Advanced Topic in Computational Vision

Miniproject in Computational Vision

CS 202-1-4171

Computer Science Department, BGU

Ohad Ben-Shahar
### Some necessary administrivia

<table>
<thead>
<tr>
<th>Lecturer</th>
<th>Ohad Ben-Shahar</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Contact info:</strong></td>
<td><strong><a href="mailto:ben-shahar@cs.bgu.ac.il">ben-shahar@cs.bgu.ac.il</a></strong> / (08-64) 77868</td>
</tr>
<tr>
<td><strong>Office:</strong></td>
<td>37/114 (Alon High-Tech Building)</td>
</tr>
<tr>
<td><strong>Office hours:</strong></td>
<td>Sundays 12:00-14:00 (or email me for an appointment)</td>
</tr>
<tr>
<td><strong>TA:</strong></td>
<td>Ilan Kadar (<a href="mailto:ilankad@cs.bgu.ac.il">ilankad@cs.bgu.ac.il</a>)</td>
</tr>
<tr>
<td><strong>Course schedule:</strong></td>
<td>Intro to Computer Vision (weeks 2+3+4)</td>
</tr>
<tr>
<td></td>
<td>Intro to Android Programming (weeks 4+5+6)</td>
</tr>
<tr>
<td></td>
<td>Submission of project proposal (week 6/7)</td>
</tr>
<tr>
<td></td>
<td>Submission of project report (last week of semester)</td>
</tr>
<tr>
<td></td>
<td>Submission of project software (end of exam period)</td>
</tr>
<tr>
<td></td>
<td>Project demos (end of exam period)</td>
</tr>
<tr>
<td></td>
<td>Grading by course staff</td>
</tr>
</tbody>
</table>
References

- **A Guided Tour of Computer Vision**,  
  by V. S. Nalwa, Addison-Wesley, 1993.

- **Machine Vision**  

- **Introductory Techniques for 3D Computer Vision**  

- **Computer Vision**  

- **Computer Vision – A Modern approach**  

- **Computer Vision: Algorithms and Applications**  
  Online version available at [http://szeliski.org/Book/](http://szeliski.org/Book/)
Visual Perception

The acquisition of knowledge about objects and events in the environment through information processing of light emitted or reflected from objects
Computational vision

The ultimate goal - making computers “see”

Automatic inference of properties of the world from images

<table>
<thead>
<tr>
<th>Automatic inference:</th>
<th>• Inference without (or minimal) human intervention.</th>
</tr>
</thead>
<tbody>
<tr>
<td>The world:</td>
<td>• The real unconstrained 3D physical world</td>
</tr>
<tr>
<td></td>
<td>• Constrained/Engineered environments</td>
</tr>
<tr>
<td>Image:</td>
<td>• 2D projection of the electromagnetic signal provided by the world.</td>
</tr>
<tr>
<td>Properties:</td>
<td>• Geometric: shape, size, location, distance,</td>
</tr>
<tr>
<td></td>
<td>• Material: color, texture, reflectivity, transparency</td>
</tr>
<tr>
<td></td>
<td>• Temporal: direction of motion (in 3D), speed, events</td>
</tr>
<tr>
<td></td>
<td>• Illumination: light source specification, light source color</td>
</tr>
<tr>
<td></td>
<td>• Symbolic: objects’ class, object’s ID</td>
</tr>
</tbody>
</table>
Computational vision

Automatic inference of properties of the world from images
Computational vision

Automatic inference of properties of the world from images

- (a specific) Baby’s face
- Happy expression
- Light direction
- Baby
- Shape (convex)
- Different materials
- Rough vs. smooth textures
- Depth relationship
The computer perspective on images

This is my baby. She is sitting on a beach bench, with the sun shining from behind, her right arm on her right leg. She is smiling.
Related fields and disciplines

- Image processing
- Computer graphics
- Pattern recognition
- Artificial intelligence
- Robotics

- Physics/Optics
- Psychology (of perception)
- Physiology
- Brain studies
- Philosophy (epistemology)

