CORRECTING RS SYSTEM DETECTOR ERROR GEOMETRIC CORRECTION

Lecture 1

Correcting Remote Sensing System Detector Error

- Ideally, the radiance recorded by a remote sensing system in various bands is an accurate representation of the radiance actually leaving the feature of interest (e.g., soil, vegetation, water, or urban land cover) on the Earth's surface.
- Unfortunately, *noise* (error) can enter the data-collection system at several points.
  - For example, *radiometric error* in remotely sensed data may be introduced by the sensor system itself when the individual detectors do not function properly or are improperly calibrated.

Correcting Remote Sensing System Detector Error

- Several of the more common remote sensing system–induced radiometric errors are:
  - random bad pixels (shot noise),
  - line-start/stop problems,
  - line or column drop-outs,
  - partial line or column drop-outs, and
  - line or column striping.

Random Bad Pixels (Shot Noise)

- Sometimes an individual detector does not record spectral data for an individual pixel.
  - When this occurs randomly, it is called a ***bad pixel***. When there are numerous random bad pixels found within the scene, it is called ***shot noise*** because it appears that the image was shot by a shotgun.
  - Normally these bad pixels contain values of 0 or 255 (in 8-bit data) in one or more of the bands. Shot noise is identified and repaired using the following methodology. It is first necessary to locate each bad pixel in the band *k* dataset. A simple thresholding algorithm makes a pass through the dataset and flags any pixel ($BV_{i,j,k}$) having a brightness value of zero (assuming values of 0 represent shot noise and not a real land cover such as water). Once identified, it is then possible to evaluate the eight pixels surrounding the flagged pixel, as shown below:

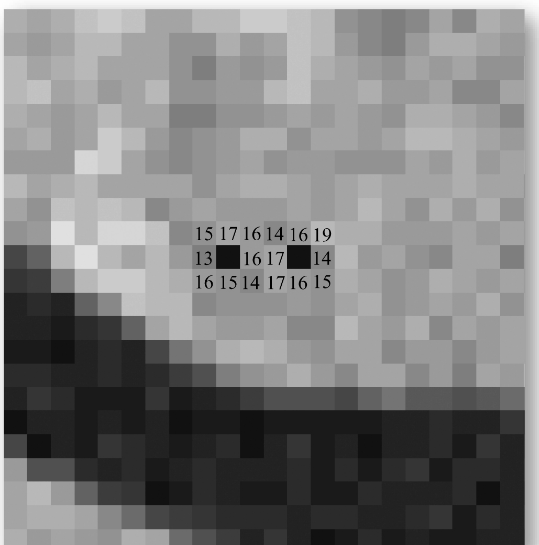
	col_{j-1}	col_j	col_{j+1}
row_{i-1}	BV_1	BV_2	BV_3
row_i	BV_8	BV_{ijk}	BV_4
row_{i+1}	BV_7	BV_6	BV_5

Random Bad Pixels (Shot Noise)

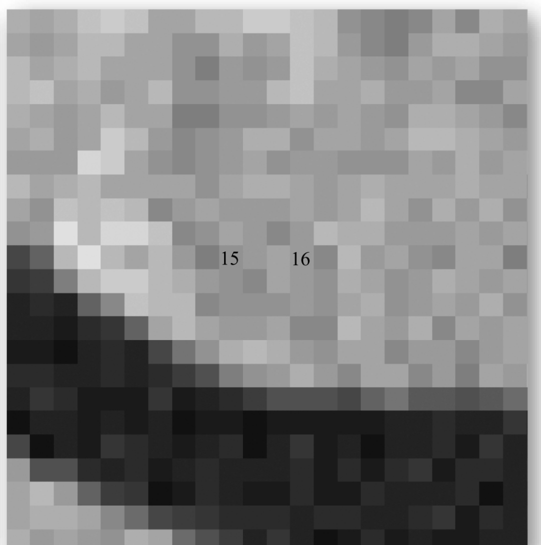
- The mean of the eight (n) surrounding pixels is computed using the equation and the value substituted for $BV_{i,j,k}$ in the corrected image:

$$BV_{i,j,k} = \text{int}\left[\frac{\sum_{n=1}^{8} BV_n}{8} \right]$$

a. Landsat TM band 7 data of the Santee Delta with shot noise.



b. Two pixels along a bad scan line with shot noise.



c. Shot noise removed.

(a) Landsat Thematic Mapper band 7 (2.08 – 2.35 µm) image of the Santee Delta in South Carolina. One of the 16 detectors exhibits serious striping and the absence of brightness values at pixel locations along a scan line.
b) An enlarged view of the bad pixels with the brightness values of the eight surrounding pixels annotated.
c) The brightness values of the bad pixels after shot noise removal. This image was not destriped.

Line or Column Drop-outs

- An entire line containing no spectral information may be produced if an individual detector in a scanning system (e.g., Landsat MSS or Landsat 7 ETM^{+}) fails to function properly.
- If a detector in a linear array (e.g., SPOT XS, IRS-1C, QuickBird) fails to function, this can result in an entire column of data with no spectral information. The bad line or column is commonly called a *line* or *column drop-out* and contains brightness values equal to 0.
- For example, if one of the 16 detectors in the Landsat Thematic Mapper sensor system fails to function during scanning, this can result in a brightness value of zero for every pixel, *j*, in a particular line, *i*. This *line drop-out* would appear as a completely black line in the band, *k*, of imagery. This is a serious condition because there is no way to restore data that were never acquired.
  - However, it is possible to improve the visual interpretability of the data by introducing estimated brightness values for each bad scan line.

Line or Column Drop-outs

- It is first necessary to locate each bad line in the dataset. A simple thresholding algorithm makes a pass through the dataset and flags any scan line having a mean brightness value at or near zero. Once identified, it is then possible to evaluate the output for a pixel in the preceding line ($BV_{i-1,j,k}$) and succeeding line ($BV_{i+1,j,k}$) and assign the output pixel ($BV_{i,j,k}$) in the drop-out line the average of these two brightness values:
- This is performed for every pixel in a bad scan line. The result is an image consisting of interpolated data every *n*th line that is more visually interpretable than one with horizontal black lines running systematically throughout the entire image. This same cosmetic digital image processing procedure can be applied to *column drop-outs* produced by a linear array remote sensing system.

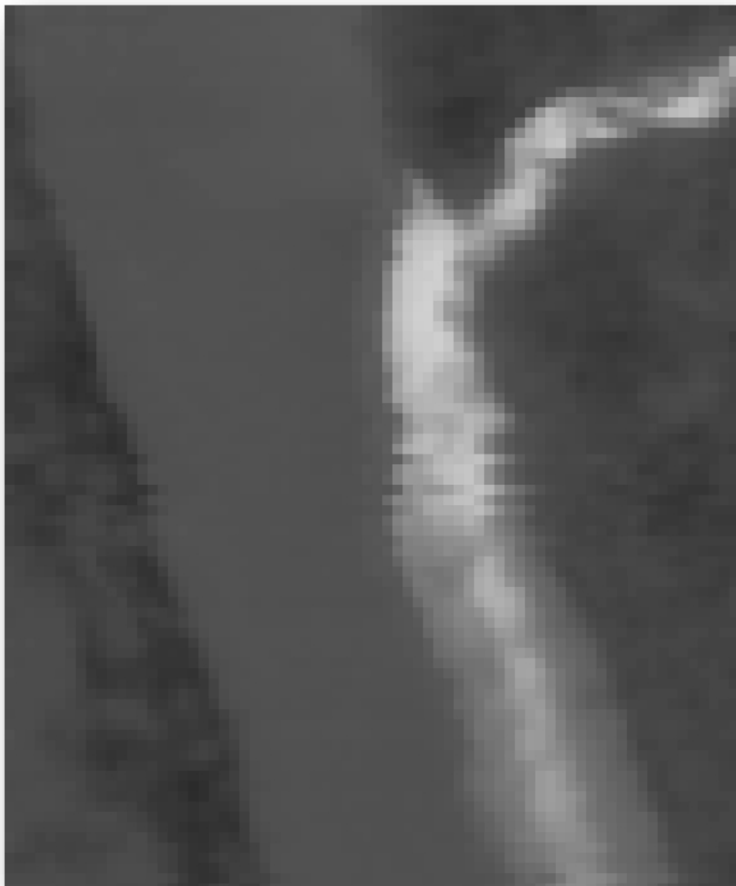
$$BV_{i,j,k} = \text{int}\left[\frac{BV_{i-1,j,k} + BV_{i+1,j,k}}{2}\right]$$

