Vision-based User Interfaces

Cham, Tat-Jen

Associate Professor / SMA-CS Fellow
School of Computer Engineering
Nanyang Technological University

School of COMPUTER ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY
Innovate

Vision as a User Interface

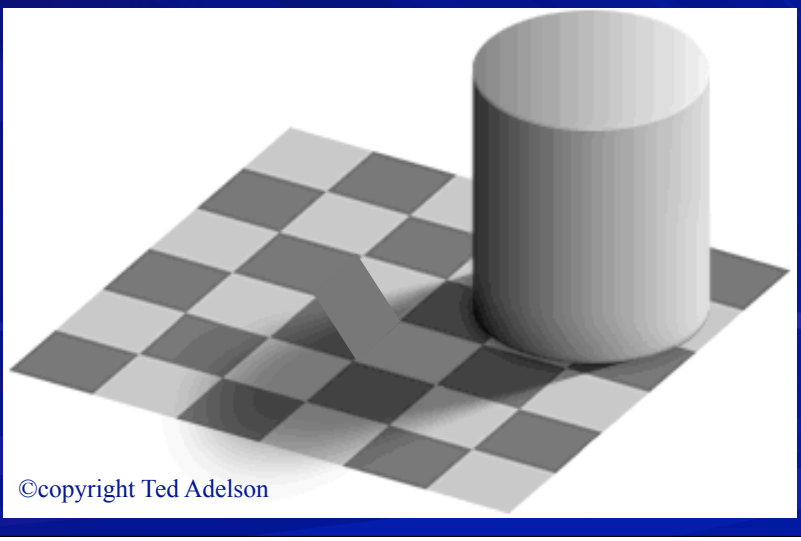
- Is vision a user interface?
- Vision is a sensing technology, not really an interface
  - Many non-interface applications (unlike speech recognition, or haptics)
- But…
  - Human communication often shows up as visual signals

Introduction

- Why vision?
  - Lots of data
    - 640x480 RGB @30fps = 27.6 Mb/s
  - Many "interesting events" occur as significant, non-random changes in light patterns
    - Larry's conjecture :)
  - Non-invasive interaction
    - What about privacy invasion?
  - Hardware can be inexpensive
    - Some webcams retailing <US$10
  - Humans have visual systems
    - we can extend intuition

Why is vision difficult?

- Humans have visual systems
  - we can extend intuition (incorrectly)



Why is vision difficult? (cont'd)

- Another example
  - Does appearance remain constant?



Why is vision difficult? (cont'd)

- The "graphics" problem
  - 3D models + illumination è rendered 2D images
  - Very well understood

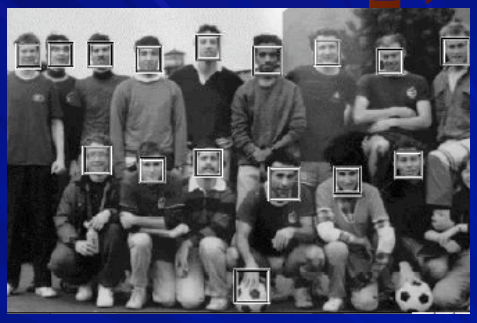
- The "vision" problem
  - Inferring 3D from 2D image is ambiguous
  - Inferring human intention from 2D image is even harder!
  - Lots of prior knowledge helps

Vision in User Interfaces

- Things to do with faces
  - Face detection
  - Face recognition
- Things to do with bodies
  - Hand gesture interpretation
  - Figure tracking
- Things to do with environment
  - Navigational aid
  - Steering

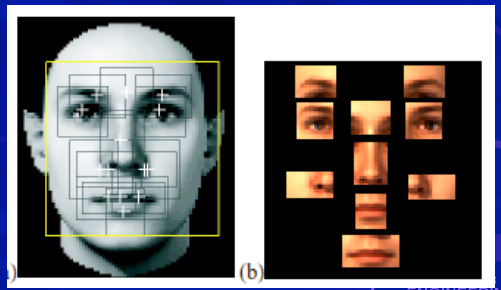
Things to do with Faces

- Face detection
  - Where are faces in an image?



Viola & Jones 2001

- Face recognition
  - Whose face is it?



