A One-Step Algorithm for Correction and Calibration of AVHRR Level 1b Data

Liping Di and Donald C. Rundquist

Abstract

NOAA-AVHRR level 1b data have been widely used for environmental research at regional and global scales. There are, however, problems in preprocessing level 1b tape data in small systems, including a general lack of AVHRR-specific software possessing suitable levels of sophistication, efficiency, and geographic coverage. This paper describes a one-step preprocessing algorithm which combines simple tape reading with geometric correction, radiometric correction, and calibration using the auxiliary parameters appended in the level 1b tape as primary input. Our algorithm is fast, memory-efficient, and PC compatible.

Introduction

In the past several years, one of the most important developments in remote sensing has been the widespread use of meteorological satellite data for non-meteorological applications. The National Oceanic and Atmospheric Administration (NOAA) series of polar-orbiting meteorological satellites have received considerable attention because of the onboard sensor; the Advanced Very High Resolution Radiometer (AVHRR). AVHRR data have been used extensively to study regional, continental, and global phenomena. The most popular application of these data has been to monitor and evaluate vegetation over the land surface.

AVHRR data have three different formats: GAC, HRPT, and LAC. GAC, or Global Area Coverage, is intended to allow worldwide coverage with a manageable amount of data. The spatial resolution for GAC is 16 km. Local Area Coverage (LAC) and High Resolution Picture Transmission (HRPT) actually are the same kind of data but differ in the manner of transmission from satellite to ground station. HRPT data are sent to Earth continuously in real time, while LAC data are selectively recorded on-board for subsequent playback. Both HRPT and LAC have a spatial resolution of 1.1 km at nadir and a spectral resolution of 10 bits (NOAA, 1990).

Because of the higher spatial and spectral resolutions than that of GAC, both HRPT and LAC data are widely used in regional or continental environmental research. However, LAC data are unique in that researchers can study geographic areas where no ground station is available. LAC data are usually stored on computer compatible tapes (CCTs) as level 1b data for distribution (NOAA, 1990). The level 1b data contain raw AVHRR spectral data as well as calibration coefficients, solar-zenith angles, Earth location, and other auxiliary data. However, the spectral data contain both geometric and radiometric errors which must be removed in order to quantitatively analyze AVHRR data and overlay other data sets. Therefore, preprocessing is a prerequisite to using AVHRR level 1b data in research.

The Problem

There are many commonly used software packages to allow users to easily read AVHRR level 1b data from a computer compatible tape (CCT) into a disk file for further analyses; for example, LDAVHRR in ERDAS (ERDAS, 1990), RFLAC in ELAS (NASA, 1990), and LACIN in LAS (USGS, 1990). However, these packages do not have any geometric and radiometric correction functions designed specifically for AVHRR data, so all auxiliary information, such as solar zenith angles, geometric locations, and calibration coefficients, are lost (i.e., these packages do not store the auxiliary data).

After the tape data have been read into a disk file by means of the software noted above, conventional geometric-correction algorithms, such as rubber sheeting and polynomial fitting, may be applied to raw AVHRR data to achieve georeferencing. One problem is that those algorithms typically involve the selection of ground control points. Because of the coarse spatial resolution of LAC, 1100 by 1100 metres at the nadir, the process of locating ground-control points (GCPs) in an AVHRR image is difficult (Peters, 1989). A second problem is that there are no commonly available software packages which provide suitable radiometric corrections, such as solar zenith angle, atmospheric attenuation, and data calibration for converting the raw digital numbers (DNs) to ground reflectance, albedo, surface temperature, and so on.

Despite the general lack of software for AVHRR preprocessing, CCTs containing AVHRR LAC data do provide the necessary parameters for both geometric and radiometric adjustments (NOAA, 1990). Instead of performing corrections after the tape has been read as in conventional methods, it is possible to use those parameters to do geometric and radiometric corrections while reading an AVHRR tape. The development of a one-step algorithm combining geometric correction, radiometric correction, and calibration with tape reading seems to be a worthwhile objective. This article introduces such an algorithm.

Geometric Correction of AVHRR Data

Remotely sensed data usually contain both systematic and nonsystematic geometric errors (Jensen, 1986). The purposes

Liping Di is with The Cooperative Institute for Research in Environmental Sciences (CIRES), University of Colorado at Boulder, Campus Box 449, Boulder CO 80309-0449
D.C. Rundquist is with The Center for Advanced Land Management Information Technologies (CALMIT), University of Nebraska-Lincoln, Lincoln, NE 68588-0517

Photogrammetric Engineering & Remote Sensing,
Vol. 60, No. 2, February 1994, pp. 165–171.

0099-1112/94/6002–165$03.00/0

of geometric correction are to remove these errors and to relate the digital remote-sensing data to a map projection.

