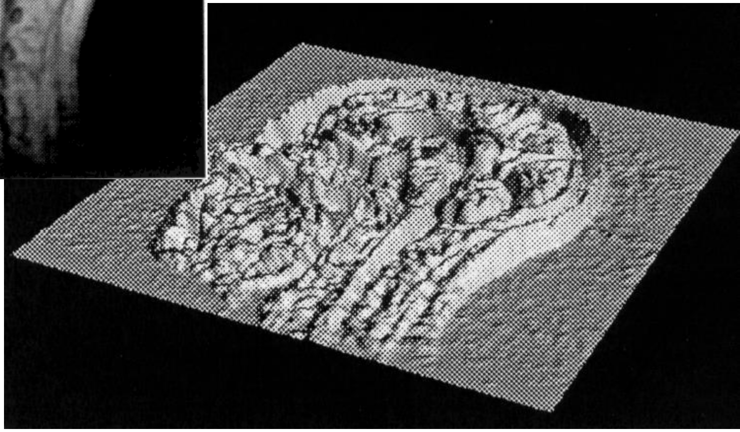
An image as a function on space

- 2D: I(x,y), typically on rectangular domain
- 3D: I(x,y,z), typically on rectilinear domain
- Both written $I(\underline{x})$
- Discrete images
  - 2D: *pixels* at sampled values of x,y
  - 3D: *voxels* at sampled values of x,y,z
  - Sampling is typically regular in each dimension
    - "Isotropic sampling", where the sampling distance is the same in each dimension, is most common
    - Non-isotropic sampling is possible
  - Both (2D or 3D) written $I(\underline{x}_i)$ or $\underline{I}(\underline{x}_i)$
    - $\underline{I}(\underline{x}_i)$ when there is more than 1 intensity value at each pixel or voxel



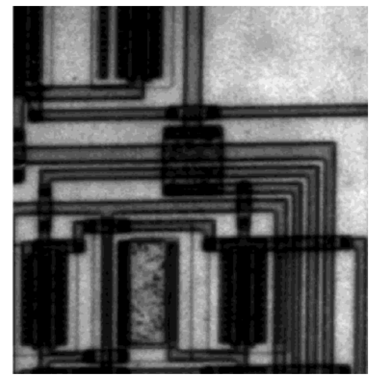
Images & Objects

Input \ Output	Image	Objects
Image	**Image Processing**	Image Analysis
Objects	Computer Graphics	

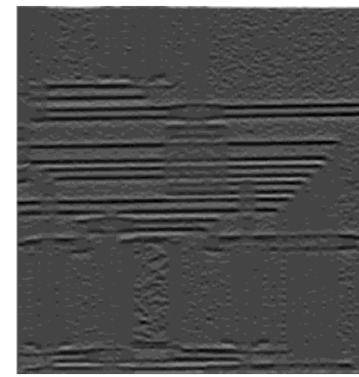
Objectives of Image Processing we will be concerned with

- Restoration
  (of blurring, noise, distortion)
- Preparation for image analysis
  - E.g., edginess, barness
- Re-formation
  - Changes in sampling
  - Normalization of translation, orientation, scaling
  - Visualizations, e.g., 3D rendering
- 3D to 2D
  - Projections, as with x-rays
  - Slices

