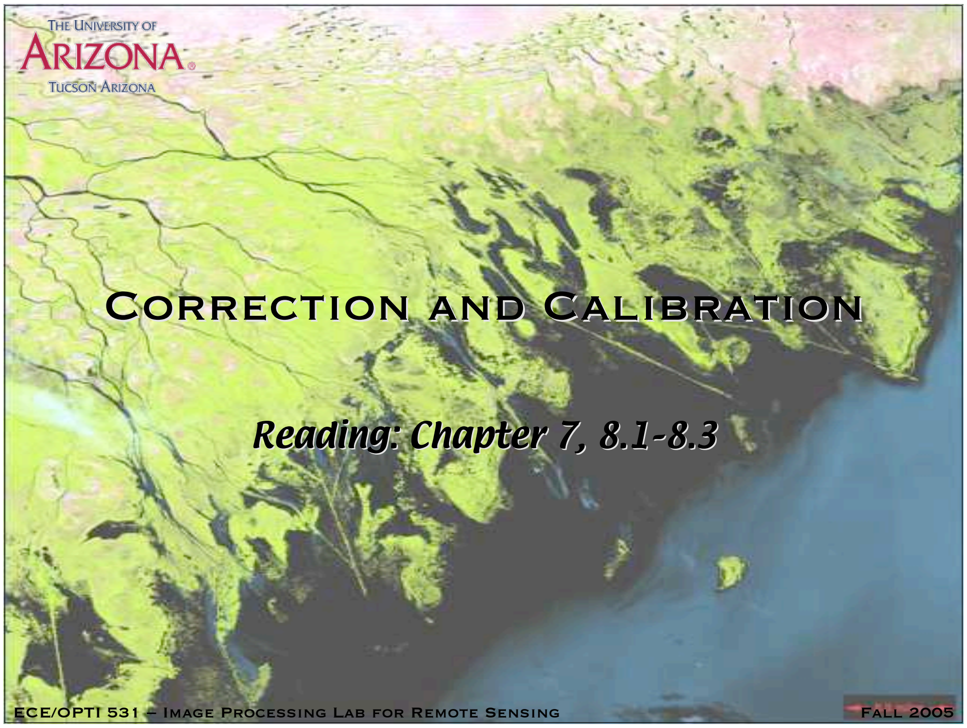
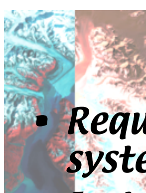


Correction and Calibration

Reading: Chapter 7, 8.1-8.3

Preprocessing



- *Required for certain sensor characteristics and systematic defects*
- *Includes:*
  - *noise reduction*
  - *radiometric calibration*
  - *distortion correction*
- *Usually performed by the data provider, as requested by the user (see Data Systems)*

Correction and Calibration

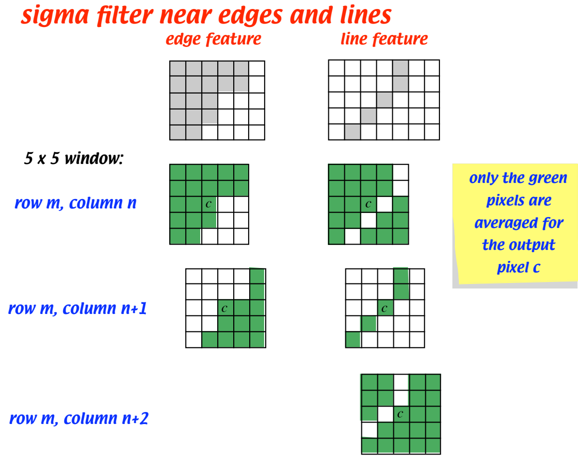
- *Noise Reduction*
- *Radiometric Calibration*
- *Distortion Correction*

Noise Reduction

- *Requires knowledge of noise characteristics*
- *Since noise is usually random and unpredictable, we have to estimate its characteristics from noisy images*
- *Global Noise*
  - *Random DN variation at every pixel*
  - *LPFs will reduce this noise, but also smooth image*
  - *Edge-preserving smoothing algorithms attempt to reduce noise and preserve signal*

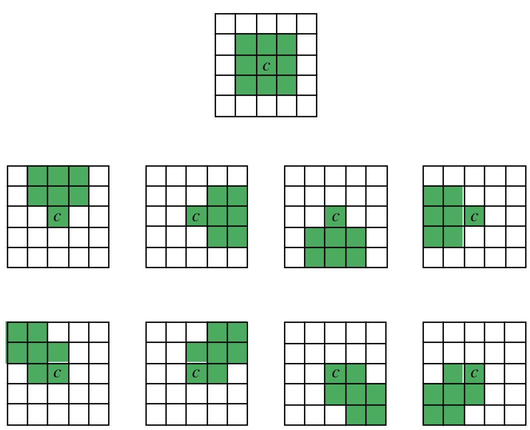
Sigma Filter

- *Average moving window pixels that are within a threshold difference from the DN of the center pixel,* $\mathrm{DN}_C \pm \Delta$

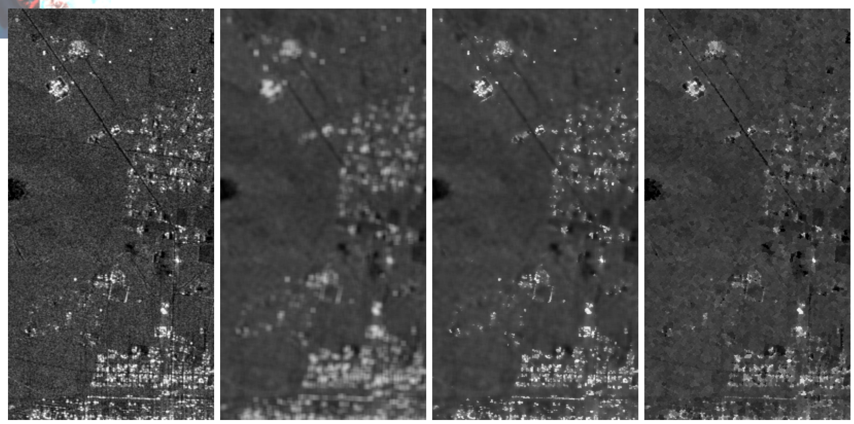


Nagao-Matsuyama Filter

- *Calculate the variance of 9 subwindows within a* 5×5 *moving window*
- *Output pixel is the mean of the subwindow with the lowest variance*



Example: SAR Noise Reduction

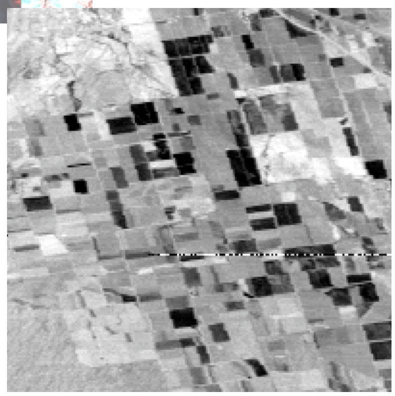


original *5 x 5 LPF* *5 x 5 sigma (k=2)* *Nagao-Matsuyama*

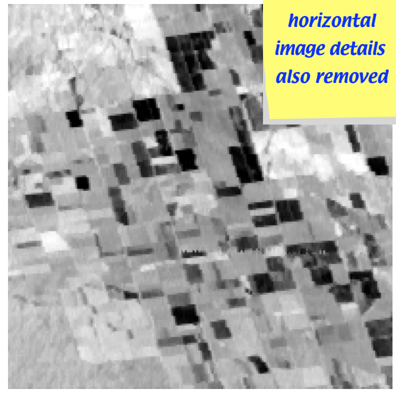
Noise Reduction (cont.)