The traditional geometric correction of remotely sensed data almost always involves relating the pixel coordinates (row and column) of ground control points (GCPs) with their corresponding map coordinates (e.g., the latitude/longitude position). A GCP is a point on the surface of the Earth that can be identified on both an image (in rows and columns) and a map (in degrees of latitude and longitude, feet, or metres). The geometric relationship between the input pixel location (row and column) and the associated map coordinate (x,y) can be determined by a group of GCPs. The typical projection equations relating the map coordinates and image coordinates are polynomials: that is,

$$\begin{aligned} x' &= a_0 + a_1x + a_2y + a_3x^2 + a_4xy + a_5y^2 + \ldots \\ y' &= b_0 + b_1y + b_2y + b_3x^2 + b_4xy + b_5y^2 + \ldots \end{aligned} \quad (1)$$

where x and y are the positions in the rectified image or map and x' and y' represent the corresponding positions in the original input image. The coefficients in Equation 1 are determined by regression analysis of GCPs. The order of the polynomial in Equation 1 is decided by both the magnitude of the distortion in the raw image and the number of available GCPs. A typical rectification of a satellite image involves third-order polynomials and 20 to 30 GCPs. With Equation 1, a pixel position in the rectified image can be projected into the distorted image coordinate and a resampling algorithm is used to retrieve the spectral value from the distorted image for the position.

Geometric correction usually is done row-by-row or block-by-block of the output (rectified) image. Because of the geometric distortion, a row in the output image may cross several rows in the distorted image. Therefore, any part of the distorted image must be directly accessible (e.g., the distorted image must be resident in a disk file or computer system memory). However, computer tapes are accessed sequentially. Therefore, it is impossible to use traditional methods to do geometric correction while reading an AVHRR tape.

AVHRR level 1b data contain Earth location information, with a fixed number of ground reference points appended to each scan line (NOAA, 1990). There are 2048 pixels in an LAC/HRPT scan line. The Earth location data (latitude and longitude) are sampled every 40 points starting at pixel 25 (25, 65, 105, ..., 1945, 1985, and 2025), so there are 51 possible ground reference values for each scan line. Latitude and longitude values are each stored in two-byte fields in 128th of a degree (East positive), which is less than the size of one pixel (about 0.0133 degree under nadir). Based on those Earth locations, we developed a method to geometrically correct AVHRR data while reading the tape.

Spatial Interpolation

Suppose that we want to obtain a rectangular area (window) bounded by minimum longitude X_{min}, maximum longitude X_{max}, minimum latitude Y_{min}, and maximum latitude Y_{max}. The image size for the window, in terms of rows and columns, can be expressed as

$$\begin{aligned} nrow &= (X_{max} - X_{min})/P_x + 0.5 \\ ncolumn &= (Y_{max} - Y_{min})/P_y + 0.5 \end{aligned} \quad (2)$$

where P_x is the pixel size in the X direction and P_y is the pixel size in the Y direction.

Suppose a scan line is read from the tape, allowing us to obtain the 51 Earth locations sampled at every 40 pixels. For the jth pixel of the scan line, which is located between Earth location i and $i+1$, a linear interpolation equation can be used to determine the geographic location of the pixel: that is,

$$\begin{aligned} X_j &= (X_{i+1} - X_i)(j - \text{column}(i))/40 + X_i \\ Y_j &= (Y_{i+1} - Y_i)(j - \text{column}(i))/40 + Y_i \\ column(i) &= (i - 1) * 40 + 25 \qquad i = 1,2,\ldots,50 \\ j &= \text{column}(i),\text{column}(i) + 1,\ldots,(i + 1) \end{aligned} \quad (3)$$

where

X_j = the interpolated Earth location (longitude) for the jth pixel of the current scan line,
Y_j = the interpolated Earth location (latitude) for the jth pixel of the current scan line,
X_i = the ith appended Earth location (longitude), and
Y_i = the ith appended Earth location (latitude).

Equation 3 is applicable from pixel 25 (column(1)) to pixel 2025 (column(51)) of the current scan line. For pixels 1 to 24, another interpolation equation is used:

$$\begin{aligned} X_j &= (X_2 - X_1)(j - 25)/40 + X_1 \\ Y_j &= (Y_2 - Y_1)(j - 25)/40 + Y_1 \qquad j = 1, 2, \ldots, 24 \end{aligned} \quad (4)$$

Pixels 2026 to 2048 are interpolated by equation

$$\begin{aligned} X_j &= (X_{51} - X_{50})(j - 1985)/40 + X_{50} \\ Y_j &= (Y_{51} - Y_{50})(j - 1985)/40 + Y_{50} \\ j &= 2026, 2027, \ldots, 2048 \end{aligned} \quad (5)$$

