

V1.0

천리안위성 2A호

Level 1B 자료 사용자 매뉴얼

2019. 5.

위성운영과



국가기상위성센터

작성 이력

연번	날짜	변경 내용(버전)	작성자/검토자
1	'19. 5. 16.	초안 작성(V1.0)	문거성/오현종



목 차

I. 개요

1. 개요	2
1.1 목적	2
1.2 약어	2
1.3 참고문헌	2
2. Level 1B 자료 산출 과정	3
2.1 관측위성	3
2.2 Level 1B 자료 가공 절차	3

II. Level 1B 자료 특성

1. 관측 센서 파장 및 해상도	5
1.1 Advanced Meteorological Imager	5
2. 관측 영역 및 주기	6
2.1 관측 타임라인	6
3. 파일 명명규칙	7
3.1 명명규칙	7
3.2 파일명 예시	8



목 차

III. Level 1B 자료 파일 구조

1. 파일 형식	10
1.1 Level 1B 자료 파일 형식	10
1.2 자료 형태	10
2. Level 1B 자료 상세 – Global Attributes	11
3. Level 1B 자료 상세 – Variables	12
4. 관측데이터	15
4.1 image_pixel_values	15
5. 데이터 품질 표식	15
5.1 데이터 품질 표식	15
6. 픽셀당 데이터 비트 개수	16
6.1 픽셀당 데이터 비트 개수	16



목 차

IV. Level 1B 자료 값 추출 방법

1. 사용자 맞춤형 영상처리도구	18
2. Panoply	20
3. 샘플코드1 처리 순서도	22
3.1 입력파일 영역	22
3.2 내부처리 영역	22
3.3 출력파일 영역	23
4. 샘플코드2 처리 순서도	24
4.1 입력파일 영역	24
4.2 내부처리 영역	24
4.3 출력파일 영역	25
5. 물리값 변환	26
5.1 Conversion table을 이용한 방법	26
6. 샘플코드1	27
7. 샘플코드2	29

붙임

붙임 1. NetCDF4	36
붙임 2. 샘플코드 구동 환경	37
붙임 3. 샘플코드1 출력파일(ASCII)	39
붙임 4. Conversion Table	40



표 목차

표 1. 약어	2
표 2. AMI, ABI, AHI 중심 파장	5
표 3. 천리안위성 2A호 관측영역 및 주기, 채널별 영상의 픽셀 수	6
표 4. 파일 명명규칙	7
표 5. 기본영상 16종 전구 영역의 파일명 예시	8
표 6. 천리안위성 2A호 Level 1B NetCDF4 파일의 자료 형태	10
표 7. 천리안위성 2A호 Level 1B 자료 상세: Global Attributes	11
표 8. 천리안위성 2A호 Level 1B 자료 상세: Variables	12
표 9. image_pixel_values 행렬	15
표 10. 데이터 품질 표식의 범위와 의미	15
표 11. 채널별 픽셀당 데이터 비트 개수	16
표 12. 샘플코드1 구동 환경 및 입출력 파일 정보	37
표 13. 샘플코드2 구동 환경 및 입출력 파일 정보	38
표 14. GK-2A AMI Calibration Coefficients and Calibration Equation	41



그림 목차

그림 1. 천리안위성 1호 및 2A호 자료 서비스 일정	3
그림 2. 위성으로부터 수신한 자료의 가공 절차	3
그림 3. 천리안위성 2A호 관측 timeline	6
그림 4. 천리안위성 2A호 Level 1B 자료의 구성	10
그림 5. 사용자 맞춤형 영상처리도구의 특징	18
그림 6. 사용자 맞춤형 영상처리도구 - image_pixel_values를 이미지로 표출	18
그림 7. 사용자 맞춤형 영상처리도구 - image_pixel_values 값 표출	19
그림 8. 사용자 맞춤형 영상처리도구 - RGB Composite 이미지 생성	19
그림 9. Panoply - 천리안위성 2A호 샘플 데이터 불러오기	20
그림 10. Panoply - image_pixel_values를 이미지로 표출	20
그림 11. Panoply - image_pixel_values 값 표출	21
그림 12. 샘플코드1 처리 순서도	22
그림 13. 샘플코드2 처리 순서도	24
그림 14. Conversion Table을 이용한 image_pixel_values의 변환	26
그림 15. NetCDF-4/HDF5 Data Model	36
그림 16. 샘플코드1 구동 디렉토리 정보	37
그림 17. 샘플코드2 구동 디렉토리 정보	38
그림 18. Conversion Table 원본 캡처	40



I

개요



1. 개요

1.1 목적

이 문서는 천리안위성 2A호의 Level 1B 자료를 사용하기 위해 필요한 기초 지식 및 파일 구조에 대한 설명과 입출력 샘플코드를 제공함을 목적으로 한다.