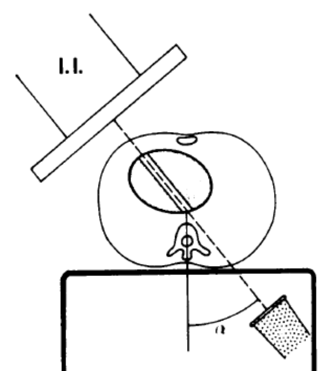


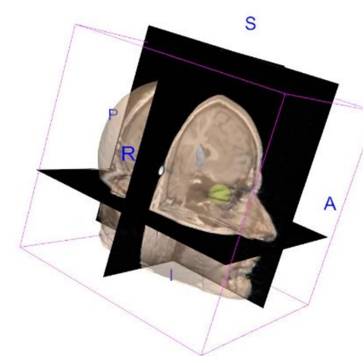


Original



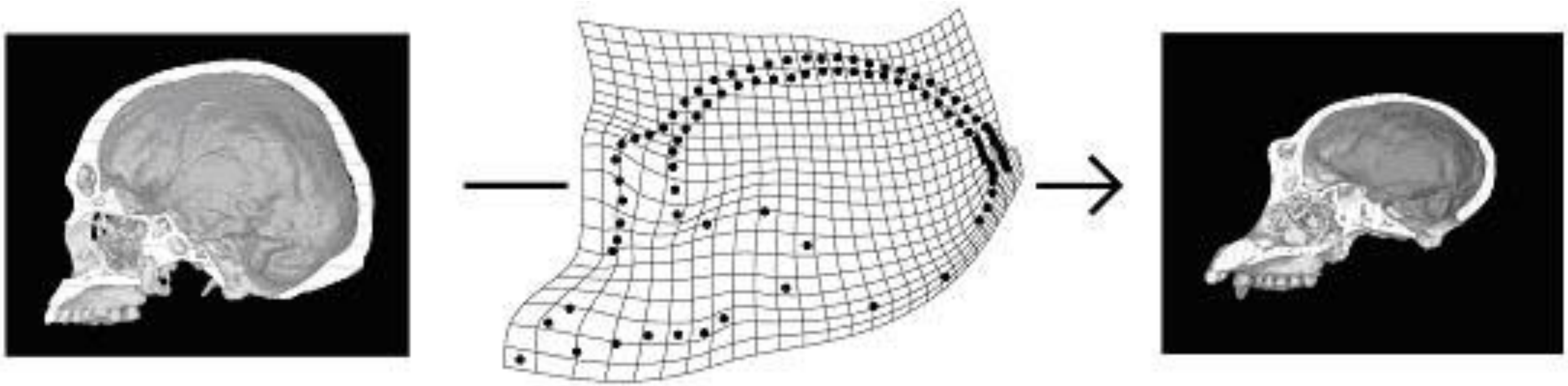
Horizontal Edginess





Other Objectives on Images we will be concerned with

- Storage and Transmission
  - Thus compression
- Registration of images



- Display
  - 2D images: resampling, geometric transformations
  - Creation of 3D images: geometric transformations

Prerequisites

- Ability to program
  - Best if you know or learn matlab
  - If not, email whether you plan to use C++ or Python
- Calculus of many variables
  - Partial derivatives, directional derivatives, gradient
- Matrix multiplication, matrix times vectors, vector dot products, matrix determinants
  - You need not have a full course in linear algebra, **though having one will ease your way**
  - We will make use of eigenvector and eigenvalue decomposition, but I will teach that
  - We will interpret these matrix and vector operations geometrically, but I will teach that

The Math Needed to Understand Image Processing

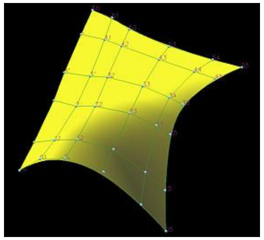
- Representation of images as Taylor series
  - Thus computation of image spatial derivatives
- Invariant operators: to shift, rotation, scale
- Shift-invariant, linear operators: convolution
  - Point and line spread functions and other convolution kernels
  - Representation of images via orthogonal basis functions, esp. sinusoids (Fourier basis functions)
  - Understanding convolution and derivatives via Fourier basis functions
- Imaging properties: resolution, noise, distortion
- Representation of images as pixels or voxels: understanding sampling effects

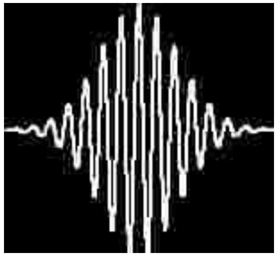
The Math Needed to Understand Image Processing cont.

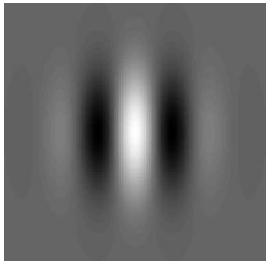
- Spatial scale
  - Two aspects
    - Levels of detail
    - Gaussian apertures and spatial scale
  - Intensity noise vs. scale
- Multiscale representations
  - Splines
  -
  - Gaussian derivatives
  - Gabor wavelets
  - Orthogonal wavelets

Little noise

noisy

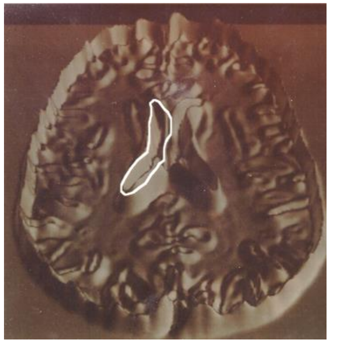




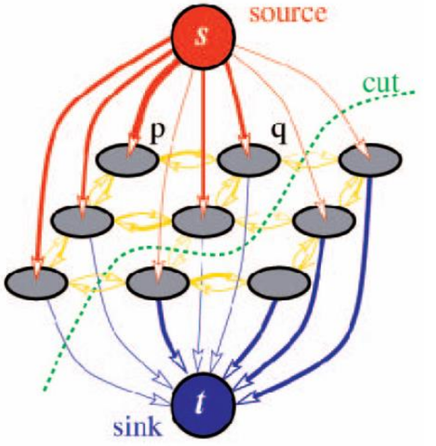


The Math Needed to Understand Image Processing$^{cont.}$

- Measures of edge and bar strength via derivatives
- Ridges in images,
  towards finding edges and bars