Line-start Problems

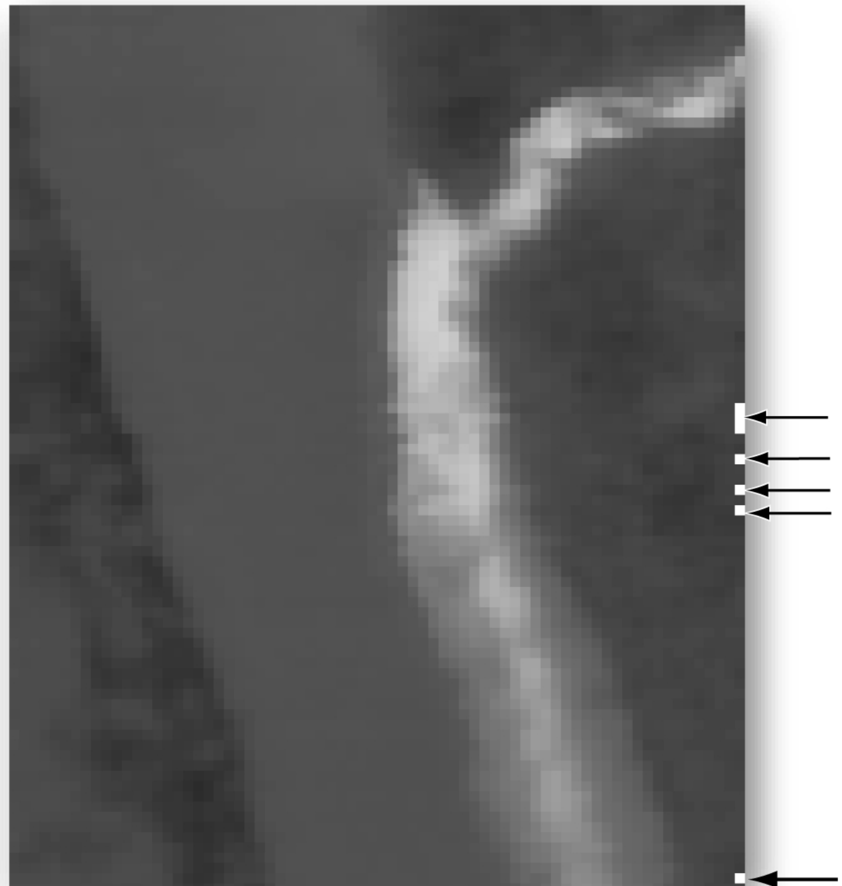
- Occasionally, scanning systems fail to collect data at the beginning of a scan line, or they place the pixel data at inappropriate locations along the scan line.
- For example, all of the pixels in a scan line might be systematically shifted just one pixel to the right. This is called a *line-start* problem.
- Also, a detector may abruptly stop collecting data somewhere along a scan and produce results similar to the line or column drop-out previously discussed. Ideally, when data are not collected, the sensor system would be programmed to remember what was not collected and place any good data in their proper geometric locations along the scan. Unfortunately, this is not always the case.
  - For example, the first pixel (column 1) in band *k* on line *i* (i.e., $BV_{i,1,k}$) might be improperly located at column 50 (i.e., $BV_{i,50,k}$). If the line-start problem is always associated with a horizontal bias of 50 columns, it can be corrected using a simple horizontal adjustment. However, if the amount of the line-start displacement is random, it is difficult to restore the data without extensive human interaction on a line-by-line basis. A considerable amount of MSS data collected by Landsats 2 and 3 exhibit line-start problems.

Line-start Problems

Line-start Problems



a. Predawn thermal infrared imagery of the Savannah River with line-start problems.



b. Seven line-start problem lines were translated one column to the left.

N-line Striping

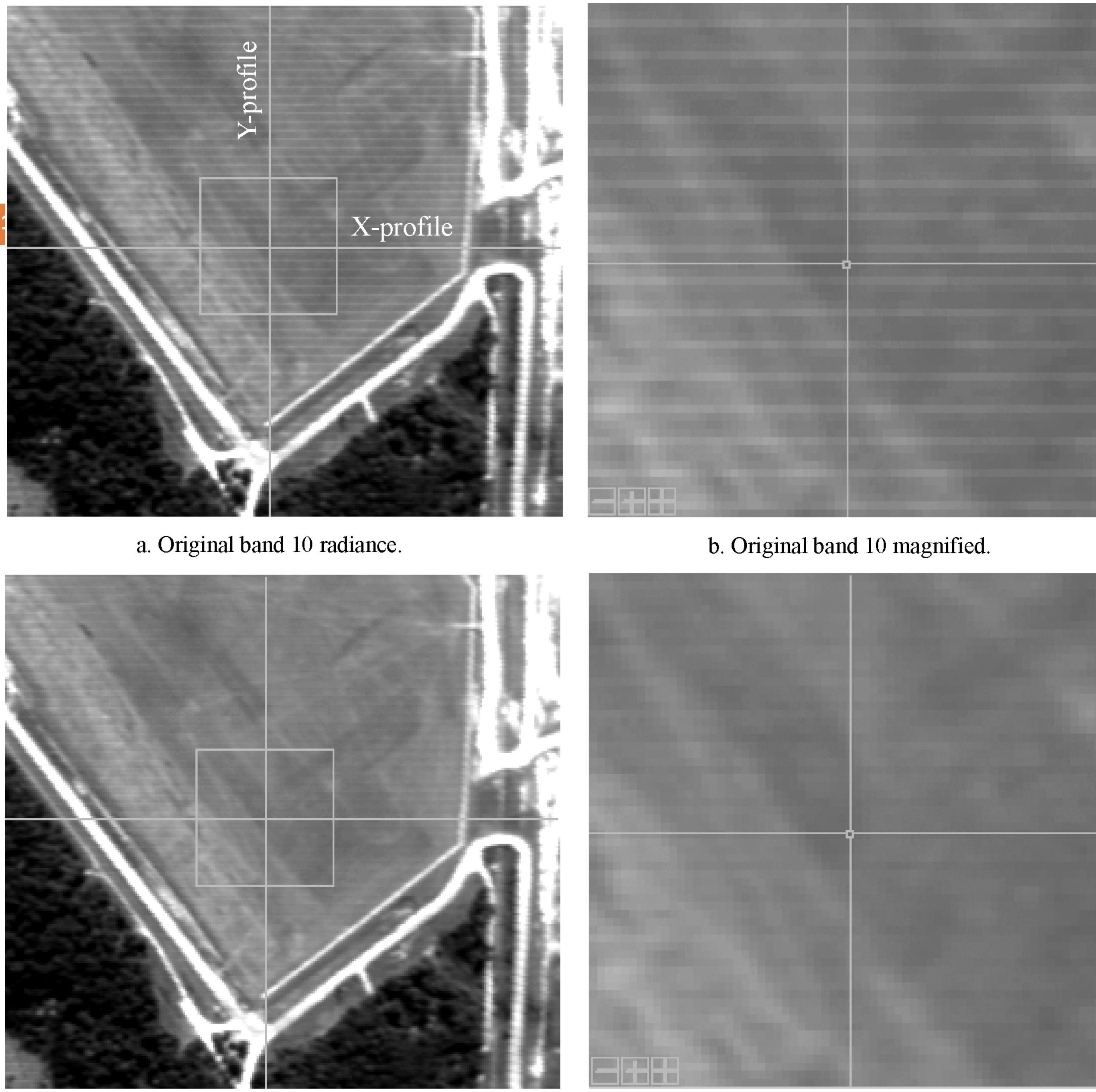
- Sometimes a detector does not fail completely, but simply goes out of radiometric adjustment. For example, a detector might record spectral measurements over a dark, deep body of water that are almost uniformly 20 brightness values greater than the other detectors for the same band. The result would be an image with systematic, noticeable lines that are brighter than adjacent lines.
- This is referred to as *n-line striping*. The maladjusted line contains valuable information, but should be corrected to have approximately the same radiometric scale as the data collected by the properly calibrated detectors associated with the same band.

N-line Striping

Striping

a. Landsat TM band 3 data of the Santee Delta.

b. Landsat TM band 3 data destriped.



a. Original band 10 radiance.

b. Original band 10 magnified.

c. Destriped band 10 radiance.

