} \begin{pmatrix} X \\ y \end{pmatrix}$$

Ideal point:

$$\begin{pmatrix} X \atop Y \atop 0 \end{pmatrix}$$
Projective Lines

- Projective line ~ a plane through the origin
  \[ \mathbf{l}^T \mathbf{x} = \mathbf{x}^T \mathbf{l} = AX + BY + CZ = 0 \]

- Ideal line ~ the plane parallel to the image

Duality: For any 2d projective property, a dual property holds when the role of points and lines are interchanged.

\[ l = X_1 \times X_2 \]

The line joining two points

\[ X = l_1 \times l_2 \]

The point joining two lines
Projective transformations

- Homographies, collineations, projectivities
- $3 \times 3$ nonsingular $H$
  \[ \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \]
  maps $P^2$ to $P^2$
  8 degrees of freedom
  determined by 4 corresponding points

- Transforming Lines?
  \[ x' = Hx \]
  subspaces preserved
  \[ x^Tl = 0 \]
  \[ x'^Tl' = 0 \]
  substitution
  \[ x^TH^Tl' = 0 \]
  dual transformation
  \[ l' = H^{-T}l \]
Homographies a generalization of affine and Euclidean transforms

<table>
<thead>
<tr>
<th>Group</th>
<th>Transformation</th>
<th>Invariants</th>
<th>Distortion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Projective</strong></td>
<td>$H_P = \begin{bmatrix} A &amp; t \ v^T &amp; v \end{bmatrix}$</td>
<td>• Cross ratio</td>
<td>![Image]</td>
</tr>
<tr>
<td>8 DOF</td>
<td></td>
<td>• Intersection</td>
<td>![Image]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Tangency</td>
<td>![Image]</td>
</tr>
<tr>
<td><strong>Affine</strong></td>
<td>$H_A = \begin{bmatrix} A &amp; t \ O^T &amp; 1 \end{bmatrix}$</td>
<td>• Parallelism</td>
<td>![Image]</td>
</tr>
<tr>
<td>6 DOF</td>
<td></td>
<td>• Relative dist in 1d</td>
<td>![Image]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Line at infinity $l_\infty$</td>
<td>![Image]</td>
</tr>
<tr>
<td><strong>Metric</strong></td>
<td>$H_S = \begin{bmatrix} sR &amp; t \ O^T &amp; 1 \end{bmatrix}$</td>
<td>• Relative distances</td>
<td>![Image]</td>
</tr>
<tr>
<td>4 DOF</td>
<td></td>
<td>• Angles</td>
<td>![Image]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Dual conic $C^*_\infty$</td>
<td>![Image]</td>
</tr>
<tr>
<td><strong>Euclidean</strong></td>
<td>$H_E = \begin{bmatrix} R &amp; t \ O^T &amp; 1 \end{bmatrix}$</td>
<td>• Lengths</td>
<td>![Image]</td>
</tr>
<tr>
<td>3 DOF</td>
<td></td>
<td>• Areas</td>
<td>![Image]</td>
</tr>
</tbody>
</table>

2 dof $l_\infty$  
2 dof $C^*_\infty$
How to define a visual task?
Visually defined alignment: basic

point-to-point: $e_{pp}(y) = y_2 - y_1$

or (homogenous coord) $e_{pp}(y) = y_2 \times y_1$

point-to-line: $e_{pl}(y) = y_1 \cdot (y_2 \times y_3)$
Some more visual alignments

parallel lines: $e_{\text{par}}(l) = (l_1 \times l_2) \times (l_3 \times l_4)$

line-to-line: $e_{\text{ll}}(y) = y_1 \cdot (y_3 \times y_4) + y_2 \cdot (y_3 \times y_4)$

point-to-ellipse: $e_{\text{pe}}(y) = y_1^T C_{\text{ellipse}} y_1$
Now: Language for visual alignments
What else do we need?

Need: 1. Some way of entering alignments in images
2. Video tracking to perform servoing!

Registration trackers:

Feature trackers:

Download: http://webdocs.cs.ualberta.ca/~vis/mtf/
Visual ambiguity

- Will the scissors cut the paper in the middle?
• Will the scissors cut the paper in the middle? **NO!**
Task Ambiguity

• Is the probe contacting the wire?
Task Ambiguity

- Is the probe contacting the wire? NO!
Solve the cut in the middle task?

• Compute paper midpoint. How?
Solve the cut in the middle task?

- Compute paper midpoint.
  (Are we done yet?)

\[ x_m = \left( l_1 \times l_2 \right) \]
Solve the cut in the middle task?

- Compute vanishing point $X_{\infty}$,
- Intersect $X_{\infty}$ w. midpt $X_m$

Alternative formulations?

$l_m = (X_{\infty} \times X_m)$
What information do we use to move?

Mobile robot navigation: From appearance to metric SLAM

- Recognizable Locations
- Metric Topological Maps
- Topological Maps
- Fully Metric Maps

How about other applications?
Beyond projective camera vision and screen GUI: Pointing

The human by pointing instructs the robot which ingredients are to be placed on the pizza.
Questions?

Camera-based 3D capture system

• Downloadable renderer+models: www.cs.ualberta.ca/~vis/ibmr
• Capturing software + IEEE VR tutorial text: www.cs.ualberta.ca/~vis/VR2003tut
• Main references for this talk:
  Jagersand et al. "Three Tier Model" 3DPVT 2008
  Jagersand "Image-based Animation..." CVPR 1997
• More papers: www.cs.ualberta.ca/~jag
Making Pizza with my robot
3rd Prize ICRA’15 Video competition

Questions?

The human by pointing instructs the robot which ingredient to place on the pizza.

Change a Lightbulb
ICRA’97 Video