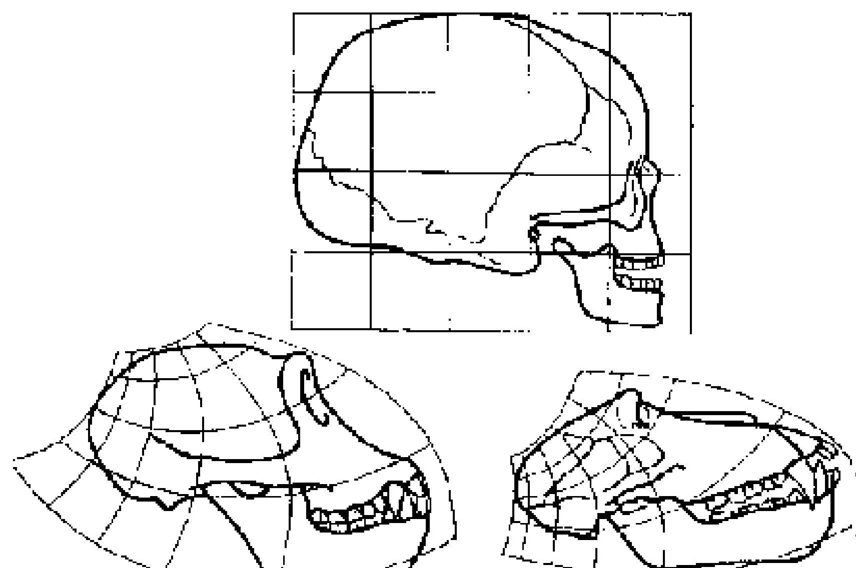


- Interpolation of discrete images
  - Via convolution; via orthogonal basis functions
  - Via splines
  - Via least-squares approximations
- Discrete images as algebraic graphs,
  with objects as graph cuts



The Math Needed to Understand Image Processing concl.

- Affine geometric transformations, in 2D and **3D**
  - Rigid transformations
    - Rotation: via matrices, **via quaternions**
  - Similarity transformations
  - Perspective transformations, homogeneous coords.
- Geometric distortion and deformation



The Math Needed to Understand Image Processing concl.

- Optimization methods
  - Gradient based; Lagrange multipliers
- [Information theory: entropy and mutual information; Levenberg-Marquardt optimization]

The Course Processes

- These slides are the text; they will appear on Sakai well before the class in which they are used
  - You probably will want to take notes on them
  - They will be updated when fixes are needed
  - Some assigned reading in Wikipedia and reserve book
    - Esp. consider purchasing *Image Processing and Analysis* by Stan Birchfield, Cengage Learning, 2016
- Since there is only this abbreviated text material, lecture attendance is critical
  - If you must miss a lecture, immediately viewing recorded lecture, getting class notes from another student, reading the ppt slides covered, studying them, followed by meeting with me is important

The Course Processes (2)

- For getting questions asked,
  - Use class: ask for clarifications, make corrections
  - Use my office hours: Tuesdays: 3:30-4:30, Fridays: 3:30-4:30
    - Including this week, remotely
    - Normally in my office: 222 SN
  - Course zoom site is https://unc.zoom.us/j/964026521
  - A Piazza site will be set up for you answer each other's questions
    - OK for homework interpretation but not for homework solutions
    - Please identify yourself in your Piazza entries
    - LAs will monitor that and try to help, but if you want an answer from me, send email

Homework

- Assignments will typically be 2 weeks long
- Most assignments will involve writing programs (best in Matlab) using pre-programmed functions or directly running pre-programmed functions
  - Get right version of Matlab on your laptop; UNC has a license
    - LAs will help
  - These will be done in teams of 2 (assigned by me at random and re-assigned every 2 assignments)
    - Each of you needs to write your own program (incl. your own internal documentation), and you will find problems in each other's programs
    - You will pass in separate programs to the unit testing system, which will do the grading of correctness; manual grading of documentation will be applied
- A few assignments will be trials of questions that are like on what the exam will be: on properties of math and algorithms and on concepts
  - These you will do alone

Grading

- Two exams, both open "book" and notes: 2/3 of average of best two units of
  - Midterm: 1 unit
  - Final, in classroom at scheduled time: 2 units
  - That is, if final score > midterm score, exam score := final score, else exam score := (midterm score + final score)/2
- Homework score: 33%
- Class participation: upgrader for grades near inter-grade boundary
  - Class attendance, incl. possible pop quizzes
  - Asking questions and making corrections in class
  - Coming to my office hours and those of TA
  - Participation in Piazza discussions

Homework Grades

- Sum of weighted individual assignment scores
  - I expect 6-7 programming assignments
    - Pass in via Gradescope and/or Matlab testing program
    - Run that program for yourself on your own data as check
  - Weights
    - on time: 100%
    - more than 1 week late: 50%
    - less than 1 week late: 90%
      - The most advantageous of these will have a weight of 100%