1.2 약어

표 1 약어

약어	의미
ABI	Advanced Baseline Imager (GOES-16/17 탑재)
AHI	Advanced Himawari Imager (Himawari-8/9 탑재)
AMI	Advanced Meteorological Imager (GK-2A 탑재)
BT	Brightness Temperature, 밝기온도
COMS	천리안위성 1호 (Communication, Ocean and Meteorological Satellite)
DQF	Data Quality Flag
ELA	Extended Local Area, 확장국지영역
FD	Full Disk, 전구영역
GEOS	geostationary view, 정지궤도위성사상
GK-2A	천리안위성 2A호 (GEO-KOMPSAT-2A: GEOstationary KOREA Multi-Purpose SATellite-2A)
GSICS	Global Space-based Inter-Calibration System
IOT	In-Orbit Test, 궤도상시험
L1B LV1B	Level 1B
LA	Local Area, 국지영역
LCC	Lambert Conformal Conic projection, 램버트정형원주사상
NetCDF	Network Common Data Form
OS	Operation System, 운영체제

1.3 참고문헌

기상청. (2018.4). 천리안위성 2A호 전처리 자료 형식(안).

한국항공우주연구원. (2018.4.20). GEO-KOMPSAT-2 INR Software Output Image ICD.

2. Level 1B 자료(채널별 기본 관측 자료) 산출과정

2.1 관측위성

천리안위성 2A호는 천리안위성 1호(COMS)의 기상관측 역할을 승계하는 차세대 정지궤도 기상위성으로 2018년 12월 5일에 발사되었다. 발사 후 궤도상시험(IOT)을 거쳐 2019년 7월부터 기본영상 및 합성영상 자료를 제공할 예정이다. 그리고 천리안위성 1호(COMS)는 2020년 3월까지 운영될 예정이며, 그 자료는 2019년 12월까지 제공될 예정이다.

그림 1 천리안위성 1호 및 2A호 자료 서비스 일정



2.2 Level 1B 자료 가공 절차

GK-2A는 총 16개의 관측 채널을 가지고 있다. 각 채널별로 관측한 자료는 지상으로 송신되어 일련의 처리 과정(전처리)을 거친다. 전처리가 완료되면 각 16개 채널별로 L1B 기본영상 자료가 생산된다.

L1B 자료는 기상센서가 관측한 원시자료에 복사 및 위치보정을 수행하여 정확도를 확보한, 사용자가 실제 사용이 가능한 자료이다. L1B 자료는 관측값 보정자료(GSICS)와 자료의 품질관련 정보를 파일 내에 포함하고 있다.

그림 2 위성으로부터 수신한 자료의 가공 절차





II

Level 1B 자료 특성



1. 관측 센서 파장 및 해상도

1.1 Advanced Meteorological Imager (AMI)

GK-2A의 AMI 센서는 16개의 채널을 가지고 있다. 3번 채널은 0.5km, 1, 2, 4번 채널은 1km, 나머지 채널은 2km의 공간해상도를 지닌다. 중심파장 및 ABI/AHI와의 비교는 표2에서 확인할 수 있다.