Both Equations 4 and 5 are extensions of Equation 3. The geographic location of the jth pixel of the current scan line is then converted to the row and column numbers of the rectified output image based on window coordinate and pixel size: that is,

$$\begin{aligned} L_j &= (Y_{max} - Y_j)/P_y + 0.5 \\ C_j &= (X_j - X_{min})/P_x + 0.5 \end{aligned} \quad (6)$$

where L_j is the row number for pixel j in the rectified output image and C_j is the column number. Note that a row increment in the output image is the inverse of the Y (latitude) increment. The first row in the output image has the maximum latitude possible in the image. The row and column numbers are then used to compare with the specified image area (window) to decide whether or not pixel j is in the selected output area:

$$\begin{cases} j \in \text{window} & 1 \le L_j \le \text{nrow} \\ & 1 \le C_j \le \text{ncolumn} \\ j \notin \text{window} & \text{otherwise} \end{cases} \quad (7)$$

Once pixel j of the current scan line is in the output window area, the (L_j, C_j) location of the rectified output image will be assigned the intensity value of the input pixel j.

In order to reduce the calculation time, it is wise to check the 51 points of Earth locations first. If none are in the output window, discard the current scan line and read next scan line.

The procedure for spatial interpolation of AVHRR data can been described as follows:

(1) Read in one scan line from an AVHRR tape. If there are no scan lines left in the tape, exit the loop.
(2) Check whether or not the scan line falls within the selected window. We might only check the 51 Earth locations of the scan line. If none falls within the window area, go to step (1).

(3) Use the equations described above to compute the output location for each pixel in the current scan line and place pixels in the windowed area in their corresponding positions in the output image. Go to step (1).

A problem with the above procedure is that the program reads tape until the end even if all scan lines in the designated window have already been processed. Although this problem cannot cause output data error, it certainly takes a lot of unnecessary time to wait for the finish of the processing if the data for the area of interest are located at the beginning of the tape. Adding a logical variable to the above procedure will remedy this problem. In the beginning of the procedure, the logical variable is assigned a value of .FALSE., which means that the entire window is still not yet segmented. Once the first scan line within the windowed area is identified, the variable is switched to .TRUE.. If the variable is .TRUE., and the current scan line is out of the windowed area, it must be the first line beyond the selected area. Because the satellite continues to fly forward, all successive scan lines will be beyond the current scan line and none of them will fall into the windowed area. Therefore, the program should stop reading tape. All scan lines beyond the current one will not be read by the program, and time can be saved.

Because of the geometric distortion in the original AVHRR data set, there is no one-to-one pixel correspondence between distorted input and rectified output images. It is possible for several pixels in the input image to be projected to one location in the output image, or for neighboring input pixels to no longer be neighbors in the output image. The output image produced by the above procedure may contain many blank pixels (i.e. those pixels have no directly corresponding pixels in the raw AVHRR image). The blank pixels must be assigned values by means of intensity interpolation instead of resampling as used in the traditional geometric correction methods.

Intensity Interpolation

The process of intensity interpolation fills all blank pixels in the output image by extracting a brightness value from a location in the original (distorted) input image and relocating it to the appropriate coordinate location in the rectified output image. The process is similar to resampling in traditional geometric correction. However, traditional resampling methods cannot be used in our case for two reasons. First, it is impossible to exactly know the location in the original (distorted) input image for those blank pixels of the output image. With the usual method, the geometric relationship between the distorted and rectified image is represented by a single polynomial equation; thus, it is possible to get an exact distorted position from the rectified position through the equation. In our case, pixels in the distorted image are relocated to locations in the rectified image by different linear interpolation equations. There are no linear interpolation equations relating locations of blank pixels in the rectified output image back to the locations in the distorted image. Second, we do not keep the distorted image during the processing, even if we could get distorted locations for these blank pixels. Unlike disk files which might be accessed randomly, files in a tape only can be read sequentially; for AVHRR, scan line by scan line. However, we do keep pixel intensities of the distorted image in their corresponding position of the rectified image before the intensity interpolation. For these blank pixels, intensity values of their neighboring pixels already exist in the rectified output image. Instead of extracting an intensity value from the raw (distorted) image, a new method, "linear filling," was developed to extract intensity values for these blank pixels.

The basic assumption for linear filling is that intensity values for blank pixels should be similar to their neighbors. The method fills blank pixels row by row. Suppose that the kth pixel of the current row in the rectified output image is a blank pixel and the pixel's left non-blank pixel neighbor is the ith pixel and the pixel's right non-blank pixel neighbor is the jth pixel. The intensity value for the blank pixel located at the kth position of the current row can be linearly interpolated by its two non-blank neighbors i and j: that is,

$$g_k = (g_j - g_i)(k - i)/(j - i) + g_i + 0.5 \qquad i < k < j \qquad (8)$$

where

g_k = the grey level for the kth pixel of the current row,
g_i = the grey level of ith pixel which is the non-blank left neighbor of blank pixel k, and
g_j = the grey level of jth pixel which is the non-blank right neighbor of blank pixel k.

