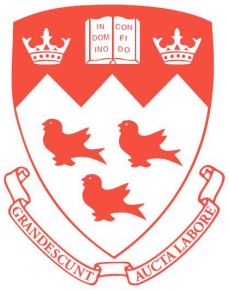
Fundamentals of Computer Vision

COMP 558 Course notes for Prof. Siddiqi's class.

taken by

Ruslana Makovetsky

(Winter 2012)

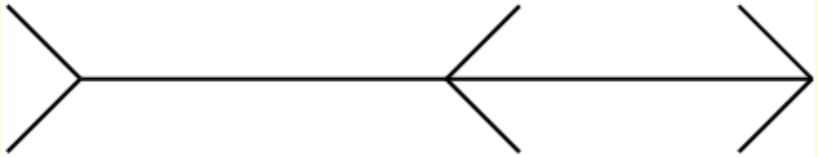


What is computer vision?

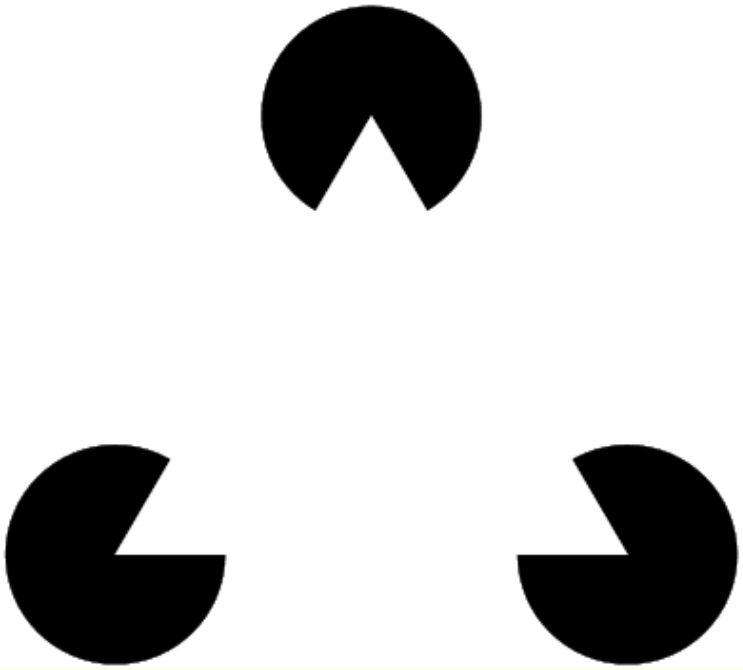
- Broadly speaking, it has to do with making a computer "see".
- The complexity of this task is enormous!
- It is about understanding visual input and making inferences about the visual world.
- What you see is NOT necessarily what is there!

Visual illusions

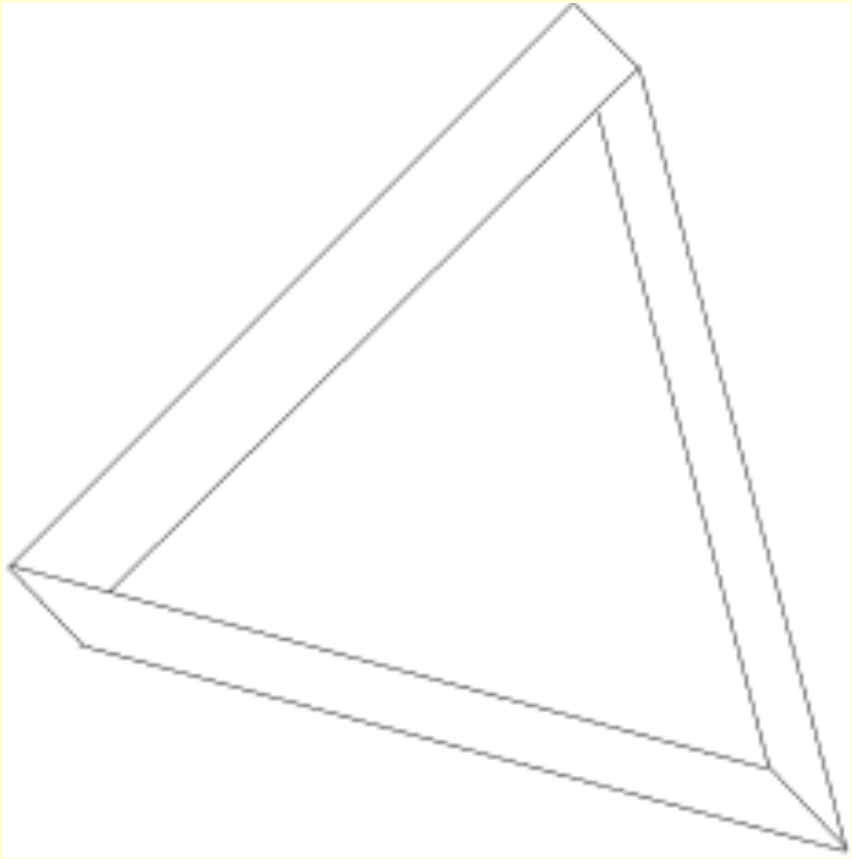
The Mueller–Lyer illusion



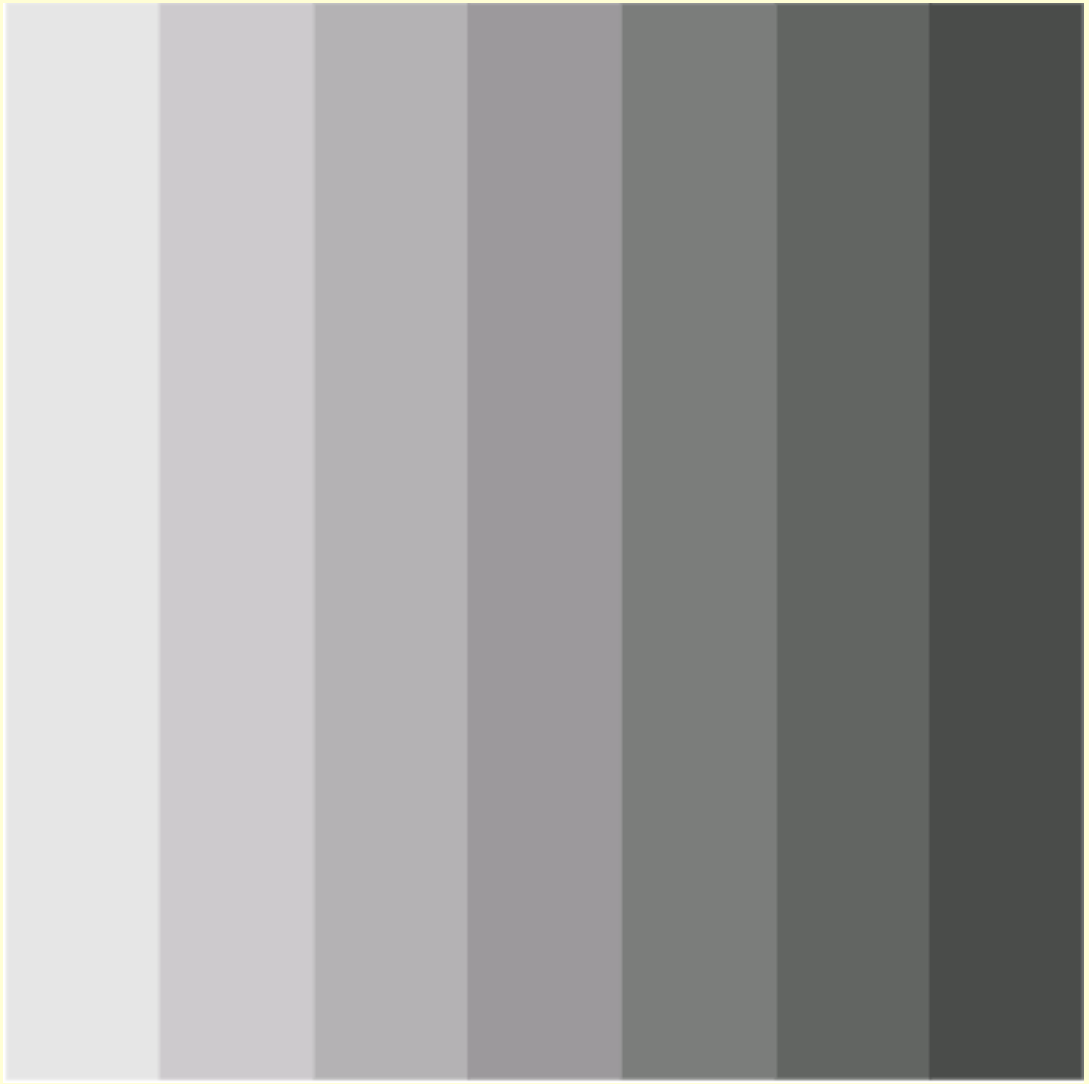
The Kanizsa Triangle



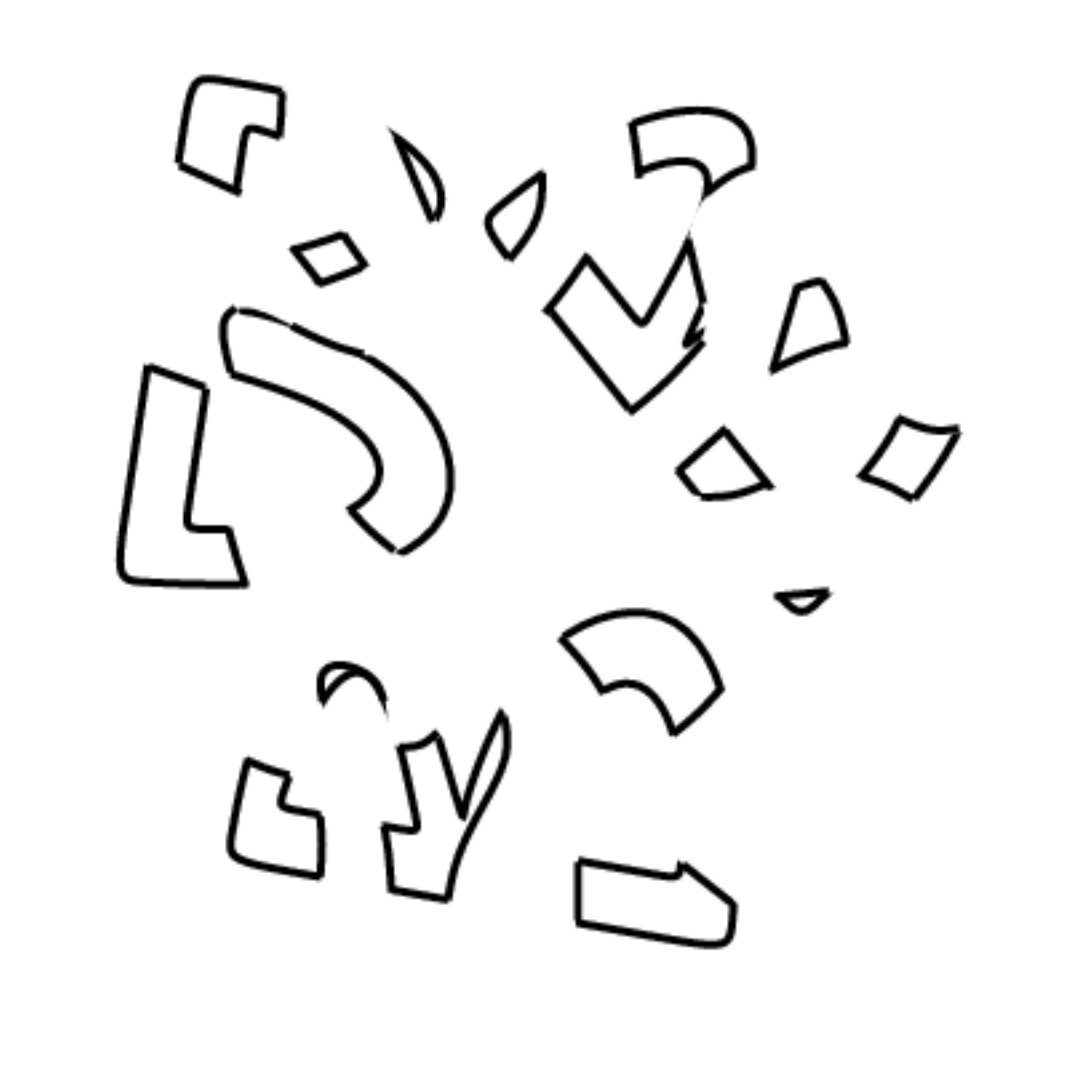
Visual illusions



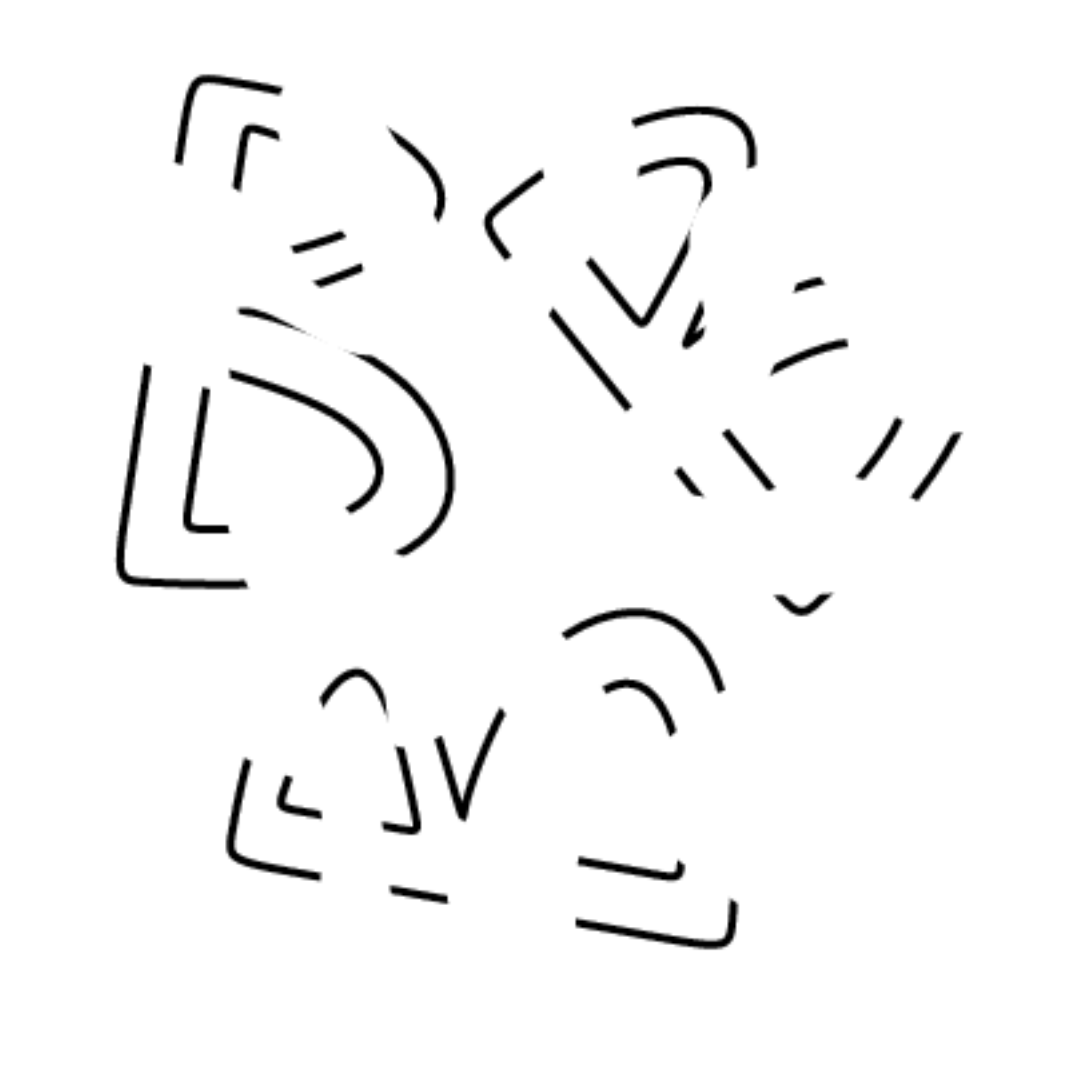
Visual illusions



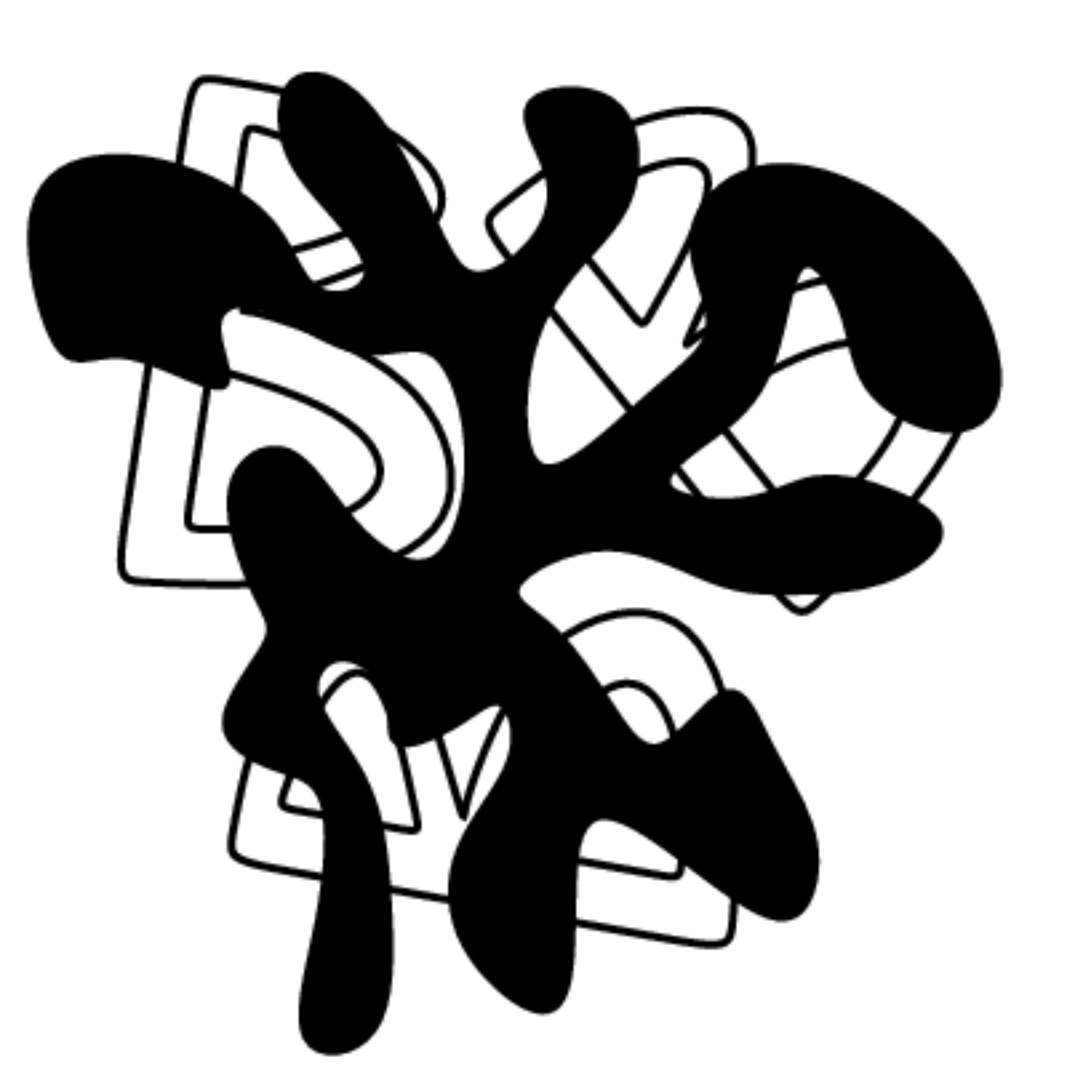
What do you see?