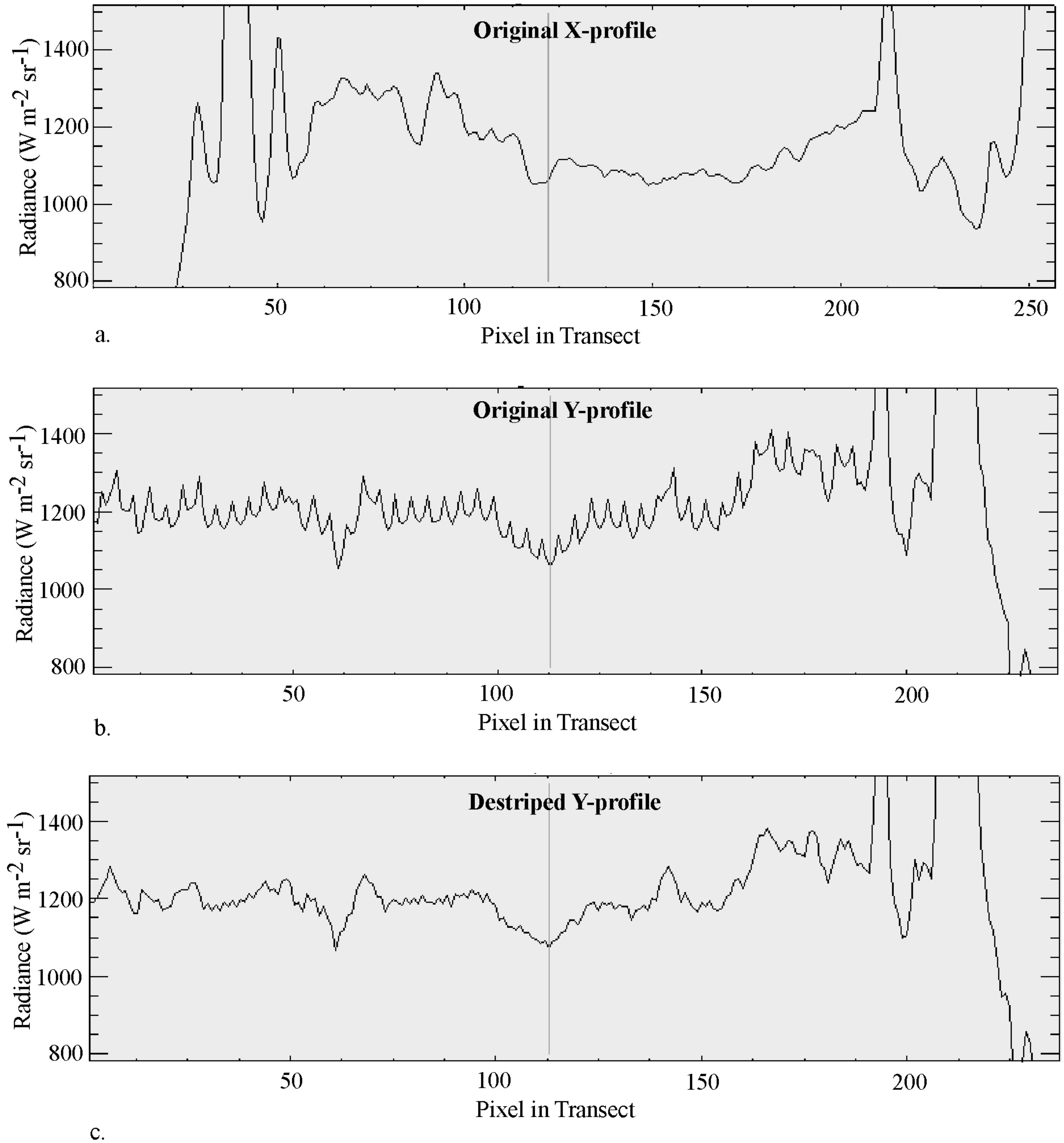
d. Destriped band 10 magnified.

a) Original band 10 radiance ($W\ m^{-2}\ sr^{-1}$) data from a GER DAIS 3715 hyperspectral dataset of the Mixed Waste Management Facility on the Savannah River Site near Aiken, SC. The subset is focused on a clay-capped hazardous waste site covered with Bahia grass and Centipede grass. The 35-band dataset was obtained at 2×2 m spatial resolution. The radiance values along the horizontal (*X*) and vertical (*Y*) profiles are summarized in the next figure.

b) Enlargement of band 10 data.

c) Band 10 data after destriping.

d) An enlargement of the destriped data



a) The radiance values along the horizontal (*X*) profile of the original band 10 radiance values in the previous figure. b) The radiance values along the vertical (*Y*) profile of the original band 10 radiance values in the previous figure. c) The radiance values along the vertical (*Y*) profile of the destriped band 10 radiance values. Note the reduction of the saw-toothed pattern in the destriped data

Geometric Correction

- It is usually necessary to *preprocess* remotely sensed data and remove geometric distortion so that individual picture elements (pixels) are in their proper planimetric (*x*, *y*) map locations. This allows remote sensing–derived information to be related to other thematic information in geographic information systems (GIS) or spatial decision support systems (SDSS).
- *Geometrically corrected imagery* can be used to extract accurate distance, polygon area, and direction (bearing) information

Geometric Correction

- Remotely sensed imagery typically exhibits *internal* and *external geometric error*. It is important to recognize the source of the internal and external error and whether it is *systematic* (predictable) or *nonsystematic* (random). Systematic geometric error is generally easier to identify and correct than random geometric error.

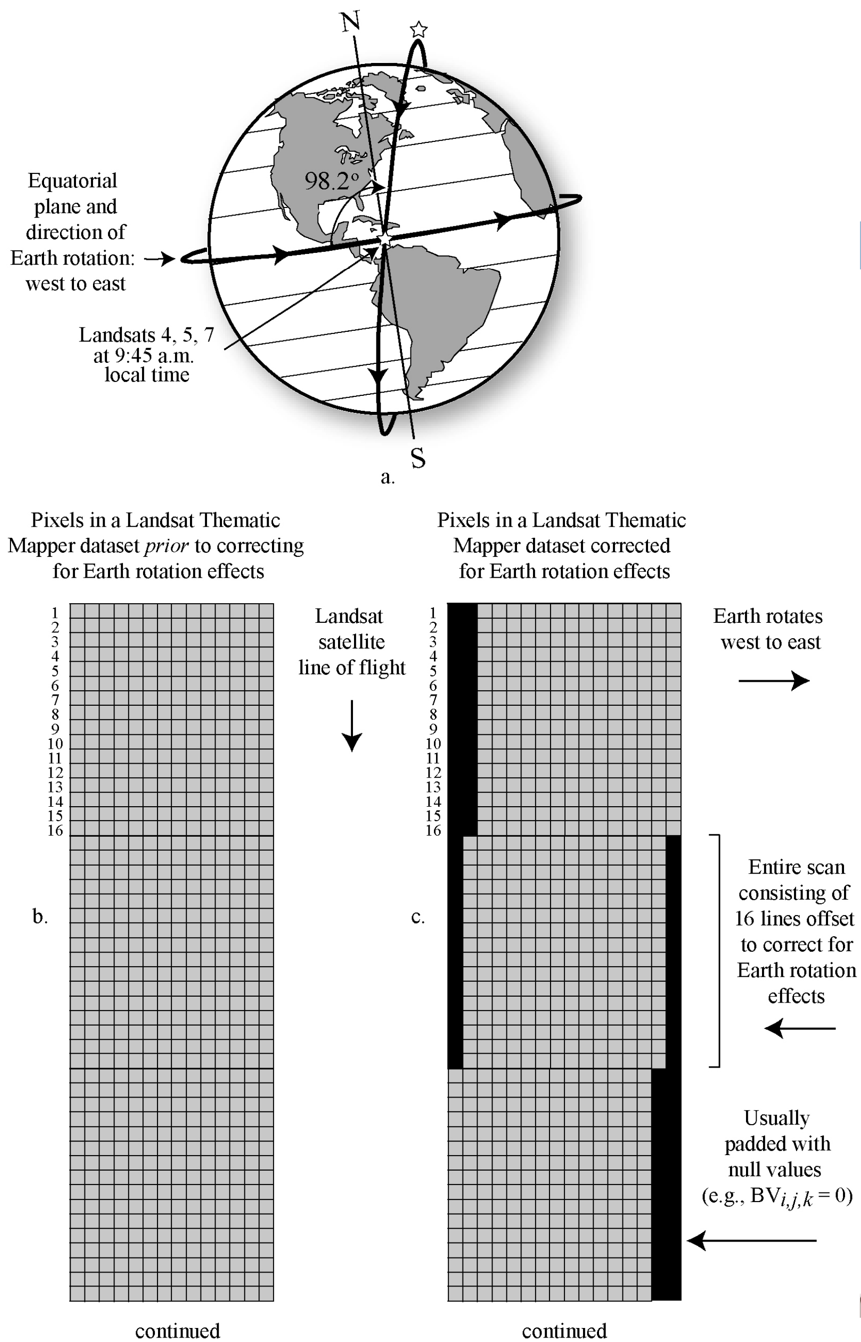
Internal Geometric Error

- ***Internal geometric errors*** are introduced by the remote sensing system itself or in combination with Earth rotation or curvature characteristics. These distortions are often *systematic* (*predictable*) and may be identified and corrected using pre-launch or in-flight platform ephemeris (i.e., information about the geometric characteristics of the sensor system and the Earth at the time of data acquisition). Geometric distortions in imagery that can sometimes be corrected through analysis of sensor characteristics and ephemeris data include:

  - skew caused by Earth rotation effects,
  - scanning system–induced variation in ground resolution cell size,
  - scanning system one-dimensional relief displacement, and
  - scanning system tangential scale distortion.