- ***Local Noise***
  - ***Individual bad pixels or lines***
  - ***Often DN = 0 ("pepper"), 255 ("salt") or 0 and 255 ("salt and pepper")***
  - ***Median filter reduces local noise because it is insensitive to outliers (Chapter 6)***
    - ***For bad lines, set filter window vertical***

Example: Bad-Line Removal



single noisy partial scanline
(Landsat MSS)



horizontal image details also removed

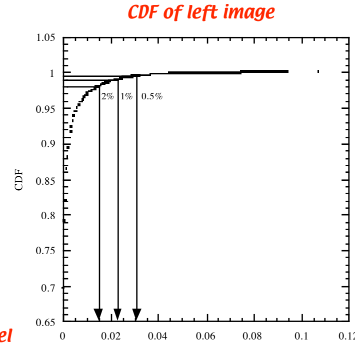
after 3 x 1 median filter

Selective Median Filter

- Detection by spatial correlation
  - Use global spatial correlation statistics to limit median filter

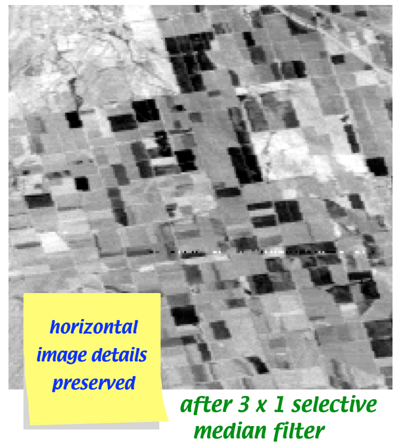
squared-difference image Δ_{ij} between each pixel and its immediate neighbor in the preceding line

CDF of left image



Selective Median Filter (cont.)

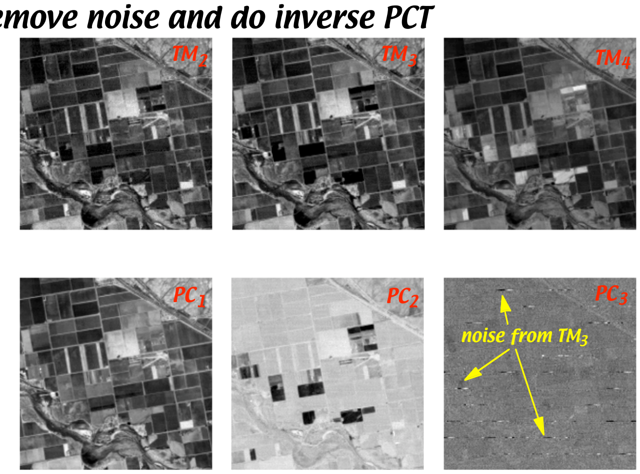
98% threshold mask
for median filter



after 3 x 1 selective
median filter

PCT Component Filter

- *Detection by spectral correlation*
  - *Use PCT to isolate spectrally-uncorrelated noise into higher order PCs*
  - *Remove noise and do inverse PCT*



Periodic (Coherent) Noise

- Repetitive pattern, either consistent throughout the image (global) or variable (local)
- Destriping
  - Striping is caused by unequal detector gains and offsets in whiskbroom and pushbroom scanners
  - Destripe before geometric processing
  - Global, linear detector matching
    - Adjust pixel DNs from each detector i to yield the same mean and standard deviation over the whole image

$$\mathrm{DN}_i^{new} = \frac{\sigma_{ref}}{\sigma_i}\left(\mathrm{DN}_i - \mu_i\right) + \mu_{ref}$$

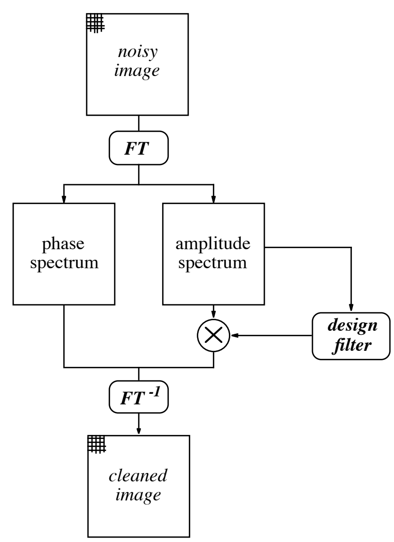
  - Nonlinear detector matching
    - Adjust each detector to yield the same histogram over the whole image (CDF reference stretch)

$$\mathrm{DN}_i^{new} = \mathrm{CDF}_{ref}^{-1}\left[\mathrm{CDF}_i(\mathrm{DN}_i)\right]$$

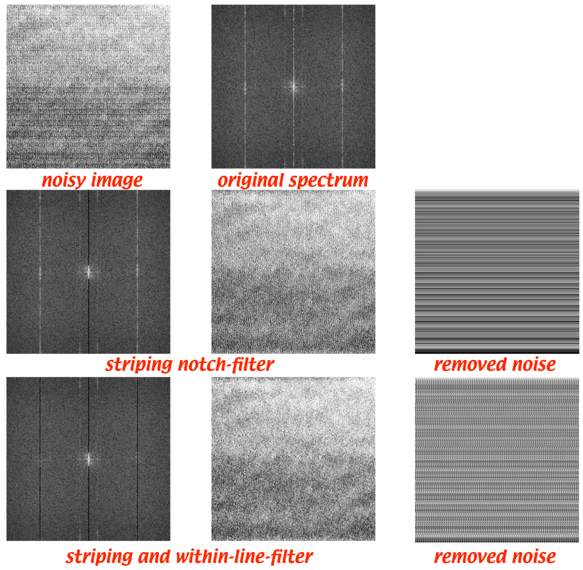
Periodic Noise (cont.)

- Spatial filtering approaches
  - Detector striping causes characteristic "spikes" in the Fourier transform of the image
  - For line striping, the spike is vertical along the v-axis of the Fourier spectrum
  - A "notch" amplitude filter (removes selected frequency components) can reduce striping without degrading image

data flow for amplitude Fourier filtering

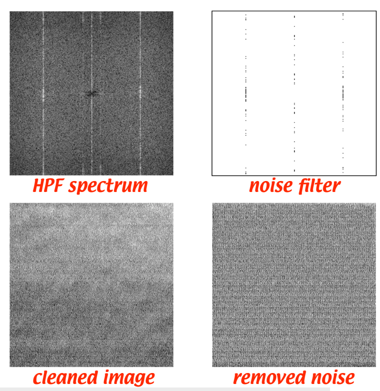


Destriping Examples



Automatic Destriping

- ***"Automatic" periodic noise filter design***
  - ***Attempt to avoid manual noise identification in spectrum***
  - ***Requires two thresholds***