After linear filling, all pixels in the output image have values. At this point in processing, the output image in a specific window can be extracted and georeferenced theoretically. However, the above algorithm requires a tremendous amount of computer memory for extracting a regional-scale area from AVHRR tape. In order to handle the problem, a memory handling algorithm was developed.

Multi-Block Virtual Memory Algorithm

The procedure to geometrically correct AVHRR data while reading an AVHRR tape has been presented above. However, a requirement implied in the procedure is that the output image must stay in the system memory (RAM) during the rectification. Because the orbit of a NOAA satellite is not orthogonal to the equator, scan lines are not parallel to lines of latitude on the Earth. Under the nadir point, an AVHRR scan line meets a latitude line with an angle which is a function of both orbit inclination angle and latitude. The angle is about 8.9 degrees at the equator and increases towards the poles. Because of both the Earth's curvature and the large scan angle of AVHRR, the angle between the scan line and the latitude line also increases toward both the beginning and ending points of a scan line.

Because rows of the rectified output image are parallel to lines of latitude and columns to lines of longitude, scan lines appear as slanted, curved lines on the rectified output image. In other words, a scan line might cross many rows in the output image. The number of rows in the output image crossed by a scan line depends on the width of the window, its Earth location, and which part of the scan line falls within the window. Because tape is read line by line, an entire output image must be kept in the system memory in order to relocate pixels to their corresponding positions in the output image. For a small window, it is possible to keep the whole output image in system memory. But for extracting an area of regional scale, the memory requirement may be far above that which a computer can normally provide. For example, a user might specify an output window of 27.3 by 27.3 degrees in size. According to Equation 2, the size of the output image will be 2048 rows by 2048 columns. A researcher may want all five channels of AVHRR data; thus, five images of 2048 by 2048 pixels each must be stored in the system memory. Because every AVHRR pixel needs two bytes to store, the total memory requirement is 40 megbytes (2048 * 2048 * 5 * 2).

To overcome the memory problem, a multi-block virtual memory algorithm has been developed. The main idea of the algorithm is to use disk as main storage and only keep a small portion of the window in the system memory. With very little increase in CPU time and no increase in I/O time, as compared with traditional algorithms, our procedure can deal with a window of virtually any size.

Suppose an output image is *n*row**n*column in size, and the image is divided into subimages (blocks) of 16 by 16 pixels. Thus, the total number of blocks for this output image is *nlb*ncb*nch*;

$$\begin{aligned} nlb &= [n\text{row}/16] \\ ncb &= [n\text{column}/16] \end{aligned} \tag{9}$$

where

nlb = the number of block rows,
ncb = the number of block columns, and
nch = the number of the output channels.

Each block has a three-dimensional coordinate:

$$(i,j,k), \quad i = 1,2,\ldots,nlb \quad j = 1,2,\ldots,ncb \quad k = 1,2,\ldots,nch \tag{10}$$

The memory requirement for each block is 512 bytes (16 × 16 × 2), which is the most efficient block size for disk I/O. Blocks are numbered by their three-dimensional coordinate (i,j,k):

$$\text{No} = (ncb*nch)*(i - 1) + ncb*(k - 1) + j \tag{11}$$

A temporary disk file of *nlb*ncb*nch* records is created, with a record length of 512 bytes. Record 'No' in the temporary file corresponds to subimage (i,j,k) in the output image according to the relationship established in Equation 11.

Consider a scan line falling into a specified window (Figure 1), where the line crosses several block rows in the output image. As shown in Figure 1, the scan line only crosses those blocks marked by the heavy line. If the angle between the scan line and the row of the output image is less than or equal to 45 degrees, each block column only has a maximum of two blocks containing the current scan line. In other words, instead of keeping the whole output image in the system memory, only 2**ncb*nch* blocks need to be kept. Note that the number of blocks to be kept is only related to the number of channels and the number of columns in the output image. Therefore, the system memory requirement is only related to the width of the window and the number of output channels. We assume that the angle between the scan line and the row of the output image is less than 45 degrees. Except at very high latitudes, all AVHRR data meet this assumption. Actually, if the angle is larger than 45 degrees, two blocks in each block row need to be kept in the system memory. The total memory for this circumstance is 2**nlb*nch* blocks.

