MODIS DATA STUDY TEAM PRESENTATION

February 16, 1990

AGENDA

1. Future AVHRR Attributes and Commonality to MODIS (Ardanuy)
2. Core Ocean Data Products (Ardanuy, Gregg)
3. Sizing the Algorithm for Computing Water-Leaving Radiance, Angstrom Exponents, and Single Scattering Aerosol Radiances (Gregg)
4. Sizing Estimates for MODIS-N Terrestrial Data Products (Riggs, Schols)
5. MODIS Data Packet Structure: Instrument-to-Ground Link (McKay)

FUTURE AVHRR ATTRIBUTES AND COMMONALITY TO MODIS[1]

THE NOAA-K SERIES IMAGERS (AVHRR-3)

A set of four AVHRR series-3 instruments will be flown on NOAA-K/L/M/N. The series-3 will begin with the launch of NOAA-K in December 1993 or early 1994. The spectral parameters for AVHRR-3 will be essentially identical to the instruments presently being flown operationally, with one notable exception: channel 3 will be split into 3A and 3B. At night, channel 3A will take data in the 3.75 μm window for surface temperature determination. During the day, when the 3.75 μm band is a mixture of Earth-emitted and solar-reflected radiances, a 1.6 μm band (channel 3A) will be utilized. However, additional applications of the channel include aerosol determination (1.6 μm) and forest fire detection and monitoring (3.75 μm). It is suggested that the channel switch not only be made at the terminator, but also upon entering and leaving the tropics, where there is little snow cover. However, aerosols will also be determined in the tropics, and will be particularly interesting near forest fires. Therefore, some conflicts requiring special mode scheduling are anticipated ("where there's smoke, there's fire").

Because most of the snow signal will be at low radiance levels (strong absorber), and due to the desire for the detection of aerosols over the oceans with higher precision, the NOAA-K channel 3A will have a split gain, with half of the 1024 counts allocated for reflectances less than 12.5%, and the other 512 counts for reflectances greater than 12.5%.

A split gain digitization will also be employed on channels 1 and 2 at 25% albedo level. For these 10-bit bands, half of the count range (512 counts) will cover the reflectance range of 0 to 25%, and half the remaining range of 25% to 100% (with some provision to avoid saturation due to anisotropic reflection).

The AVHRR-3 series will maintain the present 10-bit measurements, with local-area coverage (LAC) of 1.1 km, averaged and sub-sampled to the 4 km global-area coverage (GAC).

THE NOAA-O SERIES IMAGERS (AVHRR-4)

The final recommendations are being prepared by NESDIS for the NOAA-O series of imagers, the AVHRR-4. Scheduled for launch around the year 1997/98, these instrument are proposed to have seven spectral bands, and are similar to the AMRIR concept without the 4.3 μm CO_2 sounding bands. All bands will take data at a 12-bit resolution, as will MODIS, though specific bands will have split gains. The proposed bands for NOAA-O are as follows:

---

[1]Personal communication, Larry Stowe and Jim Fisher, NOAA/NESDIS, February 13, 1990.

Band 1: **Visible reflectance.** Center at 615 nm, with 1% cut-offs at ±10 nm (605 and 625 nm). This band is similar to the present AVHRR band 1 and the proposed 214 m MODIS band 1 at 659±5 nm with 50% cut-offs at ±25 nm (634 and 684 nm). MODIS-N will also have an 856 m channel (band 13) at 653+1-2 nm with a 15 nm bandwidth (if the gain of this ocean color channel does not cause saturation).

Band 2: **Near-infrared reflectance.** Center at 870 nm, with 1% cut-offs at ±10 nm (860 and 880 nm). This band is similar to the present AVHRR band 2 and the proposed 214 m MODIS band 2 at 865±5 nm with 50% cut-offs at ±20 nm (845 and 885 nm). MODIS-N will also have an 856 m channel (band 16) at 865±5 nm with a 15 nm bandwidth (if the gain of this ocean color channel does not cause saturation).

Band 3: **Snow/cloud discrimination.** Center at 1.61 μm, with 50% cut-offs at 1.58 and 1.64 μm. This band is identical to the band proposed for NOAA K/L/M, and similar to the proposed 856 m MODIS band 6 at 1.64 μm ±8 nm with 50% cut-offs at ±10 nm (1.54 and 1.74 μm).

Band 4: **Atmospheric window.** Center at 3.7 μm, with 50% cut-offs at 3.62 and 3.83 μm. This band is similar to the present AVHRR band 3 and the proposed 856 m MODIS band 20 at 3.75 μm ±19 nm with 50% cut-offs at ±90 nm (3.66 and 3.84 μm).

Band 5: **Split window.** Center at 4.0 μm, with 50% cut-offs at 3.92 and 4.10 μm. This band is similar to the proposed 856 m MODIS bands 22 and 23 at 3.959 μm and 4.050 μm ±20 nm with 50% cut-offs at ±25 nm (3.934 and 3.984 μm, and 4.02 and 4.08 μm). [The Atmospheric Sciences Branch of NESDIS is being asked to justify the requirements for this band.]

Band 6: **Atmospheric Window.** Center at 10.8 μm, with 50% cut-offs at 10.3 and 11.3 μm. This band is similar to the present AVHRR band 4 and the proposed 856 m MODIS band 31 at 11.03 μm ±55 nm with 50% cut-offs at ±250 nm (10.78 and 11.28 μm).

Band 7: **Split Window.** Center at 12.0 μm, with 50% cut-offs at 11.5 and 12.5 μm. This band is similar to the present AVHRR band 5 and the proposed 856 m MODIS band 32 at 12.02 μm ±60 nm with 50% cut-offs at ±250 nm (11.77 and 12.27 μm).

The AVHRR-4 series will have 12-bit quantization, and the ground and space segments of the data system with dispense with the concept of LAC and GAC. With these instruments, the full 1.1 km nadir resolution will be retained globally.

CORE OCEAN DATA PRODUCTS

1. Data Flow Scenario

Figure 1 describes a data flow scenario for obtaining ocean core data products proposed for MODIS. Although this figure may be considered a candidate processing scenario, its primary purpose is to show the interrelations among the products. The overview is compatible with the MODIS ocean core data products algorithms; however, expected changes that may result from improved algorithms are not reflected. The flight of the SeaWiFS instrument, for example, may result in some changes.

Derivation of MODIS ocean data products should begin with Level-1B calibrated radiances. A cloud filter may then be applied, diverting the data stream from ocean products to cloud products (if pixels are cloudy they will not be processed for ocean products). However, these or alternative cloud products could then be used to make determinations of surface incident photosynthetically available radiation (PAR).

If pixels pass the cloud test (if they are sufficiently clear), they should then be used, uncorrected for atmosphere, to determine sea-ice extent. They can also be corrected for water vapor to determine the sea surface temperature. These products would only be generated by MODIS-N because they require observations in thermal-infrared wavelengths.

Otherwise, the pixels will be atmospherically corrected to produce water-leaving radiances. At this time, it is expected that both MODIS-N and -T will generate water-leaving radiances and the products derived therefrom. Although this scenario involves some duplication, it also maximizes global coverage by allowing MODIS-N to fill gaps in MODIS-T observations at Level-3, and vice versa (e.g., sun-glint removal).

The atmospheric correction involves determining the Rayleigh scattering and aerosol absorption and scattering contributions to the total radiances, and removing them. Angstrom exponents (an indicator of the wavelength dependence of aerosol attenuation) come out directly in this process.

Atmospheric correction should directly generate two ocean core products: water-leaving radiances and aerosol radiances. Water-leaving radiances form the foundation for the remaining products. They would be used to directly obtain pigment concentrations (analogously to the method used by the CZCS), and chlorophyll-a pigment concentrations for Case-1 and Case-2 waters. In the generation of Case-2 pigment concentrations, a flag differentiating Case-1 from Case-2 areas will be used to determine whether to employ the Case-2 algorithm. Also derived from the water-leaving radiances are chlorophyll fluorescence, total seston (suspended particulate matter) concentrations, detached coccolith (from coccolithophores) concentrations, K(490), dissolved organic matter and phycoerythrin concentrations. Phycoerythrin is an accessory light-harvesting pigment present in specialized types of phytoplankton.

For the final ocean core data product, primary production, an empirical approach relates MODIS-generated pigment concentrations to measured in-situ primary production. Surface incident photosynthetically active radiation (PAR) and the diffuse attenuation coefficient for PAR (which are designated as special products), as well as Case-1 and Case-2 chlorophyll-a concentrations, may also be involved in this algorithm. Chlorophyll fluorescence is also envisioned as a possible method to obtain primary production in the future.

2. Implementation Scenario

In Figure 1, data dependencies and implementation interfaces become clearly evident. We see that the ocean standard product generation scenario at launch consists of a limited number of semi-independent processes:

1. Calibration and navigation of the MODIS "at-satellite" radiances.

2. Generation of the cloud flags and application of the cloud filter.

3. Production of the products dependent on the MODIS-N thermal-infrared radiances not requiring atmospheric correction.

4. Application of the atmospheric correction, yielding the angstrom exponents, aerosol radiances, and water-leaving radiances.

