Low Level Processing

CSC 249/449
Sensory Motor Systems
Randal C. Nelson

Low-Level Operations

- Typically image to image mapping
- Bottom up
- No use of detailed, high-level models
- Prior knowledge used is generic (i.e. noise levels)
- Sometimes referrred to as "image preprocessing"

Two general divisions:

- Set of general, functional operations that recur in different contexts
- Set of useful general techniques/formalisms that are useful for achieving desired functionality

Motivation: Binary Image Processing

A lot of low-level operations originated, historically, from efforts to obtain a binary (or multi-valued labeled/iconic) images. The reason is that there are a number of useful operations that are simple to carry out on binary or labeled images.

- Boundary extraction
- Connected components
- Object counting
- Euler number = number of objects - number of holes
- Computation of simple geometric properties
  - Area
  - Centroid
  - Perimeter
  - Bounding box
  - Diameter, approximate diameter
  - Convex hull
  - Moments
  - Projections
- Simple shape analysis

Functional Operations

A lot of low-level processing fits into one of a few basic, functional classes, with various methods of achieving the functionality.

- Simple segmentation
- Contrast enhancement/sharpening
- Smoothing/noise reduction
- Simple pattern matching
- Local edgel detection
- Geometric operations

Low-level Segmentation

Segmentation is the process of creating an image where the pixels are labeled with a discrete set of iconic values (e.g. foreground, background) rather than values representing a continuous space. The simplest method of segmentation is via a gray-level threshold. Often a threshold can be determined automatically by analysis of a gray-scal histogram.

- Thresholding: Set all pixels greater than threshold T to 1 (max) and all others to 0 (min).
- Every threshold defines a binary image.
- Histogram: A frequency count of gray levels over an image (similar to a probability density function).
- Find threshold by looking for bimodality in histogram: (smooth, find main peaks, set threshold at valley in between)
- Lots of tricks for doing this
- Can generalize to local threshold

Contrast Enhancement and Sharpening

Frequently used to make image more suitable for viewing. Also useful in some machine vision applications, where enhancement can make feature detection operations more stable/robust. Simple techniques inlcude the following.

- Background subtraction: Remove constant background, as might arise from dark-current in a CCD chip. Useful for correcting "washed out" appearance.
- Histogram Equalization: Make full use of available gray levels by spreading image histogram out evenly while preserving relative order. Often increases apparent contrast. Generalizes to mapping histogram into specified functional form
- Log transform: In images that are linearly related to brightness, this transforms values to ones more directly related to reflectance - a more invariant property. Can increase apparent contrast. However, most images obtained from ccd video cameras already have a non-linear transform incorporated.
- Sharpening: Various techniques for increasing sharpness of graded boundaries (as might result from slight blurring). Can enhance visual clarity, make boundaries easier to detect compitationally. Also increases noise.

Smoothing and Noise Reduction

Operationally similar techniques that serve two distinct functions:

- Reduction of fluctuations due to uncertainty in measurement process
- Reduction in “aliasing” effects due to digitization and quantization processes.

Simple techniques (all blur boundaries)

- Block averaging: Set each pixel to neighborhood average.
- Weighted average: Set pixel to weighted neighborhood average.
- Gaussian filter: Weights are determined by circular Gaussian.
- Function fitting: Fit function to neighborhood and interpolate.

“Edge Preserving” techniques

- Median filter: Set pixel to median neighborhood value. Somewhat more expensive than averaging, eliminates shot noise completely.
- Adaptive neighborhood filters: determine neighborhood membership using some rule (e.g. close to central pixel in value) and run smoothing operation only on that subset of neighbors.

Simple Pattern Matching

Basic idea is to determine how well one image patch matches another. Generally a model patch or *template* is compared against image locations. Usually a metric (distance) of some sort. Yes/no decisions generally made by thresholding metric

Metrics vary greatly in complexity. At the simple end are metrics that are specified as a direct computation. More complex metrics may be specified in terms of unknown parameters; frequently some search procedure is employed in order to determine the value of these parameters.

Simple metrics - compare one patch set directly on top of another. Typically degrade gracefully with expected distortion in value space.

- Hamming distance: Count number of pixels where labels disagree in binary or iconic images.
- SSD, SAD: Sum of squared differences, sum of absolute differences. Permit graded comparison of gray-scale images.
- Correlation: (Cross correlation) High (rather than low) value for good match. Bias due to overall image brightness. Generally some sort of normalization is needed to correct this.

"Flexible" metrics - metrics designed to degrade gradually with expected distortion in geometric space. All these involve (implicitly or explicitly) some unknown parameters, that are located by a search procdure.