As stated above, a temporary disk file, which holds all output image data, is created during the geometric correction. The 2**nlb*nch* blocks of the system memory are allocated as workspace for rectification. Suppose a pixel in the current scan line is in the window area, through Equations 3 to 6, the location of the pixel in the output image will be (L, C). Thus, the pixel will be in the subimage (i,j,k), where

$$\begin{aligned} i &= [(L - 1)/16] + 1 \\ j &= [(C - 1)/16] + 1 \\ k &= 1,2,\ldots,nch \end{aligned} \tag{12}$$

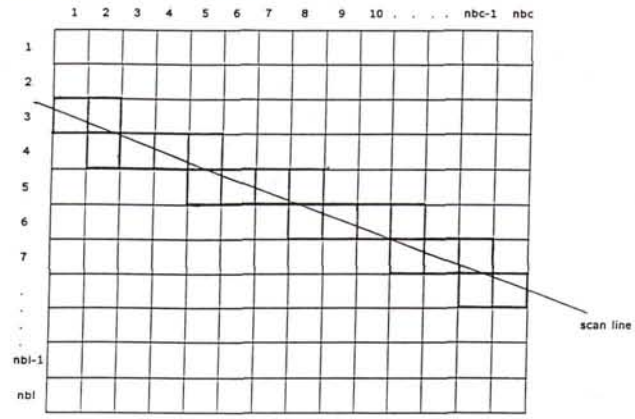

Figure 1. An example of a scan line crossing block images of a window.

and the coordinate for the pixel in the subimage (i,j,k) is (sl, sc), where

$$\begin{aligned} sl &= L - (i - 1)*16 \\ sc &= C - (j - 1)*16 \end{aligned} \tag{13}$$

Then, the block row number i is compared with the block numbers of two blocks in the jth block column kept in the system memory. If one of them is matched, the intensity of the input pixel will be assigned to the location (sl,sc) of the matched block. If none of them is matched, the location of current pixel is not in the system memory. In this circumstance, the block with two differences from the block row number of the current pixel should be written to the disk according to the record number obtained from Equation 11. Because the satellite always goes forward, the block to be written out to disk will never be readdressed any more. Therefore, it will not be retrieved to the system memory again. The memory space holding the block just written out will be used to hold the block which contains the current pixel (this block becomes a current block).

The multi-block virtual memory algorithm can be described as follows:

(1) Use Equations 3 through 6 to get the row and column in the output image for the current input pixel.
(2) Use Equations 12 and 13 to compute the subimage coordinate and the location of the current input pixel at the subimage.
(3) Compare the block row number of the current pixel with two blocks in same block column number being held in the system memory. If one of them is matched, go to step (6) (the matched block is called the current block).
(4) If none of them is matched, write the block in system memory which has a difference of two in block row number with the block row number of the current pixel to its corresponding record of the disk file according to the relationship established in Equation 11. The block then becomes the current block.
(5) Change the block coordinate of the current block to the block coordinate of the current input pixel. Clean the current block to zero.
(6) Assign the location of (sl, sc) in the current block with the intensity value of the current input pixel.Go to step (1) for the next pixel.

The memory requirement for the above algorithm is 2*ncb*nch blocks. If a program is designed to implement a rectified output image with five channels (maximum number of channels for AVHRR data) and 2048 columns in width, the system memory requirement is 640 kbytes (2 * 128 * 5 * 512). If the block size of 128 bytes (8 * 8 * 2) is selected, the amount of required memory is 320 kbytes. There are no limitations on the number of rows being processed. Such a system memory requirement can be met by a personal computer.

Compared with algorithms holding whole the output image in the system memory, the calculations of Equations 12 and 13 for every input pixel are extra computations in the virtual memory implementation. Therefore, four integer multiplication/division and four integer addition/subtract operations are applied to each pixel. The amount of I/O is not increased when using a virtual memory implementation.

The output image in the temporary disk file is stored as a series of subimages. To retrieve it as a whole image, an array with 16 rows and *nch*ncb**16 columns in size is required. We can use the same memory holding virtual blocks to hold this array. When the temporary disk file is sequentially read into the array, the first row of the array holds all "*nch*" channels of the first row of the rectified image. These data are processed in the linear filling algorithm to provide numbers for the blank pixels, which can be used in atmospheric attenuation correction, cloud detection, and calculation of albedo.

Radiometric Correction and Calibration

The radiance measured by any remote sensing system over a given object is influenced by such factors as change in scene illumination, atmospheric conditions, viewing geometry, and instrument-response characteristics (Lillesand and Kiefer, 1987). In order to compare the data gathered at different observation times by the same sensor system (multitemporal data analysis), it is necessary to remove all errors caused by the factors previously noted (Lillesand and Kiefer, 1987). The processing to remove those errors is called radiometric correction..

As we know, most terrestrial applications of remote sensing are concerned with the physical properties of ground objects. The conversion of raw data to a meaningful physical property is called data calibration.

The AVHRR level 1b tapes provide solar zenith angle data and calibration coefficients. Based on that information, several radiometric correction and calibration processes can be carried out scan line by scan line.

Solar Zenith Angle Correction

Because of the large scanning angle of the AVHRR (about ± 55.4 degrees), the solar zenith angle varies significantly along one scan line. The large angular difference can result in quite different amounts of solar radiation received by a ground object, resulting in spectral errors in the visible and near infrared channels of AVHRR data. If the research area is larger than 10 degrees in latitude, the variation of solar-zenith angles is also larger than 10 degrees. Solar-zenith angle correction is a necessary step for regional scale applications of AVHRR data. A cosine correction can remove those errors:

$$DN_0 = DN_\theta / \cos\theta \qquad (14)$$

Equation 14 corrects a digital number with a solar-zenith angle of θ degrees (DN_θ) to a digital number with an angle of 0 degrees(DN_0).