Image processing

Images

Vision (analysis)

Computer Graphics

Vision (synthesis)
Vision is the processing on incident “light”
Spectra

Power spectrum (spectral power)

Power emitted per wavelength

Wavelength
What is light

Radiant light can be described by its spectral power

![Spectral Radiation](image)
The human eye
The human eye

Pinhole camera model: Basic geometry
The human eye

Pinhole camera model: Perspective projection

\[ \overrightarrow{OP_i} = \lambda \cdot \overrightarrow{OP} \quad \Rightarrow \quad \lambda = \frac{x_i}{x} = \frac{y_i}{y} = \frac{f}{z} \quad \Rightarrow \quad \begin{cases} x_i = f \cdot \frac{x}{z} \\ y_i = f \cdot \frac{y}{z} \end{cases} \]
Why vision is difficult?

Recovery of structure is ill defined
The human eye

The retina:
The biological image plane
The human eye

Rods and cones

Rods:
- Extremely sensitivity to light
- Single photon response
- Low spatial resolution
- Single response profile
- B&W night vision (scotopic)

Cones:
- Relatively insensitive to light
- 100 photos for response comparable with rods
- High spatial resolution
- Different (3 types of) response profiles
- Color daylight vision (photopic)
The human eye

Rods and cones
Artificial light-capturing devices (a.k.a. cameras)

Photodiodes

R, G, B
This is my baby. She is sitting on a beach bench, with the sun shining from behind, her right arm on her right leg. She is smiling.
**Word of caution**

Visual interpretation of stimuli is so much more than receptor/pixel responses…
Vision as a modular process

High Level Vision

Intermediate Level Vision

Early (low level) Vision and Image Processing

Shape inference
Object recognition
Classification
Visual reasoning
Visual cognition

Grouping
Segmentation
Figure ground segregation
Shape completion
Depth ordering

Denoising
Feature extraction
Edge detection
Elementary motion analysis
Etc…
Edge detection
Edge detection

Edges represent important physical events and remain stable across viewing conditions.
Edge detection

Edge appearance in images

\[ \alpha(t) \]

\[ I_{\text{right}}(x, y) \]

\[ I_{\text{left}}(x, y) \]
**Edge detection**

**Edge appearance in images**

- **Step edge**
- **Ramp edge**
- **Roof edge**
- **Bright line**
- **Dark line**
Edge detection

Step edge localization and detection – continuous signals

Edge detection heuristics

Given $I(x)$, its edge points occur at the (local) maxima of $|I'(x)|$

Given $I(x)$, its edge points occur at the zero crossing of $I''(x)$
**Edge detection**

Gradient-based edge detection in images

\[ \nabla I = \left( \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) = \left( I_x, I_y \right) \]

Proposition:

The gradient vector points in the direction where the image changes “the most”.

**Edge detection heuristic #3**

Given \( I(x,y) \), its edge points occur at local (directional) maxima of \( |\nabla I| \)
Edge detection

Gradient-based edge detection in images

\[ \theta(x, y) = \tan^{-1} \left( \frac{I_y}{I_x} \right) \]

\[ \theta^\perp (x, y) = \tan^{-1} \left( -\frac{I_x}{I_y} \right) \]
Edge detection

Gradient-based edge detection in images

\[ |\nabla I| = \sqrt{I_x^2 + I_y^2} \]

\[ I_x = \frac{I(x+1,y) - f(x-1,y)}{2} \]

\[ I_y = \frac{I(x, y+1) - f(x, y-1)}{2} \]
Edge detection

Convolution

\[ f(t) = x(t) * h(t) = \int_{-\infty}^{+\infty} x(s) \cdot h(t - s) \, ds \]

\[ f(x, y) = I(x, y) * h(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} I(x', y') \cdot h(x - x', y - y') \, dx' \, dy' \]

\[ f(x, y) = I(x, y) * h(x, y) = \sum_{x' = -\infty}^{+\infty} \sum_{y' = -\infty}^{+\infty} I(x', y') \cdot h(x - x', y - y') \]
**Edge detection**