표 2 AMI, ABI, AHI 중심 파장(Center wavelength)

AMI (Resolution)		Center wavelength (μm)	ABI	AHI
1 blue		0.47 (1km)	0.47	0.46
2 green		0.511 (1km)		0.51
3 red		0.64 (0.5km)	0.64	0.64
4		0.856 (1km)	0.865	0.86
5		1.38 (2km)	1.378	
6		1.61 (2km)	1.61	1.6
			2.25	2.3
7		3.830 (2km)	3.90	3.9
8		6.241 (2km)	6.185	6.2
9		6.952 (2km)	6.95	7.0
10		7.344 (2km)	7.34	7.3
11		8.592 (2km)	8.50	8.6
12		9.625 (2km)	9.61	9.6
13		10.403 (2km)	10.35	10.4
14		11.212 (2km)	11.2	11.2
15		12.364 (2km)	12.3	12.3
16		13.31 (2km)	13.3	13.3

2. 관측 영역 및 주기

2.1 관측 타임라인

GK-2A는 10분간 전구 영역을 1회, 확장 국지 영역을 5회, 국지 영역을 5회 관측한다. 국지 영역은 태풍, 화산폭발 등의 현상이 발생한 특정지역의 관측에 이용될 수 있다.

그림 3 천리안위성 2A호 관측 timeline

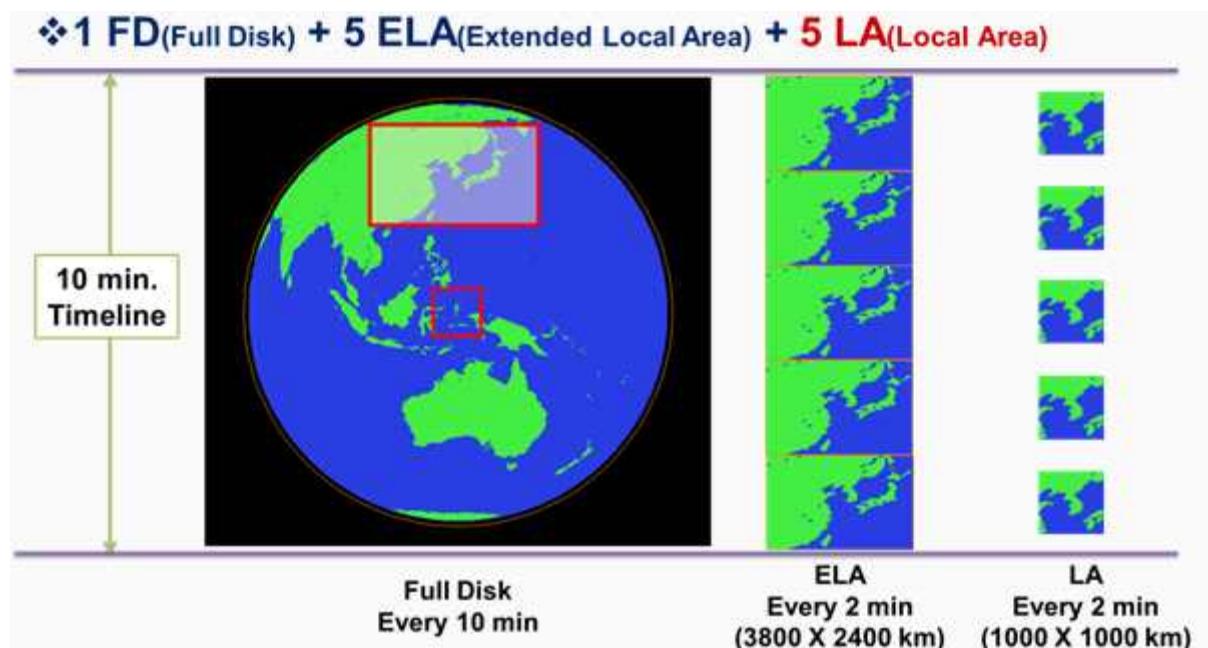


표 3 천리안위성 2A호 관측영역 및 주기, 채널별 영상의 픽셀 수

관측영역		Map Projection	Timeline 당 관측횟수	관측 주기	일 관측횟수	채널 번호	픽셀 수	
							동서방향	남북방향
전구 영역 (FD)	고정 영역	GEOS	1	10분	143	3	22,000	22,000
						1,2,4	11,000	11,000
						5~16	5,500	5,500
확장 국지 영역 (ELA)	고정 영역	LCC	5	2분	715	3	7,600	4,800
						1,2,4	3,800	2,400
						5~16	1,900	1,200
국지 영역 (LA)	유동 영역	GEOS	5	2분	715	3	2,000	2,000
						1,2,4	1,000	1,000
						5~16	500	500