The solar-zenith angle data required in Equation 14 could be obtained from an AVHRR level 1b tape. Specifically, there are 2048 pixels in an LAC/HRPT scan line and the solar-zenith angles are sampled every 40 pixels starting at pixel 25 (25, 65, 105,..., 1945, 1985, 2025). Therefore, there are 51 such values possible and each of them requires one byte, and they are stored as degrees*2. The angle for every pixel along a scan line could be interpolated from the 51 points of solar-zenith angles by using a first-order interpolation.

Calibration of AVHRR Data

AVHRR thermal data (channels 3 and 4, and 5 when present) may be converted to temperature, and the visible and near-infrared data (channels 1 and 2) may be converted to reflectance based on the conversion coefficients appended on the AVHRR tape. The calibration procedures are described herein.

The AVHRR tape provides the calibration coefficients, consisting of slope and intercept values for each of the five channels, which are located in byte 13 to byte 52 of each logical record of an AVHRR 1b tape with four bytes for each value (NOAA, 1990).

Once the calibration coefficients have been extracted, they must be scaled. The slope values must be divided by 2^{30} and the intercept values by 2^{22} (NOAA, 1990). The scaled slopes and intercepts then are used to calibrate the AVHRR data.

Thermal Channel Calibration

The scaled thermal channel slope values are in units of (milliwatts/m²)/(steradians·cm⁻¹) per count and the intercept is in (milliwatts/m²)/(steradians·cm⁻¹) (NOAA, 1990). The equation to convert DNs to their brightness temperature is

$$T_s(E) = C_2\nu / \ln(1 + C_1\nu^3/E) \qquad (15)$$

where T_s = the surface brightness temperature (K),
E = the energy value (irradiance at instrument aperture),
ν = the central wave number of channel filter (cm^{-1}),
C_1 = $1.1910659 * 10^{-5}$ mW/(m² · sr · cm⁻⁴), and
C_2 = 1.438833 (cm · K).

The radiant energy can be obtained by converting the digital numbers

$$E = c^*TIR + d \qquad (16)$$

where TIR = a digital number from AVHRR channel 3, channel 4, or channel 5, and
c, d = constants appended on the NOAA-AVHRR level 1b tape.

Note that the temperatures obtained by this procedure are not yet corrected for atmospheric attenuation.

In Equation 15, a variable called central wave number is used. The central wave numbers for channels 3, 4, and 5 vary with temperature ranges and satellites. NOAA provides detailed values for various satellites and temperature ranges (NOAA, 1990) .

Visible Channel Calibration

The scaled visible channel slope values are in units of percent reflectance/count for slopes and in percent reflectance for the intercepts. The percent reflectance measured by the sensor channel i is computed as a linear function of the input data value as follows (NOAA, 1990):

$$R_i = S_i \cdot C + I_i \quad (17)$$

where R_i = the percent reflectance measured by channel i,
C = the input data value,
S_i = the slope coefficient of channel i, and
I_i = the intercept coefficient of channel i.

The visible channels (1 and 2) are calibrated using Equation 17 to obtain the percent reflectance.

Calculation of Albedo

The overall reflectance in the visible and near-infrared portion of an EM wave, or albedo, may be useful in many areas of research such as calculating the energy balance of the Earth's surface.

The AHVRR has two channels of data in the reflected portion of an EM wave. Although the spectral range of the instrument does not cover the entire visible and near infrared EM wave, we may combine the reflectance in channels 1 and 2 to estimate albedo in the range of 0.58 to 1.10 micrometres by

$$\alpha = C_1{}^*r_1 + C_2{}^*r_2 \quad (18)$$

where
r_1 = the reflectance of channel 1,
r_2 = the reflectance of channel 2,
C_1 = proportional constants for channel 1, and
C_1 = proportional constants for channel 2.

The quantities C_1 and C_2 can be computed by the ratios of the solar energy in each of channels to the total solar radiant energy in the 0.58 to 1.10 μm range. The coefficients are 0.322 and 0.688, respectively (He *et al.*, 1987). Then, albedo can be obtained by

$$\alpha = 0.322^*r_1 + 0.678^*r_2. \quad (19)$$

Atmospheric Attenuation Correction for Retrieving Surface Temperature

Terrestrial surface temperature measurements made by remote sensors are attenuated by the Earth's atmosphere. The atmosphere modifies observed brightness temperatures by either increasing or decreasing upwelling radiation received by the sensor.