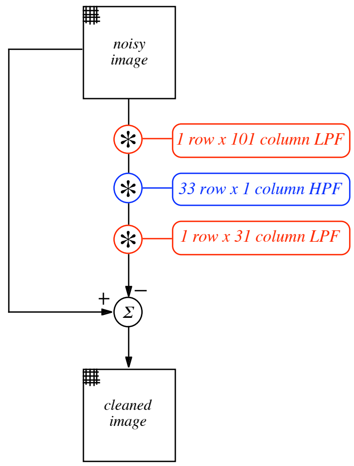


Automatic Periodic Noise Filter

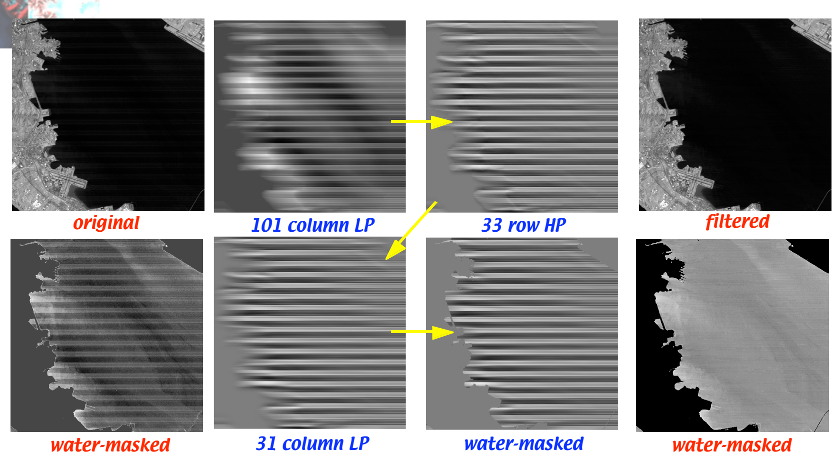
1. Apply "soft" (Gaussian) high-pass filter to noisy image to remove image components
2. Threshold HPF-filtered spectrum to isolate noise frequency components
3. Convert thresholded spectrum to 0 (noise) and 1 (non-noise) to create noise amplitude notch filter
4. Apply filter to noisy image

Debanding

- *Use combination of LPFs and HPFs (Crippen, 1989)*



Example: TM Debanding

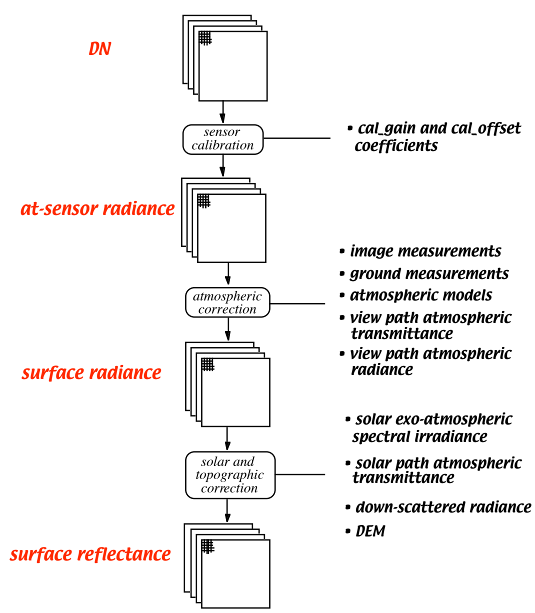


Correction and Calibration

- *Noise Reduction*
- *Radiometric Calibration*
- *Distortion Correction*

Radiometric Calibration

- *Many remote sensing applications require calibrated data*
  - *comparison of geophysical variables over time*
  - *comparison of geophysical variables derived from different sensors*
  - *generation of geophysical variables for input to climate models*
- *Extent of calibration depends on the desired geophysical parameter and available resources for calibration*



Sensor Calibration

- *Calibrate sensor gain and offset in each band (and possibly for each detector) to get at-sensor radiance*
- *Use pre-launch or post-launch measured gains and offsets*

$$L_b^s = \text{cal\_gain}_b \cdot \text{DN}_b + \text{cal\_offset}_b$$

pre-flight TM calibration coefficients

	Landsat-4		Landsat-5	
Band	cal_gain	cal_offset	cal_gain	cal_offset
1	0.672	-3.361	0.642	-2.568
2	1.217	-6.085	1.274	-5.098
3	0.819	-4.917	0.979	-3.914
4	0.994	-9.936	0.925	-4.629
5	0.120	-0.7208	0.127	-0.763
6	0.0568	1.252	0.0552	1.238
7	0.0734	-0.367	0.0677	-0.0338

Atmospheric Correction

- *Estimate atmospheric path radiance and view-path transmittance to obtain at-surface radiance (sometimes called surface-leaving radiance)*

At-sensor: $L_b^s(x,y) = \tau_{vb} L_b(x,y) + L_b^{sp}$

- *Solve for at-surface radiance*

Earth-surface: $L_b(x,y) = \dfrac{L_b^s(x,y) - L_b^{sp}}{\tau_{vb}}$

- *In terms of calibrated at-sensor satellite data*

Earth-surface: $L_b(x,y) = \dfrac{\text{cal\_gain}_b \cdot \text{DN}_b + \text{cal\_offset}_b - L_b^{sp}}{\tau_{vb}}$

In-Scene Methods

- ***Path radiance can be estimated with the Dark Object Subtraction (DOS) technique***
  - *In-scene method assumes dark objects have zero reflectance, and any measured radiance is attributed to atmospheric path radiance only*
  - *Subject to error if object has even very low reflectance*
  - *View-path atmospheric transmittance is not corrected by DOS*

```
Dark-Object Subtraction

1.  Identify "dark object" in the scene
2.  Estimate lowest DN of object, DN_0b
3.  Assume DN_0b ≡ L_b^sp
4.  DN values (calibrated to at-sensor radiance) within
    the dark object assumed to be due only to atmospheric
    path radiance
5.  Subtract DN_0b from all pixels in band b
```


Atmospheric Modeling

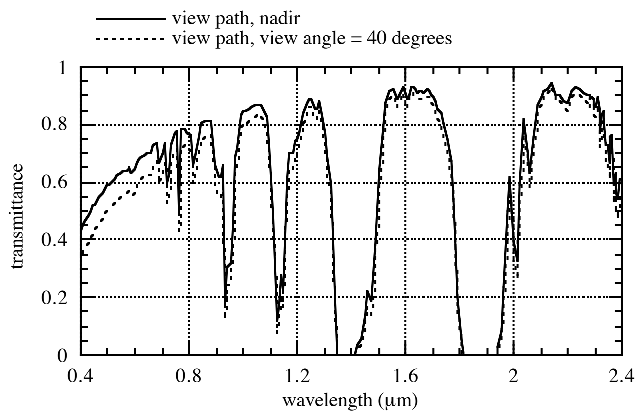
- ***DOS can be improved by incorporating atmospheric models***
  - *Reduces sensitivity to dark object characteristics*
  - *For example, coarse atmospheric characterization (Chavez, 1989)*

visibility	scattering model
very clear	$\lambda^{-4.0}$
clear	$\lambda^{-2.0}$
moderate	$\lambda^{-1.0}$
hazy	$\lambda^{-0.7}$
very hazy	$\lambda^{-0.5}$

  - *Can also fit DN_{0b} with λ^{-K} function to determine K, and use fitted model instead of DN_{0b}*
- ***View-path atmospheric transmittance can be estimated using atmospheric models, such as MODTRAN***

Atmospheric Modeling (cont.)