- 일 관측횟수는 위성 관제일정에 따라 변동될 수 있음 (휠오프로딩 시 관측 미수행, 10분/1일)

3. 파일 명명규칙

3.1 명명규칙

기본형태 : **gk2a_ami_le1b_BAND_scnRESpr_yyyyymmddhhmn.nc**

- 파일명에서 채널, 관측영역, 도법, 관측시간 등에 따라 이탤릭체만 달라지고, 볼드체는 고정

표 4 파일 명명규칙

문자	상세설명		
<i>BAND</i>	채널&파장	가시 채널	vi004 vi005 vi006 vi008
		근적외 채널	nr013 nr016
		단파적외 채널	sw038
		수증기 채널	wv063 wv069 wv073
		적외 채널	ir087 ir096 ir105 ir112 ir123 ir133
		전구 영역	fd
		확장 국지 영역	ela
		국지 영역	la
		확장 북반구 영역	enh
		아시아 영역	ea
<i>scn</i>	관측영역	한반도 영역	ko
		태풍 감시 영역	tp
		0.5km	005
<i>RES</i>	공간해상도	1km	010
		2km	020
		GEOS projection	ge
<i>pr</i>	도법	LCC projection	lc
<i>yyyy</i>	관측시작시간(년)		
<i>mm</i>	관측시작시간(월) (01 - 12)		
<i>dd</i>	관측시작시간(일) (01 - 31)		
<i>hh</i>	관측시작시간(시) (00 - 23)		
<i>mn</i>	관측시작시간(분) (00 – 59)		

3.2 파일명 예시

아래는 전구 영역에 해당하는 기본 영상 16종의 파일명 예시이다.

표 5 기본영상 16종 전구 영역의 파일명 예시

파일명 예시	해상도		비고
	공간	픽셀수	
gk2a_ami_le1b_vi004_fd010ge_201903010900.nc	1km	11000 x 11000	CH1
gk2a_ami_le1b_vi005_fd010ge_201903010910.nc	1km	11000 x 11000	CH2
gk2a_ami_le1b_vi006_fd005ge_201903010920.nc	0.5km	22000 x 22000	CH3
gk2a_ami_le1b_vi008_fd010ge_201903010930.nc	1km	11000 x 11000	CH4
gk2a_ami_le1b_nr013_fd020ge_201903010940.nc	2km	5500 x 5500	CH5
gk2a_ami_le1b_nr016_fd020ge_201903010950.nc	2km	5500 x 5500	CH6
gk2a_ami_le1b_sw038_fd020ge_201903011000.nc	2km	5500 x 5500	CH7
gk2a_ami_le1b_wv063_fd020ge_201903011100.nc	2km	5500 x 5500	CH8
gk2a_ami_le1b_wv069_fd020ge_201903011200.nc	2km	5500 x 5500	CH9
gk2a_ami_le1b_wv073_fd020ge_201903011300.nc	2km	5500 x 5500	CH10
gk2a_ami_le1b_ir087_fd020ge_201903011400.nc	2km	5500 x 5500	CH11
gk2a_ami_le1b_ir096_fd020ge_201903011500.nc	2km	5500 x 5500	CH12
gk2a_ami_le1b_ir105_fd020ge_201903011600.nc	2km	5500 x 5500	CH13
gk2a_ami_le1b_ir112_fd020ge_201903011700.nc	2km	5500 x 5500	CH14
gk2a_ami_le1b_ir123_fd020ge_201903011800.nc	2km	5500 x 5500	CH15
gk2a_ami_le1b_ir133_fd020ge_201903011900.nc	2km	5500 x 5500	CH16