- Neighborhood search: Count template pixels for which a matching pixel can be found in the image within some neighborhood (analog of Hamming distance). Or add up best match within neighborhood (analog of SSD or SAD).
- Hausdorf distance: For each pixel in the template, add up distances to nearest matching pixel in image.
- Template warping: perform parametric warp of template (e.g. affine) to obtain best fit. Can be computationally expensive.

Neighborhood metrics sensitive to "checkerboard" and multiple match problems. Smoothing filters can help with both issues, and enforcing 1-1 matching can ameliorate multiple match dificulties.

Local Edge Detection

Detect *edgels* or assign "edginess" measure to pixels in image. Idea is to flag pixels likely to be on or adjacent to image locations corresponding to physical boundaries in the world. Often both intensity (confidence) and directionality values are associated with edgels.

Large number of methods aimed at intensity changes.

- 1st derivative methods: Find image gradient by correlating with difference templates (e.g. Sobel operators). Provides direction and magnitude. Typically needs to be thinned (e.g. by non-maximum supression in directon of gradient)
- 2nd derivative methods: Place edges at 0 crossings of digital Laplacian. Theoretically produces closed contours, in practice this is not important. Usually necessary to combine with first derivative information on magnitude and direction.
- Template matching methods: Correlate with several templates representing step edges in different directions (Sobel, Kirsch operators). Take maximum (maybe interpolate).
- Other local measures: (e.g. directional variance)
- Gradient measures can be generalized to color image: Direction is that producing fastest change in color vector. "Magnitude" is 3D vector indicating color change.

But what is an edge? Various methods optimized for detection of various mathematical abstractions (e.g. step edge, one-sided uniform region), but none of these is particularly precise model for images of physical, real-world boundaries.

Geometric operations

Simple geometric operations

- Translation / windowing
- Rescale (magnify, reduce)
- Rotation (about some point)
- Affine transformation (general linear + translation)
- Non-linear warp ("morphing")

Resampling is main issue. Typically sample points or neighborhoods for new image are mapped into old image, and values computed. Avoiding aliasing is significant problem.

- Interpolation: Used when sample neighborhood is smaller than or close in size to original neighborhoods (e.g. magnifying image). Bilinear, bicubic, are popular.
- Averaging: Used when sample neighborhood subsumes several neighborhoods in original image. (e.g. reducing image)
- Intermediate cases can be messy.

General Techniques

- Histograms
- Convolution / Correlation
- Fourier domain / Wavelets
- Morphology

Histograms

Types

- Gray-scale / one-dimensional
- Color / three-dimensional
- Other properties (gradient, motion, texture)

Uses

- Use for finding thresholds in 1D.
- Use for simple clustering in 1 and higher dimensions. Local maxima form seeds.
- Multi-dimensional histograms (e.g. color) are sometimes useful for recognition

Convolution and Correlation

Convolution is defined by integral. $f * g(x) = \int_{-\infty}^{\infty} f(s)g(x-s)ds$ Intuitively $f * g(x)$ can be considered a weighted average of f about x with *kernal* g representing the weighting function (flipped). Correlation is just convolution with axis flipped. Kernal can be thought of as a *point spread function*, modelling e.g., a lens out of focus.

Properties:

- Linear
- Shift-invariant
- Commutative
- Associative
- Generalizes to 2 and higher dimensions

Used for:

- Differential operations
- Smoothing operations
- Pattern matching

The *delta function* or (impulse function) is defined as the function δ such that $f * \delta = f$. Not a true function; but an example of a *generalized function*

Fourier Transform and Frequency Domain

Generalization of Fourier series, defined by the integral
$\mathcal{F}[f(x)] = f(u) = int_{-\infty}^{\infty} f(x) e^{-2\pi iux} dx$

- Information in f(x) is preserved, but is now represented in *frequency domain*
- Can be thought of as decomposing signal into a sum of sin waves of (all) different frequencies.
- Note that transform is complex function. Corresponds to sin and cos terms in Fourier series. Often represented using magnitude (power spectrum) and phase components.
- Generalizes directly to 2D (images)
- Digital version known as the *Fast Fourier Transform* (FFT) runs in $O(n \log(n))$ time (instead of naive n^2).

Digital Filtering in Frequency Domain

- High pass filters (generally sharpening effect)
- Low pass filters (generally smoothing effect)
- Important property is the *convolution theorem*. States that convolution in the spatial domain is equivalent to multiplication in the Fourier domain. That is $\mathcal{F}(f * g) = \mathcal{F}(f) \cdot calF(g)$
- Permits large convolutions to be performed efficiently
- Permits correction of degraded (e.g. blurred) images via "deconvolution" with blurring function - division in Fourier domain.

Morphology

Generalized (usually binary) notion of local neighborhood operation. Big literature, lots of image processing applications (e.g. thinning via “erosion” and “dilation” operations). Related to cellular automata (e.g. “Game of Life”).