- *Atmospheric modeling requires knowledge of many parameters*
  - *Imaging geometry*
  - *Atmospheric profile and aerosol model*



Solar Geometry and Topography

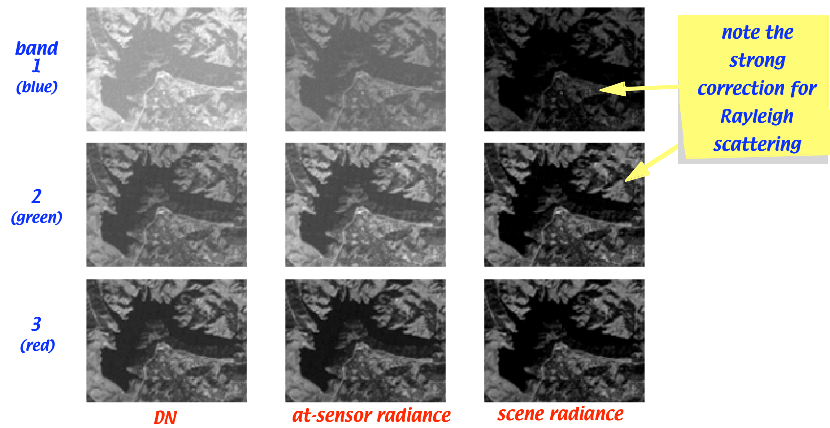
- *Further calibration to reflectance requires 3 more parameters*

$$\rho_b(x,y) = \frac{\pi L_b(x,y)}{\tau_{sb} E_b^0 \cos\left[\theta(x,y)\right]}$$

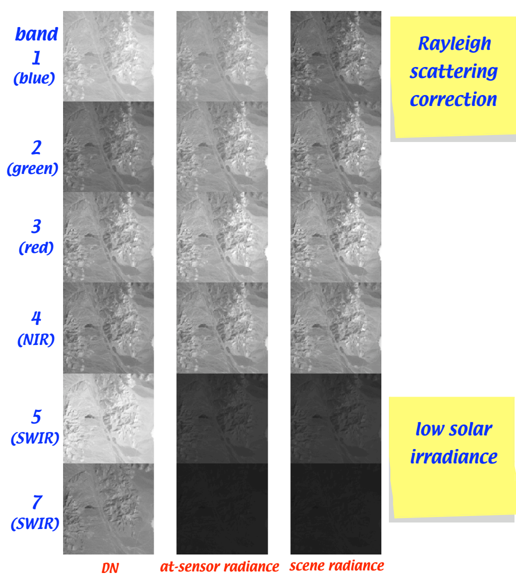
  - *solar path atmospheric transmittance (from model or measurements)*
  - *exo-atmospheric solar spectral irradiance (known)*
  - *incident angle (from DEM)*

Example: MSI Calibration

- *Partial calibration with correction for sensor gains and offsets and DOS*

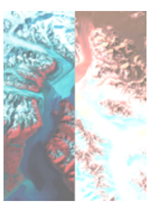


Example: Partial Calibration



Hyperspectral Normalization

- *Calibration of hyperspectral imagery is particularly important because:*
  - *High sensitivity to narrow atmospheric absorption features or the edges of broader spectral features*
  - *Spectral band shift in operating sensor*
  - *Need for precise absorption band-depth measurements*
  - *Computational issues*
- *A number of "normalization" techniques that use scene models have been developed to partially calibrate hyperspectral data*



Normalization (cont.)

technique	*description*	*reference*
residual image	scale each pixel's spectrum by a constant such that the value in a selected band equals the maximum value in that band for the entire scene subtract the average normalized radiance in each band over the entire scene from the normalized radiance in each band	(Marsh and McKeon, 1983)
continuum removal	generate a piecewise-linear or polynomial continuum across "peaks" of image spectrum and divide each pixel's spectrum by the continuum	(Clark and Roush, 1984)
internal average relative reflectance (IARR)	divide each pixel's spectrum by the average spectrum of the entire scene	(Kruse, 1988)
empirical line	band-by-band linear regression of pixel samples to field reflectance spectra for dark and bright targets	(Kruse *et al.*, 1990)
flat-field	divide each pixel's spectrum by the average spectrum of a spectrally-uniform, high reflectance area in the scene	(Rast *et al.*, 1991)

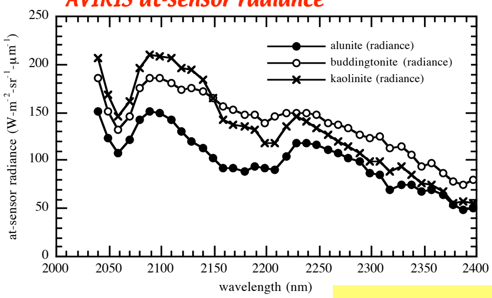
effectiveness of various normalization techniques for calibration

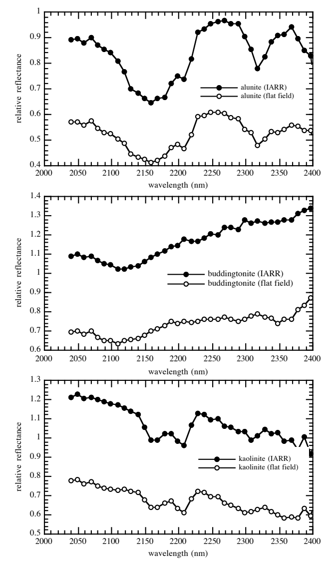
technique	*view-path radiance*	*topography*	*solar irradiance*	*solar path atmospheric transmittance*
residual images	✓	✓	✓	✓
continuum removal	✗	✗	✓	✗
IARR	✗	✗	✓	✓
empirical line	✓	✗	✓	✓
flat field	✗	✗	✓	✓

Example: HSI Normalization

- *Radiance normalized by flat field and IARR techniques*

AVIRIS at-sensor radiance

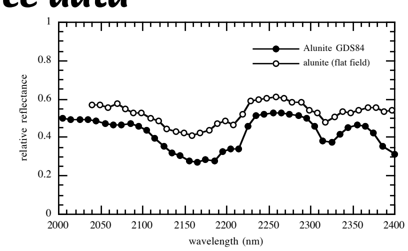




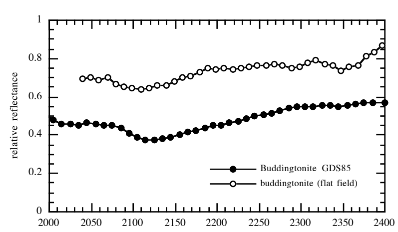
mineral discrimination confused by solar background and topographic shading

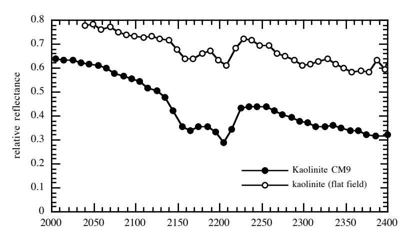
Example (cont.)

- *Radiance normalized by flat fielding, compared to reflectance data*



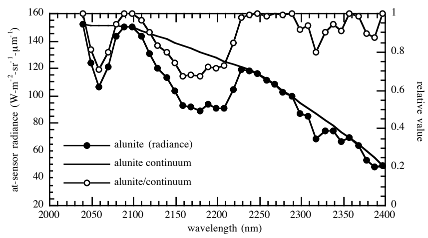
similarity implies we can use reflectance library data for image class identification



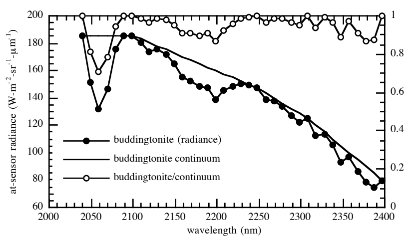


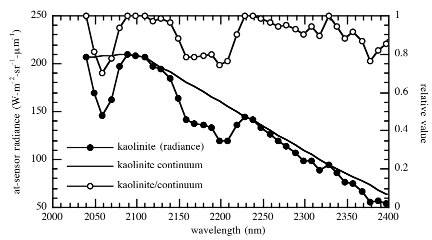
Example (cont.)

