Computer Vision - An Introductory Tour

The goal of Computer Vision is to process images acquired with cameras in order to produce a representation of objects in the world. -- Fua

Some Distinctions

- Vision: $image \Rightarrow world$
- Image Processing: $image \Rightarrow image$
- Graphics: $world \Rightarrow image$

Who needs Computer Vision?

Robot manipulation and navigation, safety monitoring, unmanned vehicles, surveillance, biometrics, animators, advanced communications, organizing and understanding image and video data

Example: Microsoft/UW PhotoSynth

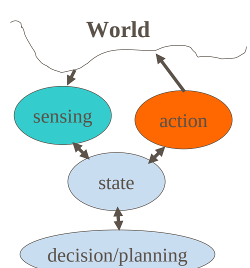
- Organizing and viewing image data
  - Obtain large collection of photos of a place or an object
  - Analyze them for similarities,
  - Displays the photos in a reconstructed **three-dimensional space**, showing you how each one relates to the next.
    - Fly through, zoom, find similar, exploit annotations
- http://labs.live.com/photosynth/whatis/

- Start with a collection of images, possibly from different cameras, positions, lighting, dates, times of day, resolutions
- Apply computer vision algorithms to extract hundreds of distinctive features.

- Images which share features can be associated.
- Seeing the same world point in multiple images (correspondence) allows us to compute its 3D location.

- Reconstructed depth points form a 3D point cloud.
- Images can be associated with 3D viewpoints and other similar images.

AI and Computer Vision



- Computer Vision is a common, powerful sensing modality.
- The field includes extraction and representation of information about world state.

Challenges of Vision

- Projection inherently discards information
  - **Infinite number of scenes could give rise to a particular image.**
  - Requires **constraints** and **assumptions** to infer knowledge of the imaged scene
- Interaction with the world involves uncertainty, not just errors.
  - We have apocryphal models of the physical world – incomplete, inaccurate, unmeasured
  - Real sensors have many sources of noise which must be modeled or suppressed.
- Biological vision systems => existence proof (?)

Slides: Computer Vision -- A modern approach

Working Systems

- Vision systems work in constrained environments:
  - Zip code readers (hand writing recognition)
  - Supermarket scanners
  - Circuit board solder inspection
  - Fingerprint identification (?)
  - Face recognition security systems (?)

Slides: Computer Vision -- A modern approach

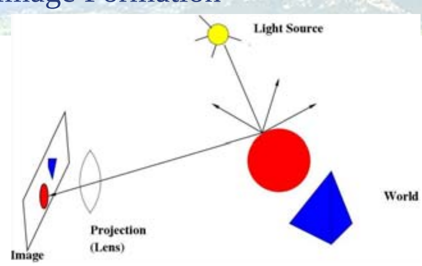
What is an image?

- Digital images are generally stored as matrices of unsigned char.

202	203	193	189	194	183	56	60	44
207	189	192	201	183	178	79	47	87
194	210	198	202	194	170	91	83	94
169	204	188	193	185	144	94	113	98
179	160	141	155	129	103	95	87	84
231	179	111	90	86	84	83	80	69
255	240	164	93	86	86	78	93	85
247	255	249	168	94	86	90	78	83
251	250	247	242	174	77	87	83	83

Slides: Computer Vision -- A modern approach

Image Formation



- Everything we image (see) is the result of emitted or reflected light entering the lens (eye).