5. Comparisons of the water-leaving radiances and in-situ data to determine calibration adjustment data sets; application of the calibration adjustment data sets to revise the water-leaving radiances.

6. Generation (in a "parallel" sense) of a wide variety of organic and inorganic ocean data products from the water-leaving radiances, including:

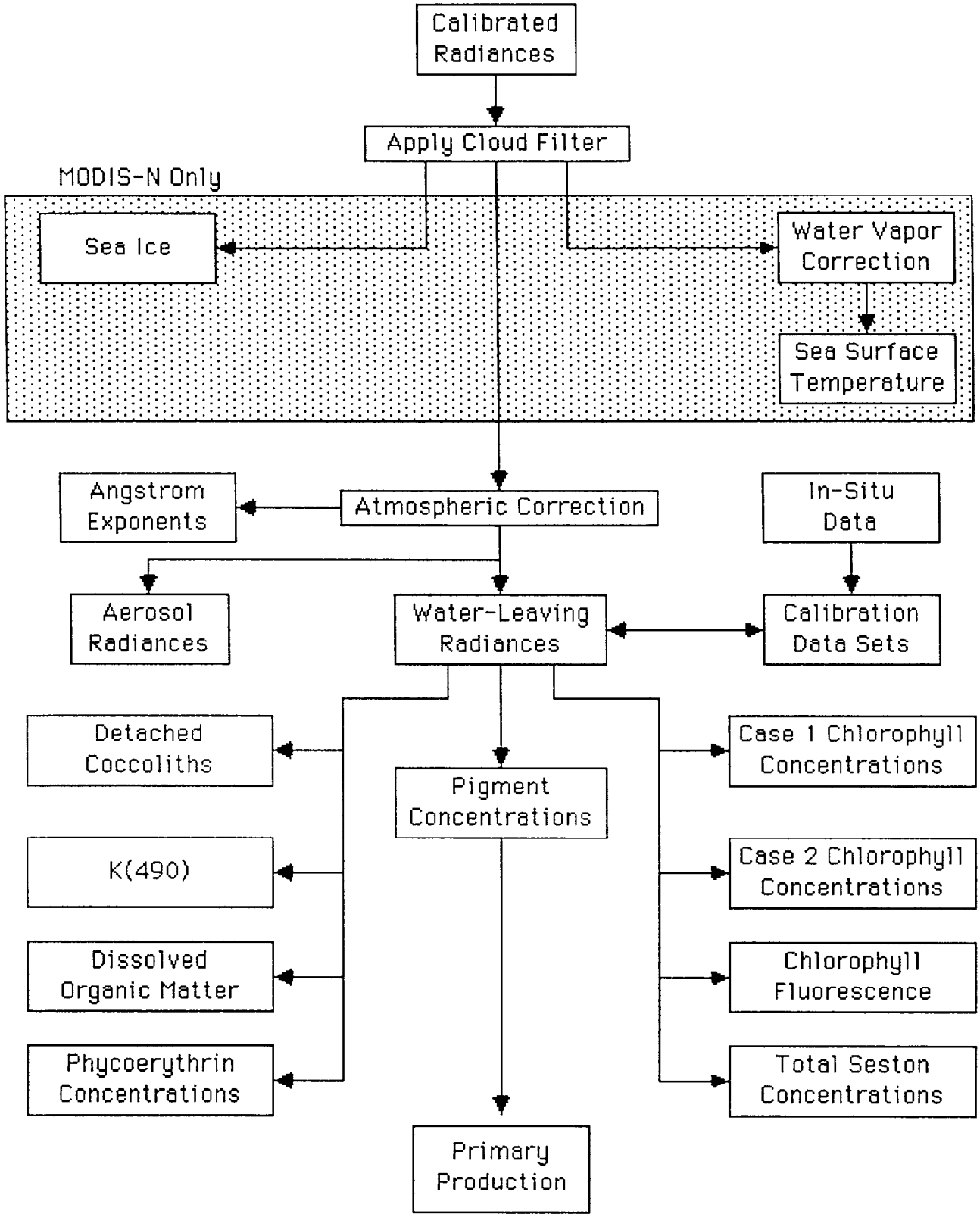
   - detached coccoliths
   - K (490 nm)
   - dissolved organic matter
   - phycoerythrin concentrations
   - pigment concentrations (CZCS)
   - case-1 chlorophyll concentration
   - case-2 chlorophyll concentration
   - total seston consentration
   - currents, fronts, and eddies (qualitative/Level-4; images)

7. Generation of the primary production.

Each of the seven classes of elements, and the elements within each class, form a logically (semi-)independent algorithm. As such, the scenario for implementation of the ocean at-launch standard data product software in the CDHF is naturally and logically decomposed.

When MODIS data product software prototyping is considered, each of these algorithms and their related code may be visualized and realized as a separate subroutine. The calling sequence for the subroutines is identical to the scenario illustrated in Figure 1, with a single exception. The calibration adjustments generated through a comparison of the water-leaving radiances and the in-situ data will likely require human intervention. The resultant data set would then be applied to adjust the water-leaving radiances, either for future processing or for reprocessing. The time-scale for this step could be from weeks to months initially, and then from months to annual well after launch.

MODIS-N and T Ocean Standard Product Processing Scenario



Sizing the Algorithm for Computing Water-leaving Radiance, Angstrom Exponents, and Single Scattering Aerosol Radiances

INTRODUCTION

Water-Leaving Radiances, Angstrom exponents, and single scattering aerosol radiances are MODIS Ocean Standard Products that derive directly from atmospheric correction. Thus to estimate sizing requirements it is useful to consider them together. In estimating the sizing requirements, we have followed the proposed method of atmospheric correction of Gordon (1989). Since this method differs substantially from that currently in use for the CZCS, we have estimated sizing requirements by writing prototype computer code (FORTRAN 77), attempting to include all the major processing and mathematical procedures. The code has been tested for completeness and data-handling and flow errors, and is presented in the Appendix. However, it must be emphasized that the code has not been optimized nor tested for realistic radiances, and should not be considered as a prototype for the atmospheric correction algorithm; it is only used here to allow a more rigorous assessment of the computational requirements for EosDIS.

ASSUMPTIONS

In operation of the code, the following required parameters/ variables are assumed to be known, either through Level-0/1 processing or are in place in storage in the case of look-up tables.

- Anchor point array element counters
- Cloud/other error flag array at every pixel
- Mean extraterrestrial solar irradiance, ozone absorption coefficients, Rayleigh optical thickness at standard pressure
- Extraterrestrial irradiance corrected for Earth-Sun distance and eccentricity (a calculation performed daily)
- Solar and spacecraft zenith and azimuth angles at every pixel
- Atmospheric pressure at anchor points
- Fourier coefficients of Rayleigh scattering, a look-up table
- Wind speed, ozone at every pixel
- At-satellite calibrated radiances at every pixel

We assume, following the CZCS heritage, that normalized Rayleigh radiances (not corrected for instantaneous irradiance) will be computed directly from the Fourier coefficients only at anchor points, then interpolated (cubic spline) to every pixel. Rayleigh radiance will then be computed at every pixel by multiplication for instantaneous irradiance.

The remaining atmospheric correction operations, estimation of sun glitter and aerosol radiances, then occurs on a pixel-by-pixel basis.

ERROR CHECKING

Since error checking is an important component of any computer algorithm, it has been included as much as we can conceive at this time. The following error checking flags are included.

- o Cloud flags
- o Tests to ensure solar and spacecraft zenith angles are within range of those included in the bi-linear interpolation of Rayleigh Fourier coefficients (≈78°)
- o Negative radiance flags
- o Sun glitter flag

A final note is that transcendental functions (e.g., e^x, sin x, etc.) were considered 10 times the number of operations of arithmetic functions (+, -, x, /).