Is it clearer now?



Bregman B's



Why study computer vision?

- Scene reconstruction
- Object recognition and identification
- Object modeling
- Robot navigation
- Medical imaging
- Image-guided neurosurgery

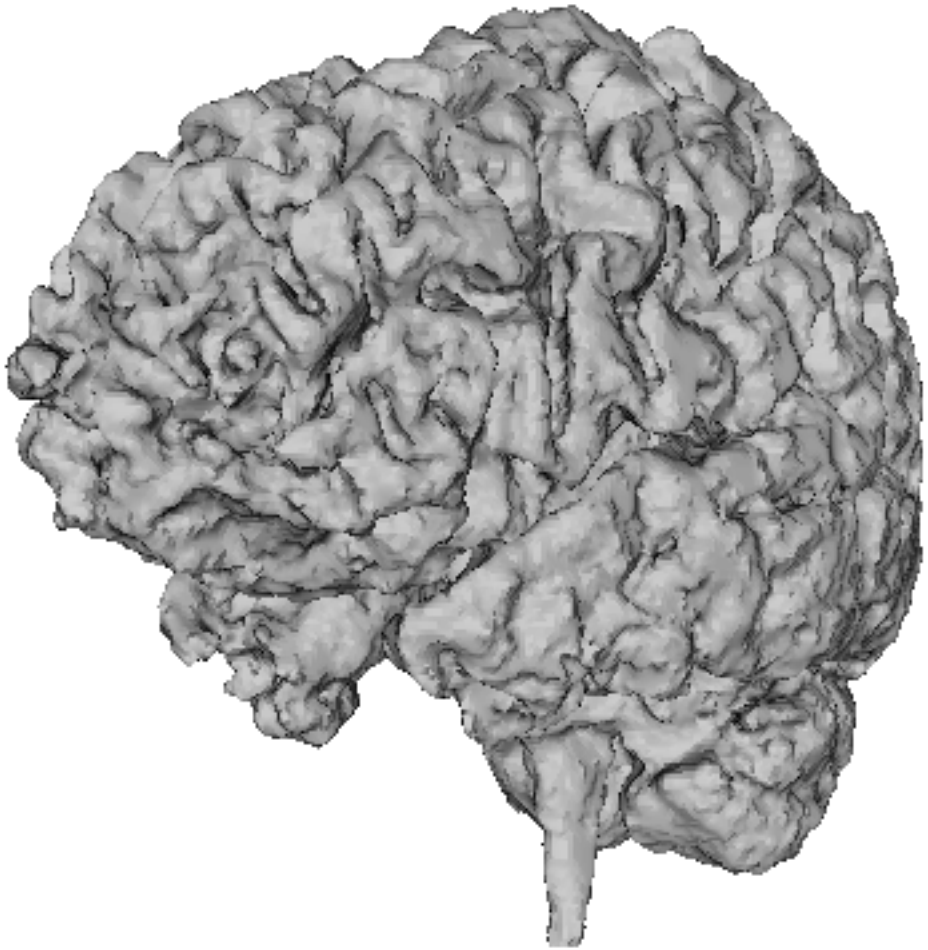
Aspects of computer vision

- Early vision: receptive fields, biological vision systems.
- Linear systems, signal processing, convolution.
- Edge detection, feature detection.
- Projective geometry and camera modeling.
- Stereo vision.

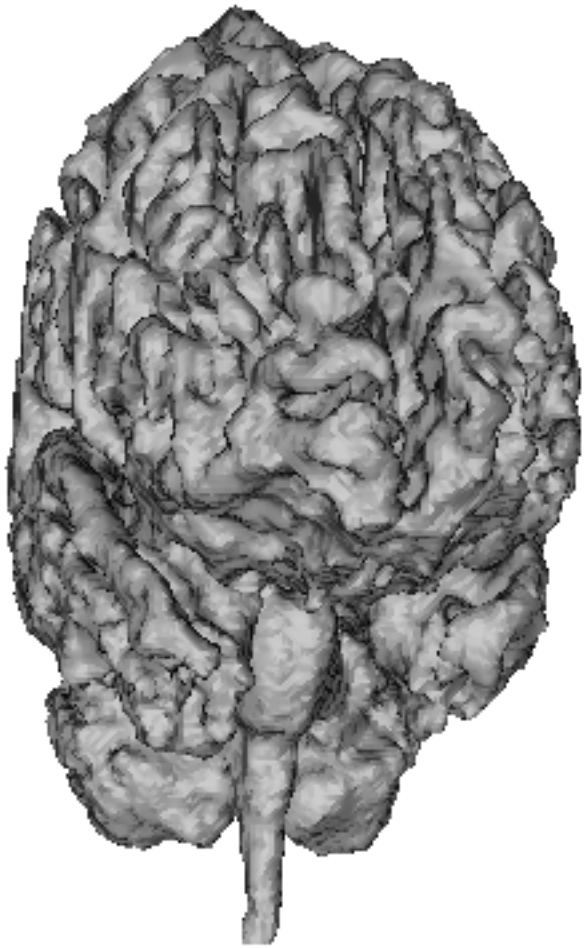
Aspects of computer vision

- Shape from shading and texture.
- Optical flow and motion analysis.
- Object representation and recognition.
- Graph-theoretic methods.
- PDEs in vision.
- Applications.

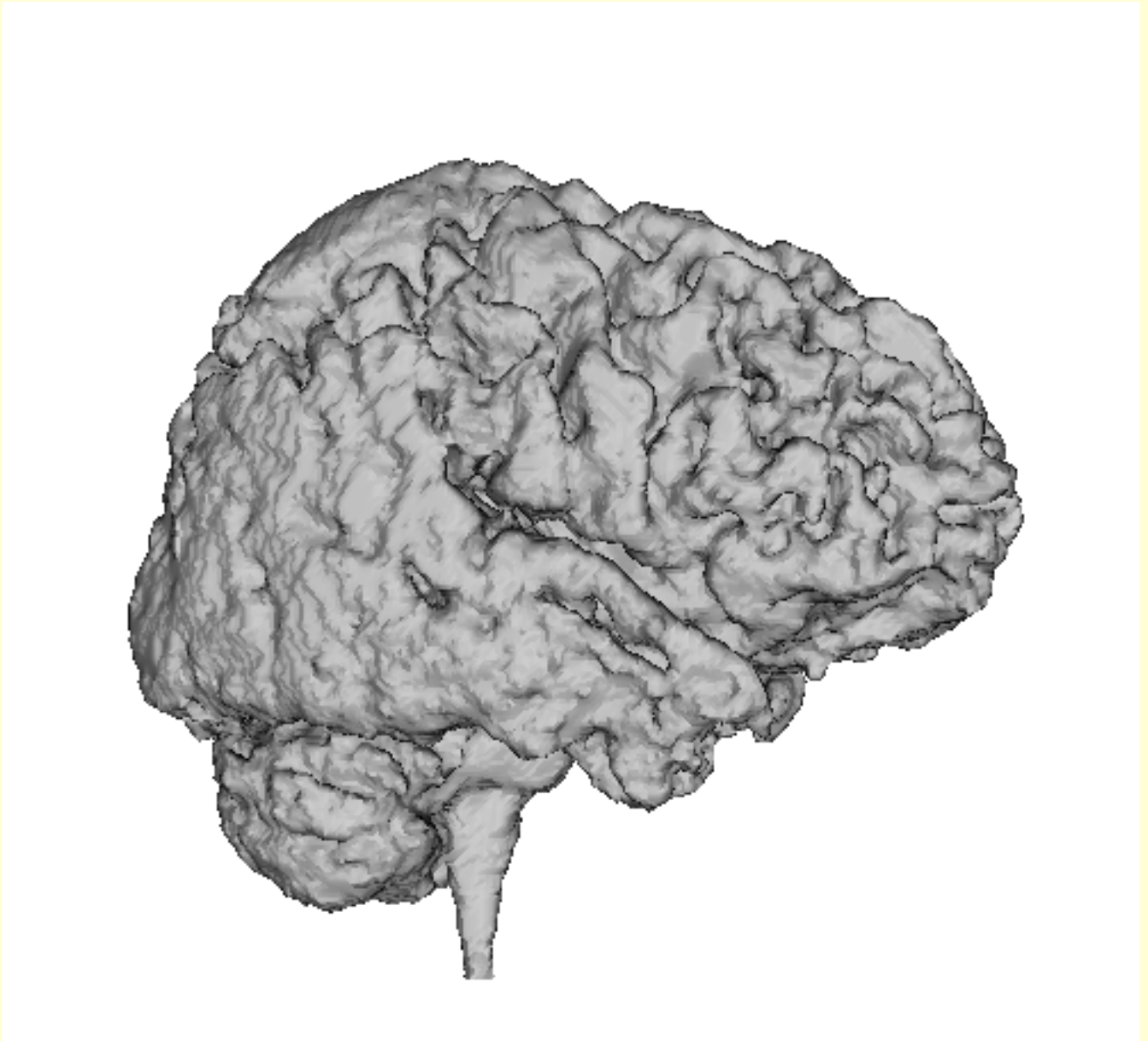
Brain Mechanisms of Vision



Brain Mechanisms of Vision



Brain Mechanisms of Vision



Cerebral Cortex

- It is the outermost layer of the cerebral hemispheres. Approx. 2mm thick, 1.5 square feet, highly folded, 10 billion neurons!
- Most sensory and motor areas of the cortex contain systematic 2D maps of the world represented. These maps are NOT spatially uniform. Much more cortex is dedicated to the fingers than to the rear-end. In vision the fovea has a representation which is approx. 35 more times more detailed than the periphery.
- Visual pathways cross so that the left half of the visual world projects to the right cerebral cortex and vice-versa.
- Operations in the visual cortex are LOCAL. In a deep sense this drives the local to global transition in computational vision.

Visual Pathway

- Messages from the light-sensitive elements, the rods and cones, traverse a few layers involving other retinal cell types (bipolar, horizontal, amacrine) before arriving at the retinal ganglion cells [See Nalwa, Chapter 1.] The rods facilitate night vision and the cones are responsible for color vision.
- 125 million rods and cones, so compaction is involved.
- The axons of the ganglion cells comprise about a million nerve fibers which are bundled together in the optic nerve.
- The optic nerve feeds to the lateral geniculate nuclei (LGN's),with partial crossing in the optic chiasm.
- The right halves of the retina project to the right LGN and right cortex, which represent the left part of the visual world (and vice-versa for the left halves of the retina).

A Receptive Field

- The mapping of the response of a neuron as a function of the amount of light incident on a retinal patch.
- Single cell recording paradigm:

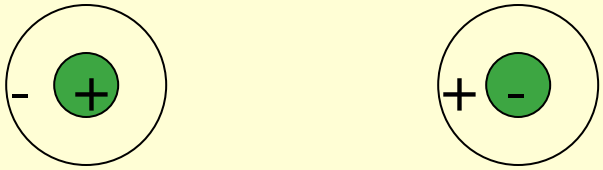
  Stimulate the retina with a pattern of light and probe the neuron with a microelectrode.

  Find the "optimal" pattern by trial and error and proceed to test hundreds of cells.

- Analogy to the impulse response of a linear time invariant system.
- Proceed from the fibers to the LGN to the primary visual cortex to deduce "function from structure".
- A receptive field is an abstraction.
- Retinotopic maps.

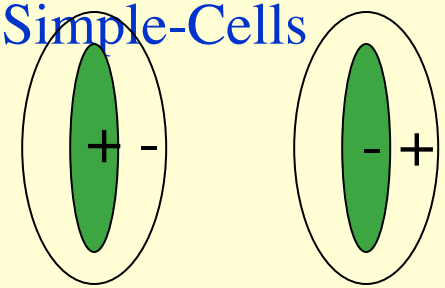
Receptive Field Types

- Center-Surround



Both retinal ganglion cells and LGN cells respond best to light of a particular size in a particular location of the retina.

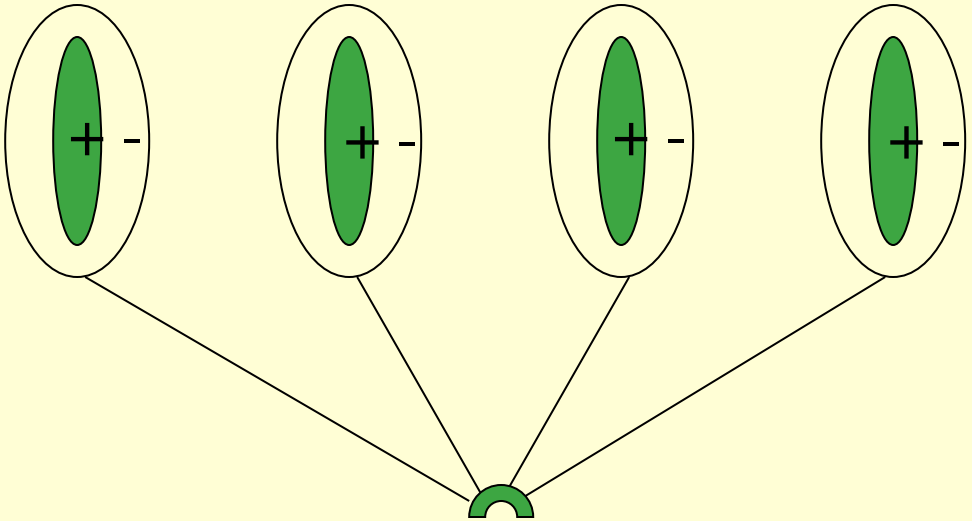
- LGN feeds into the lower part of layer IV (where the cells have the same receptive fields). LGN cells are monocular, with information from the left and right eyes kept separate.
- Simple-Cells



Cells outside layer IV behave like line detectors. They are orientation and location sensitive but are still monocular.

Receptive Field Types

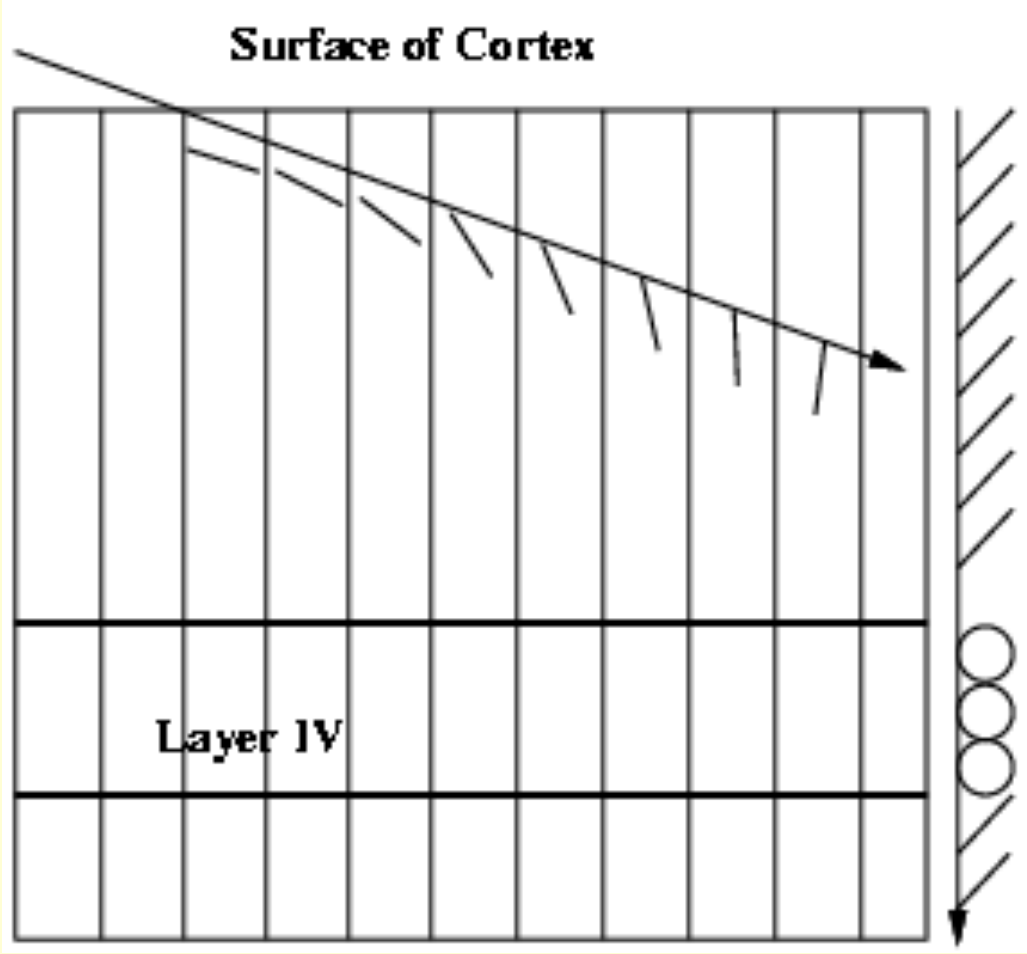
- Complex Cells



Not sensitive to exact location. Hence, they are involved in the processing of visual motion. Some stereo behavior is now observed.

Not sensitive to exact location.
Hence, they are

Orientation Columns



Approximately 10 degree shifts in orientation preference with an oblique penetration. No change in orientation preference or location for a vertical penetration (but change in size).

