Vision for Robotics
(A Gentle Introduction)

Bharath Sankaran
April 11, 2016
CS 545
Why Vision?

● What is a robot?
  ○ “A goal oriented machine that can sense, plan and act.” ~ Peter Corker (QUT).
  ○ Sense - Plan - Act view of robotics
  ○ Another view: Perception - Action - Learning view of robotics. “Autonomous movement systems can bootstrap themselves into competent behavior by trial and error learning from interacting with the environment.”

● Perception/Sensing critical
  ○ Human sensing: look (vision), touch (haptics), hearing (microphones), smell (???), taste (????????)
  ○ Almost 80% of human perception, learning, cognition and activities are mediated through vision ~ Multiple sources
  ○ Give robots the ability to see - Camera, Depth Sensors, etc
How is Vision used? (Humans & Robots)

- Visual Motor Integration: Eye - Hand, Eye - Foot and Eye - body coordination
- Visual Auditory Integration: The ability to relate and associate what is seen and heard
- Visual Memory - The ability to remember and recall information that is seen
- Visual (loop) closure - The ability to “fill gaps” or complete a visual picture based on seeing only some of the parts
- Spatial Relationships - The ability to know “where I am” in relation to objects and space around me and know where objects are in relation to one another
- Figure-Ground Discrimination - The ability to discern form and object from background
What is Computer Vision

- The science and technology of machines that can see.
- Formally, the field that includes the acquiring, processing, analyzing and understanding real world data from visual sensors.

- Is there a tree?
- How many cars are there?
- Is there an open parking spot?
- What room is this?
- Is there a refrigerator?
- Where is the sink?
- ...

- Is there a tree?
- How many cars are there?
- Is there an open parking spot?
- ...

- What room is this?
- Is there a refrigerator?
- Where is the sink?
- ...
The Pillars of Computer Vision: Areas of Research

Recognition, Reconstruction & Reorganization

Image courtesy: Jitendra Malik
Some History

- In 1966, Marvin Minsky at MIT asked his undergraduate student Gerald Jay Sussman to “spend the summer linking a camera to a computer and getting the computer to describe what it saw”. We now know that the problem is slightly more difficult than that. (Szeliski 2009, Computer Vision)
- 60 Years of Computer vision (courtesy Jitendra Malik)
  - 1960s: Beginnings in artificial intelligence, image processing and pattern recognition
  - 1970s: Foundational work on image formation: Horn, Koenderink, Longuet-Higgins
  - 1980s: Vision applied mathematics: geometry, multi-scale analysis, probabilistic modeling, control theory, optimization
  - 1990s: Geometric analysis largely completed, vision meets graphics, statistical learning approaches resurface
  - 2000s: Significant advances in visual recognition, range of practical applications, vision meets big data
  - 2010s: Deep Learning??
Main Conferences and Publications

● Main journals of the field:
  ○ IEEE Pattern Analysis Machine Intelligence (PAMI)
  ○ Int. Journal of Computer Vision (IJCV)
  ○ CVIU, IVC, PR, PRL,

● Main conferences of the field:
  ○ IEEE Computer Vision Pattern Recognition
  ○ IEEE Int. Conf. Computer Vision
  ○ European Conference Computer Vision

● Graphics and Robotics are the nearest fields, great vision papers at SIGGRAPH as well as at ICRA and Transactions of Robotics and IJRR
Visual Sensors 2D vs 3D: Different 2D cameras

Omni directional camera

Catadioptric camera

Fisheye camera
Visual Sensors 2D vs 3D: Different 3D sensors

- Lasers
- Other depth sensors, eg: time of flight
- RGBD Stereo Cameras
2D Image formation

What happens when light reflects from you?

Add a barrier to block of most of the light
- Reduces blurring
- The opening in the barrier is called an aperture
- What happens to the image? How does it transform

Images courtesy Alex Vasilescu (MIT)
2D Image formation - The Pinhole camera model

What information do we lose?
- Angles
- Distances
- Parallel lines are not parallel

Images courtesy Alex Vasilescu (MIT)

Note: Workout perspective projection math on white board. Ideas like tracking, segmentation, pixels moving together etc.
2D Image Information

What we see

What a computer sees

But why RGB?

Note: Talk about image formation on CCD, RGB bayering
Why RGB? - Minor digression

Visible spectrum incident on light sensitive retina - Cones (color)

CCDs emulate human visual system

After Isaka (2004)
3D Sensing - Basics

- Active Range Sensing: project energy (light, sonar, pulse) on the scene and detect its position to perform the measure; or exploit the effects of controlled changes of some sensor parameters (e.g. focus), ex: lidar, time of flight etc

- Passive Range Sensing: rely only on image intensities to perform the measure. Ex: stereopsis, structure from motion

Example of an active sensing system
3D Sensing - Active Sensing Basics

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} = \frac{b}{f \cot \theta - x} \begin{bmatrix}
x \\
y \\
f
\end{bmatrix}
\]
3D Sensing - Stereo Basics

\[ Z = \frac{fT}{d}, \quad d = x_r - x_l \]
RANSAC - Model fitting 3D example

- Choose a small subset of data points uniformly at random
- Fit a model to that subset
- Anything that is close the result is signal, the rest is noise
- Repeat this procedure and choose the best model
- Model examples: Plane (3 points), cylinder (2 points with normals), sphere (2 points), transformations (5 point algorithm, 8 point algorithm, 4 points + 1 direction)
Algorithm 15.4: RANSAC: fitting lines using random sample consensus

Determine:
- \( n \) — the smallest number of points required
- \( k \) — the number of iterations required
- \( t \) — the threshold used to identify a point that fits well
- \( d \) — the number of nearby points required
  to assert a model fits well

Until \( k \) iterations have occurred
- Draw a sample of \( n \) points from the data
  uniformly and at random
- Fit to that set of \( n \) points
- For each data point outside the sample
  - Test the distance from the point to the line
    against \( t \); if the distance from the point to the line
    is less than \( t \), the point is close
  end
- If there are \( d \) or more points close to the line
  then there is a good fit. Refit the line using all
  these points.
end

Use the best fit from this collection, using the
fitting error as a criterion

Image courtesy
Marc Pollefeys