- *Radiance normalized by continuum removal*



removes solar background





Correction and Calibration

- *Noise Reduction*
- *Radiometric Calibration*
- *Distortion Correction*

Distortion Correction

- *Applications*
  - *correct system distortions*
  - *register multiple images*
  - *register image to map*
- *Three components to warping process*
  - *selection of suitable mathematical distortion model(s)*
  - *coordinate transformation*
  - *resampling (interpolation)*

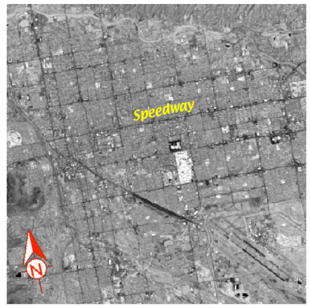
```
Distortion Correction Implementation

1.  Create an empty output image (which is in the
    reference coordinate system)
2.  Step through the integer reference coordinates, one at
    a time, and calculate the coordinates in the distorted
    image (x,y) by Eq. 7-27
3.  Estimate the pixel value to insert in the output image
    at (xref,yref) from the original image at (x,y)
    (resampling)
```


Definitions

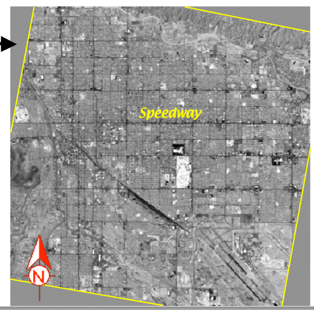
- ***Registration**: The alignment of one image to another image of the same area.*
- ***Rectification**: The alignment of an image to a map so that the image is planimetric, just like the map. Also known as **georeferencing**.*
- ***Geocoding**: A special case of rectification that includes scaling to a uniform, standard pixel GSI.*
- ***Orthorectification**: Correction of the image, pixel-by-pixel, for topographic distortion.*

Example: TM Rectification



Original (Tucson, AZ)

fill ⟶

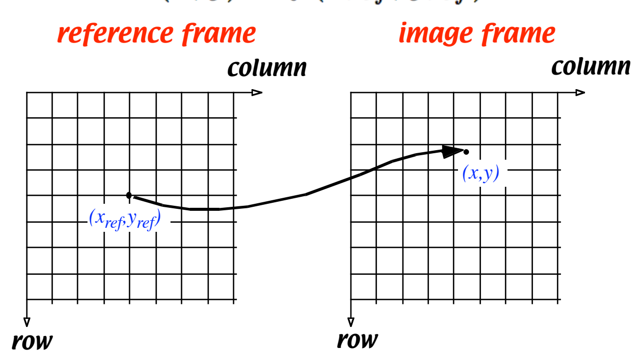


Rectified

Coordinate Transformation

- *Coordinates are mapped from the reference frame to the distorted image frame*

$$(x, y) = f(x_{ref}, y_{ref})$$



- *"Backwards" mapping (x_{ref},y_{ref}) –> (x,y) avoids "holes" or overlaying of multiple pixels in the processed image*

Polynomial Distortion Model

- *Generic model useful for registration, rectification and geocoding*
- *Known as a "rubber sheet stretch"*
- *Relates distorted coordinate system* (x,y) *to the reference coordinate system* (x_{ref},y_{ref})

$$x = \sum_{i=0}^{N} \sum_{j=0}^{N-i} a_{ij} x_{ref}^i y_{ref}^j$$

$$y = \sum_{i=0}^{N} \sum_{j=0}^{N-i} b_{ij} x_{ref}^i y_{ref}^j$$

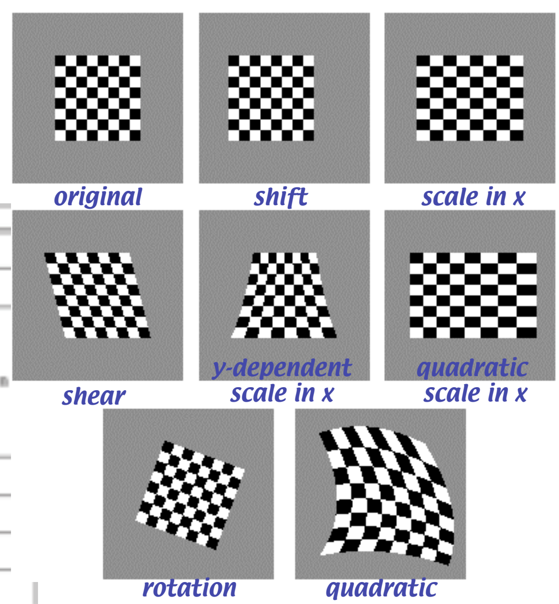
- *For example, a quadratic polynomial is written as:*

$$x = a_{00} + a_{10}x_{ref} + a_{01}y_{ref} + a_{11}x_{ref}y_{ref} + a_{20}x_{ref}^2 + a_{02}y_{ref}^2$$
$$y = b_{00} + b_{10}x_{ref} + b_{01}y_{ref} + b_{11}x_{ref}y_{ref} + b_{20}x_{ref}^2 + b_{02}y_{ref}^2$$

Distortion Types

- *The coefficients in the polynomial can be associated with particular types of distortion*

coefficient	warp component	
a_{00}	shift in x	
b_{00}	shift in y	
a_{10}	scale in x	
b_{01}	scale in y	
a_{01}	shear in x	rotation
b_{10}	shear in y	
a_{11}	y-dependent scale in x	
b_{11}	x-dependent scale in y	
a_{20}	nonlinear scale in x	
b_{02}	nonlinear scale in y	



Affine Transformation

- *A linear polynomial (number of terms K = 3)*
  - *Special case that can include:*
    - *shift*
    - *scale*
    - *shear*
    - *rotation*

$$\begin{aligned} x &= a_{00} + a_{10}x_{ref} + a_{01}y_{ref} \\ y &= b_{00} + b_{10}x_{ref} + b_{01}y_{ref} \end{aligned}$$

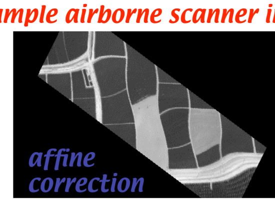
  - *In vector-matrix notation*

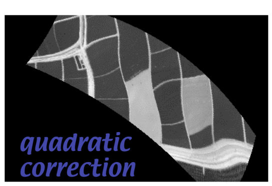
$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_{10} & a_{01} \\ b_{10} & b_{01} \end{bmatrix} \begin{bmatrix} x_{ref} \\ y_{ref} \end{bmatrix} + \begin{bmatrix} a_{00} \\ b_{00} \end{bmatrix}$$

Piecewise Polynomial Distortion

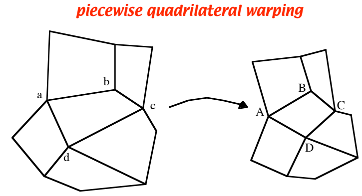
- ***For severely-distorted images that can't be modeled with a single, global polynomial of reasonable order***

Example airborne scanner image

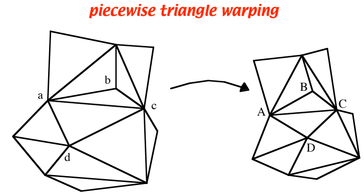




piecewise quadrilateral warping



piecewise triangle warping

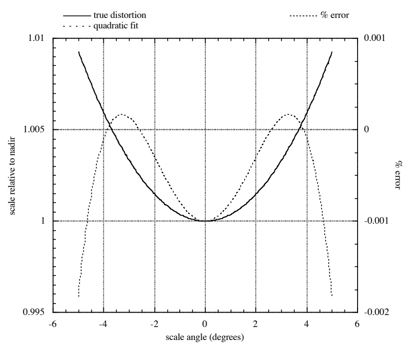


reference frame *distorted frame*

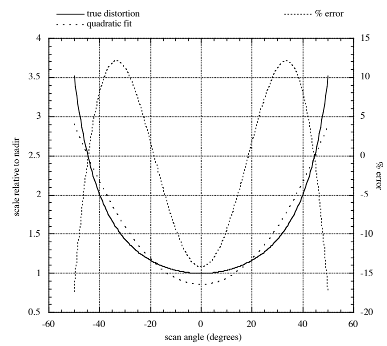
Application to Wide FOV Sensors

- *Comparison of actual distortion to global polynomial model*
  - *accurate for small FOV sensors such as Landsat, but not for large FOV scanners such as AVHRR*