RESULTS

Symbols Table

		MODIS-N	MODIS-T
I	= number of pixels along scan	1582	1007
J	= number of pixels along track	8	30
IA	= number of anchor points along scan	94	80
JA	= number of anchor points along track	2	5
L	= number of wavelengths	9	32
N	= number of operations per scan		

I. Computation of Rayleigh Scattering

A. Compute Normalized Rayleigh Radiance and Rayleigh Optical Thickness at Anchor Points

$$N = (184 + 95*L)*IA*JA$$

MODIS-N: 0.20 MFLOP/scan
MODIS-T: 1.29 MFLOP/scan

B. Compute Normalized Rayleigh Radiance and Rayleigh Optical Thickness at Along Scan at Along-Track Anchor Point Locations (24 operations to set up cubic spline interpolation; 10

operations to interpolate)

$$N = [2 + 2*IA + 2*24 + 4*I + 2*10]*L*JA$$

$$N = [70 + 2*IA + 4*I]*L*JA$$

MODIS-N: 0.19 MFLOP/scan
MODIS-T: 0.75 MFLOP/scan

C. Compute Normalized Rayleigh Radiance and Rayleigh Optical Thickness at Along Track at each Along Scan Point (24 operations to set up cubic spline interpolation; 10 operations to interpolate)

$$N = JA + \{1 + [3*JA + 2*24 + (2 + 2*10)*J]*L\}*I$$

$$N = JA + \{1 + [3*JA + 48 + 40*J]*L\}*I$$

MODIS-N: 5.33 MFLOP/scan
MODIS-T: 41.70 MFLOP/scan

II. Computation of Aerosol Characteristics, Production of Products

		T	N
N = (184 + 21*L			
+ L	(If sun glint flag)	≈ 5%	≈20%
+ 334 + 41*L)	(No flag)	≈95%	≈80%
*I*J			

MODIS-N: 10.90 MFLOP/scan
MODIS-T: 73.16 MFLOP/scan

III. Totals

MODIS-N: 16.62 MFLOP/scan
MODIS-T: 116.90 MFLOP/scan

DISCUSSION

The water-leaving radiance algorithm, which includes atmospheric correction, is the most computationally demanding of the ocean algorithms. It forms the basis for which all of the ocean color products (all but sea ice and sea surface temperature) are derived.

Thus it has a major impact on sizing estimates for MODIS.

The results per scan may be further illustrated by expressing on a per-pixel, and a per-pixel-per-wavelength basis for comparison with other sensors (Table 1).

Table 1. Assessment of sizing requirements for water-leaving radiance.

	Operations Per Scan	Operations Per Pixel	Operations Per Pixel Per Wavelength
MODIS-N	16.62×10^6	1313	146
MODIS-T	116.90×10^6	3870	121

D. Hoyt (personal communication) found that the CZCS required ≈ 200 operations per pixel per wavelength, which is greater by factors of 1.4 and 1.7 the estimates presented here for MODIS-N and MODIS-T, respectively. These differences arise from the fact that the code tested by D. Hoyt included Level-1 operations, which were not included in the present analysis. One may expect that the size requirements for MODIS are larger than for the CZCS, given that aerosol determinations are performed for MODIS on a pixel-by-pixel basis.

Finally, one may expect the atmospheric correction algorithm to improve, and probably grow, as more sophisticated methods are developed in the MODIS era. The analysis presented here applies only to current methods, but serves as a reasonable first approximation for the size requirements for MODIS water-leaving radiance algorithms.

Reference:

Gordon, H.R., 1989. Algorithm development for ocean observation with EOS/MODIS. MODIS Science Team Member Report.

Appendix. Computer code used to derive sizing estimates of MODIS Ocean Atmospheric Correction Algorithm.