Laplacian-based edge detection in images

\[
\Delta I = \nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}
\]

\[
\frac{\partial^2 I}{\partial x^2} \equiv \frac{I(x-h,y) - 2I(x,y) + I(x+h,y)}{h^2}
\]

\[
\frac{\partial^2 I}{\partial y^2} \equiv \frac{I(x,y-h) - 2I(x,y) + I(x,y+h)}{h^2}
\]

\[
\nabla^2 I \equiv \frac{I(x-h,y)+I(x,y-h)-4I(x,y)+I(x+h,y)+I(x,y+h)}{h^2}
\]

\[
\nabla^2 I \cong I \ast \Delta
\]
Edge detection

Detecting 2D edge-like structures

\[ I(x, y) \quad \left| I_x \right| \quad \left| I_y \right| \quad \left| \nabla I \right| \]
Edge detection

Effect of noise

$I(x, y)$
$|I_x|$  
$|I_y|$  
$|\nabla I|$
Edge detection

Average-based image denoising

![Graph of noisy signal and convolution kernel with \( \frac{1}{n} \)]
Edge detection

Average-based image denoising

Original  Window size 5  Window size 9  Window size 17
Edge detection

Edge detection after denoising

\[ I(x, y) \quad \text{and} \quad |\nabla I| \]
Edge detection

Combining smoothing and differentiation for edge detection

\[
\left( I \ast K_s \right) \ast K_d \\
I \ast \left( K_s \ast K_d \right)
\]
**Edge detection**

Combining smoothing and differentiation for edge detection

\[
(I * K_s) * K_d
\]

\[
I * (K_{sd})
\]
Edge detection

Sobel edge detector

\[ K_x = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} -1 & +1 \\ -1 & +1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ 1 & 0 & +1 \end{bmatrix} \]

\[ K_y = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} -1 & -1 \\ +1 & +1 \end{bmatrix} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \]

\[ I_x = I * K_x \]
\[ I_y = I * K_y \]

\[ |\nabla I| = \sqrt{I_x^2 + I_y^2} \]
Edge detection

Not all edge detectors are equal…

Roberts  Sobel  Prewitt  LOG  Canny  Logical/Linear
3D Shape Inference via Stereo Vision
Stereopsis
Stereopsis
Stereopsis

Depth via triangulation

Scene point

Left center of projection

Right center of projection
**Stereopsis**

Depth reconstruction – Rectified Calibrated Stereo

\[ p = (x, y, z) \]

\[ \frac{x + \frac{b}{2}}{z} = \frac{x_r}{f} \]

\[ \frac{x - \frac{b}{2}}{z} = \frac{x_l}{f} \]

\[ z = \frac{bf}{x_r - x_l} = \frac{bf}{D} \]
Stereopsis

The correspondence problem
Stereopsis

Establishing correspondence
Stereopsis

Epipolar geometry

Scene point

Epipolar line

Epipole
Stereopsis

Rectified epipolar geometry
**Stereopsis**

Intensity-based methods for establishing correspondence

Assume:

Intensity of surfaces is independent of viewing angle

Establish correspondence for pixel $p$ in the left image:

Define a window $W_p$ of size $(2n+1) \times (2n+1)$ around $p$

For each pixel $q$ along $p$’s epipolar line in the right image do

Define a window $W_q$ of size $(2n+1) \times (2n+1)$ around $q$

Compute similarity $S(W_p, W_q)$ beween $W_p$ and $W_q$

End

Choose $q$ which maximizes $S(W_p, W_q)$
Stereopsis

Intensity-based methods for establishing correspondence
Stereopsis

Intensity based similarity measures

\[
SSD(W_p, W_q) = \sum_{\bar{w} \in W} [I_l(\bar{p} - \bar{w}) - I_r(\bar{q} - \bar{w})]^2 \Rightarrow \text{min}
\]

\[
Cor(W_p, W_q) = \sum_{\bar{w} \in W} I_l(\bar{p} - \bar{w}) \cdot I_r(\bar{q} - \bar{w}) = \bar{W}_p \cdot \bar{W}_q \Rightarrow \text{max}
\]

\[
NC(W_p, W_q) = \frac{\left(\bar{W}_p - \bar{W}_p\right) \cdot \left(\bar{W}_q - \bar{W}_q\right)}{\|\bar{W}_p - \bar{W}_p\| \cdot \|\bar{W}_q - \bar{W}_q\|} \Rightarrow \text{max}
\]
Stereopsis