± 5° FOV



± 50° FOV



Ground Control Points (GCPs)

- *Use to control the polynomial, i.e. to determine its coefficients*
  - *Characteristics:*
    - *high contrast in all images of interest*
    - *small feature size*
    - *unchanging over time*
    - *all are at the same elevation (unless topographic relief is being specifically addressed)*
  - *Often located by visual examination, but can be automated to varying degree, depending on the problem*

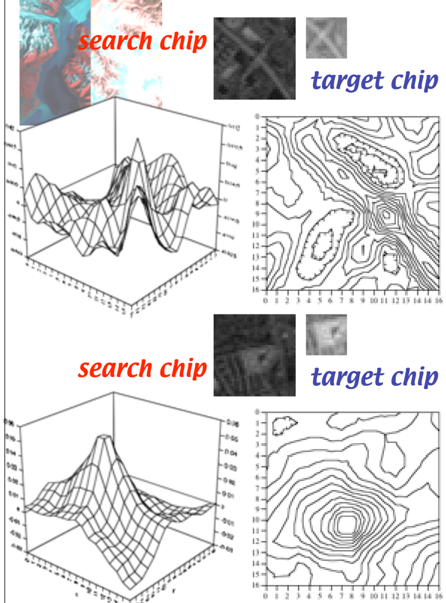
Automatic GCP Location

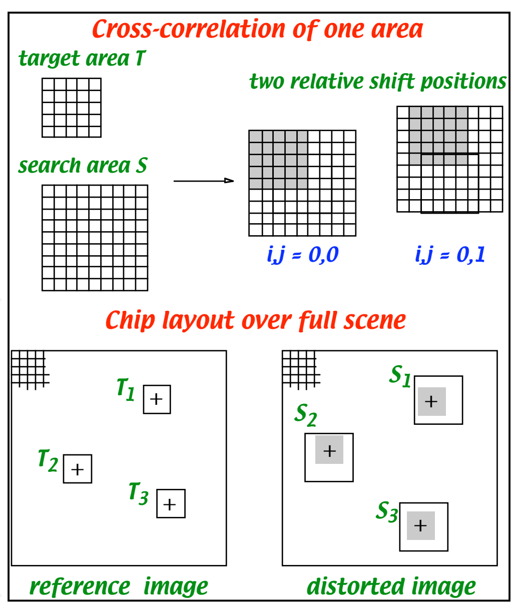
- *Use image "chips"*
  - *Small segments that contain one easily identified and well-located GCP*
- *Normalized cross-correlation between template chip T (reference) and search area S in image to be registered*

$$R_{ij} = \frac{\sum_{m=1}^{N}\sum_{n=1}^{N} T_{mn} S_{i+m,j+n}}{\sqrt{\sum_{m=1}^{N}\sum_{n=1}^{N} T_{mn}^2}\sqrt{\sum_{m=1}^{N}\sum_{n=1}^{N} S_{i+m,j+n}^2}}$$

  - *normalization adjusts for changes in mean DN within area*

Area Correlation





Finding Polynomial Coefficients

- ***Set up system of simultaneous equations using GCPs and solve for polynomial coefficients***
- ***Example with quadratic polynomial (number of terms K = 6)***
  - ***Given M pairs of GCPs***
  - ***For each GCP pair, m, create two equations***

$$\begin{aligned} x_m &= a_{00} + a_{10}x_{refm} + a_{01}y_{refm} + a_{11}x_{refm}y_{refm} + a_{20}x^2_{refm} + a_{02}y^2_{refm} \\ y_m &= b_{00} + b_{10}x_{refm} + b_{01}y_{refm} + b_{11}x_{refm}y_{refm} + b_{20}x^2_{refm} + b_{02}y^2_{refm} \end{aligned}$$

  - ***Then, for all M GCP pairs, in vector-matrix notation***

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix} = \begin{bmatrix} 1 & x_{ref1} & y_{ref1} & x_{ref1}y_{ref1} & x^2_{ref1} & y^2_{ref1} \\ 1 & x_{ref2} & y_{ref2} & x_{ref2}y_{ref2} & x^2_{ref2} & y^2_{ref2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{refM} & y_{refM} & x_{refM}y_{refM} & x^2_{refM} & y^2_{refM} \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{10} \\ a_{01} \\ a_{11} \\ a_{20} \\ a_{02} \end{bmatrix}$$

Solving for Coefficients

- ***So, for each set of GCP x- and y-coordinate pairs, we can write a linear system***
  - ***Determined case*** ***(M = K, just enough GCPs) solution:***

M = K: $\begin{aligned} \mathbf{X} &= \mathbf{WA} \\ \mathbf{Y} &= \mathbf{WB} \end{aligned}$ *solution:* $\begin{aligned} \mathbf{A} &= \mathbf{W}^{-1}\mathbf{X} \\ \mathbf{B} &= \mathbf{W}^{-1}\mathbf{Y} \end{aligned}$

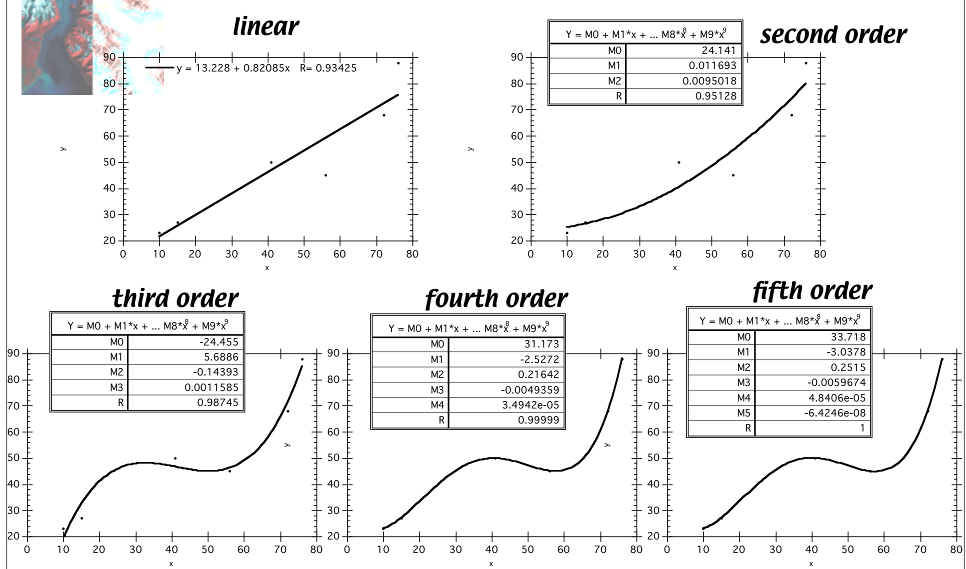
    - ***Exact solution which passes through GCPs, i.e. they are mapped exactly***
  - ***Overdetermined case*** ***(M > K, more than enough GCPs):***