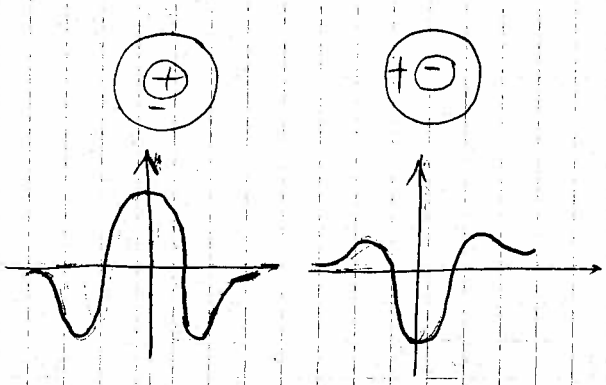
Lecture #3 (January 16)

Reading:

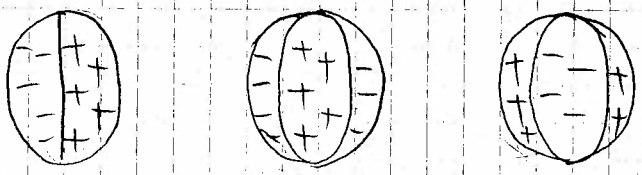
"Signals and Systems" - Chapters 2, 3, 4 (red)

Summary

- Both the retinal and LGN cells have centre-surround configuration



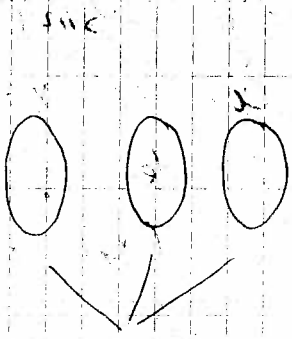
- The LGN fits into layer IV of visual cortex
- Outside Layer 4



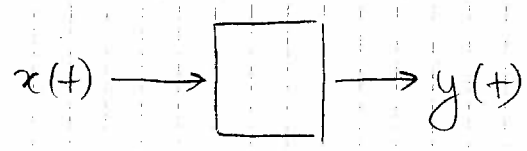
Simple cells - specific orientation and spatially localized
(orientation tuning is in increments of ~10°)

↓

Complex cells
(combination of simple cells of nearby locations)

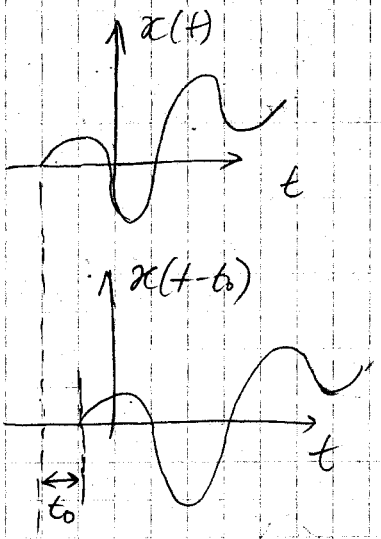


Linear System



We assume that system is **causal**, i.e.
Output at time t_0 depends only on inputs at times $t \leq t_0$.

Time Invariance (or Shift Invariance):



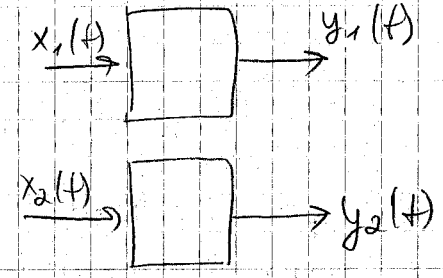
A time shift in the input signal causes corresponding shift of the output signal

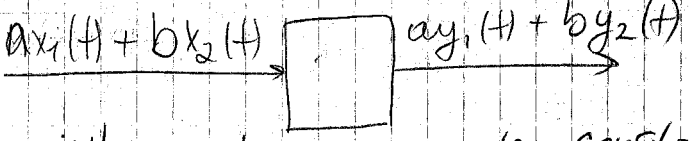
If $y(t)$ is output for $x(t)$
then $y(t-t_0)$ — // — $x(t-t_0)$

Linear system has a property of Superposition;

① Additivity

② Homogeneity





where a, b are complex consts

Examples.

Is $y(t) = \sin(x(t))$ time-invariant? Yes.

Consider $x_1(t) \rightarrow y_1(t) = \sin(x_1(t))$

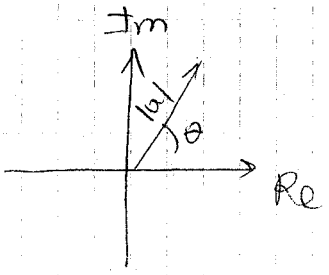
Consider $x_2(t) = x_1(t-t_0) \rightarrow y_2(t) = \sin(x_2(t)) =$

$= \sin(x_1(t-t_0)) = y_1(t-t_0)$

$y_2(t) = y_1(t-t_0)$

Complex Number

$$a = |a| e^{j\theta}$$



Real: $|a|\cos\theta$

Im $|a|\sin\theta$

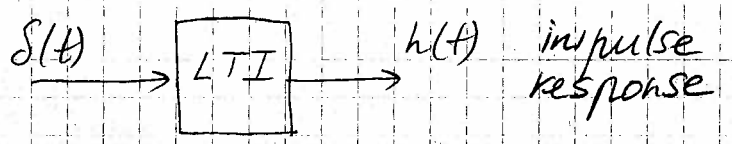
$$|a|\cos\theta + j|a|\sin\theta$$

Jan. 16

What is the basic <u>powerful</u> property of a linear time invariant (LTI) system?

Behavior is completely characterisid by transfer function, which can be easily measured.

(We can reproduce complex signal by knowing a response to simple signal only)



Delta function satisfies 3 basic assumptions

① $\delta(t-t_0) = 0, \quad t \neq t_0$

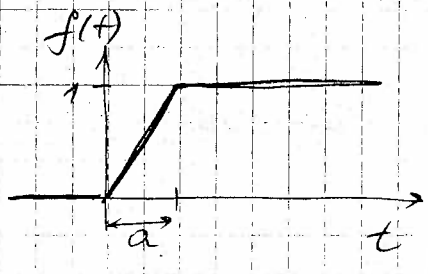
② $\int_{t_1}^{t_2} \delta(t-t_0)\,dt = 1 \qquad t_1 \leq t_0 \leq t_2$

(area under a δ function is 1)

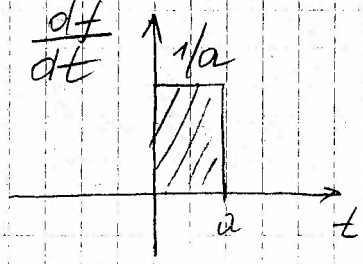
③ $\int_{-\infty}^{\infty} f(t)\,\delta(t-t_0)\,dt = f(t_0)$, $f(t)$ is continious at t_0

↓ Convolution

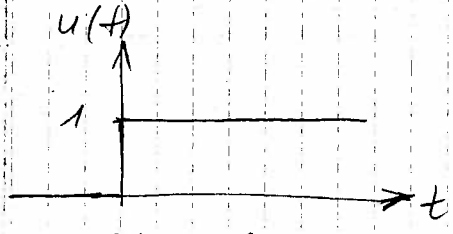
sifting property



Ramp function



$$\lim_{a\to 0}\left(\frac{df}{dt}\right) = \delta(t)$$



Step function

$$u(t) = \lim_{a\to 0} f(t)$$

$$\delta(t) = \lim_{a\to 0} \frac{df}{dt} = \frac{d}{dt}u(t)$$

LTI systems are fully characterised by their impulse ($\delta(t)$) response ($h(t)$)

Convolution theorem

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau)\, h(t-\tau)\, d\tau$$

- ~~Reflection~~
- Sliding

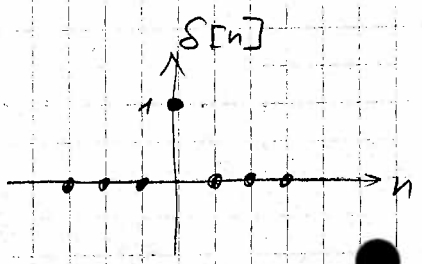
$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau) h(t-\tau)\, d\tau$$

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] h[n-k]$$

Any discrete signal can be expressed as a linear combination of shifted unit samples:

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\, \delta[n-k]$$

where $\delta[n] = \begin{cases} 1 & n=0 \\ 0 & \text{otherwise} \end{cases}$



Let $h_k[n]$ denote the response of a linear system to unit sample $\delta[n-k]$ with k fixed

Then, the response of the system to an arbitrary input $x[n]$, which itself is weighted sum of shifted samples is:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] h_k[n] \quad (*)$$

If the system is also time invariant then

$h[n] = h_0[n]$

$h_k[n] = h_0[n-k]$

(since $\delta[n-k]$ is a time shifted version of $\delta[n]$ the response $h_k[n]$ is a time shifted version of $h_0[n]$)

where $h[n] = h_0[n]$

Then (*) becomes

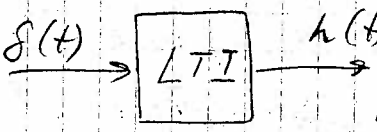
$$y[n] = \sum_{k=-\infty}^{\infty} x[k] h[n-k]$$

(This prooves the convolution theorem for the discrete case)

Lecture #4 (Jan. 18)

Review:

Linear systems (LTI) are completely characterized by their impulse response.



$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau)\, h(t-\tau)\, d\tau$$

convolution theorem

$H(\omega)$ transfer function

* Here, t is a constant, τ is integrated out.

end up with an expression depending only on t.

Notation:

$x(t)$ - continious
$x[n]$ - discrete

$$x(t) \xrightarrow[\text{Fourier Transform}]{\mathcal{F}} \int_{-\infty}^{\infty} x(t) e^{-j\omega t}\, dt$$

Fourier Transform

ω is the new, 'frequency domain' variable

$$\mathcal{F}[x(t)] = X(\omega)$$

$X(\omega)$ is defined in the frequency domain $|X(\omega)|$ and the phase of $X(\omega)$

$$\mathcal{F}^{-1}[X(\omega)] = x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t}\, d\omega$$

Inverse Fourier Transform

$\omega = 2\pi f$ common in el. eng. literature

$$x(t) * h(t) \underset{\text{Inverse Fourier transform}}{\overset{\text{Fourier transform}}{\rightleftarrows}} X(\omega) \cdot H(\omega)$$

$H(\omega) = \mathcal{F}[h(t)]$ Transfer function

* Convolution is very expensive, since it's global

Jan. 18

properties of Fourier Transformation

① Convolution property

$$x(t) * h(t) \longleftrightarrow X(\omega)\,H(\omega)$$

② Frequency domain convolution

$$x(t) \cdot y(t) \longleftrightarrow \frac{1}{2\pi} X(\omega) * H(\omega)$$

③ Frequency domain translation

$$x(t)e^{j\omega_0 t} \longleftrightarrow X(\omega-\omega_0)$$

$$\int_{-\infty}^{\infty} \underbrace{x(t)e^{j\omega_0 t}}, e^{-j\omega t}\,dt = \int_{-\infty}^{\infty} x(t)\,e^{-j(\omega-\omega_0)t}\,dt$$

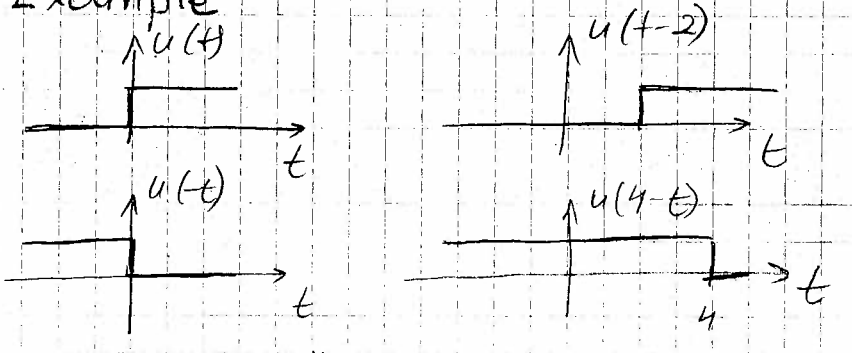
④ Scaling property

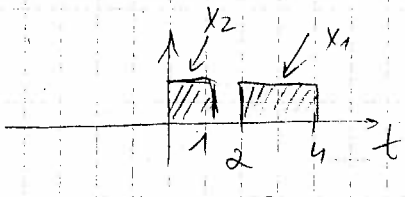
$$x(at) \longleftrightarrow \frac{1}{|a|} X\left(\frac{\omega}{a}\right)$$

⑤ Time delay

$$x(t-t_0) \longleftrightarrow X(\omega)\,e^{-j\omega t_0}$$

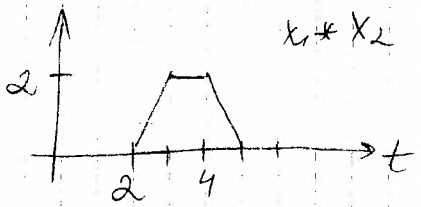
Example





$$x_2(t) = u(t)\,u(1-t)$$
$$x_1(t) = u(t-2)\,u(4-t)$$

$$x_1(t) * x_2(t) = \int_{-\infty}^{\infty} x_1(\tau)\,x_2(t-\tau)\,d\tau$$

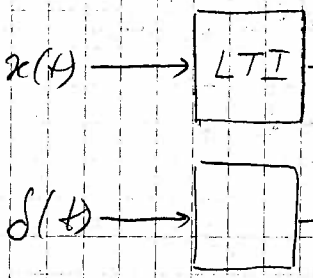


Lecture # 5 (Jan. 23)

Review

① LTI

② Convolution Theorem



$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau) h(t-\tau)\, d\tau$$

Fourier Transform

$$\mathcal{F}[x(t)] = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt = X(\omega)$$

* Java Applet for Convolution (on Calendar)

Any signal can be represented as a Fourie series, i.e. weighted combination of sines and cosines.

Fourier transforms of sines and cosines are <u>delta functions</u>!

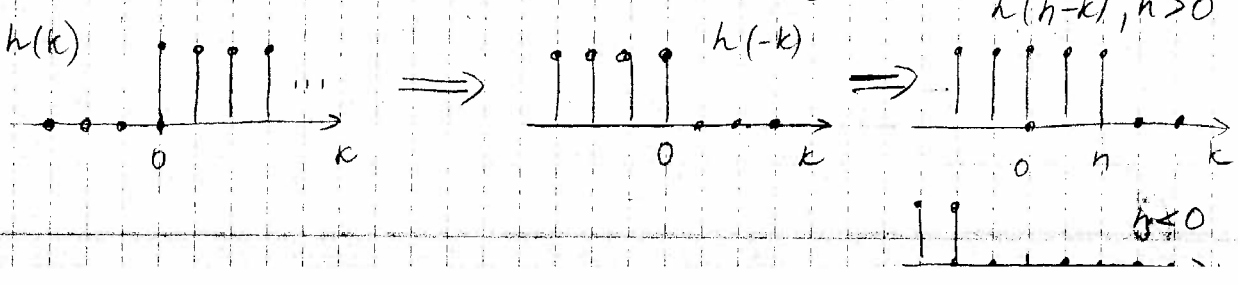
Example (discrete signal)

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]\, h[n-k]$$

$x[n] = \alpha^n u[n]$, $0 < \alpha < 1$

$h[n] = u[n]$ ← step function

(Ex. 3.7 from Oppenheint Willsky)



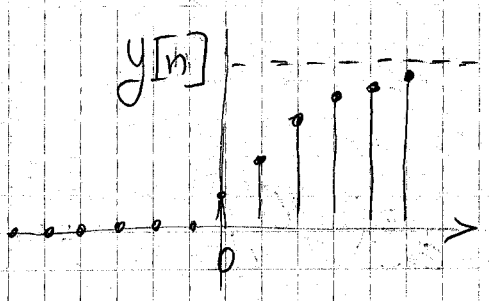
For $n<0$ there is no overlap between ~~$x(k)$ and $h(n-k)$~~ the non-zero points of $x(k)$ and $h(n-k)$, so $y(n)=0$ for $n<0$

For $n \geq 0$:

$$x[k]\,h[n-k] = \begin{cases} \alpha^k & 0 \leq k \leq n \\ 0 & \text{otherwise} \end{cases}$$

$$y[n] = \sum_{k=0}^{n} \alpha^k = \frac{1-\alpha^{n+1}}{1-\alpha}\,\underbrace{u[n]}$$

↑ to take care of $n<0$



Convolutions are expensive!

① reflection
② translation

Fourier Series

signal with finite energy

Let $f(t)$ be defined on an interval $-L \leq t \leq L$

The fourier series on $[-L, L]$ is given by

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(\frac{n\pi t}{L}\right) + b_n \sin\left(\frac{n\pi t}{L}\right)$$

$$a_0 = \frac{1}{L}\int_{-L}^{L} f(t)\,dt.$$

$$a_n = \frac{1}{L}\int_{-L}^{L} f(t)\cos\left(\frac{n\pi t}{L}\right)dt, \quad n=1,2,3,\ldots$$

$$b_n = \frac{1}{L}\int_{-L}^{L} f(t)\sin\left(\frac{n\pi t}{L}\right)dt, \quad n=1,2,3,\ldots$$

Filtering - removing certain unwanted frequencies

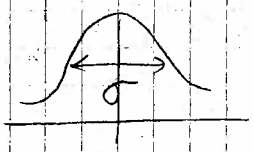
* To remove noise, Low-pass bandwidth is used

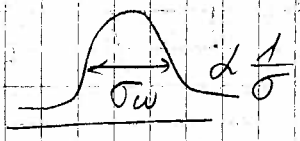
→ signal is transformed to the frequency domain,

Jan. 23 high frequencies (noise) are removed, and then the signal is transformed back. This will result is a smooth signal (noise removed).

Filters types:
- low pass
- high pass
- band pass

The only signal, whose Fourier transform is Gaussian, is Gaussia (i.e. Gaussian $\xrightarrow{\mathcal{F}}$ Gaussian)



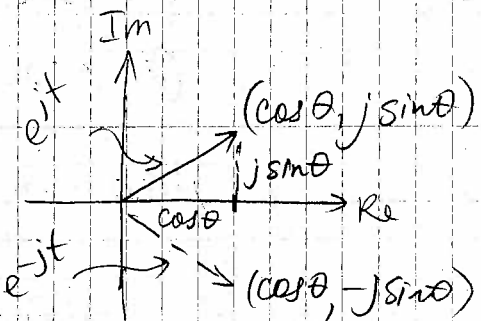


low-pass filter

Goal:

Prove that Fourier Transform of cosine or a sine is a delta function.