State-of-the-art

Ground truth disparities  State-of-the-art disparities  Bad pixels (error>0.1)
Shape from Shading
Shape from Shading
Shape from Shading
Shape from Shading

Inverting the image formation process

Image formation = “Shading from shape” (and light sources)
Shape from Shading

Image formation
Shape from Shading

Surface brightness – appearance in the *Lambertian* case and point light source

\[ L = \rho \frac{1}{\pi} E \cos \theta_L \propto \rho(\hat{N} \cdot \hat{L}) \]
Shape from Shading

Shape description – Tangent plane and normal vectors

\[ \tilde{r}_x = \left( 1, 0, \frac{\partial H}{\partial x} \right) \quad \tilde{r}_y = \left( 0, 1, \frac{\partial H}{\partial y} \right) \]

\[ \tilde{N} = \tilde{r}_x \times \tilde{r}_y = (-p, -q, 1) \]

\[ \hat{N} = \frac{\tilde{N}}{\|\tilde{N}\|} = \frac{(-p, -q, 1)}{\sqrt{p^2 + q^2 + 1}} \]
Shape from Shading

Shading on Lambertian surface – General point source

\[ I = \rho (\hat{N} \cdot \hat{L}) = \rho \frac{-p \cdot L_x - q \cdot L_y + L_z}{\sqrt{p^2 + q^2 + 1} \sqrt{L_x^2 + L_y^2 + L_z^2}} = \rho \frac{p \cdot p_L + q \cdot q_L + 1}{\sqrt{p^2 + q^2 + 1} \sqrt{p_L^2 + q_L^2 + 1}} \]
Shape from Shading

Shading on Lambertian surface – Overhead point source

\[ I(x, y) = \rho(\hat{N} \cdot [0,0,1]) = \rho \frac{1}{\sqrt{p^2 + q^2 + 1}} = R(p, q) \]
Shape from Shading

The Reflectance Map – Lambertian surface from overhead source position

\[ R(p, q) = \frac{1}{\sqrt{p^2 + q^2 + 1}} \]
Shape from Shading

The Reflectance Map – Lambertian surface from general source position

\[ R(p,q) = \frac{p \cdot p_L + q \cdot q_L + 1}{\sqrt{p^2 + q^2 + 1} \sqrt{p_L^2 + q_L^2 + 1}} \]

Gradient point of maximum brightness
Shape from Shading

The Reflectance Map – typical real surfaces

\[ R(p, q) \]
Shape from Shading

Surface orientation from shading

\[ I(x, y) = R(p, q) \]
Shape from Shading

Photometric stereo
Shape from Shading

Photometric stereo

\[ I_1(x, y) = R_1(p, q) \]
\[ I_2(x, y) = R_2(p, q) \]
Shape from Shading

Photometric stereo
Computational Motion Analysis
**Computational Motion Analysis**

**Goal of motion analysis**

Estimating the relative motion of objects with respect to each other and the camera, given two or more perspective projection images in a time sequence.

What are the uses of such information?

- Simple motion detection.
- Object tracking.
- Enhanced UI.
- Shape from motion.
- Behavior analysis.
- Augmented reality.
Computational Motion Analysis

Possible setups

1. Stationary camera, stationary objects.
2. Stationary camera, moving objects.
3. Moving camera, stationary objects.
Computational Motion Analysis

The input

The information given by the sequence of frames is represented by $E(x,y,t)$ which is the brightness level at coordinates $x$ and $y$ in the frame representing the scene at time $t$.