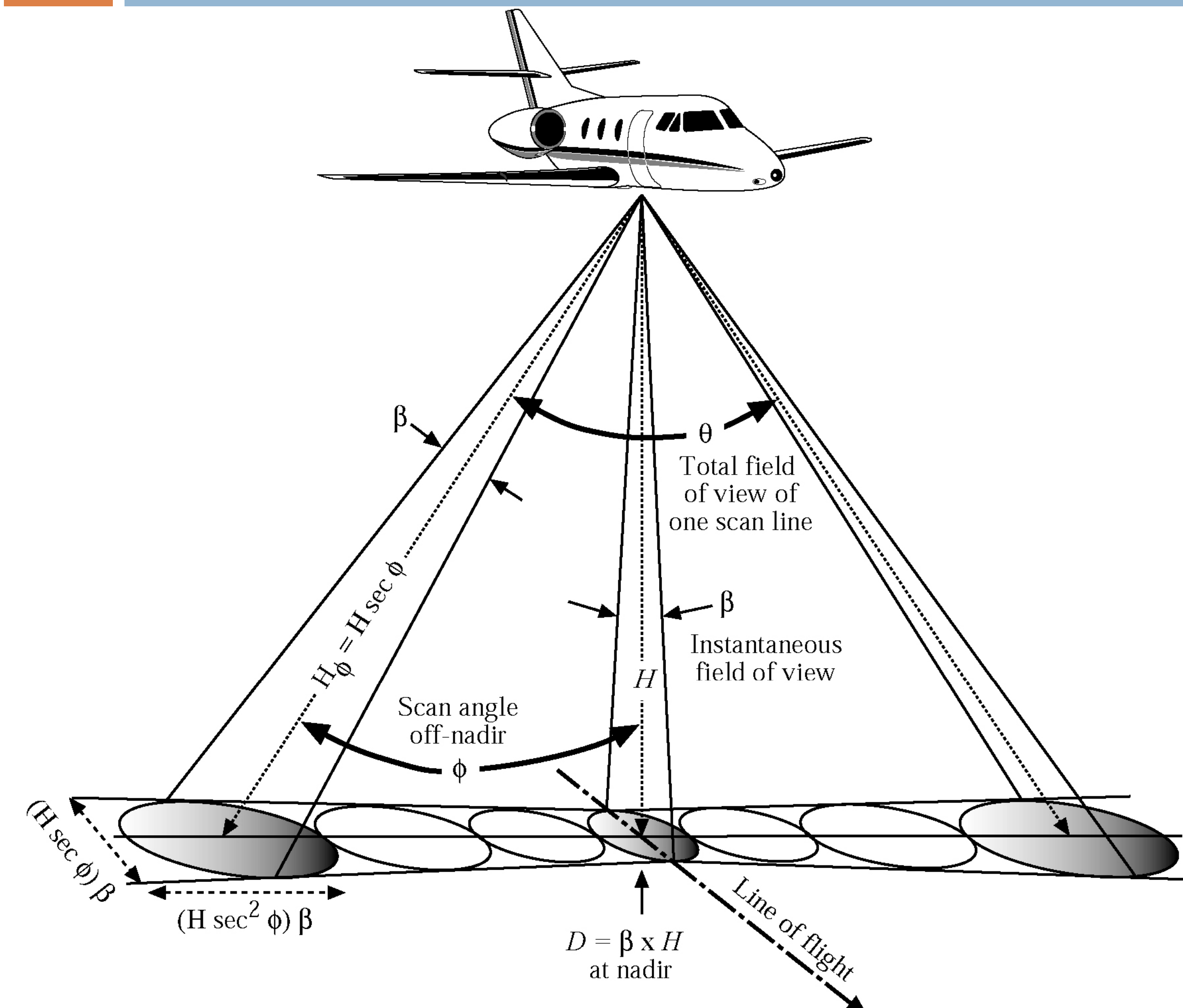
Internal Geometric Error

- Earth-observing Sun-synchronous satellites are normally in fixed orbits that collect a path (or swath) of imagery as the satellite makes its way from the north to the south in descending mode. Meanwhile, the Earth below rotates on its axis from west to east making one complete revolution every 24 hours. This interaction between the fixed orbital path of the remote sensing system and the Earth's rotation on its axis *skews* the geometry of the imagery collected.



a) Landsat satellites 4, 5, and 7 are in a Sun-synchronous orbit with an angle of inclination of 98.2°. The Earth rotates on its axis from west to east as imagery is collected.
b) Pixels in three hypothetical scans (consisting of 16 lines each) of Landsat TM data. While the matrix (raster) may look correct, it actually contains systematic geometric distortion caused by the angular velocity of the satellite in its descending orbital path in conjunction with the surface velocity of the Earth as it rotates on its axis while collecting a frame of imagery.
c) The result of adjusting (*deskewing*) the original Landsat TM data to the west to compensate for Earth rotation effects. Landsats 4, 5, and 7 use a bidirectional cross-track scanning mirror.

Scanning System-induced Variation in Ground Resolution Cell Size



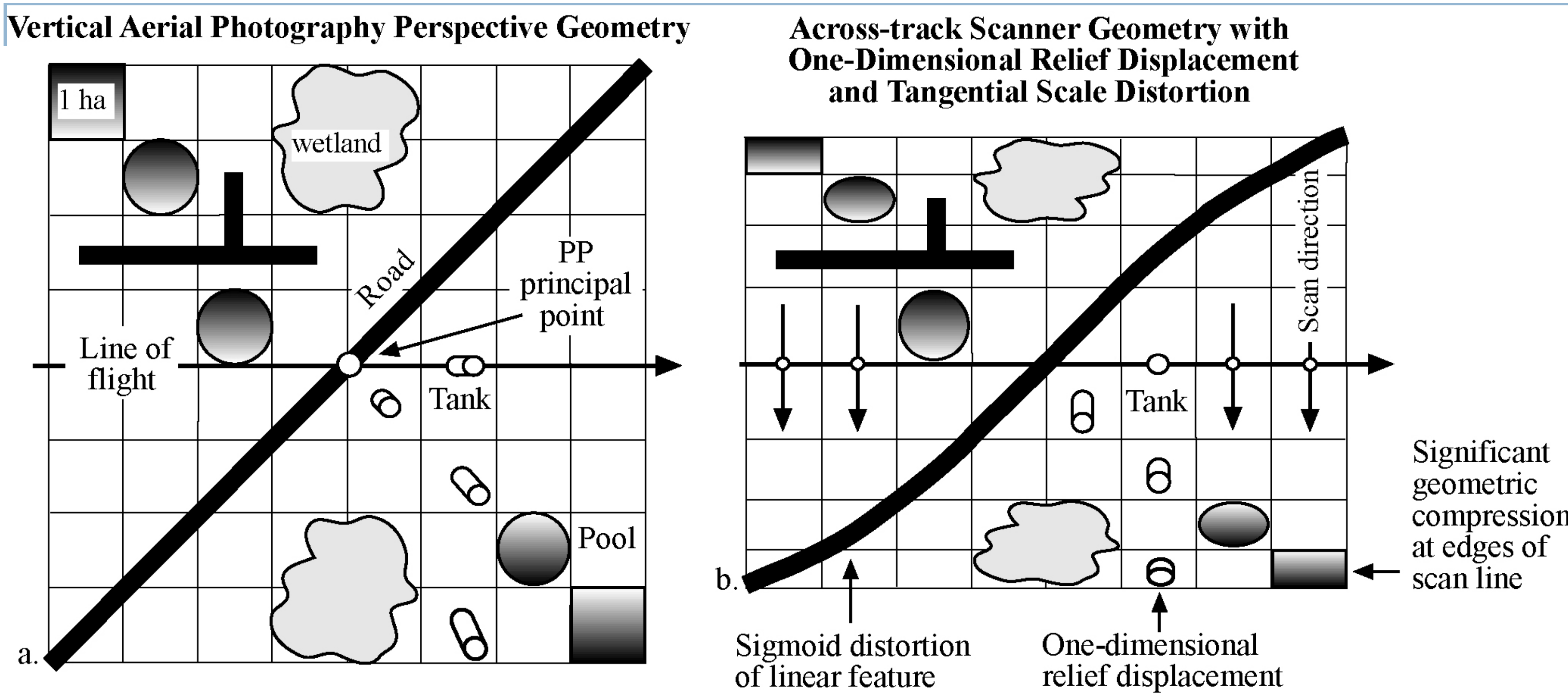
The ***ground resolution cell size*** along a single across-track scan is a function of

(a) the distance from the aircraft to the observation where *H* is the altitude of the aircraft above ground level (AGL) at nadir and *H sec* ϕ off-nadir

(b) the instantaneous-field-of-view of the sensor, **β**, measured in radians; and

(c) the scan angle off-nadir, ϕ. Pixels off-nadir have semi-major and semi-minor axes (diameters) that define the resolution cell size. The total field of view of one scan line is θ. One-dimensional relief displacement and tangential scale distortion occur in the direction perpendicular to the line of flight and parallel with a line scan.