Note: Code preceding the computation of Rayleigh radiances is merely set up to test execution of program and is not included in the sizing estimates.

```
      program setupt
c
c  Set up program for MODIS-T sizing estimates.
c
      parameter(imax=1107,jmax=30,nlt=32,iarr=80,jarr=5)
      common/blk1/rim(40,39,nlt,3)
      integer lam(nlt),ianc(iarr),janc(jarr)
      integer icerr(imax,jmax)
      real Fobar(nlt),Fo(nlt),FFo(nlt),oza(nlt),thray0(nlt)
      real thray(imax,jmax,nlt),ri(imax,jmax,nlt)
      real totl(imax,jmax,nlt)
      real th(imax,jmax),th0(imax,jmax),phi(imax,jmax),phi0(imax,jmax)
      real dbu(imax,jmax),ws(imax,jmax)
      real pres(iarr,jarr)
      real taur(nlt),thoz(nlt),rlr(nlt),rlt(nlt),rlw(nlt),rla(nlt)
      real rlwn(imax,jmax,nlt),rrla(imax,jmax,nlt),eta(3)

      double precision pi,pi2,rad
c
      pi = dacos(-1.0D0)
      rad = 180.0D0/pi
      pi2 = 2.0D0*pi
      jday = 82
      rjd = jday
c
c   Obtain extraterrestrial irradiance, ozone, and standard Rayleigh
c   optical thickness for MODIS-T
      call tdata2(lam,Fobar,oza,thray0)
c   Compute instantaneous extraterrestrial irradiance Fo
      daycor = (1.0+0.0167*cos(pi2*(rjd-3.0)/365.0))**2
      do nl = 1,nlt
       Fo(nl) = Fobar(nl)*daycor
      enddo
c
c  Obtain ancor point array data
      write(6,*)'Obtaining anchor point array counters...'
      call rdanc(ianc,janc)
c
c  Obtain pressure data at anchor points
      write(6,*)'calling rdpres...'
      call rdpres(pres)
c
c  Create Rayleigh look-up table
      write(6,*)'Creating Rayleigh look-up table...'
      call mkray
c
c  Read ancillary data
      write(6,*)'Reading wind speeds, ozone...'
      call rdowp(ws,dbu)
c
c  Obtain angles
      write(6,*)'Obtaining solar and spacecraft angles...'
      call rdang(th0,th,phi0,phi)
c
```

```
c  Obtain raw radiances (Lt)
      write(6,*)'Reading total radiances...'
      call rawdat(totl)
c
c  Interpolate Rayleigh coefficients to pixel level
      write(6,*)'Computing Rayleigh radiances...'
      call rayit(rad,ianc,janc,th0,th,phi0,phi,pres,thray0,thray,ri)
      write(6,*)'ok'
      read(5,*)
c
c  Scan loop
      do j = 1,jmax
       do i = 1,imax
     icerr(i,j) = 0
     write(6,*)j,i
     if (icerr(i,j) .ne. 0)then     !cloud/other error flag
     sunz = th0(i,j)
     suna = phi0(i,j)
     scz = th(i,j)
     sca = phi(i,j)
     cosunz = cos(sunz/rad)
     coscz = cos(scz/rad)
     do nl = 1,nlt
      rlt(nl) = totl(i,j,nl)
      thoz(nl) = oza(nl)*(dbu(i,j)/1000.0)
      FFo(nl) = Fo(nl)*exp(-thoz(nl)*
     *          (1.0/cosunz+1.0/coscz))
      rlr(nl) = FFo(nl)*ri(i,j,nl)
      taur(nl) = thray(i,j,nl)
     enddo
     call atmot(rad,sunz,suna,scz,sca,lam,Fo,FFo,rlt,rlr,taur,
     *     thoz,ws(i,j),rlw,eta,rla)
     do nl = 1,nlt
      rlwn(i,j,nl) = rlw(nl)
      rrla(i,j,nl) = rla(nl)
     enddo
     else
      do nl = 1,nlt
       rlwn(i,j,nl) = 0.0
       rrla(i,j,nl) = 0.0
      enddo
     endif
     enddo
      enddo
c
      end
```

```
      subroutine rayit(rad,ianc,janc,th0,th,phi0,phi,pres,
     * thray0,thray,ri)
c
c  Computes Rayleigh scattering normalized intensity (I in Gordon, et
c  al's 1988 notation) including multiple scattering and
c  polarization effects, according to Gordon et al. (1988).
c  Required here is bi-linear interpolation for angular
c  determinations; bi-cubic spline interpolation to obtain
c  normalized intensity at every pixel.
c
      parameter(imax=1107,jmax=30,nlt=32,iarr=80,jarr=5)
      common/blk1/rim(40,39,nlt,3)
      integer ianc(iarr),janc(jarr)
      real pres(iarr,jarr)
      real th0(imax,jmax),th(imax,jmax),phi0(imax,jmax),phi(imax,jmax)
      real thray0(nlt)
      real thray(imax,jmax,nlt),ri(imax,jmax,nlt)
      real rm1(nlt),rm2(nlt),rm3(nlt)
      real thr(iarr,jarr,nlt),rri(iarr,jarr,nlt)
      real thrr(iarr),rrri(iarr)
      real xpos1(iarr),b1(iarr),c1(iarr),d1(iarr)
      real b2(iarr),c2(iarr),d2(iarr)
      real rji(jarr),tjh(jarr)
      real xpos2(jarr),e1(jarr),f1(jarr),g1(jarr)
      real e2(jarr),f2(jarr),g2(jarr)
      double precision rad
c
      p0 = 1013.25
c
c  Interpolate through anchor point array
      do ja = 1,jarr
       j = janc(ja)
       do ia = 1,iarr
        i = ianc(ia)
        sunz = th0(i,j)
        scz = th(i,j)
c  Error check -- check to see that angles are within range
c ***************************************************************
        ierr = 0
        if (sunz .gt. 76.0)then
         ierr = 1
        endif
        if (scz .gt. 78.0)then
         ierr = 1
        endif
        if (ierr .eq. 1)go to 100
c ***************************************************************
c  If test is passed, proceed onward
        suna = phi0(i,j)
        sca = phi(i,j)
        delphi = (sca - suna)/rad
        coscz = cos(scz/rad)
c
c  Find angular position within Im array
```

```
      if (scz .eq. 0.0)then
       ist = 1
      else
       do isat = 1,40
        rscz = (isat-1)*2.0
        if (scz .le. rscz)then
         ist = isat
         go to 5
        endif
       enddo
      endif
5     continue
      if (sunz .eq. 0.0)then
       isu = 1
      else
       do isun = 1,39
        rsunz = (isun-1)*2.0
        if (sunz .le. rsunz)then
         isu = isun
         go to 6
        endif
       enddo
      endif
6     continue
c  Special case: both angles = 0
      if (ist .eq. 1 .and. isu .eq. 1)then
       do nl = 1,nlt
        rm1(nl) = rim(ist,isu,nl,1)
        rm2(nl) = rim(ist,isu,nl,2)
        rm3(nl) = rim(ist,isu,nl,3)
       enddo
       go to 10
      endif
c  Special case 2: satellite zenith = 0 -- Linearly interpolate in sunz
      if (ist .eq. 1)then
       rsunz1 = (isu-1)*2.0 - 2.0
       rsunz2 = (isu-1)*2.0
       do nl = 1,nlt
        call linterp(rsunz1,rsunz2,rim(ist,isu-1,nl,1),
     *       rim(ist,isu,nl,1),sunz,rm1(nl))
        call linterp(rsunz1,rsunz2,rim(ist,isu-1,nl,2),
     *       rim(ist,isu,nl,2),sunz,rm2(nl))
        call linterp(rsunz1,rsunz2,rim(ist,isu-1,nl,3),
     *       rim(ist,isu,nl,3),sunz,rm3(nl))
       enddo
       go to 10
      endif
c  Special case 3: sun zenith = 0 -- Linearly interpolate in scz
      if (isu .eq. 1)then
       rscz1 = (ist-1)*2.0 - 2.0
       rscz2 = (ist-1)*2.0
       do nl = 1,nlt
        call linterp(rscz1,rscz2,rim(ist-1,isu,nl,1),
     *       rim(ist,isu,nl,1),scz,rm1(nl))
```

```
      call linterp(rscz1,rscz2,rim(ist-1,isu,nl,2),
     *       rim(ist,isu,nl,2),scz,rm2(nl))
      call linterp(rscz1,rscz2,rim(ist-1,isu,nl,3),
     *       rim(ist,isu,nl,3),scz,rm3(nl))
     enddo
     go to 10
    endif
c
c  Final possibility: no angles are 0 -- bi-linearly interpolate
    rscz1 = (ist-1)*2.0 - 2.0
    rscz2 = (ist-1)*2.0
    rsunz1 = (isu-1)*2.0 - 2.0
    rsunz2 = (isu-1)*2.0
    do nl = 1,nlt
     call bilin(rscz1,rscz2,rsunz1,rsunz2,rim(ist-1,isu-1,nl,1),
     *      rim(ist,isu-1,nl,1),rim(ist-1,isu,nl,1),rim(ist,isu,nl,1),
     *      scz,sunz,rm1(nl))
     call bilin(rscz1,rscz2,rsunz1,rsunz2,rim(ist-1,isu-1,nl,2),
     *      rim(ist,isu-1,nl,2),rim(ist-1,isu,nl,2),rim(ist,isu,nl,2),
     *      scz,sunz,rm2(nl))
     call bilin(rscz1,rscz2,rsunz1,rsunz2,rim(ist-1,isu-1,nl,3),
     *      rim(ist,isu-1,nl,3),rim(ist-1,isu,nl,3),rim(ist,isu,nl,3),
     *      scz,sunz,rm3(nl))
    enddo
c
10  continue
c  Compute I
    do nl = 1,nlt
     ri0 = rm1(nl) + rm2(nl)*cos(1.0*delphi)
     *           + rm3(nl)*cos(2.0*delphi)
c  Correct for non-standard pressure
     thr(ia,ja,nl) = thray0(nl)*pres(ia,ja)/p0
     rri(ia,ja,nl) = ri0*((1.0-exp(-thr(ia,ja,nl)/coscz))/
     *               (1.0-exp(-thray0(nl)/coscz)))
    enddo
    go to 200
100 continue
c  Error trap for angles out of range
    do nl = 1,nlt
     rri(ia,ja,nl) = 0.0
     thr(ia,ja,nl) = 0.0
    enddo
200 continue
c  End anchor point array loop
      enddo
     enddo
c
c
c  Now spline to create full set of ri's
c
c   Along scan line
c   Create real position arrays from integer arrays
     do ia = 1,iarr
      xpos1(ia) = ianc(ia)
```

```
      enddo
c
c   Loop through major array, along scan lines
      do ja = 1,jarr
       j = janc(ja)
       do nl = 1,nlt
     do ia = 1,iarr
      rrri(ia) = rri(ia,ja,nl)
      thrr(ia) = thr(ia,ja,nl)
     enddo
c  Spline along scan lines
     call pspline(iarr,xpos1,rrri,b1,c1,d1)
     call pspline(iarr,xpos1,thrr,b2,c2,d2)
     ia = 1
     do iscan = 1,imax
      x = iscan
      i = iscan
c  If anchor points are zero, entire scan line section is zero
      if (i .ge. ianc(ia) .and. i .lt. ianc(ia+1))then
       if (rrri(ia) .le. 0.0 .or. rrri(ia+1) .le. 0.0)then
        ri(i,j,nl) = 0.0
        thray(i,j,nl) = 0.0
       else
        ri(i,j,nl) = seval(iarr,x,xpos1,rrri,b1,c1,d1)
        thray(i,j,nl) = seval(iarr,x,xpos1,thrr,b2,c2,d2)