Assuming that the frames are taken at the same time intervals, we assume that $t$ represents the $t^{th}$ frame of the sequence.
Computational Motion Analysis

Change detection via Binary difference picture

\[
DP_{jk}(x, y) = \begin{cases} 
1 & \text{if } |E(x, y, j) - E(x, y, k)| > \tau \\
0 & \text{otherwise}
\end{cases}
\]
Computational Motion Analysis

Change detection via Binary difference picture

Accumulative difference picture:

\[ ADP_0(x, y) = 0 \]
\[ ADP_k(x, y) = ADP_{k-1}(x, y) + DP_{1k}(x, y) \]
Computational Motion Analysis

What we actually need is … pixel-wise motion correspondence
**Computational Motion Analysis**

**Motion field** – the vector field of motion projections

\[ \mathbf{v}_i \delta t = \mathbf{v}_o \delta t \]

Scene point velocity: \( \mathbf{v}_o = \frac{d\mathbf{r}_o}{dt} \)

Image velocity: \( \mathbf{v}_i = \frac{d\mathbf{r}_i}{dt} \)

Perspective projection: \( \frac{1}{f'} \mathbf{r}_i = \frac{\mathbf{r}_o}{\mathbf{r}_o \cdot \mathbf{Z}} \)

Motion field

\[ \mathbf{v}_i = \frac{d\mathbf{r}_i}{dt} = f' \frac{(\mathbf{r}_o \cdot \mathbf{Z})\mathbf{v}_o - (\mathbf{v}_o \cdot \mathbf{Z})\mathbf{r}_o}{(\mathbf{r}_o \cdot \mathbf{Z})^2} = f' \frac{(\mathbf{r}_o \times \mathbf{v}_o) \times \mathbf{Z}}{(\mathbf{r}_o \cdot \mathbf{Z})^2} \]
Computational Motion Analysis

Can we hope to retrieve the motion field correctly?
Computational Motion Analysis

Optic Flow (a.k.a. Optical Flow)

Since we can’t retrieve the motion field, we calculate the optical flow which is the *apparent motion of the brightness pattern*.

Why this makes sense – because brightness patterns in the image *often* move in correspondence the moving objects in the scene.
Computational Motion Analysis

Can we hope to estimate the motion field correctly?

Except for special situations, the optical flow is not too different from the motion field. This will allow us to estimate relative motion by means of the changing image.
Computational Motion Analysis

Major assumption - Brightness/Color constancy
**Computational Motion Analysis**

Color constancy, small and continuous motion

- Assume brightness of patch remains same in both images:
  \[ E(x + \delta x, y + \delta y, t + \delta t) = E(x, y, t) \]

- Assume small and continuous motion: (Taylor expansion up to first order)
  \[ E(x, y, t) + \delta x \frac{\partial E}{\partial x} + \delta y \frac{\partial E}{\partial y} + \delta t \frac{\partial E}{\partial t} = E(x, y, t) \]

Optical Flow: Velocities \((u, v)\)

Displacement:

\[(\delta x, \delta y) = (u \delta t, v \delta t)\]
Color constancy, small and continuous motion

Canceling \(E(x,y,t)\):

\[
\frac{\partial E}{\partial x} \delta x + \frac{\partial E}{\partial y} \delta y + \frac{\partial E}{\partial t} \delta t = 0
\]

Dividing by \(\delta t\):

\[
\frac{\partial E}{\partial x} u + \frac{\partial E}{\partial y} v + \frac{\partial E}{\partial t} = 0
\]

The optical flow constraint equation (Horn and Schunck):

\[
E_x u + E_y v + E_t = 0
\]

The derivatives are computable but \((u,v)\) can’t be uniquely found.
Correspondence can be determined *between* brightness/color levelsets, but not *within* them
Computational Motion Analysis

The aperture problem
Computational Motion Analysis

Formal expression of the aperture problem – 2 unknowns but 1 equation

\[ E_x u + E_y v + E_t = 0 \]