$\cos(\omega_0 t)$



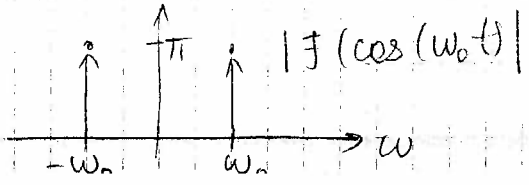
$$\frac{e^{j\theta}+e^{-j\theta}}{2} = \cos\theta$$

$$\frac{e^{j\theta}-e^{-j\theta}}{2j} = \sin\theta$$

Theorem

$$\mathcal{F}[\cos(\omega_0 t)] = \pi[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]$$

$$\mathcal{F}[\sin(\omega_0 t)] = -j\pi[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]$$



Modulation Theorem (seen last time)

$$\mathcal{F}[e^{j\omega_0 t}x(t)] = \int_{-\infty}^{\infty} x(t)e^{-j(\omega-\omega_0)t}dt = X(\omega-\omega_0)$$

$$\mathcal{F}[1] = ?$$

$$= \lim_{\alpha\to 0}\int_{-\infty}^{\infty} e^{-\alpha|t|}e^{j\omega t}dt =$$

$$= \lim_{\alpha\to 0}\left[\int_{-\infty}^{0} e^{(\alpha-j\omega)t}dt + \int_{0}^{\infty} e^{-(\alpha+j\omega)t}dt\right] =$$

$$= \lim_{\alpha\to 0}\left[\frac{e^{(\alpha-j\omega)t}}{\alpha-j\omega}\Big|_{-\infty}^{0} + \frac{e^{-(\alpha+j\omega)t}}{-(\alpha+j\omega)}\Big|_{0}^{\infty}\right] =$$

$$= \lim_{\alpha\to 0}\frac{2\alpha}{\alpha^2+\omega^2}$$

① case $\omega \neq 0$ $\quad \mathcal{F}[1] = 0$

② case $\omega = 0$ $\quad \mathcal{F}[1] = \lim_{\alpha\to 0}\frac{\alpha}{2\alpha+\omega} = \infty$ (*)

Area under $\mathcal{F}[1] = \lim_{\alpha\to 0}\int\frac{2\alpha}{\alpha^2+\omega^2}d\omega =$

$$= \lim\left(2\tan^{-1}\left(\frac{\omega}{\alpha}\right)\Big|_{-\infty}^{\infty}\right) = 2\left(\frac{\pi}{2}+\frac{\pi}{2}\right) = 2\pi$$ (**)

(*) + (**) : $\mathcal{F}[1] = 2\pi\delta(\omega)$

Theorem's proof:

① $\mathcal{F}\left[\begin{matrix}\cos\\ \sin\end{matrix}(\omega_0 t)\right] = \mathcal{F}\left[\frac{e^{j\omega_0 t} \pm e^{-j\omega_0 t}}{2j}\right] =$

$$= \mathcal{F}\left[\frac{e^{j\omega_0 t}}{2j}\cdot 1\right] + \mathcal{F}\left[\frac{-e^{-j\omega_0 t}}{2j}\right] =$$

$$= \frac{1}{2j}\cdot 2\pi\left[\delta(\omega-\omega_0) + \delta(\omega+\omega_0)\right] = j\pi\left[\delta(\omega-\omega_0 + \delta(\omega+\omega_0)\right]$$

Lecture #6 (Jan. 25)

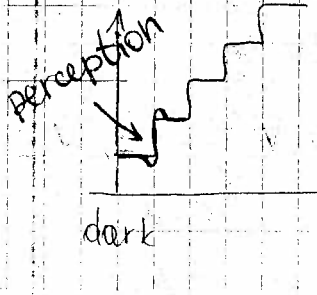
Announcements:

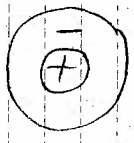
1. Assignment 1
2. Read the Marr-Holdest paper (edge detection)
3. Chapters 2,3 (Trucco and Verri - Grean)

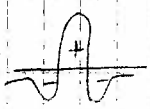
Project ideas:

1. Shape from Shading
2. Stereo → n-dimentional stereo
3. Motion or optical Flow
4. Photo stitching (panarama)
5. Object tracking
6. → Active contours or surfaces - deformable models
7. Object recognition
8. Face recognition
9. Lighting - Illumination

Mach Bands (review)



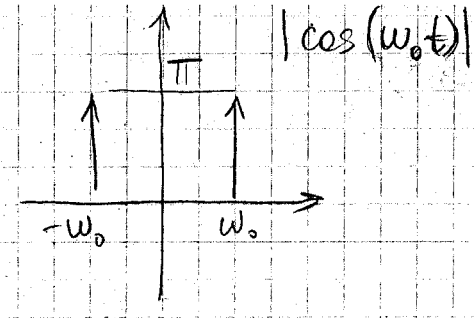


Convolve a step edge with 

(exhibition)
→ enhance contrast on edges

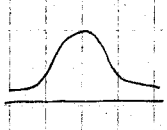
Review

1. Any continuos signal on finite interval can be represented as weighted combination of sines and cosines.
2. Convolution is a linear operation $\xleftrightarrow{\mathcal{F}}$ Faster Multiplication
3. The frequency domain leads to delta functions.



Questions we want to answer today

① When you convolve a signal with a Gaussian what is an interpretation in the frequency domain?



② What does this have to do with edge detection

contour drawings are often sufficient for recognition

Edge detection

- take derivatives → problem of noise (amplified)
- convolve with a kernel, then take derivatives and find where the derivatives are high (related to a Gaussian)

③ Narrow Gaussian = delta function, and convolving with signal we get the signal back.

The wider the Gaussian is, convolution blurs (smoothes) the signal more.

Marr-Hildeth

① Blur with the Gaussian

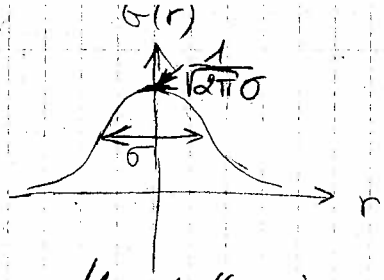
② Take first derivative

Find places where it is locally maximum

③ Zero-crossings of a second derivatives

Jan. 25

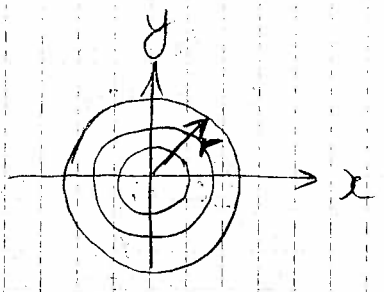
$$G(r) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{r^2}{2\sigma^2}}$$



Area under the Gaussian has the following properties:

① 95% is within 2 σ's

② 98% — 11 — 3 — 11 —

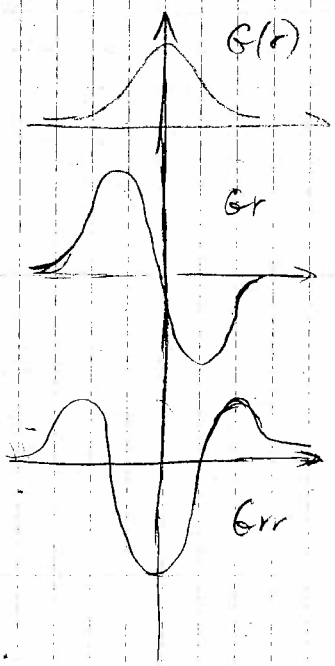


Think of 2-D Gaussians by mapping distance from the origin as r

We are going to proove that the Fourier transform of Gaussian is Gaussian.

$$G_r(r) = -\frac{r}{\sqrt{2\pi}\sigma} e^{-\frac{r^2}{2\sigma^2}}$$

$$G_{rr}(r) = -\frac{1}{\sqrt{2\pi}\sigma^3} e^{-\frac{r^2}{2\sigma^2}} \left[1 - \frac{r^2}{\sigma^2}\right]$$



Consider $G(x) = e^{-ax^2}$

$$\mathcal{F}[G(x)] = \int_{-\infty}^{\infty} e^{-ax^2} e^{-j\omega x} dx$$

Frequency: Hz or number of cycles/sec

f, where $\omega = 2\pi f$

and $f = \frac{1}{T}$ (T - period of the signal)

$$\mathcal{F}[G(x)] = \int_{-\infty}^{\infty} e^{-ax^2} e^{-j2\pi f x} dx$$

Hint: $e^{-j2\pi fx} = \cos(2\pi fx) - j\sin(2\pi fx)$

$$\Rightarrow \int_{-\infty}^{\infty} e^{-ax^2} \cos(2\pi f x)\,dx - j\underbrace{\int_{-\infty}^{\infty} e^{-ax^2} \sin(2\pi f x)\,dx}_{0}$$

① sin is odd
② e^{-ax^2} is symmetric

$$= \int_{-\infty}^{\infty} e^{-ax^2} \cos(2\pi f x)\,dx =$$

$$= \sqrt{\frac{\pi}{a}}\, e^{-\pi^2 f^2/a}$$

For normalized Gaussian $a = \frac{1}{2\sigma^2}$ and there is a constant in front of $\frac{1}{\sqrt{2\pi}\,\sigma}$

Fourier transform is

$$\frac{1}{\sqrt{2\pi}\,\sigma}\sqrt{2\pi\sigma^2}\, e^{-\pi^2 f^2 2\sigma^2} = e^{-\pi^2 f^2 2\sigma^2} =$$

$$\underset{\omega = 2\pi f}{=} e^{-\frac{1}{2}\sigma^2\omega^2} = e^{-\frac{1}{2}\frac{\omega^2}{\sigma_\omega^2}} \quad \text{where } \sigma_\omega = \frac{1}{\sigma}$$

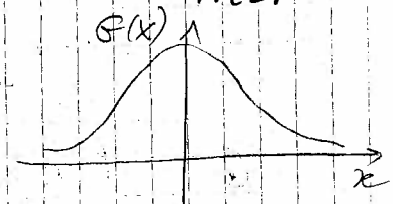
Lecture #7 (Jan. 30)

A1:

- Convolve 2D in Matlab
- Frequency Domain FFT → complex variable
- Problem 3 (Q2) – Mar-Hildreth paper! summarize the argument

Main ideas from the paper:

- Take derivatives of "smoothed" images
  convolution with a Gaussian ⟷ Low pass Filter



- Gaussian has the property that its Fourier Transform is a Gaussian.

$$\sigma_\omega \propto \frac{1}{\sigma}$$

spread in freq. domain; spread of original Gaussian

- Zero-crossings of a second derivative - 'like' process should give edges.

What is the challenge of computing second derivatives of a 2D image?
Need to define direction!!! (Humans have discretizing by 10° orientations)

$I(x,y)$ is the 'image'

second derivative → $D^2[G_\sigma * I(x,y)] = [D^2 G_\sigma * I(x,y)]$ ← zero-crossings of this!

Laplacian is a principled way to compute second derivatives

$$\Delta G(x,y) = G_{xx} + G_{yy}$$

where x, y are 2 ⊥ directions (rotation invariant)

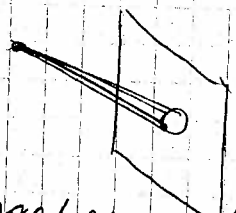
She had it last lecture with polar coordinates

This only holds if <u>linear variation</u> condition holds.

Image formation

Aperture problem

We'd like to have a hole diameter → 0. However, it's not possible, since no light will go in. Thus we need finite size aperture, but light direction is not unique.



Read <u>chapter 2</u> of background reading.

① Perspective Projection

By similar triangles

✓ $x_i = \frac{f}{z_0} x_0$

✓ $y_i = \frac{f}{z_0} y_0$

f - focal length of the camera (easy)

z_0 - 3D depth of the object point (completely unknown)

(x_0, y_0, z_0) - object coordinates in a world

2 equation in 3 variables, that is why the vision is a hard problem

② Orthographic Projection

$$\frac{f}{z_0} = 1$$

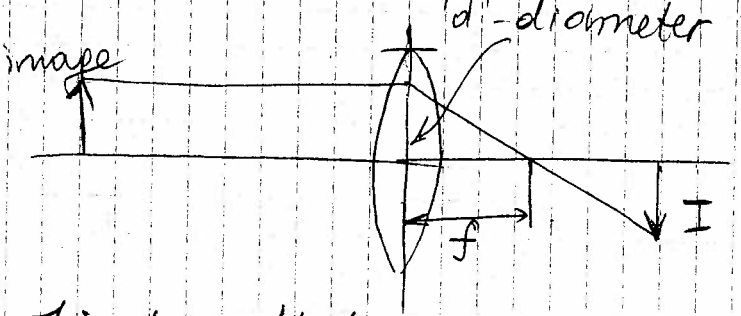
What if the object was 'far away' and relatively small?

③ We assume that $\frac{f}{z_0}$ is a constant → weak perspective (which is orthographic followed by scaling)

(object appears flat, changes in depth are negligible)

Jan. 30

④ Need a lens!



⑤ Thin Lens Model

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v}$$

→ Homogeneous coordinates - will need them later

- Optical Parameters (of camera)
- Photometric Parameters (light, reflection)
- Geometric Parameters (camera placement over scene)

$$E(p) = L(P)\frac{\pi}{4}\left(d/f\right)^2 \cos^4\alpha$$

$E(p)$: we measure this from the image

α: angular offset (angle between optical axis and light direction)

Lecture #8 – Feb. 1

Homogeneous Coordinates

Map each 3D point to a line in 4D using a dummy variable

$$\underbrace{(X, Y, Z, W)}_{\text{new coord.}} = (wx, wy, wz, w) \qquad w \neq 0$$

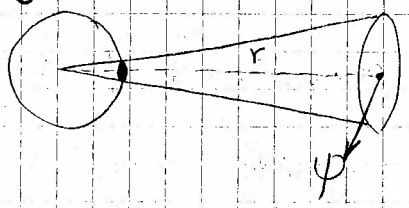
(x, y, z) is a 3D point

$$(x, y, z) = \left(\frac{X}{w}, \frac{Y}{w}, \frac{Z}{w}\right)$$

In Figure 2.7 the goal is to express the amount of light incident per unit area of the image, image irradiance ($E(\vec{p})$) in terms of scene quantities and scene radiance ($L(\vec{p})$), also called reflectance
Here, f is a focal length, d is a diameter of the lens
Z is a distance from the lens to scene point (P)

For <u>Lambertian</u> surfaces: $L(\vec{p}) \propto \vec{n} \cdot \vec{l}$

solid angle stands for <u>area</u>



$$E(\vec{p}) = L(\vec{p}) \cos^4\alpha \left(\frac{d}{f}\right)^2 \frac{\pi}{4}$$

See handouts!!!

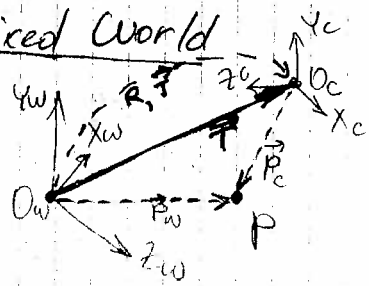
treat as a constant

How to relate 'scene' points with 'image' points?

① Computer vision systems assume a <u>fixed world coordinate system</u>, by fixing

O_w, $\underbrace{X_w, Y_w, Z_w}_{\text{axes}}$

origin



② Individual cameras have their own camera coordinate systems

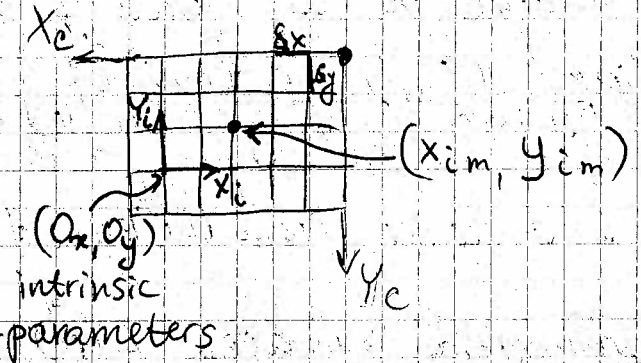
③ The two are related by a 3×3 rotation matrix R and a translation vector $\vec{T}$

$\vec{P}_c$ as a function of $\vec{P}_w$

$$\vec{P}_c = R(\vec{P}_w - \vec{T}) \quad (1)$$

$R, \vec{T}$ are extrinsic parameters

In Trucco and Verri text horizontal and vertical axes of the image and the camera reference frames are <u>oppositely orientated.</u>



(O_x, O_y) - coordinates of the image reference frame in the camera reference system

$\Delta x, \Delta y$ are the dimensions of the pixel

Last intrinsic parameter is a focal length, f

$(O_x, O_y, \delta x, \delta y, f)$ intrinsic parameters of the camera

Coordinates of $\vec{P}$ in the camera reference frame are given by:

$$x = -(x_{im} - O_x)\delta_x$$
$$y = -(y_{im} - O_y)\delta_y \quad (2)$$

Remember that under perspective projection

$x = \frac{f}{z} X$ and $y = \frac{f}{z} Y$

Plug (1) and (2) into this, we get

$$-(x_{im} - O_x)\delta_x = f \frac{R_1^T (\vec{P}_w - \vec{T})}{R_3^T (\vec{P}_w - \vec{T})}$$

Feb. 1

$$-(y_{im} - o_x) s_y = f \frac{R_2^T (\vec{P_w} - \vec{T})}{R_3^T (\vec{P_w} - \vec{T})}$$

(R_i is the i-th row of R)

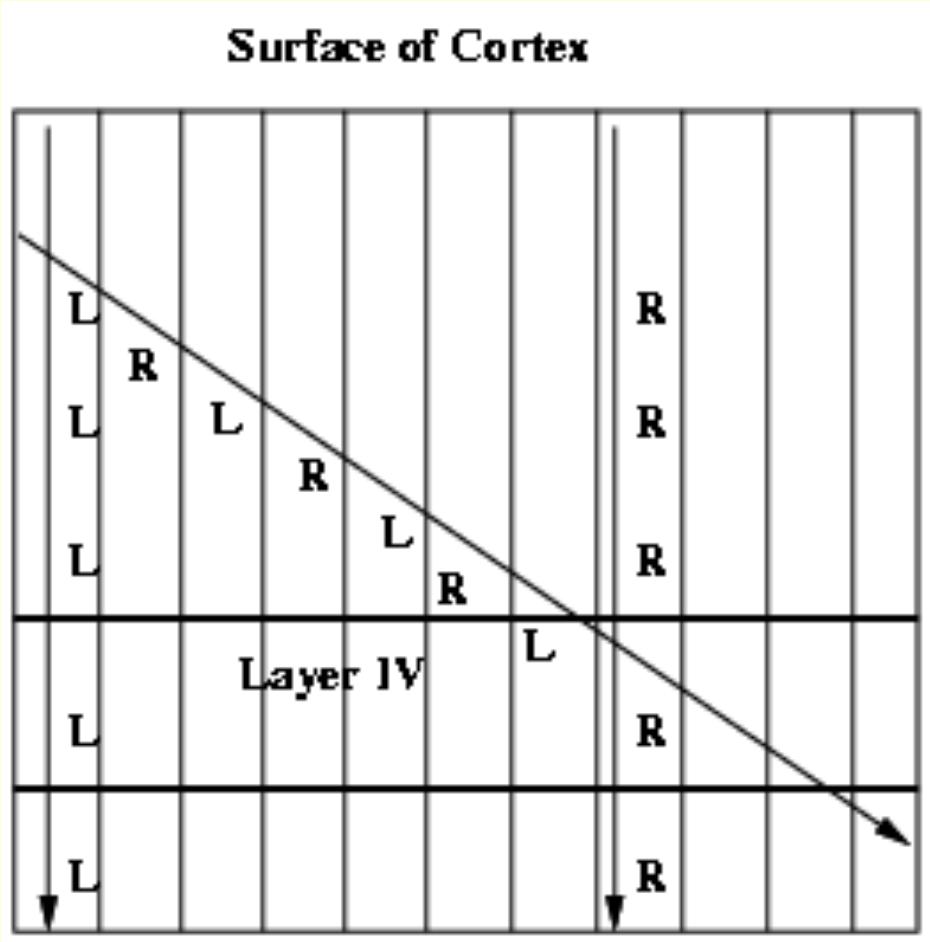
Note, that $\frac{s_x}{f}$ and $\frac{s_y}{f}$ can be treated together

Let

$$M_{int} = \begin{pmatrix} -f/s_x & 0 & o_x \\ 0 & -f/s_y & o_y \\ 0 & 0 & 1 \end{pmatrix}$$

$$M_{ext} = \begin{pmatrix} & & \\ & & \\ & & \end{pmatrix}$$

Ocular Dominance Columns



Vertical penetrations reveal columns which receive most of their input from one eye, but not the other. Staining techniques have confirmed the anatomy.