Slides: Computer Vision -- A modern approach

Image formation (cont'd)

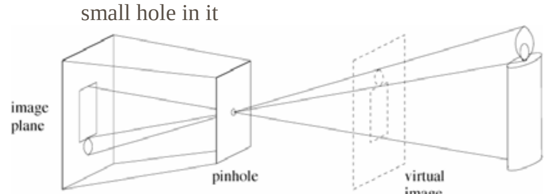
- Sensor –
  - Position, orientation, settings, projection model, sensor etc
- Light Sources –
  - Position, color, intensity
- Surfaces –
  - Pose, color, reflectance properties

➔Assume all surfaces Lambertian and all light sources distant (and diffuse)

Slides: Computer Vision -- A modern approach

Pinhole cameras

- Abstract camera model - box with a small hole in it
- Pinhole cameras work in practice

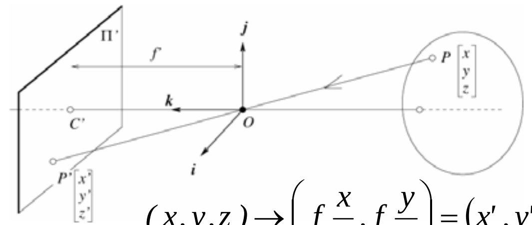


Slides: Computer Vision -- A modern approach

Observations

- Projection inverts the image
  - We generally use the *virtual image* which is equivalent
- Pinholes are not infinitely small, so a cone of light strikes every point in the image.
- Small pinhole implies very little light strikes the image
  - Lenses collect light and allow improved focus.

The equation of projection



$$(x, y, z) \rightarrow \left(f\frac{x}{z}, f\frac{y}{z} \right) = (x', y')$$

More Challenging Problems

- Reconstructing the 3D world
  - from multiple views
  - from shading
- Acting in the world – robots
  - Obstacle avoidance
  - Safe terrain identification
  - Reference from observation to known maps or models
  - Place recognition
- Recognition-- draw distinctions between what is seen, by constructing discriminative representations

Possible Constraints

- Models (templates)
  - Identify faces, printed characters etc by comparing to a model or reference image
- Using multiple images (separated in time)
  - Temporal image sequence:
    - Use temporal differences ie people tracking, motion detection, surveillance
  - Structure from motion, SLAM
- Using multiple views (separated by space)
  - Stereo, space carving

Correspondence (AI = Search)

- Template matching, stereo matching, tracking etc all require identifying points in different images which represent the same world point.
- **Correspondence**
  - What point (etc.) in this image corresponds to what point (etc.) in that?
- The better our description of points, the easier this is
- Local representations of image properties make things easier
  - identify points which are easily localised
  - compare with points in next image
  - search radius is constrained by geometry

Features

- Images are large (640x480x3=921Kbytes)
  - Is every pixel important?
  - Can we discard much of this raw data?
- Feature extraction attempts to identify 'interesting' points in the image.
  - Edges: transitions in intensity or color suggest object boundaries.
  - Regions: similar color, texture etc imply coherent surfaces
  - Principal components,

Representations of image properties

- Filter outputs
  - essentially form a dot-product between a pattern and an image, while shifting the pattern across the image
  - strong response -> image locally looks like the pattern
  - e.g. gradients – spatial derivatives
  - e.g. spots and bars
- A collection of filter outputs provides a comprehensive description of a point in an image
  - And so makes correspondence easier

Convention: filter outputs are displayed with zero as mid-grey, strong positive response as white, strong negative response as black

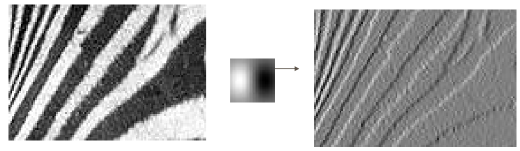


Image Filter Result

The result has a large value at points where the image "looks like" the filter kernel

Filter kernels that are larger see effects at coarser scales -- the filter on the left responds to the zebra's whiskers, that on the right to its stripes

We can use filters to obtain an image gradient - vectors are overlaid here on the image

Edges

Sobel

- Use intensity gradient
- How 'strong' must intensity change be?
- Which neighbor pixels form an edge?