We have a constraint line:

\[ v = -\frac{E_x}{E_y} u - \frac{E_t}{E_y} \]
Computational Motion Analysis

Combining the brightness assumption (the data term) with regularization

The error in the optical flow constraint:

\[ e_c = \iint_{\text{image}} (E_x u + E_y v + E_t)^2 \, dx \, dy \]

Smoothness constraint – the motion field varies smoothly in the image:

\[ e_s = \iint_{\text{image}} (u_x^2 + u_y^2) + (v_x^2 + v_y^2) \, dx \, dy \]

Find \((u,v)\) at each image point that minimizes:

\[ e = e_s + \lambda e_c \]

Weighting factor
Computational Motion Analysis
Computational Motion Analysis

Horn & Schunck 1981

Ground Truth

Sun et al., 2010
Object identification and recognition
Object identification and recognition

What does it mean to “recognize and object”?

- Retrieving information associated with an object that is not provided in the raw data (the image) itself
  - Name
  - Type
  - Class
  - Function
  - What would it do to me if it caught me

- Matching against a knowledge base (memory)
Object identification and recognition

Recognition as classification

- Hierarchical
  - My car
  - Jeep
  - Car
  - Vehicle
  - Man-made object
  - :

- Classification level depends on application or circumstances.
Object identification and recognition

How are objects recognized?

• Characteristic shape or structure
• Relative location
• Characteristic motion
• Color
• Texture
Object identification and recognition

Shape representation – viewer-centered templates
Object identification and recognition

Shape representation – viewer-centered feature vectors

\[(v_1, v_2, \ldots, v_n)\]
Object identification and recognition

Shape representation – viewer-centered feature vectors

\( (v_1, v_2, \ldots, v_n) \)
Object identification and recognition

Shape representation – object-centered components
Object identification and recognition

Main approaches to recognition

• Appearance-based
• Feature alignment
• Parts and structural matching
• Shape invariances
Appearance-Based Recognition

Recognition by appearance matching

DB

Direct comparison

?
Appearance-Based Recognition

Abstraction - images as vectors

\[ \Rightarrow \vec{v}_I \]

\[ \Rightarrow I \]
Appearance-Based Recognition

Abstraction - images as vectors

Normalized images $\|\vec{v}_I\| = 1$
Appearance-Based Recognition

Recognition by appearance matching

\[
\text{arg min}_{i} D(\vec{v}_{I_i}, \vec{v}_q) = \text{arg min}_{i} \| \vec{v}_{I_i} - \vec{v}_q \| \\
= \text{arg min}_{i} (\hat{v}_{I_i} \cdot \hat{v}_q)
\]
Appearance-Based Recognition

Problems in naive appearance matching

Variability due to pose/viewing
Appearance-Based Recognition

Problems in naive appearance matching

Variability due to pose/viewing
Appearance-Based Recognition

Problems in naive appearance matching

Image size $N \times N$ pixels, $B$ bytes/pixel

$N_V$ viewing directions per object

$N_L$ illumination directions per object

$N_o$ objects in database

\[
N_o \cdot B \cdot N^2 \cdot N_V \cdot N_L \quad \text{bytes}
\]

\[
\begin{align*}
N &= 256 \\
B &= 1 \\
N_V &= 64 \quad \Rightarrow \quad 12.5\text{GBytes} \\
N_L &= 32 \\
N_o &= 100
\end{align*}
\]
Appearance-Based Recognition

Problems in naive appearance matching

\[ \| \vec{v}_{I_i} - \vec{v}_q \| = \sqrt{\sum_{k=1}^{N^2} (\vec{v}_{I_i}(k) - \vec{v}_q(k))^2} \]

\[ \downarrow \]

\( O(N^2) \) operations / image

Cost of matching
Mobile Vision Miniproject
Mobile Vision Miniproject

What if you also had multiple cameras and controllable light source?

What if you have a camera and general purpose computing power that goes with you everywhere you want?
**Mobile Vision Miniproject**

- Recognition in the wild
- Programmable/smart camera
- Mobile vision games
- Social applications
- What could you do with a pair of wirelessly-connected mobile programmable cameras?
- And… wherever your imagination takes you…