Lecture #8

sometimes *radiance* refers to the energy radiated from a surface (emitted or reflected), whereas *irradiance* refers to the energy incident on a surface.

Surface Reflectance and Lambertian Model. A model of the way in which a surface reflects incident light is called a *surface reflectance model*. A well-known one is the *Lambertian model*, which assumes that each surface point appears equally bright from all viewing directions. This approximates well the behavior of rough, nonspecular surfaces, as well as various materials like matte paint and paper. If we represent the direction and amount of incident light by a vector $\mathbf{I}$, the scene radiance of an ideal Lambertian surface, L, is simply proportional to the dot product between $\mathbf{I}$ and the unit normal to the surface, $\mathbf{n}$:

SCENE RADIANCE

$$L = \rho \mathbf{I}^T \mathbf{n} \tag{2.4}$$

with $\rho > 0$ a constant called the surface's *albedo*, which is typical of the surface's material. We also assume that $\mathbf{I}^T \mathbf{n}$ is *positive*: that is, the surface faces the light source. This is a necessary condition for the ray of light to reach $\mathbf{P}$. If this condition is not met, the scene radiance should be set equal to 0.

surface and light source

We will use the Lambertian model in several parts of this book: for example, while analyzing image sequences (Chapter 8) and computing shape from shading (Chapter 9). Intuitively, the Lambertian model is based on the exact cancellation of two factors. Neglecting constant terms, the amount of light reaching *any* surface is always proportional to the cosine of the angle between the illuminant and the surface normal $\mathbf{n}$ (that is, the effective area of the surface as seen from the illuminant direction). According to the model, a Lambertian surface reflects light in a given direction $\mathbf{d}$ proportionally to the cosine of θ, the angle between $\mathbf{d}$ and $\mathbf{n}$. But since the surface's area seen from the direction $\mathbf{d}$ is inversely proportional to $\cos\theta$, the two $\cos\theta$ factors cancel out and do not appear in (2.4).

Linking Surface Radiance and Image Irradiance. Our next task is to link the amounts of light reflected by the surfaces, L, and registered by the imaging sensor, E.

Assumptions and Problem Statement

Given a thin lens of diameter d and focal length f, an object at distance Z from the lens, and an image plane at distance Z' from the lens, with f, Z, and Z' as in (2.1), find the relation between image irradiance and scene radiance.

In order to derive this fundamental relation, we need to recall the geometric notion of *solid angle*. The solid angle of a cone of directions is the area cut out by the cone on the unit sphere centered in the cone's vertex. Therefore, the solid angle $\delta\omega$ subtended by a small, planar patch of area δA at distance r from the origin (Figure 2.6) is

solid angle

$$\delta\omega = \frac{\delta A \cos\psi}{r^2} \tag{2.5}$$

(1) define $\delta\omega$ and ask why r^2?

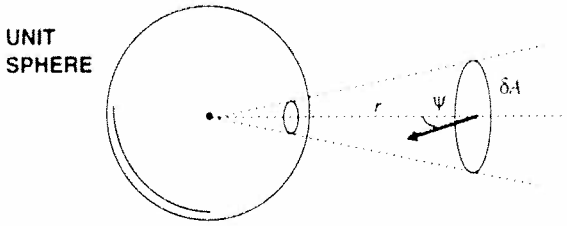


Figure 2.6 The definition of solid angle.

with ψ the angle between the normal to δA and the ray that points from the origin to δA. The factor $\cos\psi$ ensures the proper foreshortening of the area δA as seen from the origin.

We now write the image irradiance at an image point, **p**, as the ratio between δP, the power of light over a small image patch, and δI, the area of the small image patch:

(2) Define image irradiance

$$E = \frac{\delta P}{\delta I}. \tag{2.6}$$

If δO is the area of a small surface patch around **P**, L the scene radiance at **P** in the direction toward the lens, $\Delta\Omega$ the solid angle subtended by the lens, and θ the angle between the normal to the viewed surface at **P** and the principal ray (Figure 2.7), the power δP is given by $\delta O L \Delta\Omega$ (the total power emitted in the direction of the lens) multiplied by $\cos\theta$ (the foreshortening of the area δO as seen from the lens):

(3) Explain all the terms in 2.7

$$\delta P = \delta O L \Delta\Omega \cos\theta. \tag{2.7}$$

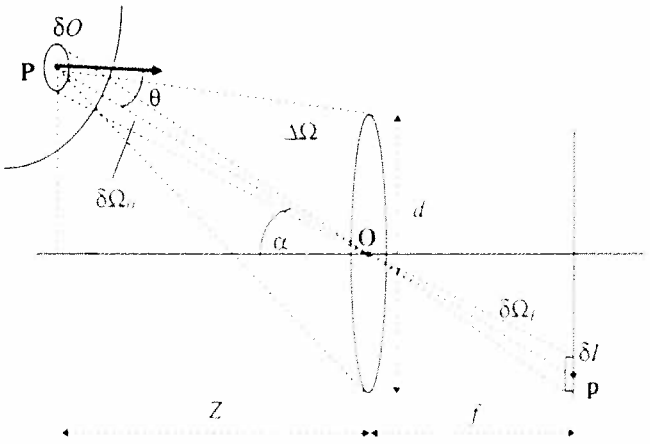


Figure 2.7 Radiometry of the image formation process.

Combining (2.6) and (2.7), we find

$$E = L\Delta\Omega\cos\theta\frac{\delta O}{\delta I}. \tag{2.8}$$

We still need to evaluate $\Delta\Omega$ and $\delta O/\delta I$. For the solid angle $\Delta\Omega$ (Figure 2.7), (2.5) with $\delta A = \pi d^2/4$ (lens area), $\psi = \alpha$ (angle between the principal ray and the optical axis), and $r = Z/\cos\alpha$ (distance of P from the lens center) becomes

$$\Delta\Omega = \frac{\pi}{4}d^2\frac{\cos^3\alpha}{Z^2}. \tag{2.9}$$

For the solid angle $\delta\Omega_I$, subtended by a small image patch of area δI (see Figure 2.7), (2.5) with $\delta A = \delta I$, $\psi = \alpha$, and $r = f/\cos\alpha$ gives

$$\delta\Omega_I = \frac{\delta I\cos\alpha}{(f/\cos\alpha)^2}. \tag{2.10}$$

Similarly, for the solid angle $\delta\Omega_O$ subtended by the patch δO on the object side, we have

$$\delta\Omega_O = \frac{\delta O\cos\theta}{(Z/\cos\alpha)^2}. \tag{2.11}$$

It is clear from Figure 2.7 that $\delta\Omega_I = \delta\Omega_O$; hence, their ratio is 1, so that dividing (2.11) by (2.10) we obtain

$$\frac{\delta O}{\delta I} = \frac{\cos\alpha}{\cos\theta}\left(\frac{Z}{f}\right)^2 \tag{2.12}$$

Ignoring energy losses within the system, and plugging (2.9) and (2.12) into (2.8), we finally obtain the desired relation between E and L.

The Fundamental Equation of Radiometric Image Formation

$$E(\mathbf{p}) = L(\mathbf{P})\frac{\pi}{4}\left(\frac{d}{f}\right)^2\cos^4\alpha. \tag{2.13}$$

Equation (2.13) says that *the illumination of the image at* **p** *decreases as the fourth power of the cosine of the angle formed by the principal ray through* **p** *with the optical axis*. In the case of small angular aperture, this effect can be neglected; therefore, the image irradiance can be regarded as uniformly proportional to the scene radiance over the *whole* image plane.

The nonuniform illumination predicted by (2.13) is hard to notice in ordinary images, because the major component of brightness changes is usually due to the spatial gradient of the image irradiance. You can try a simple experiment to verify the effect predicted by (2.13) by acquiring an image of a Lambertian surface illuminated by diffuse light (see Exercise 2.2).

Lecture #9 (Feb. 6)

Major journals for CV:

PAMI

IJCV

Major conferences:

CVPR

ICCV

ECCV

Camera Coordinates, projection (perspective projection), Image Coordinates $\longrightarrow$ 3D coordinates

$$-(x_{im} - O_x)\, s_x = \frac{f\, R_1^T (\vec{P}_w - \vec{T})}{R_3^T (\vec{P}_w - \vec{T})}$$

$$-(y_{im} - O_y)\, s_y = \frac{f\, R_2^T (\vec{P}_w - \vec{T})}{R_3^T (\vec{P}_w - \vec{T})}$$

$$M_{int} = \begin{pmatrix} -f/s_x & 0 & O_x \\ 0 & -f/s_y & O_y \\ 0 & 0 & 1 \end{pmatrix}$$

$$M_{ext} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & -R_1^T \vec{T} \\ r_{21} & r_{22} & r_{23} & -R_2^T \vec{T} \\ r_{31} & r_{32} & r_{33} & -R_3 \vec{T} \end{pmatrix}$$

Camera Calibration - recovering intrinsic and/or extrinsic parameters from test images.

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \underbrace{M_{int} M_{ext}}_{M} \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}$$

Linear Matrix Equation of Perspective Projection

Here $\frac{x_1}{x_3} = x_{im}$ ← in pixels

$\frac{x_2}{x_3} = y_{im}$ ←

If we let $M = M_{int} M_{ext}$

Let's assume that pixels are square, and that $s_x = s_y = 1$

and that $(o_x, o_y) = (0, 0)$

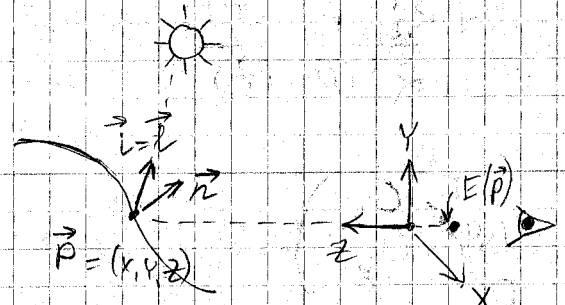
Then

$$M = \begin{pmatrix} -f r_{11} & -f r_{12} & -f r_{13} & f R_1^T \vec{T} \\ -f r_{21} & -f r_{22} & -f r_{23} & f R_2^T \vec{T} \\ r_{31} & r_{32} & r_{33} & -R_3^T \vec{T} \end{pmatrix}$$

is the projection matrix

Shape from Shading

Given a model of image formation we seek to recover the depth or shape of surfaces in the scene from an observed pattern of intensity variations



We assume that the light source is far away, and thus we have || light directions

z will refer to the depth at an image point.

$$L(\vec{p}) = \lambda \rho \vec{i}^T \vec{n}$$

- λ: incident illumination
- ρ: albedo or reflectance (applies to Lambertian surfaces), $0 \leq \rho \leq 1$

Recall, that brightness of the scene point does not depend on viewing direction (for Lambertian surface)

ρ may not be known and $\vec{i}$ may not be known!

$\lambda = 1$

Feb. 6 $R_{\rho,\vec{i}}(\vec{n}) = \rho \vec{i}^T \vec{n}$ surface reflectance map

Notes:

$\vec{i}^T$ is 1×3

$\vec{n}$ is 3×1

Recall that $E(\vec{p}) = L(\vec{P}) \frac{\pi}{4} \left(\frac{d}{f}\right)^2 \cos^4 \alpha$

We assume that the optical system is calibrated so that $E(\vec{p}) = L(\vec{P}) = R_{\rho,\vec{i}}(\vec{n})$ shape from shading eq.

Puzzle

The equation says that for a fixed ρ and $\vec{i}$ $E(\vec{p})$ relates to $\vec{n}$. How do we get z, the depth of the scene point?

Answer:

$$\vec{n} = \frac{[-p, -q, 1]}{\sqrt{p^2+q^2+1}} \quad \text{where} \quad p = \frac{\partial z}{\partial x}, \quad q = \frac{\partial z}{\partial y}$$

We have set up the coordinate axes so that

① $z = z(x,y)$ ... depth map

② The surface is far away so that

$x = f/z_0 \, X \rightarrow X = \frac{z_0}{f} x$ (weak perspective)

$y = f/z_0 \, Y \rightarrow Y = \frac{z_0}{f} y$

Now:

$$\vec{P} = (x, y, z(x,y))$$

Normal is $\perp$ to the tangent plane, and thus cross product of 2 vectors on the tangent plane gives the normal vector

Let $\vec{v}_1 = \left(1, 0, \frac{\partial z}{\partial x}\right)$ ← p

$\vec{v}_2 = \left(0, 1, \frac{\partial z}{\partial y}\right)$ ← q

Take the cross product, get that

$$\vec{n} = \frac{[-p, -q, 1]}{\sqrt{p^2+q^2+1}}$$

↓ Check slides on 'Calendar'

On the image $\cos\alpha = \vec{\ell} \cdot \vec{n}$

(not the same α as before!!!)

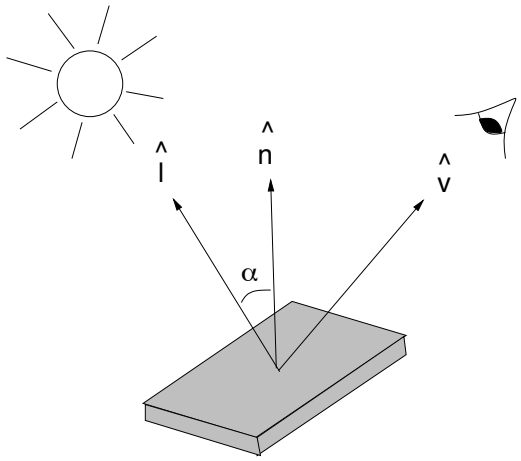
$$E(x,y) = \rho\lambda\cos(\alpha)$$

What about $\vec{\ell}$?

Pick z as $\vec{\ell}$ direction. (Exactly above or below)

$\vec{\ell} = (0,0,1)$

Goal: To recover surface depth from shading in a single image.



$\hat{l}$: light source direction

$\hat{n}$: normal to a surface patch

$\hat{v}$: viewing direction

α: angle between $\hat{n}$ and $\hat{l}$

For a lambertian surface:

$$E(x,y) = \rho\lambda\hat{l}\cdot\hat{n}, \tag{9}$$

where $E(x,y)$ is the image intensity, λ is the incident illumination, ρ is the albedo of the surface.

Let $\hat{l}$ coincide with $\hat{v}$:

$$E(x,y) = \frac{1}{\sqrt{1+z_x^2+z_y^2}}, \tag{10}$$

where $z(x,y)$ is the depth of the surface.

$z(x,y)$ must be recovered from $E(x,y)$ and initial conditions.

Proposed Solution: Evolve a curve such that it tracks the height contours of $z(x, y)$. [Kimmel *et al.*, IJCV95]

Height climbed while progressing a distance $|\Delta C|$ in the direction $\widehat{n}$ in the (x, y) plane is given by $|\Delta C| = |\Delta z| \cot(\alpha)$.

Let z denote time in the course of evolution, *i.e.*, $z = t$. Since $E = \rho\lambda \cos(\alpha)$, we have

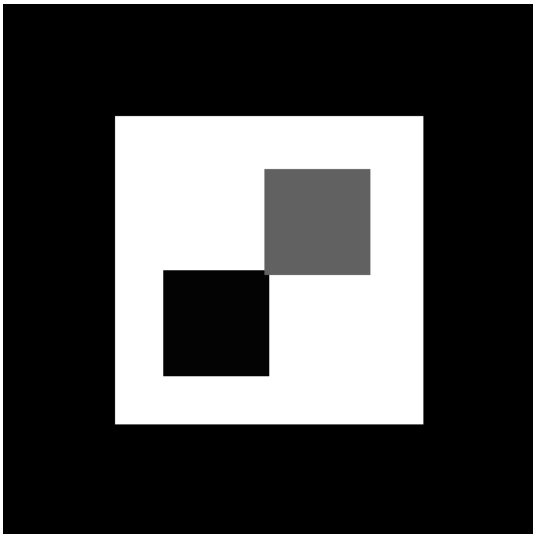
$$\left|\frac{\Delta C}{\Delta t}\right| = cot(\alpha) = \frac{E/\rho\lambda}{\sqrt{1 - (E/\rho\lambda)^2}}. \qquad (11)$$

The curve evolution equation is:

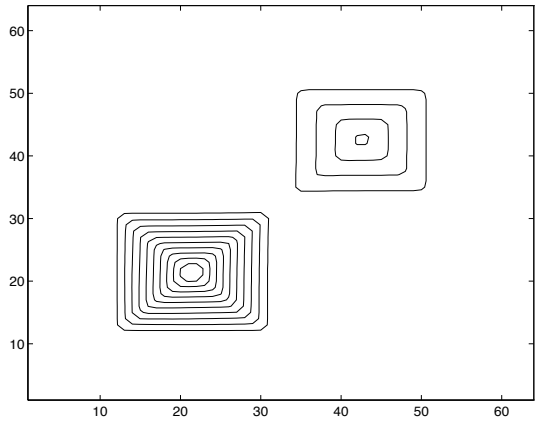
$$\begin{cases} \frac{\partial \mathcal{C}}{\partial t} & = \frac{E/\rho\lambda}{\sqrt{1-E^2/(\rho\lambda)^2}}.\widehat{n}, \\ \mathcal{C}(s,0) & = \mathcal{C}_0(s). \end{cases} \tag{12}$$

where the initial curve, $\mathcal{C}_0$, is a known height contour of z.