       endif
      else
       ia = ia+1
       ri(i,j,nl) = rrri(ia)
       thray(i,j,nl) = thrr(ia)
      endif
     enddo
       enddo
      enddo
c
c
c  Now spline along track
      do ja = 1,jarr
       xpos2(ja) = janc(ja)
      enddo
c   Spline along track, along scan
      do iscan = 1,imax
       i = iscan
       do nl = 1,nlt
     do ja = 1,jarr
      j = janc(ja)
      rji(ja) = ri(i,j,nl)
      tjh(ja) = thray(i,j,nl)
     enddo
     call pspline(jarr,xpos2,rji,e1,f1,g1)
     call pspline(jarr,xpos2,tjh,e2,f2,g2)
     ja = 1
     do j = 1,jmax
      y = j
```

```
 if (j .ge. janc(ja) .and. j .lt. janc(ja+1))then
  if (rji(ja) .le. 0.0)then
   ri(i,j,nl) = 0.0
   thray(i,j,nl) = 0.0
  else
   ri(i,j,nl) = seval(jarr,y,xpos2,rji,e1,f1,g1)
   thray(i,j,nl) = seval(jarr,y,xpos2,tjh,e2,f2,g2)
  endif
 else
  ja = ja+1
  ri(i,j,nl) = rji(ja)
  thray(i,j,nl) = tjh(ja)
 endif
enddo
  enddo
 enddo
c
 return
 end
```

```
      subroutine atmot(rad,sunz,suna,scz,sca,lam,Fo,FFo,rlt,
     * rlr,thray,thoz,ws,rlwn,eta,rla)
c
c  Atmospheric correction algorithm for MODIS-T sizing estimates.
c  Computes on a per-pixel basis.
c
      parameter(nlt=32)
      integer lam(nlt)
      real Fo(nlt),FFo(nlt),thoz(nlt),thray(nlt)
      real t(nlt),t0(nlt),eta(3)
      real rlt(nlt),rlr(nlt),rlg(nlt),rla(nlt),rlwn(nlt)
      double precision pi,rad
c
      pi = dacos(-1.0D0)
      cosunz = cos(sunz/rad)
      coscz = cos(scz/rad)
c
c  Set up glitter parameters
      isg = 0
c   Re-orient satellite azimuth to be with respect pixel
      tsca = sca+180.0
      if (tsca .gt. 360.0)then
       rsca = tsca - 360.0
      else
       rsca = tsca
      endif
c    Glint angles
      delphi = (rsca - suna)/rad
      sinfn = sin(scz/rad)*sin(sunz/rad)*cos(delphi)
      cos2w = coscz*cosunz + sinfn
      wl = acos(cos2w)
      w = 0.5*wl          !1/2 the angle between sun and sat
      arg = (coscz+cosunz)/(2.0*cos(w))
      thetan = acos(arg)
      thn = thetan*rad
c    Wave slope distribution
      targ = tan(thetan)*tan(thetan)
      sig2 = 0.003 + 0.00512*ws
      pws = 1.0/(pi*sig2) * exp(-targ/sig2)
      sgfac = 1.0/(coscz*cosunz*cosunz*cosunz*cosunz)
c  Glint flag
      t0(7) = exp(-(0.5*thray(7)+thoz(7))/cosunz)
      rlgn = Fo(7)*t0(7)/4.0*sgfac*0.021*pws
      if (rlgn .gt. 1.0)then
       isg = 1
       do nl = 1,nlt
        rlwn(nl) = 0.0
       enddo
       go to 100
      endif
c
c  Compute diffuse transmittance
      if (thray(1) .ne. 0.0)then       !angles flag
       do nl = 1,nlt
```

```
        t(nl) = exp(-(0.5*thray(nl)+thoz(nl))/coscz)
        t0(nl) = exp(-(0.5*thray(nl)+thoz(nl))/cosunz)
c
c  Compute glint
        rlgn = Fo(nl)*t0(nl)/4.0*sgfac*0.021*pws
        rlg(nl) = rlgn*t(nl)
       enddo
      else
       do nl = 1,nlt
        rlwn(nl) = 0.0
       enddo
       go to 100
      endif
c
c  Compute aerosols
      call aerot(lam,FFo,rlt,rlr,rlg,rla,eta)
c
c  Compute water-leaving radiances
      do nl = 1,nlt
       if (rla(nl) .ge. 0.0)then     !check for negative radiance
        tlw = rlt(nl) - rlr(nl) - rlg(nl) - rla(nl)
        rlw = tlw/t(nl)
        rlwn(nl) = rlw/(cosunz*t0(nl))
       else
        rlwn(nl) = 0.0
       endif
      enddo
c
c  Final check for negative radiances
      do nl = 1,nlt
       if (rlwn(nl) .lt. 0.0)then
        rlwn(nl) = 0.0
       endif
      enddo
c
100   continue
      return
      end
```

```
      subroutine aerot(lam,FFo,rlt,rlr,rlg,rla,eta)
c
c  Gordon's method for computing aerosols for MODIS-T.
c
      parameter(nlt=32)
      integer lam(nlt)
      real FFo(nlt),rlt(nlt),rlr(nlt),rlg(nlt),rla(nlt),eta(3)
c
c  Compute aerosols
      do nl = nlt,18,-1
       rla(nl) = rlt(nl) - rlr(nl) - rlg(nl)
      enddo
      s1 = rla(24)/rla(nlt)
      eps1 = s1*FFo(nlt)/FFo(24)
      tmp = alog10(eps1)
      rlam = lam(24)
      tmp2 = alog10(rlam/lam(nlt))
      eta(1) = tmp/tmp2
      s2 = rla(18)/rla(nlt)
      eps2 = s2*ffo(nlt)/FFo(18)
      tmp = alog10(eps2)
      rlam = lam(18)
      tmp2 = alog10(rlam/lam(nlt))
      eta(2) = tmp/tmp2
      eta(3) = 0.5*(eta(1)+eta(2))
      do nl = 1,17
       rlam = lam(nl)
       eps = (rlam/lam(nlt))**eta(3)
       s = eps*FFo(nl)/FFo(nlt)
       rla(nl) = s*rla(nlt)
      enddo
c
      return
      end
```

```
      subroutine tdata2(lam,Fobar,oza,thray)
c
c  Opens and reads MODIS-T light data, and creates Rayleigh optical
c  thickness data, specifically for the sizing estimates.
c
      parameter(nlt=1)
      character*50 title
      integer lam(nlt)
      real Fobar(nlt),oza(nlt),thray(nlt)
c
c  Light Data
      write(6,*)'Reading light data...'
      open(4,file='ett.dat',status='old',form='formatted')
      read(4,'(a50)')title
      read(4,'(a50)')title
      read(4,'(a50)')title
      do nl = 1,nlt
       read(4,10)lam(nl),Fbar,oza(nl)
       Fobar(nl) = Fbar*1000.0      !convert to mW/cm2/um
      enddo
      close(4)
c
c  Compute Rayleigh
      ex2 = -2.0
      ex4 = -4.0
      do nl = 1,nlt
       rlam = lam(nl)/1000.0
       arg1 = 1.0 + 0.0113*rlam**ex2 + 0.00013*rlam**ex4
       thray(nl) = arg1*0.008569*rlam**ex4
      enddo
c
c
10    format(i4,2f9.4)
20    format(i4,f10.4,f10.5,f10.3)
      return
      end
```

```
      subroutine bilin(x0,x1,y0,y1,f00,f10,f01,f11,x,y,b)
c
c  Bi-linear interpolation.  Origin at lower-left.
c  Requires that x, y position of the variable in question is c c
c  known.
c
      dx = x1-x0
      dy = y1-y0
      dxdy = dx*dy
c
      b1 = f00*(y1-y)*(x1-x) + f10*(y1-y)*(x-x0)
     *     + f01*(y-y0)*(x1-x) + f11*(y-y0)*(x-x0)
      b = b1/dxdy
c
      return
      end
```

```
      subroutine linterp(x0,x1,f0,f1,x,b)
c
c  Linear interpolation.  Origin at lower-left corner.
c
      dx = x1-x0
      b = (f0*(x1-x) + f1*(x-x0))/dx
c
      return
      end
```

```
      SUBROUTINE PSPLINE(N,X,Y,B,C,D)
      INTEGER  N
      REAL X(N),Y(N),B(N),C(N),D(N)
C
C  THE COEFFICIENTS B(I), C(I), AND D(I), I=1,2,......,N ARE
C  COMPUTED
C  FOR A CUBIC INTERPOLATING SPLINE
C
C  S(X)=Y(I) + B(I)*(X-X(I)) + C(I)*(X-X(I))**2 +
C  D(I)*(X-X(I))**3
C
C  FOR X(I) .LE. X .LE. X(I+1)
C
C  INPUT...
C
C  N=THE NUMBER OF DATA POINTS OR KNOTS(N .GE. 2)
C  X=THE ABSCISSAS OF THE KNOTS IN  STRICTLY INCREASING
C  ORDER
C  Y=THE ORDINATES OF THE KNOTS
C
C  OUTPUT...
C
C  B, C, D = ARRAYS OF SPLINE COEFFICIENTS AS DEFINED ABOVE
C
C
C  USING P TO DENOTE DIFFERENTIATION,
C
C   Y(I)=S(X(I))
C   B(I)=SP(X(I))
C   C(I)=SPP(X(I))/2
C   D(I)=SPPP(X(I))/6     (DERIVATIVE FROM THE RIGHT)
C
C   THE ACCOMPANYING FUNCTION SUBPROGRAM SEVAL CAN BE USED TO
C   EVALUATE THE SPLINE
C
      INTEGER  NM1,IB,I
      REAL T
      NM1=N-1
      IF(N.LT.2) RETURN
      IF(N.LT.3) GO TO 50
C
C   SET UP TRIDIAGONAL SYSTEM
C
C   B=DIAGONAL, D=OFFDIAGONAL, C=RIGHT HAND SIDE
C
      D(1)=X(2)-X(1)
      C(2)=(Y(2)-Y(1))/D(1)
      DO 10 I=2,NM1
       D(I)=X(I+1)-X(I)
       B(I)=2.0*(D(I-1)+D(I))
       C(I+1)=(Y(I+1)-Y(I))/D(I)
       C(I)=C(I+1)-C(I)
10    CONTINUE
C
```

```
C  END CONDITIONS. THE THIRD DERIVATIVES AT X(1) AND X(N)
C  OBTAINED FROM DIVIDED DIFFERENCES
C
      B(1)=-D(1)
      B(N)=-D(N-1)
      C(1)=0.0
      C(N)=0.0
      IF(N.EQ.3) GO TO 15
      C(1)=C(3)/(X(4)-X(2))-C(2)/(X(3)-X(1))
      C(N)=C(N-1)/(X(N)-X(N-2))-C(N-2)/(X(N-1)-X(N-3))
      C(1)=C(1)*D(1)**2/(X(4)-X(1))
      C(N)=-C(N)*D(N-1)**2/(X(N)-X(N-3))
C
C  FORWARD ELIMINATION
C
15    DO 20 I=2,N
       T=D(I-1)/B(I-1)
       B(I)=B(I)-T*D(I-1)
       C(I)=C(I)-T*C(I-1)
C
20    CONTINUE
C
C  BACK SUBSTITUTION
C
      C(N)=C(N)/B(N)
      DO 30 IB=1,NM1
       I=N-IB
30    C(I) = (C(I)-D(I)*C(I+1))/B(I)
C
C  C(I) IS NOW SIGMA(I) OF THE TEXT
C
C  COMPUTE POLYNOMIAL COEFFICIENTS
C
      B(N) = (Y(N)-Y(NM1))/D(NM1) + D(NM1)*(C(NM1)+2.0*C(N))
      DO 40 I = 1,NM1
       B(I) = (Y(I+1)-Y(I))/D(I)-D(I)*(C(I+1)+2.0*C(I))
       D(I) = (C(I+1)-C(I))/D(I)
       C(I) = 3.0*C(I)
40    CONTINUE
C
      C(N) = 3.0*C(N)
      D(N) = D(N-1)
      RETURN
C
50    B(1) = (Y(2)-Y(1))/(X(2)-X(1))
      C(1) = 0.0
      D(1) = 0.0
      B(2) = B(1)
      C(2) = 0.0
      D(2) = 0.0
      RETURN
      END
```

```
      FUNCTION SEVAL(N,U,X,Y,B,C,D)
      INTEGER N
      REAL U,X(N),Y(N),B(N),C(N),D(N)
C
C
C    THIS SUBROUTINE EVALUATES THE CUBIC SPLINE FUNCTION
C
C          SEVAL= Y(I) + B(I)*(U-X(I)) + C(I)*(U-X(I))**2 +
C           D(I)*(U-X(I))**3
C
      INTEGER I,J,K
      REAL DX
      DATA I/1/
      IF(I.GE.N) I=1
      IF(U.LT.X(I)) GO TO 10
      IF(U.LT.X(I+1)) GO TO 30
C
C    BINARY SEARCH
C
10    I=1
      J=N+1
20    K=(I+J)/2
      IF(U.LT.X(K)) J=K
      IF(U.GE.X(K)) I=K
      IF(J.GT.I+1)  GO TO 20
C
C    EVALUATE SPLINE
C
30    DX=U-X(I)
      SEVAL=Y(I)+DX*(B(I)+DX*(C(I)+DX*D(I)))
      RETURN
      END
```