AVHRR channels 4 and 5 are designed to exploit atmospheric windows in the thermal region of the electromagnetic spectrum where attenuation is relatively small; but even so, the effects can induce significant error in the AVHRR-sensed surface temperature. Because the atmospheric attenuation rates for the AVHRR thermal bands (from 10.1 to 11.5 μm for channel 4, and from 11.1 to 12.6 μm for channel 5) are different, it is possible to use the difference in spectral response in the two EM windows (split-window) to derive attenuation correction algorithms. Several split-window atmospheric attenuation correction models have been developed by a number of investigators including those by Deschamps and Phulpin (1980):

$$T_c(4,5) = 2.626(\text{ch.4}) - 1.626\ (\text{ch.5}) - 1.1 \quad (20)$$

McClain *et al.* (1983):

$$T_c(4,5) = 1.035(\text{ch.4}) + 3.046\ (\text{ch.4} - \text{ch.5}) - 10.784 \quad (21)$$

Price (1984):

$$T_c(4,5) = (\text{ch.4}) + 3.33(\text{ch.4} - \text{ch.5}) \quad (22)$$

and Singh (1984):

$$T_c(4,5) = 1.699\ (\text{ch.4}) - 0.699\ (\text{ch.5}) - 0.240 \quad (23)$$

where
T_c = the atmosphere-corrected surface temperature,
ch.4 = calibrated surface temperature from channel 4, and
ch.5 = calibrated surface temperature from channel 5.

All temperatures in the above equations are in Kelvin. According to an experimental test, the McClain split-window model yields AVHRR temperatures that are consistently within the defined tolerance interval (Cooper and Asrar, 1990); therefore, it can be used to correct AVHRR thermal data for atmospheric attenuation.

Cloud Detection

It is sometimes useful to separate cloud-covered area from that which is cloud-free. Common sense tells us that a cloud is usually cooler than the land surface (Eck and Kalb, 1991). Because the AVHRR has thermal channels which record the surface temperature, it is not difficult to separate cloud from land surface based on the thermal channel calibration result of the AVHRR data. Pixels with temperatures below a selected threshold value can be thought of as cloud cover and a digital code can be assigned.

The One-Step Algorithm

Based on the discussion above, the overall one-step algorithm which combines tape reading with geometric correction, radiometric correction and calibration of AVHRR 1b data could be described as follows:

(1) Read a scan line. This scan line becomes the current scan line;
(2) Determine whether or not the current scan line is within the output window area by checking the 51 Earth locations appended on the scan line. If the current scan line is within the output window, go to step 4;
(3) Use the logical variable described in the "spatial interpolation" section to determine whether or not whole output window has been retrieved from the tape. If the answer is yes, stop tape reading and go to step 8; otherwise, go to step 1;
(4) Apply the solar zenith angle correction (Equation 14) and data calibration (Equations 15, 16, and 17) to pixels of the current scan line (if the raw AVHRR data are needed, skip this step);
(5) Project pixels in the current scan line to locations in the output window by applying Equations 3, 4, 5, 6 and 7;
(6) Place pixels of the current scan line using the multiple block virtual memory algorithm;
(7) Repeat steps 4 through 6 until all pixels of the current scan line, which fall into the output window, have been processed. Go to step 1;
(8) Retrieve a block row of data from the temporary disk file created by the virtual memory algorithm (see section "multiple block virtual memory algorithm");
(9) Fill the blank pixels by linear filling (Equation 8);
(10) Calculate albedo (Equation 18), correct for atmospheric attenuation (Equations 20, 21, 22, or 23) , and detect the cloud-covered area;
(11) Write the result to the result file;
(12) Repeat steps 8 through 11 until all block rows have been processed.

Figure 2 shows the general flow chart for the algorithm.

"NAPS"

Based on the algorithm described in this article, a comprehensive AVHRR tape reading and preprocessing package was

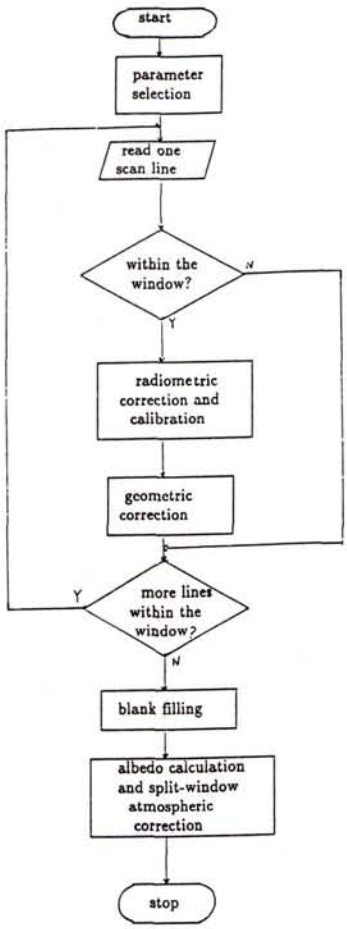

Figure 2. General flow chart of the one-step AVHRR preprocessing algorithm.