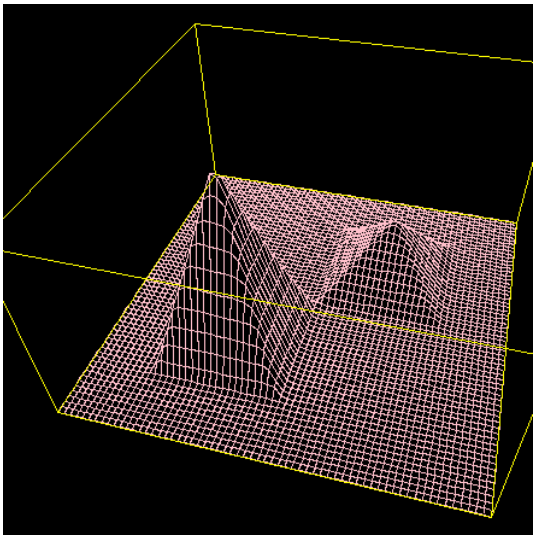
Examples - Pyramids



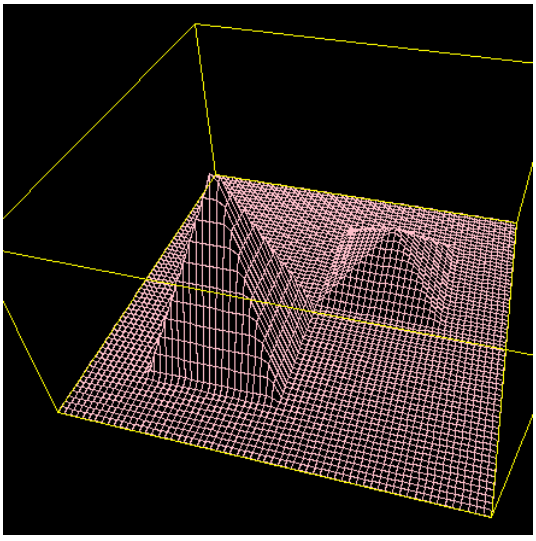
shaded image



equal height contours

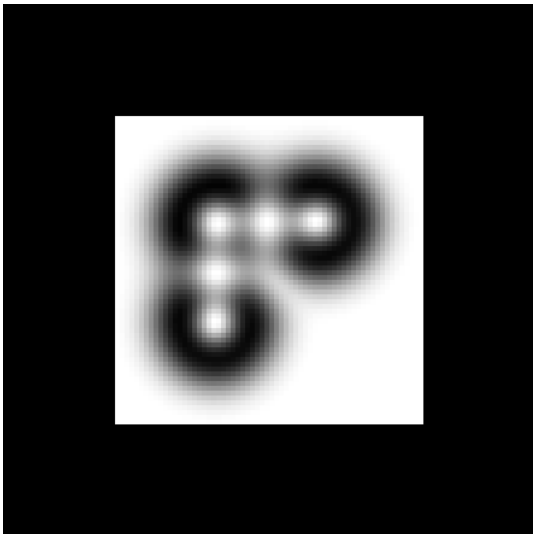


numerical solution

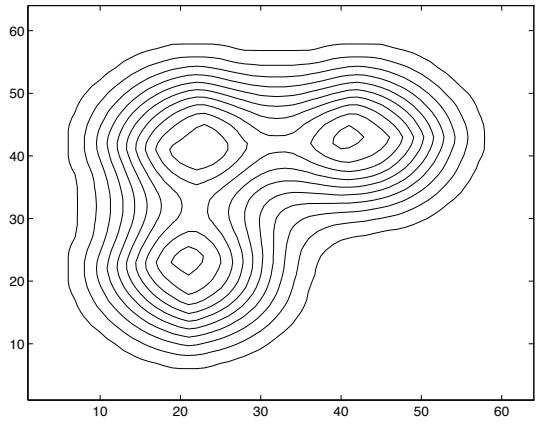


true surface

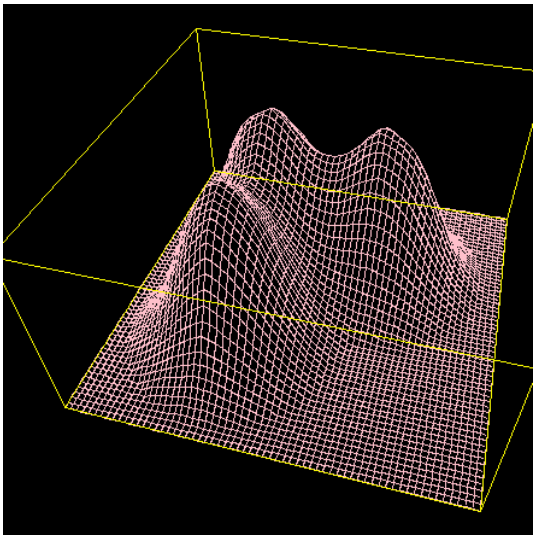
Examples - Three Mountains



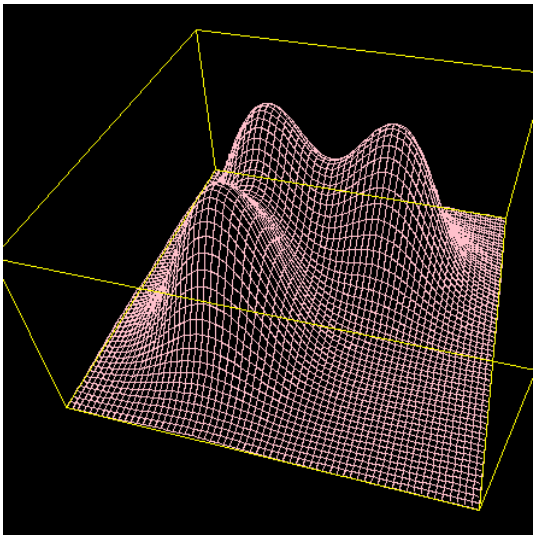
shaded image



equal height contours

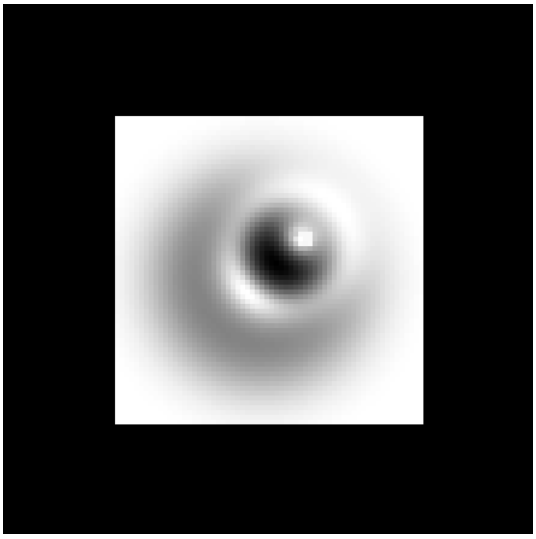


numerical solution

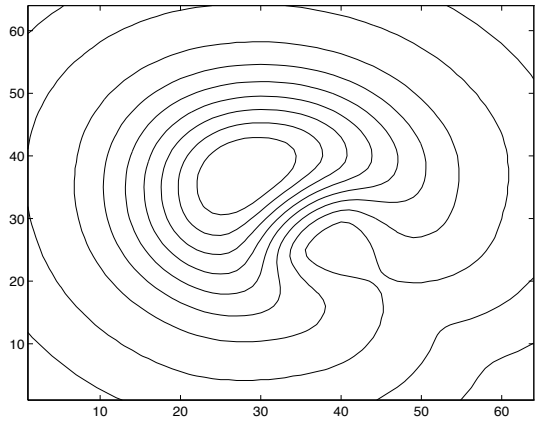


true surface

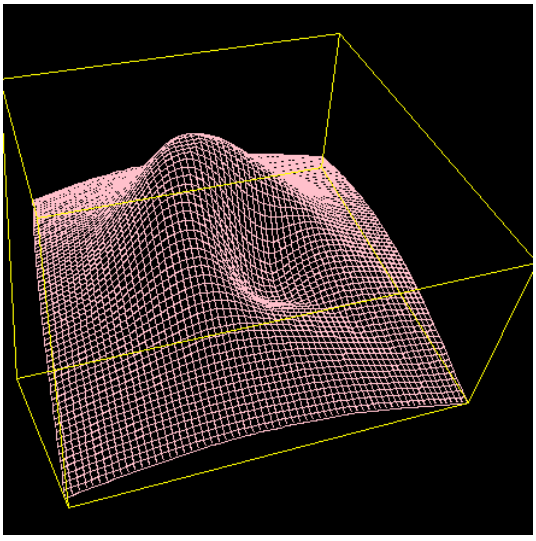
Examples - Volcano



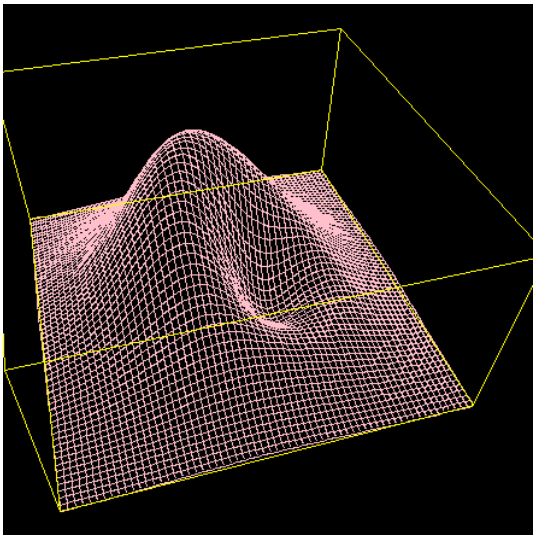
shaded image



equal height contours



numerical solution



true surface

Lecture #10 (Feb. 8)

Review:

$$E(x,y) = \rho\lambda \langle \vec{i}, \vec{n} \rangle = \rho\lambda \left\langle i, \frac{[-p,-q,1]}{\sqrt{p^2+q^2+1}} \right\rangle$$

(dot product)

- $E(x,y)$: intensity in the image
- ρ: albedo
- $\vec{i}$: incident illumination

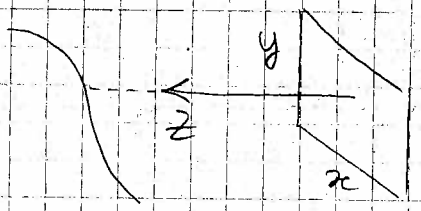
where

$$p = \frac{\partial z}{\partial x}$$

$$q = \frac{\partial z}{\partial y}$$

Assumed:
1. weak perspective
2. Lambertian surfaces

$z(x,y) \Rightarrow$ depth



$$\begin{pmatrix} v_1 = (1, 0, p) \\ v_2 = (0, 1, q) \end{pmatrix}$$

Shape From Shading problem

Issues:
- Not all points on the surface will be lit
- p and q will be treated as an independent quantities

Note that $p_y = q_x$

* p and q are not independent, and we should add constraint since model doesn't include this.

Given the reflectance map of the viewed surface

$$R_{\rho,\vec{i}} = R_{\rho,\vec{i}}(\vec{n}) = R_{\rho,\vec{i}}(p,q)$$

and knowledge of ρ, $\vec{i}$,

reconstruct the surface, for which $E(x,y) = R_{\rho,i}(p,q)$

Measuring surface Reflectance of materials is an open area of research (vision-based graphics)
- could be a project

Note, in practice we use the max of $\{E(x,y), 0\}$

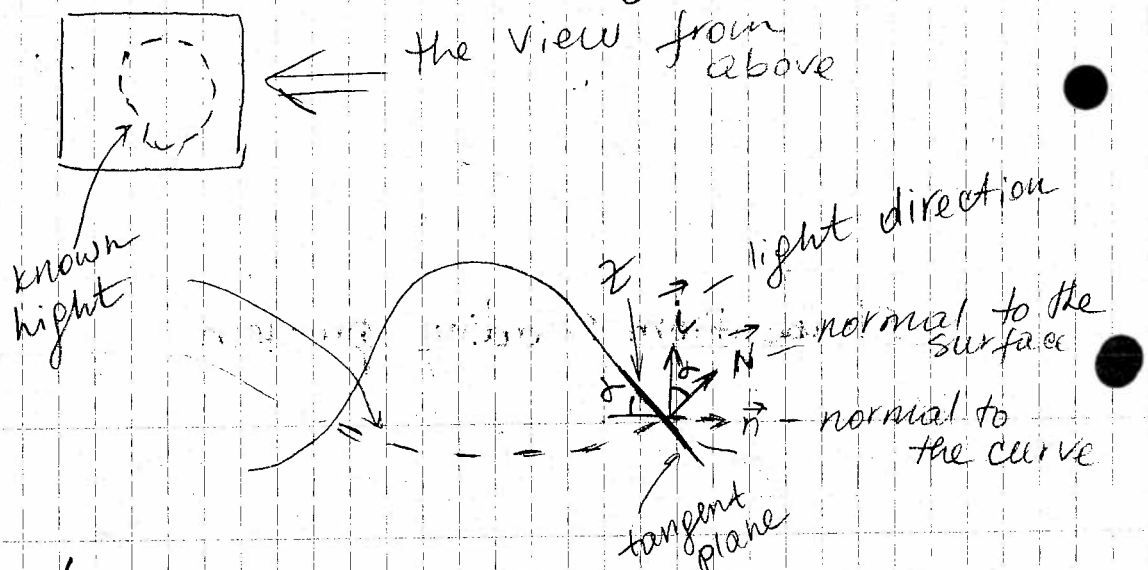
* We need initial condition to start from - z of at least one point on the surface. We assume $z=0$ for back-

In controlled situations it makes good sense to have light source $\vec{i}$ coincide with viewing direction $\vec{v}$. Then:

$$\vec{n} = \frac{[-p, -q, 1]}{\sqrt{1+p^2+q^2}}$$

$$\vec{i} = [0 \;\; 0 \;\; -1]$$

From the point of view of the image $E(x,y)$ the situation is the following



Goal:
Take the step in the direction $\vec{n}$ in the image so as to "climb" surface at the constant rate

The first algorithm for SFS

① Assumes that we always "climb" up or down
cannot convert depth!
Assumes continuous surfaces.

Background reading
① Reflectance map
② Chapter 9

Feb. 8

Horn Shape From Shading Solution

① Solve $p_y = q_x$ globally subject to certain conditions:

- smoothness condition
- integrability condition $p_y = q_x$

This is an ill posed problem:

ⓐ solution may not exist
ⓑ may not be unique
ⓒ may not vary continuously with the input data

↓ Regularization theory

trade off between smoothness of the solution and fidelity to the input data

Define an appropriate 'Energy' and minimize it

$$\mathcal{E} = \iint_{\text{image}} dx\,dy \left[\underbrace{\left(E(x,y) - R(p,q)\right)^2}_{\text{Problem fidelity}} + \underbrace{\lambda}_{\text{reg. parameter}} \underbrace{\left(p_x^2 + p_y^2 + q_x^2 + q_y^2\right)}_{\text{smoothness term}} \right]$$

Minimize $\mathcal{E}$!!

(using the Euler Lagrange equations)

$$\frac{\partial E}{\partial p} - \frac{\partial}{\partial x}\frac{\partial E}{\partial p_x} - \frac{\partial}{\partial y}\frac{\partial E}{\partial p_y} = 0$$

$$\frac{\partial E}{\partial q} - \frac{\partial}{\partial x}\frac{\partial E}{\partial q_x} - \frac{\partial}{\partial y}\frac{\partial E}{\partial q_y} = 0$$

start locally and propagate

gayane
gayane.petrosyan@mail.mcgill.ca

Lecture #11 (Feb. 13)

$$G(x,y) = e^{-\frac{1}{2}\frac{(x^2+y^2)}{\sigma^2}}$$

$$\Delta G = G_{xx} + G_{yy}$$

$$\Delta G = G_{rr} + G_r/r$$ (→ 0)

The idea of the papers only works ~~with maximum~~ under linearity assumption.

Aside

Estimates of 1st and 2nd derivative

① $f(x+h) = f(x) + h f'(x) + \frac{1}{2} h^2 f''(x) + \dots$

② $f(x-h) = f(x) - h f'(x) + \frac{1}{2} h^2 f''(x) + \dots$

① − ② $\quad \frac{f(x+h) - f(x-h)}{2h} = f'(x)$

① + ② $\quad \frac{f(x+h) + f(x+h) - 2f(x)}{h^2} = f''(x)$

Think of h as Δx or Δy

Horn Shape From Shading

$$e = \iint_{y\,x} \underbrace{[(E(x,y) - R(p,q))^2}_{\text{SFS part (fidelity)}} + \underbrace{\lambda}_{\in (0,1)} \underbrace{(p_x^2 + p_y^2 + q_x^2 + q_y^2)}_{\text{smoothness part}}] \, dx\,dy$$

where $p = \frac{\partial z}{\partial x}$, $q = \frac{\partial z}{\partial y}$ and we are seeking an assignment of p, q throughout the image

Variational Approach

Seeks to 'globally' minimize an energy by 'locally' satisfying the Euler-Lagrange equations

→ left with PDEs to solve locally
(partially differential equations)

Think of $e = e(p(x,y), q(x,y), p_x, p_y, q_x, q_y)$

E-L equations:

$$\frac{\partial E}{\partial p} - \frac{\partial}{\partial x}\frac{\partial e}{\partial p_x} - \frac{\partial}{\partial y}\frac{\partial e}{\partial p_y} = 0$$

$$\frac{\partial E}{\partial q} - \frac{\partial}{\partial x}\frac{\partial e}{\partial q_x} - \frac{\partial}{\partial y}\frac{\partial e}{\partial q_y} = 0$$

Fact: R is the only function of p and q. Neither E nor R depend on p_x, q_x, p_y, q_y.

Plug-in e, we get

$$-2(E-R)\frac{\partial R}{\partial p} - \underbrace{2\lambda p_{xx} - 2\lambda p_{yy}}_{-2\lambda\Delta p} = 0$$

Euler Lagrange Equations (E-L)

$$-2(E-R)\frac{\partial R}{\partial q} - \underbrace{2\lambda q_{xx} - 2\lambda q_{yy}}_{-2\lambda\Delta q} = 0$$

Use

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}$$

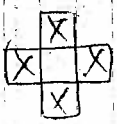
$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

Plug-in to E-L

(i,j) are pixel coordinates, $h = \Delta x = \Delta y = 1$

$$p_{i,j}^{k+1} = \overline{p_{i,j}}^{k} + \frac{1}{4\lambda}(E-R)\frac{\partial R}{\partial p}\Big|^{k}$$

$$q_{i,j}^{k+1} = \overline{q_{i,j}}^{k} + \frac{1}{4\lambda}(E-R)\frac{\partial R}{\partial q}\Big|^{k}$$



average of 4 neighbors

- at $k=0$ we need to fix p and q somewhere in the scene
- keep iterating untill your updates for p and q are small

PROBLEM!!!

The fact that p and q should be related by the same depth z is not accounted for.