SIZING OF AT-LAUNCH LAND DATA PRODUCTS

These algorithms and sizing estimates are based on information presented in MODIS Science Team member proposals, discussions at the 31 January - 2 February 1990 MODIS Science Team meeting, published literature, and empirical equations presented in the open literature, as such they are tentative, unstructured, and subject to extensive revision, yet serve as a beginning for the sizing task.

VEGETATION INDEX

Normalized Difference Vegetation Index (NDVI) calculated as ratio of reflectances, (near infrared - red)/(near infrared + red). Use MODIS-N bands 1 and 2, at 214 m spatial resolution.

Assume that each pixel in a scan is processed and that data supplied to the algorithm is land leaving radiance (atmospherically corrected to some degree), earth located, masked for land, and cloud flagged. for clouds.

Pseudo code for NDVI:

```
READ land mask, and cloud screen flag
IF land, AND cloudless THEN continue
ELSE examine next pixel
READ latitude, and solar zenith angle, and scan angle
IF latitude, AND solar zenith angle, AND scan angle within
   specified limits THEN proceed to calculate NDVI
ELSE skip calculating NDVI
READ land leaving radiances for Band 1 and Band 2
NDVI = (Band 2 - Band 1)/(Band 2 + Band 1)
IF NDVI < -1 OR > 1  set an error flag to indicate out of range
   NDVI values (NDVI ranges from -1.0 - 1.0)
WRITE the NDVI values to the data cube
```

Pseudo code for weekly NDVI composite:

```
READ previous day NDVI from memory
IF current NDVI > previous NDVI in this pixel THEN replace with
   current NDVI
ELSE compare NDVIs for next pixel
WRITE updated NDVI image to memory
```

Operation	#	# pixels/scan	# operations/scan
+	1	202496	202496
-	1	202496	202496
÷	1	202496	202496
READ	8	202496	1619968
IF	7	202496	1417472
WRITE	3	202496	607488
Total operations			4254416

MFlop/scan 4.3

SNOW COVER

Snow cover algorithm is proposed to produce a weekly, global map of snow cover. Snow cover algorithm may require data from reflective and thermal bands, and include a discrimination between snow and clouds. Assume that snow cover determined for each pixel, nominal 1 km spatial resolution, in each orbit, although geographic and temporal flags may be set and used to limit processing during different seasons. Presence of snow cover determined through reflective and thermal properties of snow.

Pseudo code for snow cover:

```
  READ date and geographic location
     (geographic location e.g. lat. long. elev.)
  IF date and geographic location indicate possible snow THEN
     proceed
  ELSE exit snow cover algorithm
  READ in reflective bands (2) and thermal bands (3)
  IF check reflective bands, visible and near infrared for
     reflectance typical of snow THEN flag as snow
  ELSE not snow
  IF snow THEN read surface brightness temperatures, bands 21,31,32
     use split window technique to correct for water vapor effect
     on temperature
discriminate between snow and clouds
  IF brightness temperature ≤ specified temperature THEN flag as
     cloud   (maybe other thermal and reflectance checks)
  ELSE snow
  WRITE to pixel snow map for scan
degrade spatial resolution of the pixel snow map to say 10 km
  Avgsnow = (4 across track pixels + 4 along track pixels)/16
  WRITE the 10 km snow cover map to memory
average/manipulate the 10 km snow cover maps to produce weekly 10
     km snow cover map
```

Operation	#	# pixels/scan	# operations/scan
+	16	12656	202496
÷	1	12656	12656
READ	9	12656	113904
IF	9	12656	113904
WRITE	2	12656	15312
split window (unspecified)	7	12656	88592
Total operations for one band			546864
MFlop/scan 0.55			

LAND COVER TYPE

The nature of the land cover type map is not well defined at this time. It seems that its production will require the categorization of earth vegetation into "natural units" to be utilized as a as a base map to monitor vegetation dynamics and studying interactions of vegetation with climate and geophysical parameters. Earth's vegetation has been categorized in the past and world land cover maps do exist, yet the MODIS land team scientists suggest that existing global land cover types are not the most suitable for proposed MODIS studies and that new land cover types need to be defined.

Pseudo code for land cover type:

Vector or raster mapping of defined "natural units" of land cover
Assume a cover type is entered for each pixel, raster mapping

Operation	#	# pixels/scan	# operations/scan
READ			
214 m	1	202496	202496
428 m	1	50624	50624
856 m	1	12656	12656
Total for one band			12656-202496
depends on resolution			

LAND SURFACE TEMPERATURE

To derive land-surface temperatures from satellite observations of terrestrial radiation requires accurate knowledge of land-surface boundary conditions, such as emissivity, topography, and surface heterogeneity. It is also required that corrections for atmospheric effects in the radiance signal received by the satellite be applied.

The effects of the uncertainties in both land-surface boundary conditions and atmospheric conditions in the satellite data must be understood and quantified before the selection of the algorithms, to obtain land-surface temperatures, can be made. Availability of the variables which describe both these conditions has a determination on algorithm selection. The effect of uncertainty in surface emissivity on the accuracy of surface temperature is very strong, e.g. the error in land-surface temperature associated with a 1 % inaccuracy in surface emissivity was larger than the error in the atmospheric correction model (Wan and Dozier, 1989). When the emissivity of the land surface is not known, corrections for atmospheric effects and surface emissivity variations must be made concurrently, and necessitates that some assumptions be made about the spectral variation in surface emissivity. At this time there are few accurate data on spectral emissivity for many land covers; this is a fundamental block to the accurate measurement of land-surface temperature (Wan and Dozier, 1989).

In order to give an estimate of the sizing of the algorithm for land-surface temperatures we consider the following cases:

- Case 1 : Known land-surface emissivities.

Assuming that spectral emissivity values for the land surface are known temperature can be determined by statistical analysis of a computer emissivity-radiance simulation model (Wan and Dozier, 1989). In this model three LANDSAT thermal infrared bands were used to infer a blackbody-radiance in channel 4, corrected for atmosphere and surface emissivity. The land-surface temperature is then obtained through inversion of Planck's equation. It takes 70 operations to determine the land-surface temperature.

-Case 2 : Unknown land-surface emissivities.

When the emissivity of the land surface is not known, the problem of determining surface temperature may be approached by focussing on broad categories of emissivity, and assume that thermal data are combined with other remotely sensed data to classify the surface into these broad emissivity categories. Wan and Dozier (1989) developed a regression analysis to analyze results in two broad emissivity groups, into which five land surfaces; clay, fine sands, coarse sands, tree leaf, and snow were separated. Group 1 included clay and sands and represented surfaces with spectral emissivity features. Group 2 included tree leaves and snow and represented spectrally flat surfaces. The resulting equations involve a quadratic expression of 4 thermal bands and thus contain more than 30 terms. However, they contain no explicit emissivity terms.
No accurate sizing estimate, in terms of number of operations, is possible at this time but, speculate on the order of 700 operations per pixel.

Wan, Z. and Dozier, J. 1989. Land-surface temperature measurements from space:physical principles and inverse modeling. IEEE Trans. Geosci. Remote Sens. 27:268-278.

Pseudo code for land surface temperature
none developed at this time because of uncertainties involved in determining surface temperature from satellite data.

Operation	#	# pixels/scan	# operations/scan
Case 1		12656	890000
Case 2		12656	8900000

Total operations per band
MFlop/scan 0.89 - 8.9

THERMAL ANOMALIES

Product for use in detecting fires occurrence and extent. At present it appears that detection of thermal anomalies requires checking surface temperatures, or radiances, for those out of normal range. Done for every pixel, MODIS-N bands 21, 31, and 32, with day and night scans required. Option appears to exist for using either radiance or calculated surface temperature to detect anomalies.