Scanning System One-Dimensional Relief Displacement

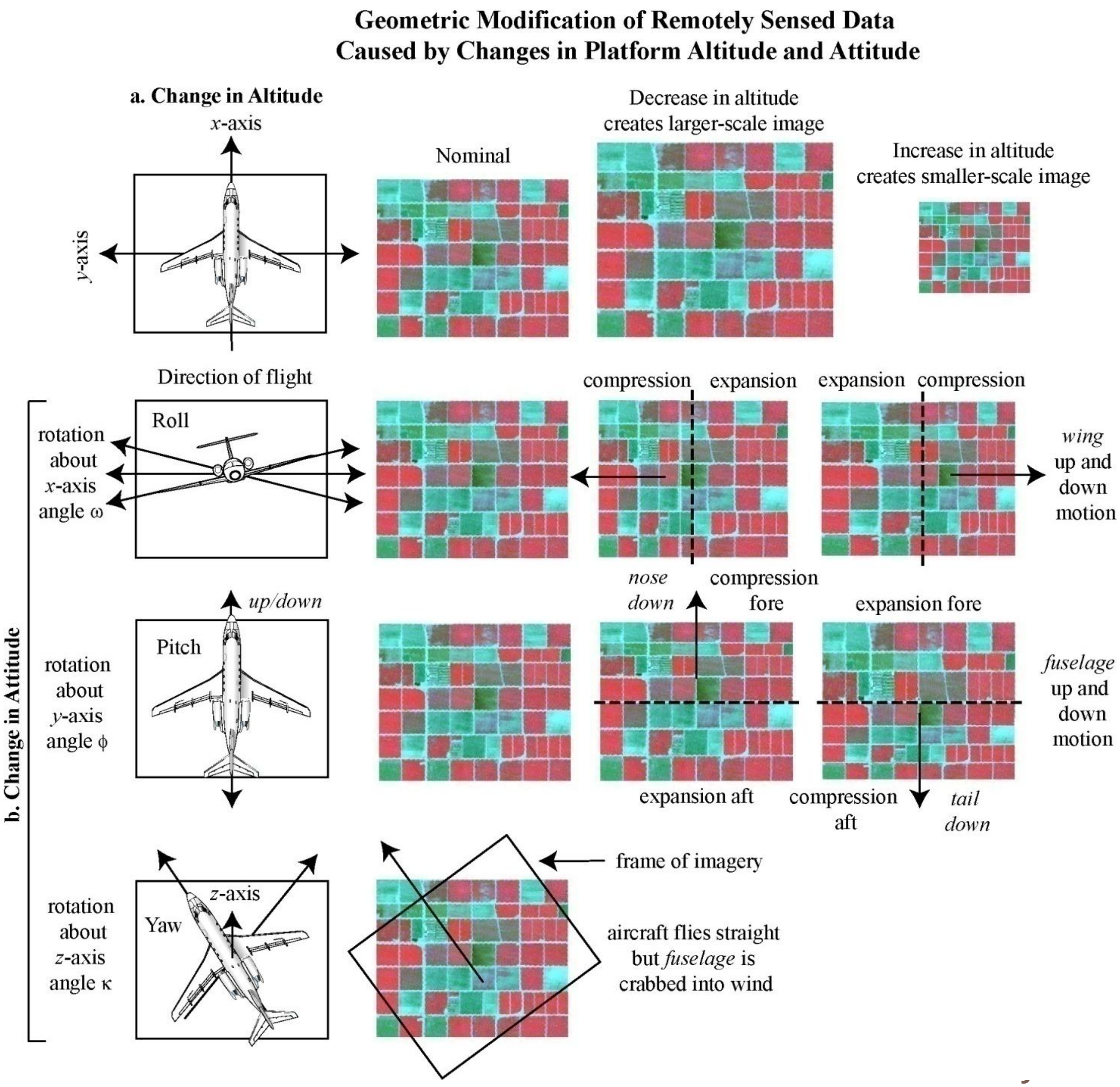


One-dimensional relief displacement is introduced in both directions away from nadir for each sweep of the across-track mirror. Linear features trending across the terrain are often recorded with *s*-shaped or *sigmoid curvature* characteristics due to tangential scale distortion and image compression.

External Geometric Error

- *External geometric errors* are usually introduced by phenomena that vary in nature through space and time. The most important external variables that can cause geometric error in remote sensor data are random movements by the aircraft (or spacecraft) at the exact time of data collection, which usually involve:

  - *altitude* changes, and/or
  - *attitude* changes (roll, pitch, and yaw).



(a)Increasing altitude results in smaller-scale imagery while decreasing altitude results in larger-scale imagery.
b) Geometric modification may also be introduced by aircraft or spacecraft changes in *attitude*, including roll, pitch, and yaw. An aircraft flies in the *x*-direction. Roll occurs when the aircraft or spacecraft fuselage maintains directional stability but the wings move up or down, i.e. they rotate about the *x*-axis angle (omega: ω). Pitch occurs when the wings are stable but the fuselage nose or tail moves up or down, i.e., they rotate about the *y*-axis angle (phi: ϕ). Yaw occurs when the wings remain parallel but the fuselage is forced by wind to be oriented some angle to the left or right of the intended line of flight, i.e., it rotates about the z-axis angle (kappa: κ). Remote sensing data often are distorted due to a combination of changes in *altitude* and *attitude* (roll, pitch, and yaw).

Ground Control Points

- Geometric distortions introduced by sensor system *attitude* (roll, pitch, and yaw) and/or *altitude* changes can be corrected using ground control points and appropriate mathematical models. A ***ground control point*** (GCP) is a location on the surface of the Earth (e.g., a road intersection) that can be identified on the imagery and located accurately on a map. The image analyst must be able to obtain two distinct sets of coordinates associated with each GCP:

  - *image coordinates* specified in i rows and j columns, and
  - *map coordinates* (e.g., x, y measured in degrees of latitude and longitude, feet in a State Plane coordinate system, or meters in a Universal Transverse Mercator projection).

- The paired coordinates (i, j and x, y) from many GCPs (e.g., 20) can be modeled to derive *geometric transformation coefficients*. These coefficients may be used to geometrically rectify the remote sensor data to a standard datum and map projection.

Types of Geometric Correction

- Commercially remote sensor data (e.g., SPOT Image, DigitalGlobe, Space Imaging) already have much of the *systematic error* removed. Unless otherwise processed, however, *unsystematic random error* remains in the image, making it non-planimetric (i.e., the pixels are not in their correct *x*, *y* planimetric map position). Two common geometric correction procedures are often used by scientists to make the digital remote sensor data of value:
  - *image-to-map rectification*, and
  - *image-to-image registration*.
- The *general rule of thumb* is to rectify remotely sensed data to a standard map projection whereby it may be used in conjunction with other spatial information in a GIS to solve problems. Therefore, most of the discussion will focus on *image-to-map rectification*.

Image to Map Rectification

- ***Image-to-map rectification*** is the process by which the geometry of an image is made planimetric. Whenever accurate area, direction, and distance measurements are required, image-to-map geometric rectification should be performed. It may not, however, remove all the distortion caused by topographic relief displacement in images. The *image-to-map rectification* process normally involves selecting GCP image pixel coordinates (row and column) with their map coordinate counterparts (e.g., meters northing and easting in a Universal Transverse Mercator map projection).

Image to Image Registration

- *Image-to-image registration* is the translation and rotation alignment process by which two images of like geometry and of the same geographic area are positioned coincident with respect to one another so that corresponding elements of the same ground area appear in the same place on the registered images. This type of geometric correction is used when it is *not* necessary to have each pixel assigned a unique *x*, *y* coordinate in a map projection. For example, we might want to make a cursory examination of two images obtained on different dates to see if any change has taken place.