Integrability constrain $p_y = q_x$

Chellape + Frankot proposed a solution

Feb. 13 Turns out that if you want to require that z gives a consistent p and q, there is implied relationship between p, q, and z in Fourier domain

Let p, q, and z be expressed as inverse Fourier transforms (discrete domain)

$$p = \sum \underbrace{c_p(\omega_x, \omega_y)}_{\text{Fourier transform of } p} e^{i(\omega_x x + \omega_y y)}$$

$$q = \sum c_q(\omega_x, \omega_y) e^{i(\omega_x x + \omega_y y)}$$

$$z = \sum c(\omega_x, \omega_y) e^{i(\omega_x x + \omega_y y)}$$

$$\frac{\partial z}{\partial x} = \sum \underbrace{i\omega_x c(\omega_x, \omega_y)}_{} e^{i(\omega_x x + \omega_y y)}$$

$$\frac{\partial z}{\partial y} = \sum \underbrace{i\omega_y c(\omega_x, \omega_y)}_{} e^{i(\omega_x x + \omega_y y)}$$

Observe that

$$c_p(\omega_x, \omega_y) = i\omega_x c(\omega_x, \omega_y)$$
$$c_q(\omega_x, \omega_y) = i\omega_y c(\omega_x, \omega_y)$$

$\times i\omega_x$:
$$i\omega_x c_p(\omega_x, \omega_y) = -\omega_x^2 c(\omega_x, \omega_y)$$

$\times i\omega_y$:
$$i\omega_y c_q(\omega_x, \omega_y) = -\omega_y^2 c(\omega_x, \omega_y)$$

$+ i\omega_x$

$$c(\omega_x, \omega_y) = \frac{i\omega_x c_p(\omega_x, \omega_y) + i\omega_y c_q(\omega_x, \omega_y)}{\omega_x^2 + \omega_y^2} \quad (4)$$

This is the constraint we were after

More complex version:

① Iterate the discrete equations

② Set z to be inverse discrete F.T. using (4) constraint

③ Reset p and q using c_p and c_q

Lecture #12 (Feb. 15)

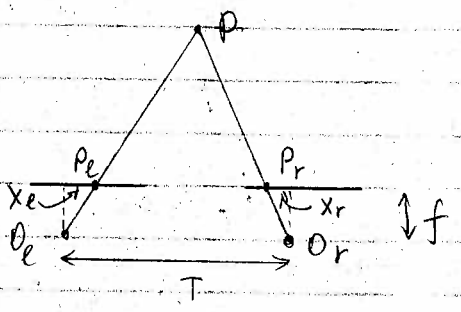
Section 7 in background reading

Dense Stereo

① Pairwise matching between the left and the right images (correspondance problem)

② Given the correspondances and knowledge of camera parameters, use triangulation to reconstruct depth. (reconstruction problem)

Matching can be easy to hard



$\Delta O_l P O_r$ and $\Delta p_l P p_r$ are similar

$$\frac{T}{z} = \frac{T + x_l - x_r}{z - f}$$

depth of the scene point $z = f\frac{T}{d}$ $\quad d = x_r - x_l$ (disparity)

Camera Calibration and Stereo Geometry
(1980's up to mid 90's
3b Computer Vision (by Faugeras)

1) Each camera has a reference frame with the origin at that camera's center O_e, O_R
2) $\vec{P}_l, \vec{P}_r$ refer to the same 3D point $\vec{P}$ in two different coordinate systems.
3) $\vec{P}_r = R(\vec{P}_l - \vec{T})$ ⊛
4) $\vec{e}_l, \vec{e}_r$ are the **epipoles** where the line connecting O_l and O_r intersects the image planes
5) $\vec{p}_l = \frac{f_l}{z_l}\vec{P}_l$, $\vec{p}_r = \frac{f_r}{z_r}\vec{P}_r$ (perspective projection)

6) O_l, P, O_r define epipolar plane $\rightarrow \Pi_p$

Corresponding points must lie on conjugated epipolar lines
(Epipolar constraint)

Goal
Define Essential Matrix and Fundamental Matrix

Note that the following vectors are co-planar
$\vec{P}_e$, $\vec{T}$, and $\vec{P}_e - \vec{T}$

$(\vec{P}_e - \vec{T})$ is perpendicular to $\vec{T} \times \vec{P}_e$

$(\vec{P}_e - \vec{T})$: in the epipolar plane (Π_p)

$\vec{T} \times \vec{P}_e$: orthogonal to Π_p

$$(\vec{P}_e - \vec{T})^T \cdot (\vec{T} \times \vec{P}_e) = 0$$ (dot product)

Use (*)
$$(\vec{P}_e - \vec{T}) = R^{-1}\vec{P}_r = R^T \vec{P}_r$$

$$(R^T \vec{P}_r)^T \cdot (\vec{T} \times \vec{P}_e) = 0 \quad (**)$$

$$\vec{T} \times \vec{P}_e = S\vec{P}_e \text{ where } S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \quad (***)$$

S has rank 2
(rank deficient), so it's not invertable

Combine (**) and (***)

$$(R^T \vec{P}_r)^T S \vec{P}_e = 0 \Rightarrow \vec{P}_r^T R S \vec{P}_e = 0$$ holds for all points

$RS = E$ Essential Matrix

Recall that $\vec{P}_r^T E \vec{P}_e = 0$ (due to perspective projection)

Feb. 15

Let M_ℓ and M_r be the matrices of the intrinsic parameters.

$$M = \begin{bmatrix} -f/s_x & 0 & o_x \\ 0 & -f/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

pixel coordinates

$\rightarrow \bar{p}_\ell = M_\ell \vec{p}_\ell$ Eq. 2

$\rightarrow \bar{p}_r = M_r \vec{p}_r$ Eq. 3

Substitute Eq. 2 and Eq. 3 into Eq. 1

$\bar{p}_r^T F \bar{p}_\ell = 0$ where

$F = (M_r^{-1})^T E M_\ell^{-1}$ Fundamental Matrix

Lecture #13 (Feb. 29)

8-point Algorithm

* Matching or Correspondence is a separate issue.

Camera Coordinates

$$\underset{1\times 3}{\vec{p}_r^{\,T}}\;\underbrace{E}_{3\times 3}\;\underbrace{\vec{p}_e}_{3\times 1} = 0$$

$$\vec{u}_r = E\vec{p}_e$$

E = essential matrix

projective line in a right image plane that goes through $\vec{p}_r$ and $\vec{e}_r$

Camera coordinates are not convinient to work with

Pixel Coordinates

$$\bar{p}_r^{\,T} F \bar{p}_e = 0$$

F = fundamental matrix

$$F = (M_r^{-1})^T E M_e^{-1}$$

where

$$M = \begin{pmatrix} -f/s_x & 0 & o_x \\ 0 & -f/s_y & o_y \\ 0 & 0 & 1 \end{pmatrix}$$

Aside:
OpenCV has a good impelementation of an algorithm to estimate the intrinsic parameters

Linear Algebra Review

① Inverse of a square nxn matrix
(not all square matrices are invertible!)

- joint the columns of nxn identity matrix to get the following

$$[A ; I]$$

- convent the first n columns of [A:I] to I via row interchange operations
- at the end we get $[I : A^{-1}]$

Example:

$$\left[\begin{array}{ccc:ccc} 2 & 3 & 1 & 1 & 0 & 0 \\ 4 & 6 & 0 & 0 & 1 & 0 \\ -1 & 1 & 4 & 0 & 0 & 1 \end{array}\right] \longrightarrow \begin{bmatrix} 12/5 & -11/10 & -3/5 \\ -8/5 & 9/10 & 2/5 \\ 1 & -1/2 & 0 \end{bmatrix}$$

② The determinant of A is computed using the method of cofactors

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \rightarrow \det A = A_{11} \det S_{11} - A_{21} \det S_{21} \ldots + (-1)^{n+1} A_{n1} \det S_{n1}$$

S_{ij} - matrix when row i and column j are illeminated

③ The null space of A is the set of vectors x such that $Ax = 0$

④ The rank of a $n \times n$ matrix is a number of linearly independent rows.

⑤ Invertible matrix is not singular (i.e. has full rank)

⑥ Singular Value Decomposition

$$A_{m \times n} = U_{m \times m} D_{m \times n} V^T_{n \times n}$$

- columns of U are mutually $\perp$ unit vectors
- — " — V — " —
- the matrix D is diagonal where $d_1 \geq d_2 \geq \ldots \geq d_n \geq 0$
  d_i are called singular values
  (If A is singular → one of singular values will be 0)

A 'match' gives us a $\bar{p}_e$ and $\bar{p}_r$ - pixel coordinates of corresponding points.

$E = RS$ has rank 2 (not invertible!)

F has rank 2 as well (singular)

Step ① Start with $n \geq 8$ matches,

Step ② and set up a homogeneous system using

$$\bar{p}_r^T F \bar{p}_e = 0 \leftarrow \text{this is repeated } n \text{ times}$$

~~Step ③~~

F has 9 entries $\begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$

Step ③
Let A be the $n \times 9$ matrix of coefficients of the system.

Step ④ Rewrite A as

$$SVD(A) = UDV^T$$

Step ⑤ The entries of F are the components of the column of V corresponding to the least singular value of A (d_1)

Step ⑥ In order to make F singular:

1) $F = UDV^T$
2) sets the smallest singular value of D to 0 to get D'
3) $F' = UD'V^T$

estimated F

* In theory F should be singular, but in practice because of not perfect matches, we get non singular F, so we need to fix it.

Lecture #13

Step 4: Classifying Local Shape. Finally, the shape classification given in Chapter 4 is achieved by defining two further quantities, the *mean curvature, H,* and the *Gaussian curvature, K*:

$$H = -\frac{k_1 + k_2}{2} \qquad K = k_1 k_2.$$

One can show that *the Gaussian curvature measures how fast the surface moves away from the tangent plane around P*, and in this sense is an extension of the 1-D curvature k. The formulae giving H and K for a range surface in r_{ij} form, $(x, y, h(x, y))$ are given in Chapter 4.

References

M.P. Do Carmo, *Differential Geometry of Curves and Surfaces*, Prentice-Hall, Englewood Cliffs (NJ) (1976).

A.6 Singular Value Decomposition

The aim of this section is to collect the basic information needed to understand the *Singular Value Decomposition* (SVD) as used throughout this book. We start giving the definition of SVD for a generic, rectangular matrix A and discussing some related concepts. We then illustrate three important applications of the SVD:

- solving systems of nonhomogeneous linear equations;
- solving rank-deficient systems of homogeneous linear equations;
- guaranteeing that the entries of a matrix estimated numerically satisfy some given constraints (e.g., orthogonality).

Definition

Singular Value Decomposition

Any $m \times n$ matrix A can be written as the product of three matrices:

$$A = UDV^\top. \tag{A.6}$$

The columns of the $m \times m$ matrix U are mutually orthogonal unit vectors, as are the columns of the $n \times n$ matrix V. The $m \times n$ matrix D is diagonal; its diagonal elements, σ_i, called *singular values*, are such that $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_n \geq 0$.

☞ While both U and V are not unique, the singular values σ_i are fully determined by A.

Some important properties now follow.

Properties of the SVD

Property 1. The singular values give you valuable information on the singularity of a square matrix, *A square matrix, A, is nonsingular if and only if all its singular values are different from zero.* Most importantly, the σ_i also tell you how close A is to be singular: the ratio

$$C = \frac{\sigma_1}{\sigma_n},$$

called *condition number*, measures the *degree of singularity of* A. When $1/C$ is comparable with the arithmetic precision of your machine, the matrix A is *ill-conditioned* and, for all practical purposes, can be considered singular.

Property 2. *If A is a rectangular matrix, the number of nonzero* σ_i *equals the rank of A.* Thus, given a fixed tolerance, ϵ (typically of the order of 10^{-6}), the number of singular values greater than ϵ equals the *effective rank* of A.

Property 3. *If A is a square, nonsingular matrix, its inverse can be written as*

$$A^{-1} = VD^{-1}U^{\top}.$$

Be A singular or not, the *pseudoinverse* of A, A^{+}, can be written as

$$A^{+} = VD_0^{-1}U^{\top},$$

with D_0^{-1} equal to D^{-1} for all nonzero singular values and zero otherwise. If A is nonsingular, then $D_0^{-1} = D^{-1}$ and $A^{+} = A^{-1}$.

Property 4. The columns of U corresponding to the nonzero singular values span the range of A, the columns of V corresponding to the zero singular value the null space of A.

Property 5. The squares of the nonzero singular values are the nonzero eigenvalues of both the $n \times n$ matrix $A^{\top}A$ and $m \times m$ matrix $AA^{\top}$. The columns of U are eigenvectors of $AA^{\top}$, the columns of V eigenvectors of $A^{\top}A$. Morevoer, $A\mathbf{u}_k = \sigma_k\mathbf{v}_k$ and $A^{\top}\mathbf{v}_k = \sigma_k\mathbf{u}_k$, where $\mathbf{u}_k$ and $\mathbf{v}_k$ are the columns of U and V corresponding to σ_k.

Property 6. One possible distance measure between matrices can use the *Frobenius norm*. The Frobenius norm of a matrix A is simply the sum of the squares of the entries a_{ij} of A, or

$$\|A\|_F = \sum_{i,j} a_{ij}^2. \tag{A.7}$$

By plugging (A.6) in (A.7), it follows that

$$\|A\|_F = \sum_i \sigma_i^2.$$

We are now ready to summarize the applications of the SVD used throughout this book.

Least Squares

Assume you have to solve a system of m linear equations,

$$A\mathbf{x} = \mathbf{b},$$

for the unknown n-dimensional vector $\mathbf{x}$. The $m \times n$ matrix A contains the coefficients of the equations, the m-dimensional vector $\mathbf{b}$ the data. If not all the components of $\mathbf{b}$ are null, the solution can be found by multiplying both sides of the above equation for $A^\top$ to obtain

$$A^\top A\mathbf{x} = A^\top \mathbf{b}.$$

It follows that the solution is given by

$$\mathbf{x} = (A^\top A)^+ A^\top \mathbf{b}.$$

This solution is known to be optimal in the least square sense.

It is usually a good idea to compute the pseudoinverse of $A^\top A$ through SVD. In the case of more equations than unknowns the pseudoinverse is more likely to coincide with the *inverse* of $A^\top A$, but keeping an eye on the condition number of $A^\top A$ (**Property 1**) won't hurt.

☞ Notice that linear fitting amounts to solve exactly the same equation. Consequently, you can use the same strategy!

Homogeneous Systems

Assume you are given the problem of solving a homogeneous system of m linear equations in n unknowns

$$A\mathbf{x} = 0,$$

with $m \geq n - 1$ and $rank(A) = n - 1$. Disregarding the trivial solution $\mathbf{x} = 0$, a solution unique up to a scale factor can easily be found through SVD. *This solution is simply proportional to the eigenvector corresponding to the only zero eigenvalue of* $A^\top A$ (all other eigenvalues being strictly positive because $rank(A) = n - 1$). This can be proven as follows.

Since the norm of the solution of a homogeneous system of equations is arbitrary, we look for a solution of unit norm in the least square sense. Therefore we want to minimize

$$\|A\mathbf{x}\|^2 = (A\mathbf{x})^\top A\mathbf{x} = \mathbf{x}^\top A^\top A\mathbf{x},$$

subject to the constraint

$$\mathbf{x}^\top \mathbf{x} = 1.$$

Introducing the Lagrange multiplier λ this is equivalent to minimize the *Lagrangian*

$$\mathcal{L}(\mathbf{x}) = \mathbf{x}^\top A^\top A\mathbf{x} - \lambda(\mathbf{x}^\top \mathbf{x} - 1).$$

Equating to zero the derivative of the Lagrangian with respect to $\mathbf{x}$ gives

$$A^\top A\mathbf{x} - \lambda\mathbf{x} = 0.$$

This equation tells you that λ is an eigenvalue of $A^\top A$, and the solution, $\mathbf{x} = \mathbf{e}_\lambda$, the corresponding eigenvector. Replacing $\mathbf{x}$ with $\mathbf{e}_\lambda$, and $A^\top A\mathbf{e}_\lambda$ with $\lambda\mathbf{e}_\lambda$ in the Lagrangian yields

$$\mathcal{L}(\mathbf{e}_\lambda) = \lambda.$$

Therefore, the minimum is reached at $\lambda = 0$, the least eigenvalue of $A^\top A$. But from **Properties 4** and **5**, it follows that this solution could have been equivalently established as *the column of V corresponding to the only null singular value of A* (the kernel of A). This is the reason why, throughout this book, we have not distinguished between these two *seemingly* different solutions of the same problem.

Enforcing Constraints

One often generates numerical estimates of a matrix, A, whose entries are not all independent, but satisfy some algebraic constraints. This is the case, for example, of orthogonal matrices, or the fundamental matrix we met in Chapter 7. What is bound to happen is that the errors introduced by noise and numerical computations alter the estimated matrix, call it $\hat{A}$, so that its entries no longer satisfy the given constraints. This may cause serious problems if subsequent algorithms assume that $\hat{A}$ satisfies *exactly* the constraints.

Once again, SVD comes to the rescue, and allows us to *find the closest matrix to $\hat{A}$, in the sense of the Frobenius norm (**Property 6**), which satisfies the constraints exactly*. This is achieved by computing the SVD of the estimated matrix, $\hat{A} = UDV^\top$, and estimating A as $UD'V^\top$, with D' obtained by changing the singular values of D to those expected when the constraints are satisfied exactly.[4] Then, the entries of $A = UD'V^\top$ satisfy the desired constraints by construction.

References

G. Strang, *Linear Algebra and its Applications*, Harcourt Brace Jovanovich, Orlando (FL) (1988).

[4] If $\hat{A}$ is a good numerical estimate, its singular values should not be too far from the expected ones.

Lecture #14 (March 5)

mash
wireframe } check in Matlab

- 8 point Algorithm is sensitive to noisy estimates
- Matrix A has entries of varying magnitude, need to normalize! to get good results.