M ≥ K: $\begin{aligned} \mathbf{X} &= \mathbf{WA} + \varepsilon_X \\ \mathbf{Y} &= \mathbf{WB} + \varepsilon_Y \end{aligned}$ *solution:* $\begin{aligned} \hat{\mathbf{A}} &= (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{X} \\ \hat{\mathbf{B}} &= (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{Y} \end{aligned}$

    - $(\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T$ ***is called the pseudoinverse of W***
    - ***solution results in least-squares minimum error at GCPs:***

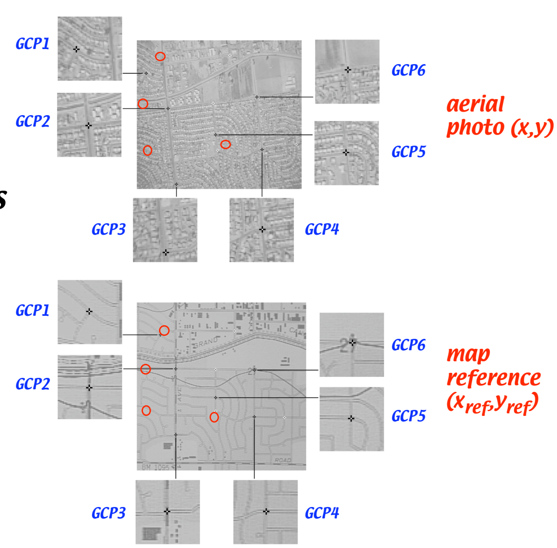
$$\begin{aligned} \min\left[\varepsilon_X^T\varepsilon_X\right] &= (\mathbf{X} - \mathbf{W}\hat{\mathbf{A}})^T(\mathbf{X} - \mathbf{W}\hat{\mathbf{A}}) \\ \min\left[\varepsilon_Y^T\varepsilon_Y\right] &= (\mathbf{Y} - \mathbf{W}\hat{\mathbf{B}})^T(\mathbf{Y} - \mathbf{W}\hat{\mathbf{B}}) \end{aligned}$$

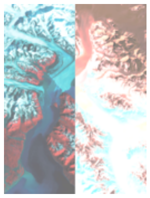
Example: 1-D Curve Fitting



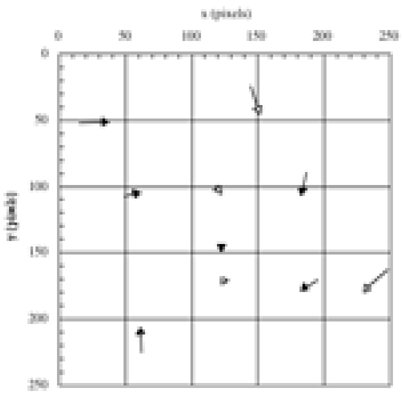
Example: Registration

- ***Register aerial photo to map in an urban area***
- ***Locate 6 GCPs in each (black crosses)***
- ***Locate 4 Ground Points (GPs) in each (red circles)***
  - ***not used for control, only for testing***



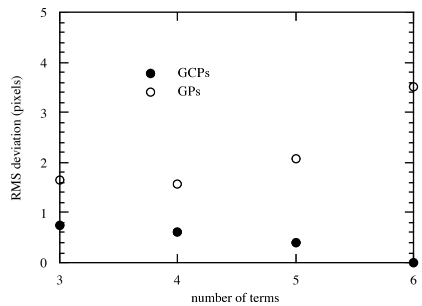


GCP Error



Mapping of GCPs

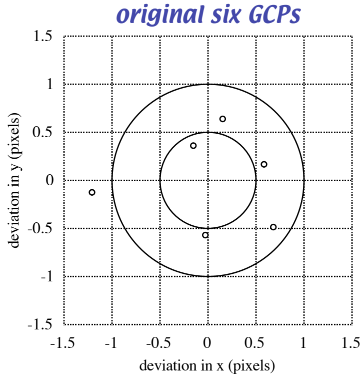
Error at GCPs and Ground Points (GPs) as function of polynomial order



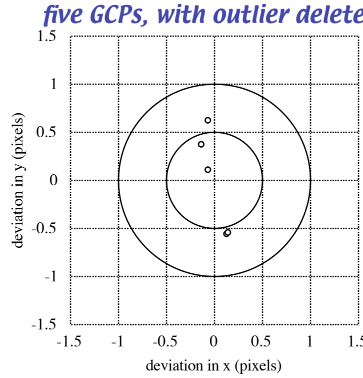
GCP Refinement

- *Analyze GCP error and remove outliers*

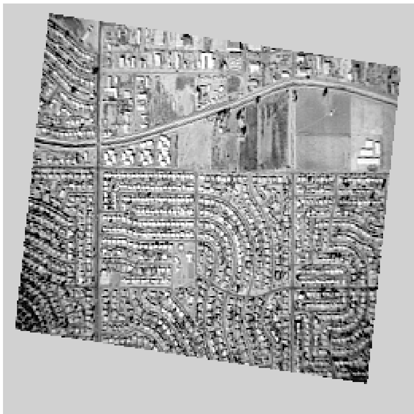
original six GCPs

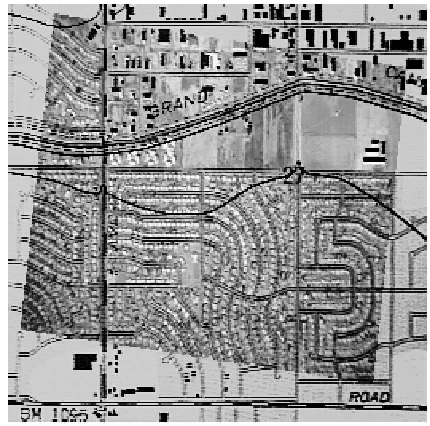


five GCPs, with outlier deleted



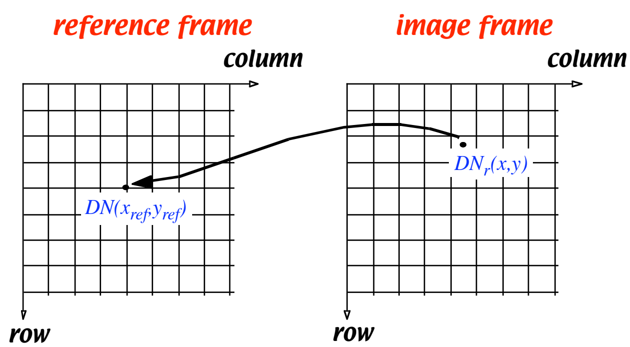
Final Registration Result

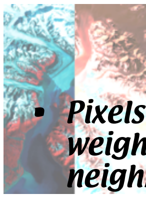




Resampling

- ***The (x,y) coordinates calculated by*** $(x, y) = f(x_{ref}, y_{ref})$ ***are generally between the integer pixel coordinates of the array***
- ***Therefore, must estimate (interpolate or resample) a new pixel at the (x,y) location***





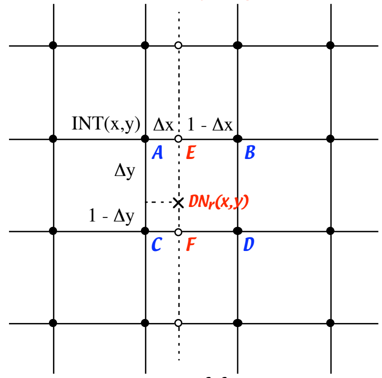
Resampling (cont.)