Pseudo code for thermal anomalies:

```
  READ pixel lat. long. land/ocean flag
  READ in surface temperatures from Surface Temperature Product
OR READ in radiances, bands 21, 31, 32
  READ in atmospheric water vapor
correct radiances for water vapor absorption in each band e.g.
  DO for 3 bands
     Tw = EXP(-0.2385*a_wλ*W*M/(1+20.07*a_wλ*W*M)^0.45)
     Twrad = Rad * EXP(-Tw/COSθ)
  IF radiances unusual flag as anomaly
or calculate a temperature from the radiance value
if surface temperatures were used then check them for anomalies
  IF surface temperature is outside normal expected values THEN
     flag pixel
  ELSE next pixel
  WRITE to fire flags data layer in scan data cube
```

Operation	#	# pixels/scan	# operations/scan
+	3	12656	37968
*	21	12656	265776
÷	6	12656	75936
EXP	9	12656	113904
COS	3	12656	37968
READ	8	12656	101248
IF	3	12656	37968
WRITE	1	12656	12656
Total operations per band			663424

MFlop/scan 0.66

LAND LEAVING RADIANCE (Atmospheric Corrections)

Atmospheric corrections required to produce a land leaving radiance from the measured at satellite radiance for land surfaces are very difficult to define for MODIS-N at the present time. Presented here in crude, segmented, and incomplete form are calculations that may be used to correct for atmospheric effects. The purpose here is to begin building an estimate of the processing requirements for atmospheric corrections.

Pseudo code for land leaving radiance:

```
  READ in the satellite radiance value for bands 1, 2 and others
     tbd
  READ in for each pixel, solar zenith angle, look zenith angle and
     look azimuth angle
expression for radiance at the satellite may take the form proposed
by Y. Kaufman of
  L_s = L_o + F_d*T*ρ/(1-s*ρ)
calculated for each band and sun-surface-sensor geometry

  READ in ρ from geographic referenced known surface target, or
     determine from image itself
  READ in F_d the extraterrestrial irradiance, wavelength dependent,
     possibly supplied from look up tables

calculate the transmittance functions for T
calculate Rayleigh transmittance coefficient
  READ in surface pressure data
  Ray0 = 0.008569*λ^-4 *(1+0.0113*λ^-2 + 0.00013*λ^-4)
  Raytrans = (surface pressure/1013.25) * Ray0
  L_sRay = F_d*ρ*EXP^-(Raytrans/COSθ)

calculate ozone transmittance coefficient
  READ in atmospheric ozone amount
  READ in solar zenith angle
  Ozonetrans = EXP(-a_oλ * O_3 * M_0)
  M_0 = (1+h_o/6370)/(COS^2(Z)+2*h_o/6370)^0.5
  L_sOz = F_d*ρ*EXP^-(Ozonetrans/COSθ)

calculate water vapor transmittance coefficient
  READ in precipitable water
  Watertrans = EXP(-0.2385*a_wλ * W*M/(1+20.07*a_wλ*W*M)^0.45)
  L_sWater = F_d*EXP^(-Watertrans/COSθ)

calculate aerosol transmittance and scattering
  calculate transmittance of aerosols for each band
  τ_aλ = EXP(-β*λ^-α * m_a)
  β = (0.55)^α(3.912/Vis - 0.01162)[0.02472(Vis-5)+1.132]
  calculate scattering function
  Id = Io*COS(θ)τmax*[Fc*ω(1-τa)τray]
     (some further solution required)
  WRITE calculated land leaving to file
```

Assume 214 m resolution

Operation	#	# pixels/scan	# operations/scan
+	7	202496	1417452
-	4	202496	809984
*	33	202496	6682368
÷	9	202496	1822464
EXP	13	202496	2632448
COS	5	202496	1012480
READ	11	202496	2227456
IF	0 (will be some with error trapping)		
WRITE	1	202496	202496
Total operations for <u>one</u> band			16807148
MFlop/scan	16.8		

MODIS Data Packet Structure: Instrument-to-Ground Link

(Preliminary)

In September of 1989 the MODIS Data Team prepared an informal recommendation on data packet structure for use on the MODIS-T downlink. The primary consideration governing the recommendation was a perceived need to selectively route MODIS data to various locations within the EosDIS. Delivery of unprocessed (i.e. Level-0) data to destinations needing only data from certain spectral bands is facilitated if each packet contains only data from a single spectral band, so that whole packets can be routed to a destination without the need to first decommutate band-interleaved data and select the required data items.

During the MODIS Science Team Meeting from January 31 through February 2, it was pointed out that a spectrally-oriented packet structure may increase processing losses for derived products that require the simultaneous availability of data from several spectral bands to complete processing. Since potential data losses were not considered in the earlier data packet analysis, this document extends that analysis to include consideration of data losses. At present some of the specifics of data link performance are not known (to this author, at least), so that we shall begin with a development of the mathematics of data loss. Specific recommendations will be reserved for a follow-on report to be written once additional information is developed and "best estimate" performance parameters are available for the Eos downlink.

<u>Probability of data loss.</u>

For the moment, suppose that MODIS data loss occurs only at the packet level, so that either a complete data packet is available for use or all information in that packet is lost and not available for use. (The appropriateness of this assumption is one of the issues to be resolved with the Eos data link designers.) Suppose that the probability that a data packet is lost is given by ϵ_{packet}. If processing for given pixel of a given product requires the simultaneous availability of data for n spectral bands, and these data are distributed in n separate packets, then a loss of any of the required packets can prevent the completion of processing for the pixel in question. Processing can be prevented if any single packet is missing, if any two of the required packets are missing (in the n required bands), if three are missing, etc. A direct approach to the problem of computing the probability that a pixel cannot be processed would require the enumeration of all possible ways that processing can be blocked and the assignment of a probability to each event. Fortunately, a mathematical trick can considerably simplify the computations.

Consider the complementary event to the event that processing is blocked, i.e. consider the probability that all required bands are

present and processing can proceed. The probability that the first required band is present is given by

$$[1 - \epsilon_{packet}]$$

and assuming that packet losses are independent events, the probability that all n of the required packets are available is

$$[1 - \epsilon_{packet}]^n$$

But we are interested in the probability that a pixel cannot be processed, ϵ_{pixel} say, and this event is the complement of the event that all required packets are available, so that

$$\epsilon_{pixel} = 1 - [1 - \epsilon_{packet}]^n \qquad (1)$$

This is the desired relationship relating pixel loss rate to packet loss rate when n packets must simultaneously be available before pixel processing can be completed.

General Nature of Results and Equation Reduction for Small ϵ_{packet}

Figure 1 is a plot showing pixel loss (in percent) as a function of packet loss (also in percent) for n = 1,2,3,5,7,10. For n > 1, and high probability of packet loss, the curves show a very high probability that at least one of the required n packets will be lost, and processing cannot be completed. For a real data system, the probability of packet loss is (hopefully) very small. For $\epsilon_{packet} << 1$, the binomial theorem can be applied to the right hand side of Equation 1, and

$$\epsilon_{pixel} = n\ \epsilon_{packet} + o(\epsilon^2_{packet});\ \epsilon_{packet} << 1 \qquad (2)$$

where $o(\epsilon^2_{packet})$ indicates terms of order ϵ^2_{packet} and above. To the order of this approximation, pixel loss is a linear function of packet loss and the loss is directly proportional to n, i.e. the number of pixels lost if n packets are required to complete processing is n times the number lost if all information required to complete processing is available in a single packet and simultaneous data availability is not required. This is the basic result that must be applied to the problem of data packet structure. It is thought that the packet loss rate for Eos will place operation within the linear region surrounding the origin in Figure 1. For reference, an expanded view of this region is shown in Figure 2. The line slopes in this figure are nearly identical to the value of n applied.

Products Listed by Number of Bands Required.

Besides the value of ϵ_{packet} for Eos (yet to be determined), the expected pixel loss will depend on n for the specific product in question. TABLE 1 lists estimated number-of-band requirements for MODIS ocean, land, and atmospheric products. It is apparent that more than a few bands are required for several MODIS products, and, if ϵ_{packet} is not very small for the Eos downlink, additional data

loss due to single-band packetization of MODIS data could be appreciable. Specific numerical estimates will be developed in a future report.