Heisele, Ho & Poggio 2001

Things to do with Bodies

- Gesture recognition
  - recognizing *American Sign Language (ASL)*

Yang & Ahuja 1999

- Body tracking
  - *transferring 3D body motion*

Cheung, Baker & Kanade 2003

Things to do with Environment

- Navigational aid (for mobile cameras)
  - Recognizing landmarks
  - Inferring camera motion ("egomotion")
- Steering
  - Mouse input
  - Device motion

Optical mouse

Ojom Mosquito

Case Studies on Localization

Localization based on…

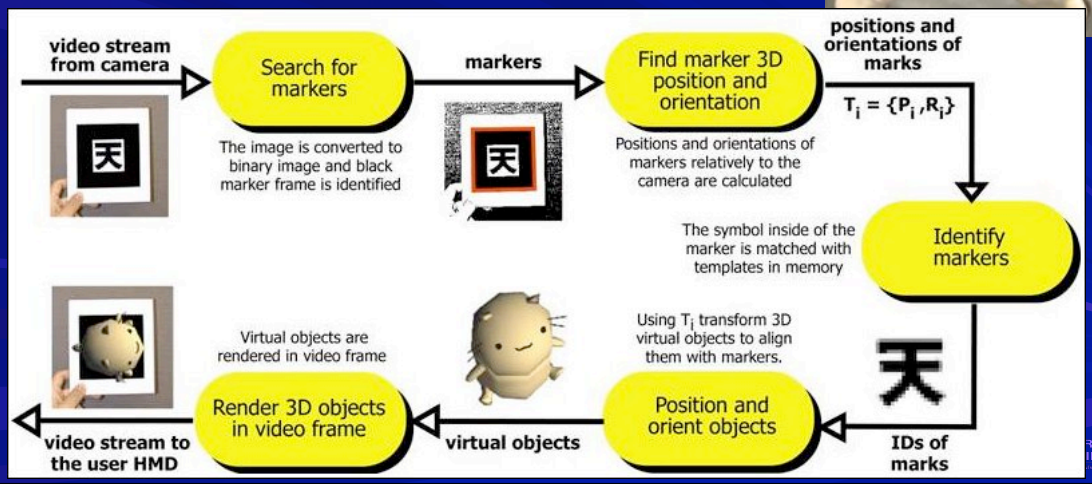
- Recognizing visual tags
- Recognizing buildings
- Understanding image motion

Visual Tag Recognition

- CyberCode and NaviCam
  - Jun Rekimoto, Sony
  - Uses special tags
    - Tags are "hyperlinks" to more informative data
  - http://www.businessweek.com/1997/25/b353217.htm

Visual Tag Recognition

- Augmented Reality Toolkit
  - Hirokazu Kato, Mark Billinghurst
  - http://www.hitl.washington.edu/artoolkit/



Location from Images from a

- Duncan Robertson & Roberto Cipolla University of Cambridge
- http://www.newscientist.com/news/news.jsp?id=ns99994857
- Use:
  - Snap a picture of nearby buildings with your phone
  - Server tells you:
    - which building
    - location and pointing direction of your camera

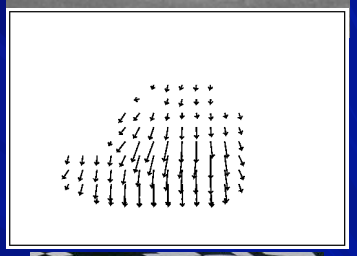
Location from Images from a Mobile Phone Camera

Visual Odometry

- Infer camera motion from a sequence of images
  - "Egomotion"
  - Technically 3D rotation + translation
- Normally assume that scene is (predominantly) static
  - To make problem tractable