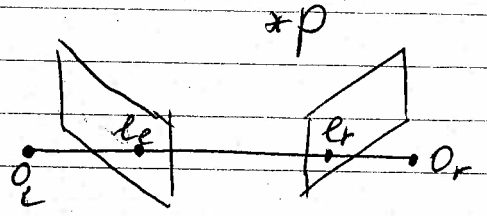
$$\underbrace{\begin{bmatrix} & \\ & \end{bmatrix}}_{\text{matrix } A \; n \times 9} \begin{bmatrix} F_{11} \\ F_{12} \\ \vdots \\ F_{33} \end{bmatrix}_{9\times 1} = \begin{bmatrix} 0 \end{bmatrix}$$

① ICCV - In defense of the 8 pt. Algorithm } useful papers
② ~~PAMI 97 19(6)~~ (by Hartley)

① Essential Matrix

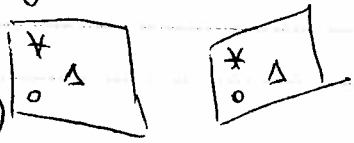
$$F = (M_r^{-1})^T E M_e^{-1} \Rightarrow E = M_r^T F M_e$$ ⇝ Open CV to get M_r, M_e

② Estimate the epipoles once you have F, using an other SVD on F (see text)

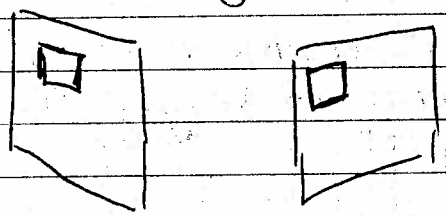


Epipolar lines are not horizontal (in general)
Rectified stereo pairs ↓ (rectification)
Epipolar lines are now horizontal

* 3D Reconstruction Problem
ray-trace back in 3D once you have the camera geometry
* Correspondence problem
SIFT features
features correspondence (sparse)



- Sparse correspondence
- dense correspondence
  (translating window that looks at neighborhoods)

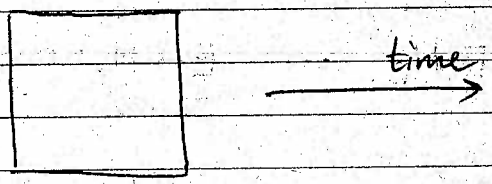


2 approaches:

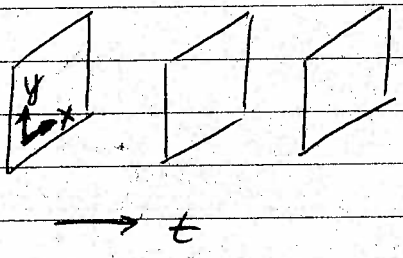
① Maximizing Cross correlation $\int u \cdot v \, dx$ (integral of a product)
② Minimizing sum of squarred distances $(u-v)^2$

Motion

Analyzing dynamic scenes



Single frame



① Gibson ~~"Reception of the visual world"~~
   Visual Flow
   Optical Flow

② Human Vision
   (Ecological Approaches)

Lecture #15 (March 7)

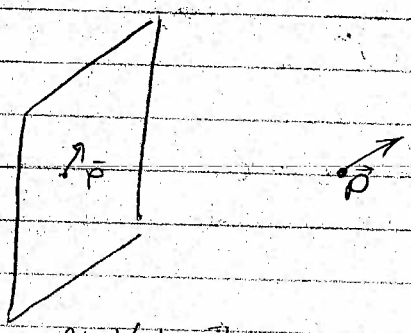
Structure from Motion

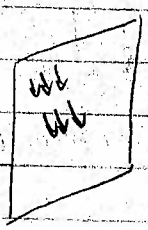
① Gibson → An ecological approach..
→ The perception of the Visual World

② Johanssen
Moving Light Displays

* BML walker demo

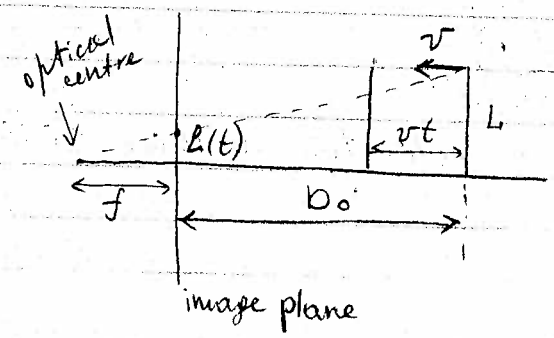
⊛ Anstis
Ramachandran → Visual Motion





Optical flow → correspondance
(object tracking)

Time to contact estimate



- Object is moving with velocity v
- after time t it has moved by $v.t$
- time taken to hit the camera is

$$\tau = \frac{D}{v}, D = D_0 + f$$

- Apparent size of the object at time t, $l(t)$

$$\frac{f}{l(t)} = \frac{D_0 - vt + f}{L} = \frac{D - vt}{L}$$

$$l(t) = \frac{f \cdot L}{D - vt} \qquad (1)$$

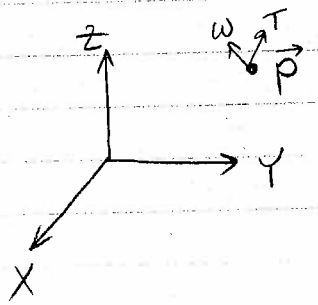
$$l'(t) = f \frac{(D-vt)\frac{df}{dt}^{0} - L\frac{d}{dt}(D-vt)}{(D-vt)^2} =$$

$$= f \frac{-L}{(D-vt)^2} \frac{d}{dt}\underbrace{(D-vt)}_{0} =$$

$$l'(t) = f \frac{Lv}{(D-vt)^2} \qquad (2)$$

$$\frac{(1)}{(2)} = \frac{D-vt}{v} = \tau - t$$

3D to 2D (motion field)



$\vec{P} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$ is a 3D point

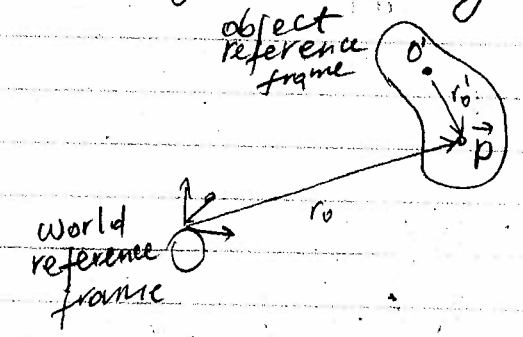
Let's assume that we work with camera's reference frame

(*) $\vec{p} = f\frac{\vec{P}}{Z} = \begin{bmatrix} x \\ y \\ f \end{bmatrix}$ homogeneous coordinates

$$\vec{v} = -\underbrace{\vec{T}}_{\text{translational}} - \underbrace{\vec{\omega} \times \vec{P}}_{\text{rotational}}$$

March 7

Aside – Angular Velocity in more detail



P is a 3D point described either by r_o' or r_o, depending on the reference frame

If we have a pure rotation of the rigid object, then the instantaneous velocity of $\vec{p}$ in the world reference frame is $\vec{v} = \vec{v}_{o'} + \omega \times r_o'$

Let $O \to O'$ then $\vec{v}_{o'} = 0$ and $r_{o'} = r_o = \vec{p}$

$\Rightarrow \vec{v} = \omega \times \vec{p}$

$$\vec{v} = -T - \vec{\omega} \times \vec{p}$$

$$\begin{bmatrix} a_x & a_y & a_z \\ \omega_x & \omega_y & \omega_z \\ X & Y & Z \end{bmatrix}$$

$$\begin{aligned} v_x &= -T_x - \omega_y Z + \omega_z Y \\ v_y &= -T_y - \omega_z X + \omega_x Z \\ v_z &= -T_z - \omega_x Y + \omega_y X \end{aligned} \quad (3)$$

$\vec{v}$ is the relative motion between $\vec{p}$ and the camera

How do we use (+)?

Take derivatives with respect to t to get velocities!

(3')

$$\frac{d\vec{p}}{dt} = \vec{v} = f\frac{(Z\vec{V} - \vec{p}v_z)}{Z^2} \quad (4)$$

because $\frac{d\vec{p}}{dt} = \vec{V}$, $\frac{d\vec{p}}{dt} = \vec{v}$

$$\frac{dz}{dt} = V_z$$

Plug (3) into (4)

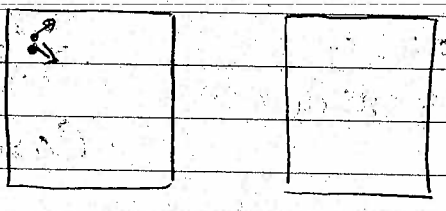
$$v_x = \frac{T_z x - T_x f}{z} - \omega_y f + \omega_z y + \omega_x \frac{xy}{f} - \omega_y \frac{x^2}{f}$$

$$v_y = \frac{T_z y - T_y f}{z} + \omega_x f -$$

Lecture #16 (March 12)

① Midterm - April 2nd - including Recognition, not including Geometric Flows
② Projects Presentations 11th and 16th
- 12 min max
③ Final report due date April 23rd
- 12 pages, single cd, single spaced
- additional stuff on Web page

Today:
① Motion field equations (again)
② Radial flows
③ Optical flow estimation] - least square approach (region based)
(object tracking)] - a variational (local)



$$v_x = \underbrace{\frac{T_z x - T_x f}{Z}}_{v_x^T} \underbrace{- w_y f + w_z y + w_x \frac{xy}{f} - w_y \frac{x^2}{f}}_{v_x^w}$$

$$v_y = \frac{T_z y - T_y f}{Z} + w_x f - w_z x - w_y \frac{xy}{f} + w_x \frac{y^2}{f}$$

(w_x, w_y, w_z) - angular velocity
(T_x, T_y, T_z) - translational

If you have no rotation, what does the motion field look like?

$$v_x = \frac{T_z x - T_x f}{Z}, \quad v_y = \frac{T_z y - T_y f}{Z}$$

① The magnitude of $\vec{v}$ is inversely proportional to z
$|\vec{v}| \propto \frac{1}{z}$
(speed of points decreases with increasing z)

② Velocity vectors are parallel?
True if $T_z = 0$

③ If $T_z \neq 0$?
Radial flow!

$\vec{\omega} = (0,0,0)$, $T_z \neq 0$
Let $\bar{p}_0 = [x_0, y_0]^T = \begin{bmatrix} f\,T_x/T_z \\ f\,T_y/T_z \end{bmatrix}$
Then,

$$\frac{v_x}{T_z} = \frac{x - x_0}{z}, \quad \frac{v_y}{T_z} = \frac{y - y_0}{z}$$ radial flow

④ If $T_z < 0$

focus of expansion

(Case ①)

⑤ If $T_z > 0$

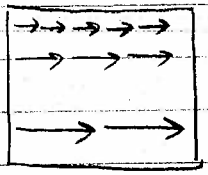
focus of contraction

(Case ②)

Case ③ $T_z = 0$

$$v_x = -f\frac{T_x}{z}, \quad v_y = -f\frac{T_y}{z}$$

a parallel motion field where the length of the vectors are inversely proportional to depth



Motion Parallax
Has to do with the instantaneous coincidence of two projected points. (2 distinctive points in 3D, but having the same projection on the image plane)

March 12

2 points: $\vec{P} = [X\ Y\ Z]^T$ and $\vec{P}_1 = [X_1\ Y_1\ Z_1]^T$

The motion field equations:

$$\left.\begin{aligned} v_x &= v_x^T + v_x^\omega \\ v_y &= v_y^T + v_y^\omega \end{aligned}\right\} \text{ for } \vec{P}$$

$$\left.\begin{aligned} v_{1x} &= v_{1x}^T + v_{1x}^\omega \\ v_{1y} &= v_{1y}^T + v_{1y}^\omega \end{aligned}\right\} \text{ for } \vec{P}_1$$

Take the difference between these 2:

$$\Delta v_x = (T_z x - T_x f)\left(\frac{1}{Z} - \frac{1}{Z_1}\right)$$

$$\Delta v_y = (T_z y - T_y f)\left(\frac{1}{Z} - \frac{1}{Z_1}\right)$$

Note: A rigid body has the property that $v_x^\omega = v_{1x}^\omega$, $v_y^\omega = v_{1y}^\omega$

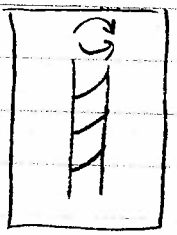
$$\frac{\Delta v_y}{\Delta v_x} = \frac{T_z y - T_y f}{T_z x - T_x f} = \frac{\left(y - \frac{T_y f}{T_z}\right)}{\left(x - \frac{T_x f}{T_z}\right)} = \frac{y - y_0}{x - x_0}$$

Summary:

For all rotational motions, the vector $[\Delta v_x, \Delta v_y]$ points in to the direction of $\vec{p}_0$, the instantaneous epipole

Optical Flow

Estimation of the 2D flow field from a sequence of frames.



The actual movement is left/right but the perception is up/down

"aperture problem"
(Barber Pole Illusion)

If intensities of the edge is constant, there is ambiguity in solving correspondance

Aperture problem:
① ② and ③ all have the same 'normal' (⊥ to the edge)

Optical Flow can only recover the normal component of the velocity field $\vec{v}$
(tangential component is ambiguous)

Image Brightness Constancy Equation:

$$E(x(t), y(t), t) = \text{constant}$$

$$\frac{dE}{dt} = 0 \Rightarrow \frac{\partial E}{\partial x}\frac{dx}{dt} + \frac{\partial E}{\partial y}\frac{dy}{dt} + \frac{\partial E}{\partial t} = 0$$

$$\nabla E = \left(\frac{\partial E}{\partial x}, \frac{\partial E}{\partial y}\right) \rightsquigarrow \text{spacial gradient}$$

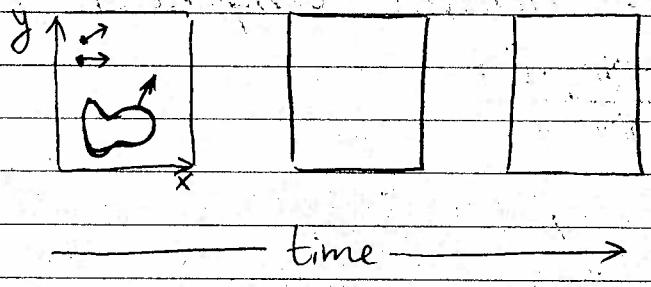
$$\vec{v} = \left(\frac{dx}{dt}, \frac{dy}{dt}\right)$$

$$(\nabla E)^T \vec{v} + \frac{\partial E}{\partial t} = 0$$

Lecture #17 (March 14)

Optical Flow (under the constant brightness assumption)

$E(x(t), y(t), t) = \text{Constant}$.



$$\frac{dE}{dt} = 0 = \frac{\partial E}{\partial x}\frac{dx}{dt} + \frac{\partial E}{\partial y}\frac{dy}{dt} + \frac{\partial E}{\partial t} = \langle \nabla E, \vec{v} \rangle + \frac{\partial E}{\partial t}$$

spatial gradient: $\nabla E = \left(\frac{\partial E}{\partial x}, \frac{\partial E}{\partial y}\right)$

this is what we want to estimate: $\vec{v} = \left(\frac{dx}{dt}, \frac{dy}{dt}\right)$

Goal of optical flow is to recover $\vec{v}$!

We cannot actually recover the true $\vec{v}$ due to the aperture problem.

At a location (x,y) ∇E points in the direction $\perp$ to the isophote (level curve of constant intensity at (x,y))

$$|\vec{v}_n| = \frac{\langle \nabla E, \vec{v} \rangle}{\|\nabla E\|} = -\frac{\frac{\partial E}{\partial t}}{\|\nabla E\|}$$

* The direction of $|v_n|$ is ∇E

optical flow can give this to us

Quick assessment of how faithful the constant brightness assumption is.

① Fair for translational motions
② Bad if we have rotations, too

Error

① Forward model of image formation
$E = \rho \lambda \langle \vec{I}, \vec{n} \rangle$
Let the object undergo rotation and calculate the normal component $\vec{v}_n$

② $|\vec{v}_n| = \frac{\langle \nabla E, \vec{v} \rangle}{\|\nabla E\|}$

$$\text{Error} \propto \frac{1}{\|\nabla E\|} \frac{\vec{I}^T \cdot (\vec{\omega} \times \vec{n})}{\|\nabla E\|}$$

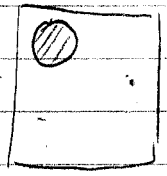
(high spatial gradient, or edge-like regions give more faithful estimates of $|v_n|$)

First idea

Find $\vec{v}$ such that $\frac{\langle \nabla E, \vec{v} \rangle}{\|\nabla E\|} + \frac{\frac{\partial E}{\partial t}}{\|\nabla E\|} = 0$

Local solution → motion estimates might be very different in nearby location!!!

Let's define the neighborhood size and estimate a single velocity $\vec{v}$ for all locations within this neighborhood (patch-based approach)
→ For robustness, use circular neighborhood



For all patches in a frame
For each pixel (i,j) in the patch
Compute $\nabla E(i,j)$
Compute $E_t(i,j)$ difference in time
Find a single v that minimizes the following

$$\sum_{(i,j) \in \text{patch}} \frac{\langle \nabla E, v \rangle + \frac{\partial E}{\partial t}}{\|\nabla E\|}$$

March 14

Turns out that the best $\vec{v}$ in a least squares sence can be directly computed!

$\longrightarrow \vec{v} =$ (multiplication of matrices) Eq. 8.25.

$$\vec{v} = (A^T A)^{-1} A^T \vec{b}$$ (linear least squares approach)

where

$$A = \begin{bmatrix} \nabla E(p_1) \\ \vdots \\ \nabla E(p_N) \end{bmatrix}, \quad b = [E_t(p_1) \dots E_t(p_N)]^T$$

vector of $\frac{\partial E}{\partial t}$

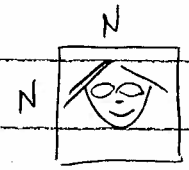
the vector of spatial gradients

Modification: Weighted least squares solution (more weight to points near the center)

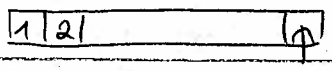
Lecture #18 (March 19)

* Background reading "Recognition By Components"
* Check slides for "Appearance-based recognition" on course website (calendar)

Faces example



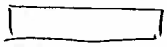
$N = 256$



256^2 entries

$n = N \times N$

↓ PCA



~20 components

Image of a particular person can be represented as a linear combination of PCs.

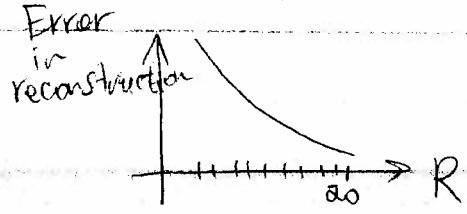
→ Same PCs for all - but different weights for different individuals.

Once we find weights for each image in db, we can throw away the images, and store only the weights (and PCs)

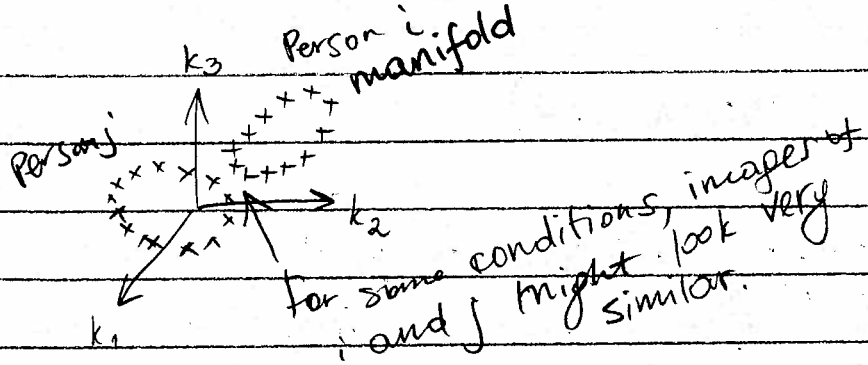
We can acquire images by varying lighting conditions (L) and head pose (R).

⊕ Images should have the faces at the same scale!

Algorithm cannot distinguish between foreground/background pixels.



For about 10-20 people, with 5-10 images of each.



Covariance matrix $C = [x x^T]$

⊛ $C_{N^2 \times N^2}$ – huge dimension

Naive eigen value decomposition will fail !!!

⊛ If new object appears we need to recompute covariance matrix (no grace method to add eigenvectors!)

⊛ Doing PCA on FFT solves the problem of different viewpoints (rotations)

⊛ In a projected space, the distance metric is not Euclidian (though it can be used by a naive algorithms)

March 26

Optical Flow
(Horn + Schunck)

The equation: $E(x(t), y(t), t)$

$$E_x \frac{dx}{dt} + E_y \frac{dy}{dt} + E_t = \frac{dE}{dt}$$

This allaos distinct image points to move independently (local solution, no smoothness constraints)

We are looking for:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \vec{v} = \begin{pmatrix} dx/dt \\ dy/dt \end{pmatrix}$$

Solution:

$$(E_x u + E_y v + E_t)^2 = \varepsilon_b^2$$

$$\underbrace{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2}_{|\nabla u|^2} + \underbrace{\left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2}_{|\nabla v|^2} = \varepsilon_c^2$$

$$\varepsilon^2 = \iint (\alpha^2 \varepsilon_c^2 + \varepsilon_b^2)\, dx\, dy$$

↓ Euler-Lagrange equations

$$E_x^2 u + E_x E_y v = \alpha^2 \nabla^2 u - E_x E_t$$

$$E_x E_y u + E_y^2 v = \alpha^2 \nabla^2 v - E_y E_t$$

Assume some initial values for u and v and then iterate locally (like shape from shading)

The algorithm makes sence when the movement is smooth (moving surface) and doesn't make sence for discontinuos movements