Image to Map Geometric Rectification Logic

- Two basic operations must be performed to geometrically rectify a remotely sensed image to a map coordinate system:

  - *Spatial interpolation*, and
  - *Intensity interpolation*.

Spatial Interpolation

- The geometric relationship between the input pixel coordinates (column and row; referred to as x', y') and the associated map coordinates of this same point (X, Y) must be identified. A number of GCP pairs are used to establish the nature of the geometric coordinate transformation that must be applied to rectify or fill every pixel in the output image (x, y) with a value from a pixel in the unrectified input image (x', y'). This process is called ***spatial interpolation***.

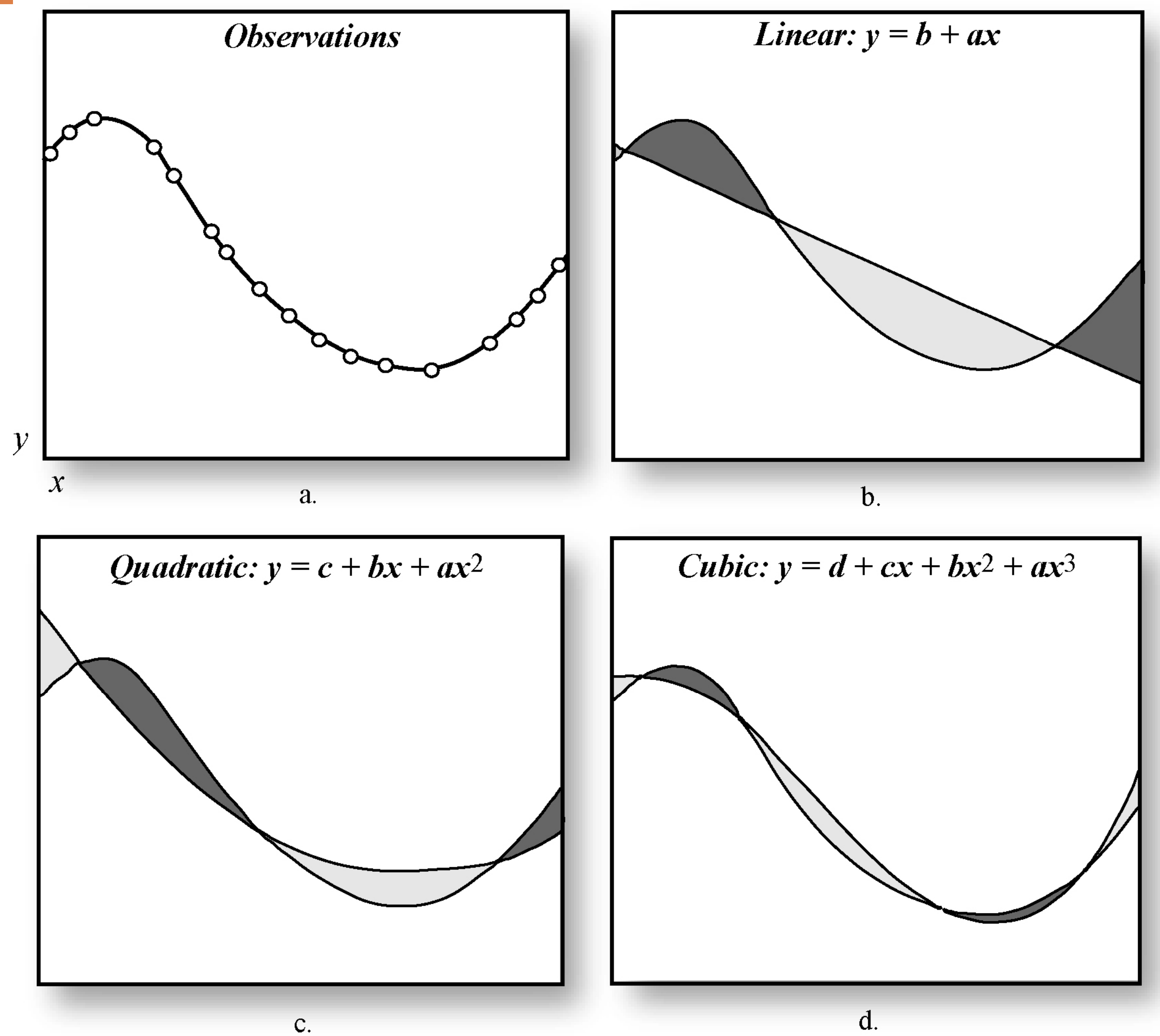
Intensity Interpolation

- Pixel brightness values must be determined. Unfortunately, there is no direct one-to-one relationship between the movement of input pixel values to output pixel locations. It will be shown that a pixel in the rectified output image often requires a value from the input pixel grid that does not fall neatly on a row-and-column coordinate. When this occurs, there must be some mechanism for determining the brightness value (*BV*) to be assigned to the output rectified pixel. This process is called ***intensity interpolation***.

Spatial Interpolation Using Coordinate Transformations

- ***Image-to-map rectification*** requires that polynomial equations be fit to the GCP data using least-squares criteria to model the corrections directly in the image domain without explicitly identifying the source of the distortion. Depending on the distortion in the imagery, the number of GCPs used, and the degree of topographic relief displacement in the area, *higher-order polynomial equations* may be required to geometrically correct the data. The *order* of the rectification is simply the highest exponent used in the polynomial.

Spatial Interpolation



Concept of how different-order transformations fit a hypothetical surface illustrated in cross-section:

(a) Original observations.

b) First-order linear transformation fits a plane to the data.

c) Second-order quadratic fit.

d) Third-order cubic fit.

Spatial Interpolation Using Coordinate Transformations

- Generally, for moderate distortions in a relatively small area of an image (e.g., a quarter of a Landsat TM scene), a *first-order, six-parameter, affine (linear) transformation* is sufficient to rectify the imagery to a geographic frame of reference.

- This type of transformation can model six kinds of distortion in the remote sensor data, including:

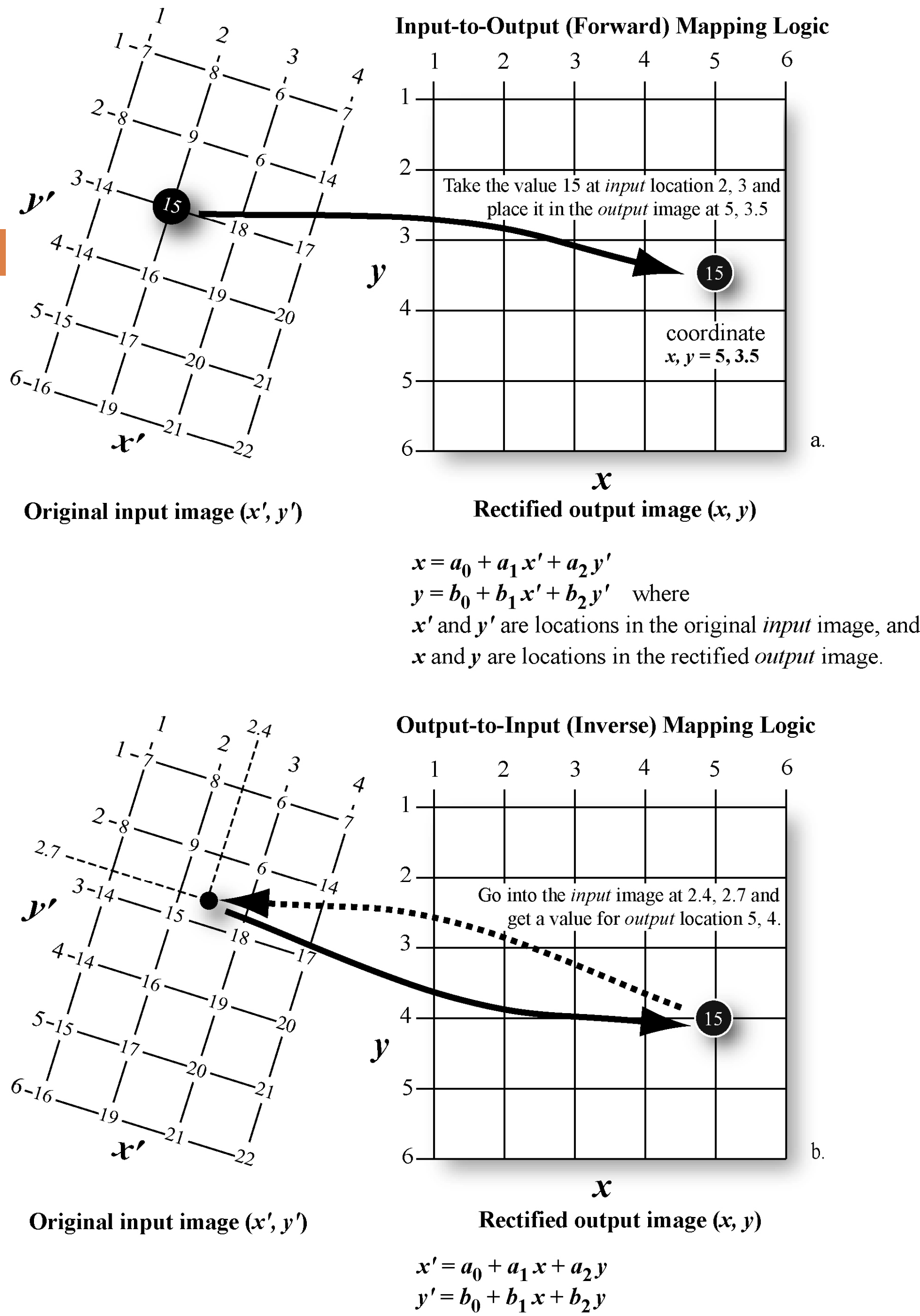
  - *translation* in *x* and *y*,
  - *scale* changes in *x* and *y*,
  - *skew*, and
  - *rotation*.