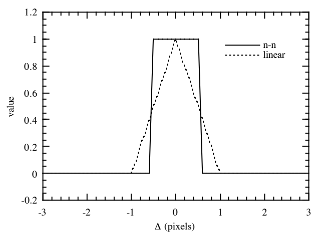
- ***Pixels are resampled using a weighted-average of the neighboring pixels***
- ***Common weighting functions:***
  - ***nearest-neighbor**: fast, but discontinuous*

$$DN_r = DN_C$$

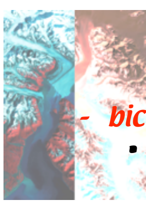
  - ***bilinear**: slower, but continuous*

$$DN_r = [\Delta x DN_B + (1 - \Delta x) DN_A](1 - \Delta y) + [\Delta x DN_D + (1 - \Delta x) DN_C]\Delta y$$





- ***Implement 2-D bilinear resampling as two successive 1-D resamplings***
  - *resample E between A and B*
  - *resample F between C and D*
  - *resample DN(x,y) between E and F*



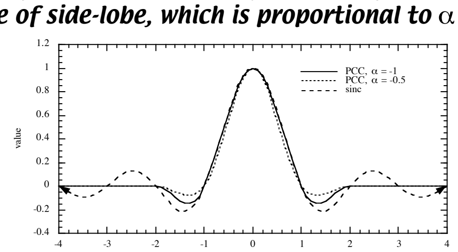
Resampling (cont).

- ***bicubic**: slowest, but results in sharpest image*
  - *piecewise polynomial; special case of Parametric Cubic Convolution (PCC)*

$$DN_r(\Delta;\alpha) = PCC(\Delta,\alpha) = \begin{cases} (\alpha+2)|\Delta|^3 - (\alpha+3)|\Delta|^2 + 1, & |\Delta| \le 1 \\ \alpha(|\Delta|^3 - 5|\Delta|^2 + 8|\Delta| - 4), & 1 \le |\Delta| \le 2 \\ 0, & |\Delta| \ge 2 \end{cases}$$

where Δ is the distance from (x,y) to the grid points in 1-D

  - *"standard" bicubic is α = -1; superior bicubic is α = -0.5*
  - *High-boost filter characteristics; amount of boost depends on amplitude of side-lobe, which is proportional to α*



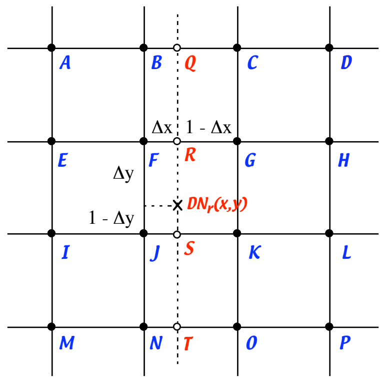
PCC Resampling

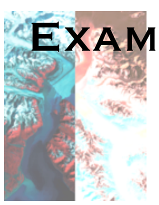
- *PCC resampling procedure*
  - *resample along each row, A-D, E-H, I-L, M-P*

$$\mathrm{DN}_Q = \mathrm{DN}_A\cdot\mathrm{PCC}(\Delta x{+}1;\alpha){+}\mathrm{DN}_B\cdot\mathrm{PCC}(\Delta x;\alpha){+}\mathrm{DN}_C\cdot\mathrm{PCC}(1{-}\Delta x;\alpha){+}\mathrm{DN}_D\cdot\mathrm{PCC}(2{-}\Delta x;\alpha)$$

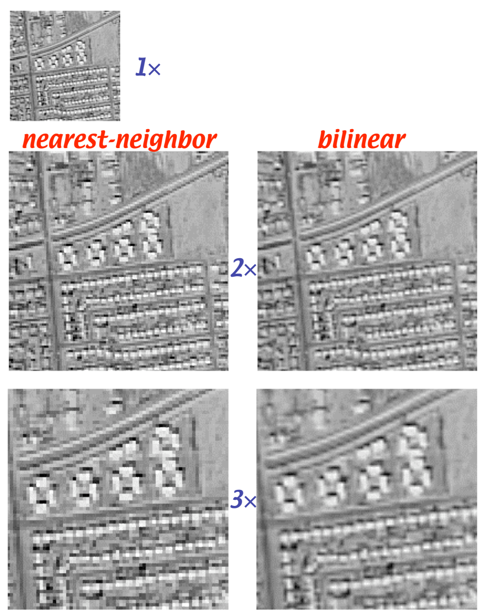
  etc.
  - *resample along new column Q-T*

$$\mathrm{DN}_r = \mathrm{DN}_Q\cdot\mathrm{PCC}(\Delta y{+}1;\alpha){+}\mathrm{DN}_R\cdot\mathrm{PCC}(\Delta y;\alpha){+}\mathrm{DN}_S\cdot\mathrm{PCC}(1{-}\Delta y;\alpha){+}\mathrm{DN}_T\cdot\mathrm{PCC}(2{-}\Delta y;\alpha)$$

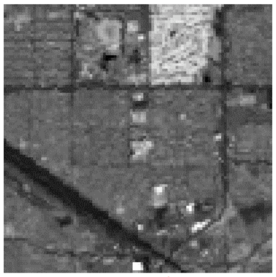




Example: Magnification ("Zoom")



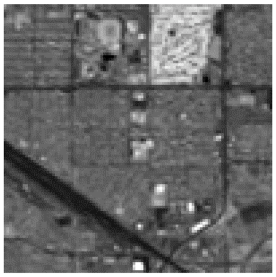
Example: Rectification



nearest-neighbor

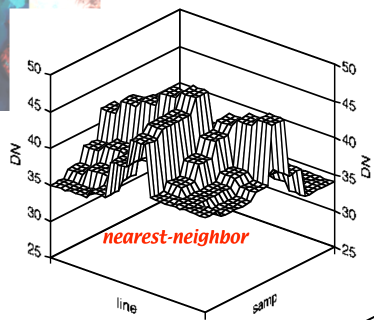
bilinear

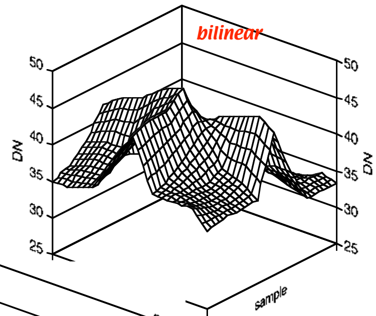
PCC (α = -0.5)



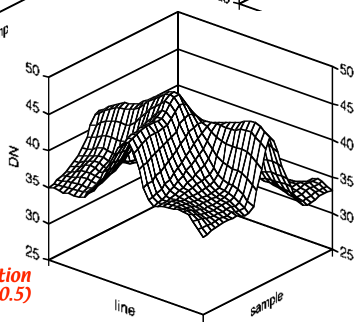
PCC (α = -1.0)

Interpolation Quality





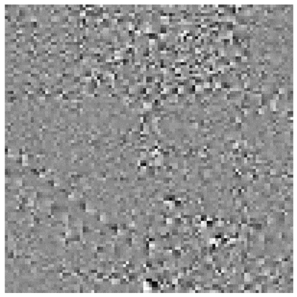
resampled image surface plots (4x zoom)



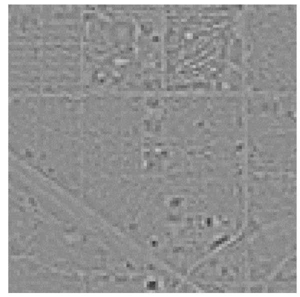
parametric cubic convolution (α = -0.5)

Resampling Differences

- *Difference images between different resampling functions*



nearest-neighbor - bilinear



bilinear - PCC

- *polynomial distortion model affects global geometric accuracy*
- *resampling function affects local radiometric accuracy*