III

Level 1B 자료

파일 구조



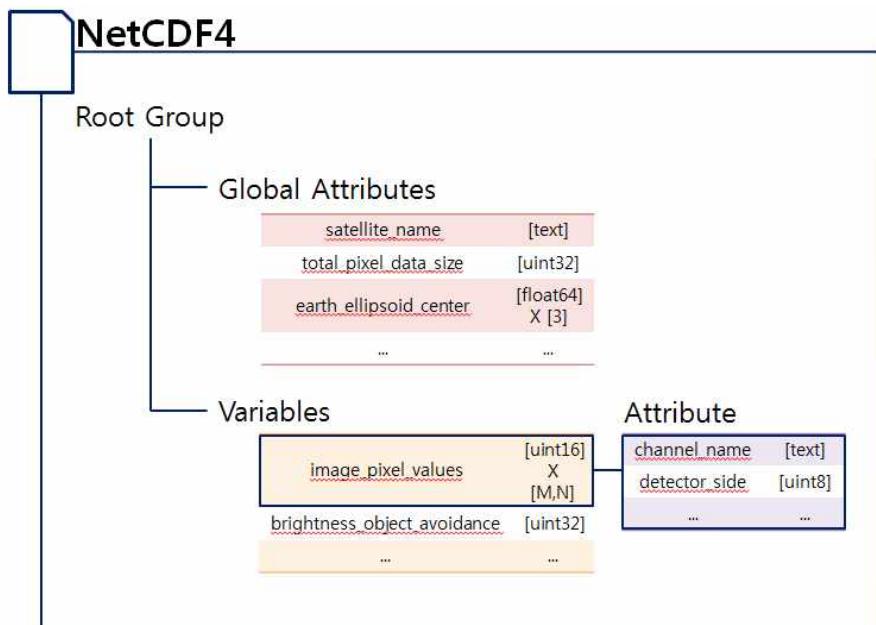
1. 파일 형식

1.1 Level 1B 자료 파일 형식

GK-2A L1B 자료의 파일 형식은 NetCDF4이며, 확장자는 nc이다. 파일의 내부 구조는 크게 Variables과 Global Attributes로 구성되어 있으며, 특히 Variables의 image_pixel_values에는 관측데이터가 픽셀별로 저장되어 있다.

Global Attributes와 Variables에 대한 상세한 정보는 각각 표-2와 표-3에 기재되어 있다.

그림 4 천리안위성 2A호 Level 1B 자료의 구성



1.2 자료 형태

NetCDF4 파일의 Attribute와 Variable은 다양한 자료 형태(Data Type)로 존재할 수 있다. GK-2A L1B 파일 내부의 자료 형태는 표 6에 기재되어 있다.

표 6 천리안위성 2A호 Level 1B NetCDF4 파일의 자료 형태(Data Type)

Format	Data Type	Format	Data Type
uint8	8bit unsigned integer	float64	64bit floating point
uint16	16bit unsigned integer	text	text format
uint32	32bit unsigned integer	format x [M,N]	Array form data