Spatial Interpolation Using Coordinate Transformations

- When all six operations are combined into a single expression it becomes:
- where x and y are positions in the *output*-rectified image or map, and x' and y' represent corresponding positions in the original *input* image. These two equations can be used to perform what is commonly referred to as *input-to-output*, or ***forward**-mapping*. The equations function according to the logic shown in the next figure. In this example, each pixel in the *input* grid (e.g., value 15 at $x', y' = 2, 3$) is sent to an x,y location in the output image according to the six coefficients shown.

$$x = a_0 + a_1 x' + a_2 y'$$

$$y = b_0 + b_1 x' + b_2 y'$$

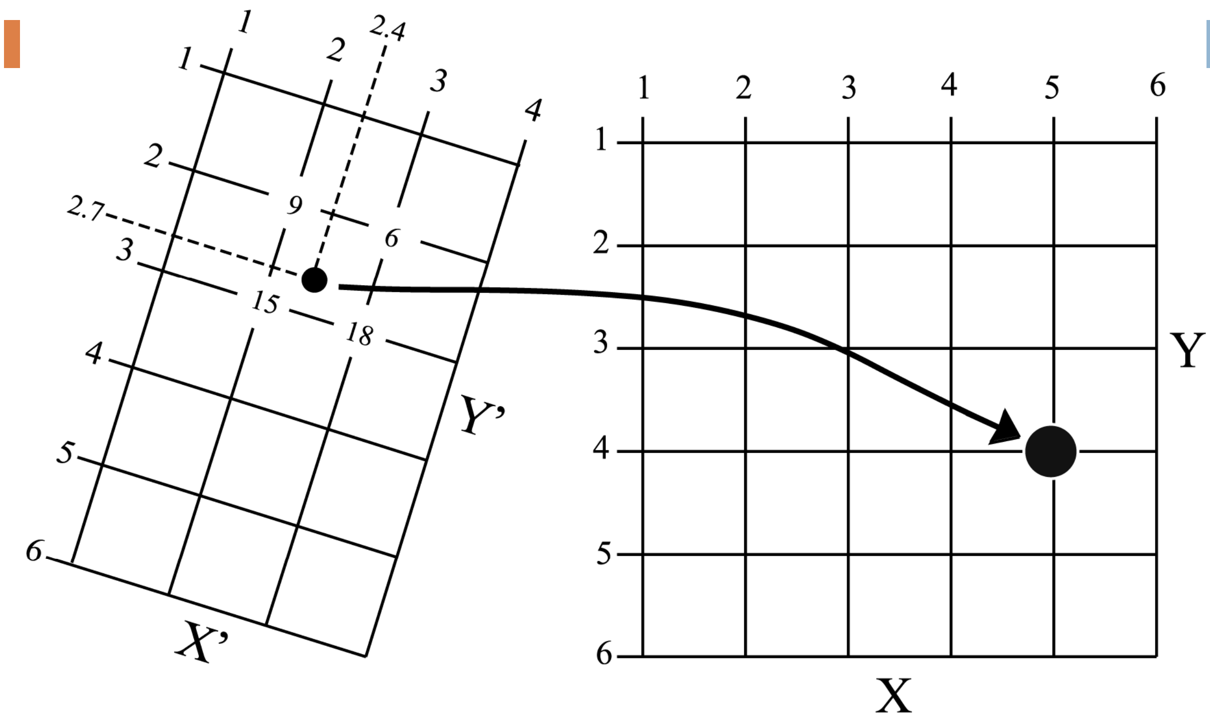


a) The logic of filling a rectified output matrix with values from an unrectified input image matrix using *input-to-output* (***forward***) mapping logic.

b) The logic of filling a rectified output matrix with values from an unrectified input image matrix using *output-to-input* (***inverse***) mapping logic and nearest-neighbor resampling.

Output-to-input inverse mapping logic is the preferred methodology because it results in a rectified output matrix with values at every pixel location.

Spatial Interpolation Logic



Original input image

(a)

Rectified output image

(b)

$$x' = a_0 + a_1 x + a_2 y$$

$$y' = b_0 + b_1 x + b_2 y$$

The goal is to fill a matrix that is in a standard map projection with the appropriate values from a non-planimetric image.

$$x' = -382.2366 + 0.034187x + (-0.005481)y$$

$$y' = 130162 + (-0.005576)x + (-0.0349150)y$$

Spatial Interpolation Logic

A way to measure the accuracy of a geometric rectification algorithm (actually, its coefficients) is to compute the ***Root Mean Squared Error*** (*RMS$_{error}$*) for each ground control point using the equation:

$$RMS_{error} = \sqrt{\left(x' - x_{orig}\right)^2 + \left(y' - y_{orig}\right)^2}$$

where:

x$_{orig}$ and ***y$_{orig}$*** are the ***original*** row and column coordinates of the GCP in the image and ***x'*** and ***y'*** are the ***computed or estimated*** coordinates in the original image when we utilize the six coefficients. Basically, the closer these paired values are to one another, the more accurate the algorithm (and its coefficients). The square root of the squared deviations represents a measure of the accuracy of each GCP. By computing ***RMS$_{error}$*** for all GCPs, it is possible to (1) see which GCPs contribute the greatest error, and 2) sum all the ***RMS$_{error}$***.

Intensity Interpolation

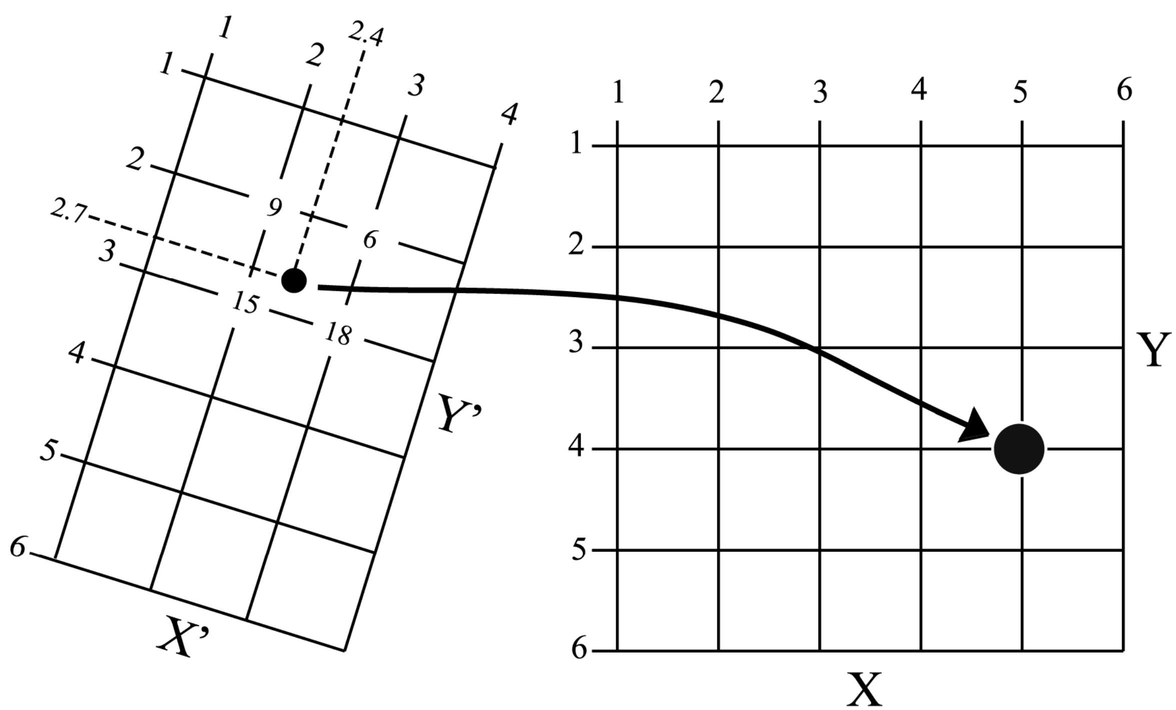
- ***Intensity interpolation*** involves the extraction of a brightness value from an x', y' location in the original (distorted) input image and its relocation to the appropriate x, y coordinate location in the rectified output image. This *pixel-filling logic* is used to produce the output image line by line, column by column. Most of the time the x' and y' coordinates to be sampled in the input image are floating point numbers (i.e., they are not integers). For example, in the Figure on the next slide we see that pixel 5, 4 (x, y) in the output image is to be filled with the value from coordinates 2.4, 2.7 (x', y') in the original input image. When this occurs, there are several methods of brightness value (*BV*) *intensity interpolation* that can be applied, including:

  - *nearest neighbor,*
  - *bilinear interpolation*, and
  - *cubic convolution.*

- The practice is commonly referred to as ***resampling***.

Nearest-Neighbor Resampling

The brightness value closest to the predicted ***x', y'*** coordinate is assigned to the output ***x, y*** coordinate.



Original input image

(a)

Rectified output image

(b)

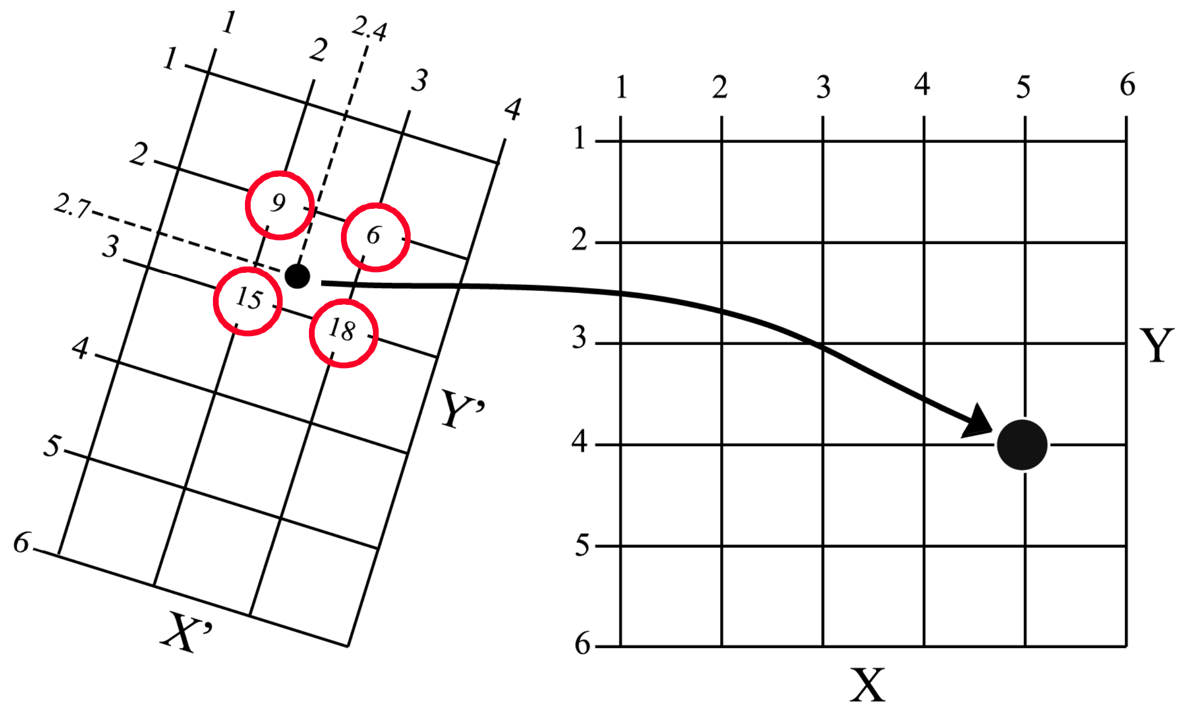
Bilinear Interpolation

Assigns output pixel values by interpolating brightness values in two orthogonal direction in the input image. It basically fits a plane to the ***4*** pixel values nearest to the desired position (***x', y'***) and then computes a new brightness value based on the weighted distances to these points. For example, the distances from the requested (***x', y'***) position at 2.4, 2.7 in the input image to the closest four input pixel coordinates (2,2; 3,2; 2,3;3,3) are computed . Also, the closer a pixel is to the desired x',y' location, the more weight it will have in the final computation of the average.

$$BV_{wt} = \frac{\sum_{k=1}^{4} \frac{Z_k}{D_k^2}}{\sum_{k=1}^{4} \frac{1}{D_k^2}}$$

where Z_k are the surrounding four data point values, and $D^2{}_k$ are the distances squared from the point in question (***x', y'***) to the these data points.

Bilinear Interpolation



Original input image

(a)

Rectified output image

(b)

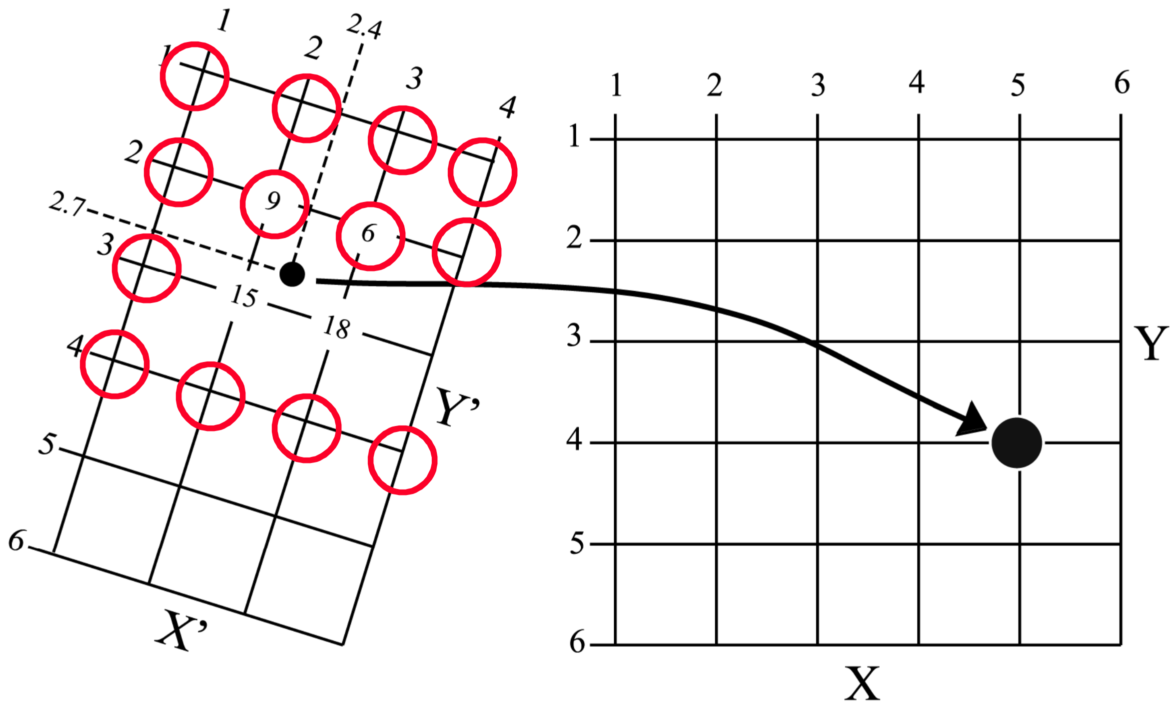
Cubic Convolution

Assigns values to output pixels in much the same manner as bilinear interpolation, except that the weighted values of ***16*** pixels surrounding the location of the desired ***x', y'*** pixel are used to determine the value of the output pixel.

$$BV_{wt} = \frac{\sum_{k=1}^{16} \frac{Z_k}{D_k^2}}{\sum_{k=1}^{16} \frac{1}{D_k^2}}$$

where $\boldsymbol{Z_k}$ are the surrounding four data point values, and $\boldsymbol{D^2_k}$ are the distances squared from the point in question (x', y') to the these data points.

Cubic Convolution



Original input image

(a)

Rectified output image

(b)