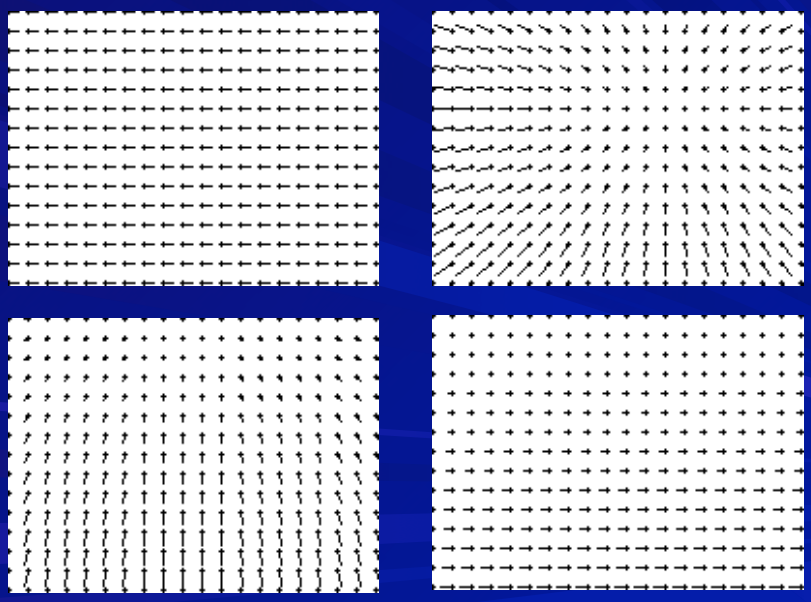
Optical Flow

- Vector field of spatial displacements
  - Assumption: pixels "move" but don't change color
- Why use optical flow?
  - Birds and bees do it too
  - We are good at interpreting it too
    - Starfield simulation
    - Biological motion
- Get camera velocity, or location of camera by summing
  - Displacement = ∫velocity



Motion from Optical Flow

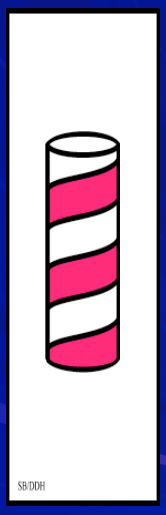
- We can figure out full 3D egomotion from optic flow field
  - Intuitive, but mathematically not trivial



Flow field © David Young, U. Sussex

Motion from Optical Flow (cont'd)

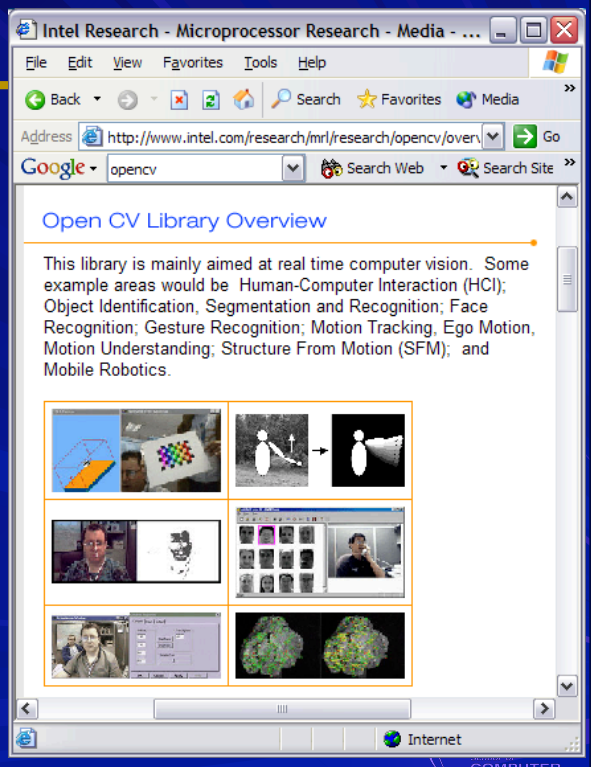
- Potential problems
  - Noisy
  - Insufficient parallax
  - Speed-scale ambiguity
  - Aperture problem
- Solution
  - More prior constraints the better
  - E.g. motion constraints



Flow field © David Young, U. Sussex

OpenCV

- Open source Computer Vision Library
  - Maintained by Intel
  - Lots of the latest vision algorithms
  - C/C++
  - Available for both Windows & Linux



More General Considerations

- Robust
  - Should work in all foreseeable conditions
  - Graceful degradation with sensor failure
- Accurate
  - Or have applications cope with reduced accuracy
- Real-time / efficient
  - Fast algorithms are key
  - Low-power consumption for mobile devices

End