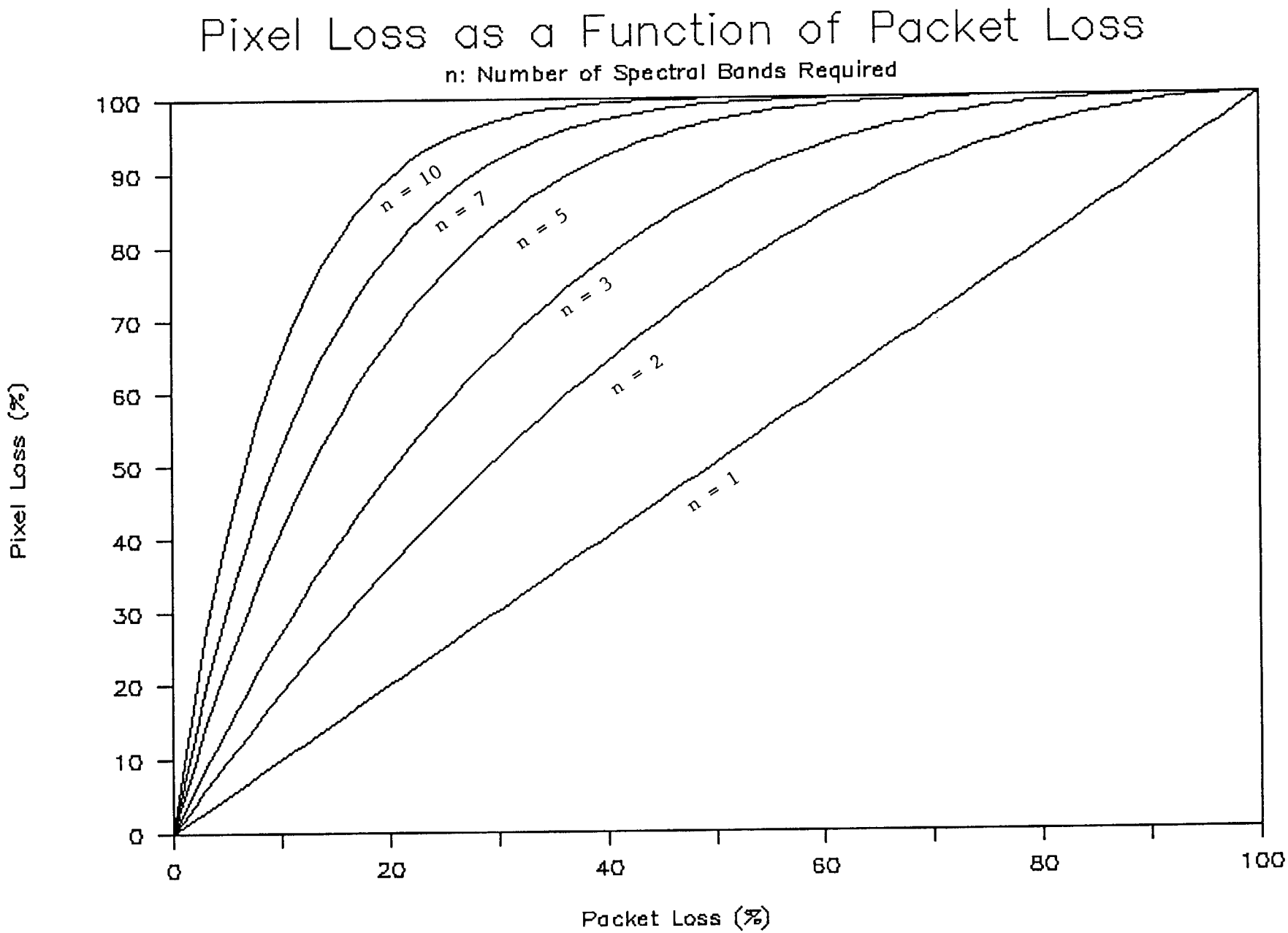


Figure 1. Pixel Loss as a Function of Packet Loss

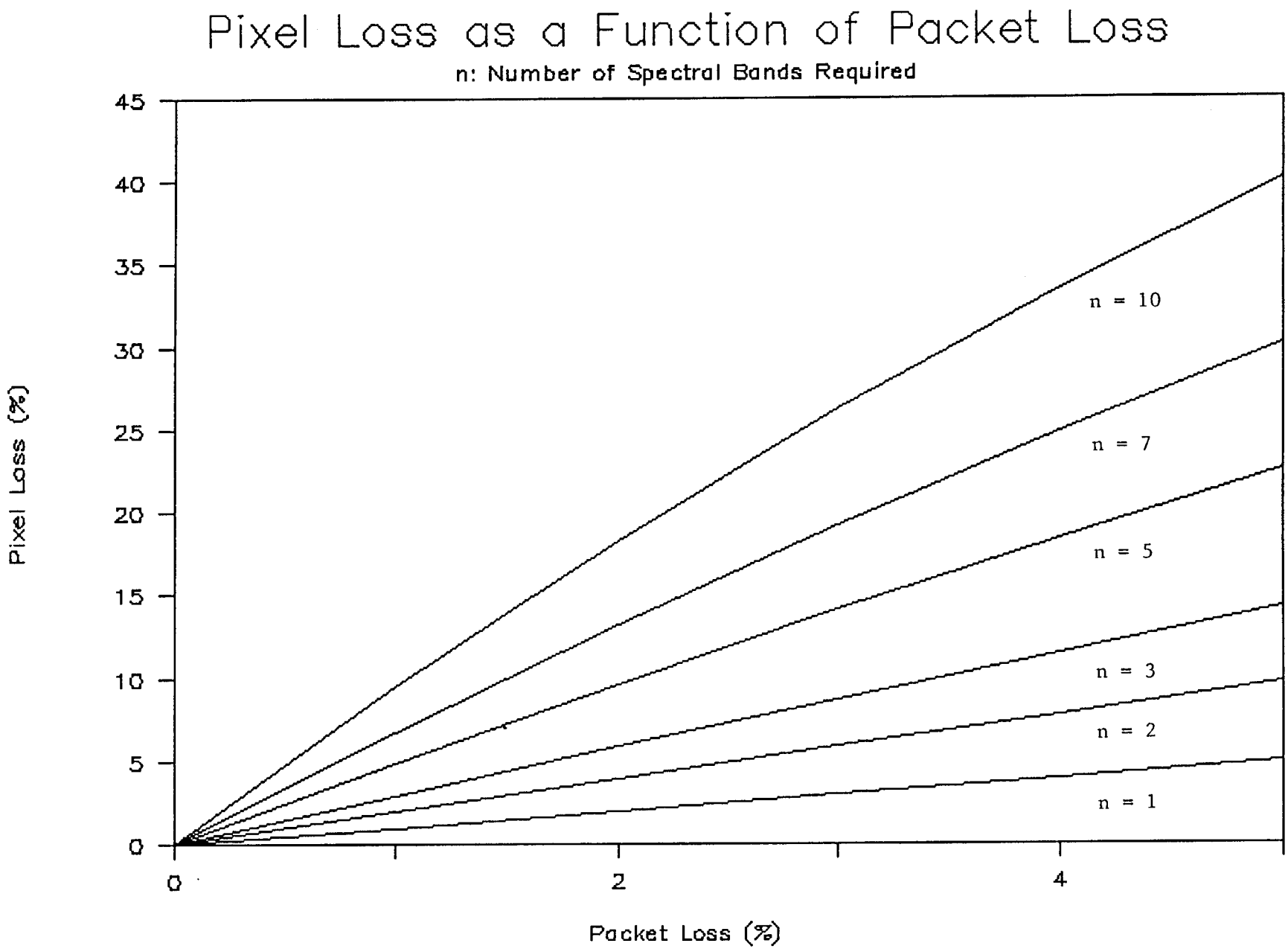


Figure 2. Pixel Loss as a Function of Packet Loss (expanded view).

TABLE 1

NUMBER OF SPECTRAL BANDS REQUIRED TO PRODUCE MODIS DATA PRODUCTS

(Current best estimates)

MODIS Ocean Data Products	
Chlorophyll A Concentrations (Case 1)	7-10
Chlorophyll A Concentrations (Case 2)	7-10
Chlorophyll A Fluoresence	4
CZCS Pigment Concentrations	6
Sea Surface Temperature	3
Water-Leaving Radiances	4
Attenuation at 490 nm	5
Detached Coccolith Concentration	4
Phycoerythrin Concentrations	6
Single Scattering Aerosol Radiances	4
Angstrom Exponents	3
Primary Production	6-10
MODIS Land Data Products	
Land-Leaving Radiances	3-8
Vegetation Index	2
Surface Temperature	6
Snow Cover	3-5
Thermal Anomalies	3
Spatial Heterogeneity	2-4
Cloud Mask, Cloud Shadow	3-6
Bidirectional Reflectance, BRDF	1
Albedo	1
Soil Brightness Index	7
MODIS Atmospheric Data Products	
Cloud Optical Thickness	3-5
Cloud Fractional Area	1-2
Cloud Effective Emissivity	4
Cloud-Top Temperature	6
Cloud-Top Pressure	4
Cloud Particle Effective Radius	3-5
Cloud Particle Thermodynamic Phase	3-5
Aerosol Optical Depth	6
Aerosol Size Distribution	6
Atmospheric Stability	6
Total Precipitable Water	6
Total Ozone	4-6
Layer Mean Temperatures	6
Layer Mean Moisture	6