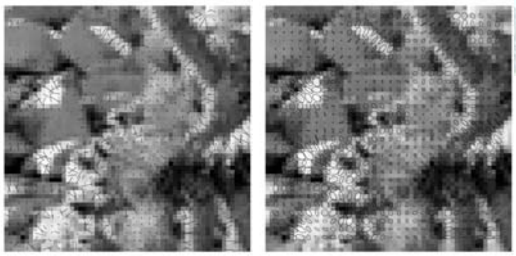
Canny

Sobel edge detector

- Smooth (blur) image with Gaussian
- Compute spatial intensity derivatives in the X and Y directions:
- Compute gradient image;
- Edge image is g>threshold

$$dy = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, dx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

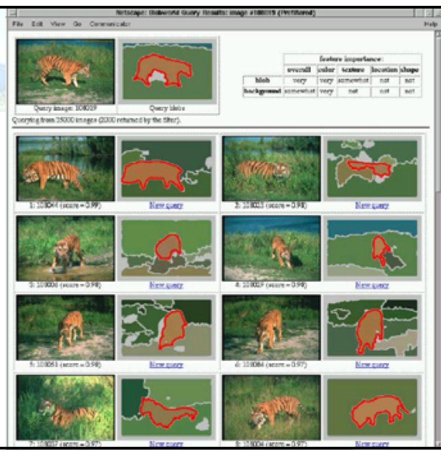
$$g = \sqrt{I_x^2 + I_y^2}$$



The covariance of the gradient in a small window gives an indication of whether the point is a corner or not - another representation that can be used for correspondence. On the left, a gradient and on the right, covariance ellipses.

Segmentation

- Which image components "belong together"?
- Belong together=lie on the same object
- Cues
  - similar colour
  - similar texture
  - not separated by contour
  - form a suggestive shape when assembled



- Use segmentation to simplify path/groundplane identification
- Use distant segments to robustly estimate motion relative to far field.

Object Recognition

- Which bits of image should be recognised together?
  - Segmentation.
- How can objects be recognised without focusing on detail?
  - Abstraction.
- How can objects with many free parameters be recognised?
  - No popular name, but it's a crucial problem anyhow.
- How do we structure very large modelbases?
  - again, no popular name; abstraction and learning come into this

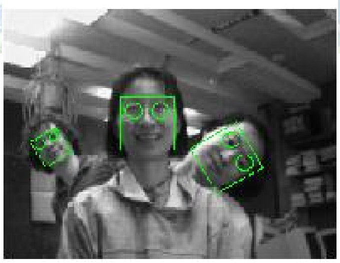
Model Based Recognition

Extract scale invariant features from template **images** and cluster in distinct views (Lowe, CVPR'01)

Template models identified in cluttered image

Recognition by matching templates

- Some objects are 2D patterns
  - e.g. faces
- Build an explicit pattern matcher
  - discount changes in illumination by using a parametric model
  - changes in background are hard
  - changes in pose are hard

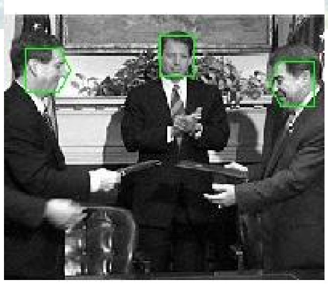


http://www.ri.cmu.edu/projects/project_271.html

Relations between templates

- e.g. find faces by
  - finding eyes, nose, mouth
  - finding assembly of the three that has the “right” relations

Probabilistic reasoning about relations makes it possible to find lateral views of faces http://www.ri.cmu.edu/projects/project_320.html



Tracking – multiple temporal images

- Use a model to predict next position and refine using next image
- Model:
  - simple dynamic models (second order dynamics)
  - kinematic models
  - etc.
- Face tracking and eye tracking now work rather well
- Common techniques Kalman filter, (Linear, extended, unscented), particle filter (regular, unscented)

Tracking by using probabilistic inference to predict next frame

work by Blake and Isard, films from the condensation algorithm home page http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/ISARD1/condensation.html

Tracking Human Pose

- Use skintone segmentation to find hands and face

- Infer global arm pose from face and hand pose

Quiz # 4

- How is Computer Vision different from Image Processing?
- What are the important factors in image formation?
- What is visual correspondence?
- Why is feature extraction useful?