developed at the Center for Advanced Land Management Information Technologies (CALMIT), University of Nebraska-Lincoln. The software, referred to as "NAPS" (Nebraska AVHRR Preprocessing Software), allows users to extract a georeferenced area from AVHRR tape with a nearly unlimited window size and selectable numbers of channels (from one to five). The output dataset is georeferenced with user-specified pixel size (default value is 0.01333 by 0.01333 degrees, near 1.1 km by 1.6 km at 40 degrees latitude). The output dataset can be either raw digital numbers or radiometrically corrected ground physical quantities such as reflectance or surface temperature. A solar zenith-angle correction option is available, and cloud detection is possible. Another option allows users to create an "albedo channel" based on the combined reflectance in both visible and near infrared channels. A split-window atmospheric correction option could be used to eliminate the vapor absorption of the thermal-infrared radiation emitting from ground objects and to allow retrieval of "true" surface temperature by using the temperature difference between AVHRR channel 4 and channel 5. Four split-window atmospheric correction models (Deshamps and Phulpin, 1980; McClain *et al.*, 1983; Price, 1984; Singh, 1984) have been encoded in the program for user selection. An option is available to allow a user to achieve decimal accuracy of output data by entering an enlargement scale.

Conclusion

The algorithm presented in this paper provides an efficient mechanism for obtaining geometrically and radiometrically corrected NOAA/AVHRR data for research. Compared with the traditional AVHRR data preprocessing methods, the algorithm not only allows one to retrieve regional-scale AVHRR data but also saves considerable time and human resources. Furthermore, the memory requirement is small enough to provide for implementation on a PC.

References

Cooper, D. I., and G. Asrar, 1989. Evaluating Atmospheric Correction Models for Retrieving Surface Temperatures from the AVHRR Over a Tallgrass Prairie, *Remote Sens. Environ.*, 27:93–102.

Deschamps, P.Y., and T. Phulpin, 1980. Atmospheric Correction of Infrared Measurements of Sea Surface Temperature Using Channels at 3.7 , 11 and 12 mm, *Boundary Layer Meteorol.*, 18:131–143.

ERDAS, 1990. *ERDAS User's Guide*, ERDAS Inc., Altanta, Georgia.

Eck, T. F., and V. L. Kalb, 1991. Cloud-Screening for Africa Using a Geographically and Seasonally Variable Infrared Threshold. *Int. J. Remote Sensing*, 12:1205–1221.

He, C., K. Kittleson, and J. Bartholic, 1987. Evapotranspiration Monitored from Satellites as an Indication of Shift and Impact of Vegetation Change, Presented at the Twenty-First International Symposium on Remote Sensing of Environment, Ann Arbor, Michigan.

Jensen, J. R., 1986. *Introductory Digital Image Processing.* Prentice-Hall, Englewood Cliffs, New Jersey.

Lillesand, T. M., and R. W. Kiefer, 1987. *Remote Sensing and Image Interpretation*, Second Edition, John Wiley & Sons, Inc., New York.

McClain, E.P., W.G. Pichel, C.C. Walton, Z. Ahmad, and J. Sutton, 1983. Multichannel Improvements to Satellite-derived Global Sea Surface Temperatures, *Adv. Space Res.* 2:43–47.

NASA, 1989. *ELAS-Earth Resources Laboratory Application Software*, NASA National Space Technology Laboratories, Earth Resources Laboratory.

NOAA, 1990. *NOAA Polar Orbiter Data User's Guide*, U.S. Department of Commerce, NOAA, NESDI, NCDC, and the Satellite Data Service Division, Washington, D. C.

Peters, A. J., 1989. *Coarse Spatial Resolution Satellite Remote Sensing of Drought Conditions in Nebraska: 1985-1988*, Ph.D Dissertation, Department of Geography, University of Nebraska-Lincoln.

Price, J. C., 1982. Estimation of Regional Scale Evapotranspiration Through Analysis of Satellite Thermal Infrared Data, *IEEE Trans. Geoscience and Remote Sensing*, GE-20(3): 286–292.

———, 1984. Land Surface Temperature Measurements from The Split-window Channels of NOAA 7 Advanced Very High Resolution Radiometer, *J. Geophys. Res.* 89:7231–7237.

USGS, 1990. *LAS Version 5*, U.S. Geological Survey, EROS Data Center, Sioux Falls, South Dakota.

Wienreb, M. P., and M. L. Hill, 1980. *Calculation of Atmospheric Radiances and Brightness Temperatures in Infrared Window Channels of Satellite Radiometers*, NOAA Technical Report NESS 80, U.S. Department of Commerce, NOAA, Washington, D.C.

(Received 26 June 1992; accepted 15 December 1992; revised 19 January 1993)