2. Level 1B 자료 상세 - Global Attributes

표 7 천리안위성 2A호 Level 1B 자료 상세: Global Attributes

Information Type	No	Global Attribute Name	Format	Description and Example
General	1	satellite_name	text	Name of Satellite (ex) GK-2A
	2	instrument_name	text	Name of Instrument (ex) AMI
	3	data_processing_center	text	Name of Processing Station Center (ex) SOC or NMSC
	4	data_processing_mode	text	Data processing mode (ex) operation, reprocessing
	5	channel_spatial_resolution	text	Channel spatial resolution [km] CH_1:1km, CH_2:1km, CH_3:0.5km, CH_4:1km, CH_5~16:2km (ex) 0.5
	6	channel_center_wavelength	text	Channel center wavelength [um] (ex) 0.47
	7	scene_acquisition_time	text	This time corresponds to OBT time of 1st swath/ 1st packet. These are applied to folder/filename name. (ex) yyyyymmdd_hhmss
	8	mission_reference_time	text	Planned observation starting time of the whole image of event file in UTC (00000000_000000 if event data not matched) (ex) yyyyymmdd_hhmss
	9	file_generation_time	text	This corresponds to time which Level 1B data are generated. (ex) yyyyymmdd_hhmss
	10	file_name	text	NetCDF file name (ex) gk2a_ami_le1b_v004_fd010ge_yyyyymmddhhmm.nc
	11	file_format_version	text	File format version (ex) 1.0.0_yyyyymmdd
	12	geometric_correction_sw_version	text	Identifies version of the operational software used to create this output (ex) 1.0.0_yyyyymmdd
	13	star_catalog_version	text	Star catalog version (ex) 1.0.0_yyyyymmdd
	14	landmark_catalog_version	text	Landmark catalog version (ex) 1.0.0_yyyyymmdd
Data	1	total_pixel_data_size	uint32	Total pixel data size [byte] (ex) 96800000
	2	number_of_total_swath	uint8	Total Number of L1A Swaths for the complete L1B output (ex) 3 or 5 or 23
	3	number_of_columns	uint32	Number of image pixel columns (ex) 22000
	4	number_of_lines	uint32	Number of image pixel lines (ex) 22000
Observation	1	observation_mode	text	Scene Imaging Mode (ex) FD or ELA or LA or Moon
	2	observation_start_time	float64	Observation start time [sec] (ex) 1.28178413000E+08
	3	observation_end_time	float64	Observation end time [sec] (ex) 1.28178413000E+08
	4	time_synchro_utc	float64	UTC Synchronization time in OBT data file which was used for processing, the current time (ex) 1.28178413000E+08
	5	time_synchro_obt	float64	On-board time corresponding to 'Time Synchronization in UTC (ex) 1.28178413000E+08
Projection	1	projection_type	text	Projection type (ex) GEOS
	2	sub_longitude	float64	Longitude of Sub Satellite Point [radian] (ex) 128.2
	3	cfac	float64	Column scaling factor
	4	lfac	float64	Line scaling factor
	5	coff	float64	Column offset
	6	loff	float64	Line offset
	7	nominal_satellite_height	float64	Nominal Satellite Height [meter] (ex) 42164000.0
	8	earth_equatorial_radius	float64	Earth Semi-major axis [meter]
	9	earth_polar_radius	float64	Earth Semi-minor axis [meter]
	10	earth_ellipsoid_center	float64 x [3]	Earth ellipsoid center coordinates in ECEF [meter] (ex) [0.0, 0.0, 0.0]
	11	image_upperleft_latitude	float64	Image upperleft pixel latitude [radian] (-100000000000000.0 if not available)
	12	image_upperleft_longitude	float64	Image upperleft pixel longitude [radian] (-100000000000000.0 if not available)
	13	image_lowerright_latitude	float64	Image lowerright pixel latitude [radian] (-100000000000000.0 if not available)
	14	image_lowerright_longitude	float64	Image lowerright pixel longitude [radian] (-100000000000000.0 if not available)
	15	image_center_latitude	float64	Image center latitude [radian] (-100000000000000.0 if not available)
	16	image_center_longitude	float64	Image center longitude [radian] (-100000000000000.0 if not available)
	17	image_upperleft_x	float64	Image upperleft pixel location in GEOS coordinate [radian] (-100000000000000.0 if not available)
	18	image_upperleft_y	float64	Image upperleft pixel location in GEOS coordinate [radian] (-100000000000000.0 if not available)
	19	image_lowerright_x	float64	Image lowerright pixel location in GEOS coordinate [radian] (-100000000000000.0 if not available)
	20	image_lowerright_y	float64	Image lowerright pixel location in GEOS coordinate [radian] (-100000000000000.0 if not available)
	21	resampling_kernel_type	text	Resampling kernel type (ex) BICUBIC, BILINEAR, SINC, ...

3. Level 1B 자료 상세 – Variables

표 8 천리안위성 2A호 Level 1B 자료 상세: Variables

Information Type	No	Variable Name	Attribute Name	Format	Description and Example
Data	1	image_pixel_values		uint16 x [M,N]	pixel value including 2-bits data quality flag. (format depends on number of image lines M and image columns N) (ex) VI004, or VI005, or ..., or IR133
	1-1		channel_name	text	
	1-2		detector_side	uint8	Detector Side (ex) 1 or 2
	1-3		number_of_total_pixels	uint32	Number of total pixels (ex) 484000000
	1-4		number_of_error_pixels	uint32	Number of error pixels
	1-5		max_pixel_value	uint16	Maximum pixel value
	1-6		min_pixel_value	uint16	Minumum pixel value
	1-7		average_pixel_value	float64	Average of all pixel values
	1-8		stddev_pixel_value	float64	Standard deviation of all pixel values
	1-9		number_of_total_bits_per_pixel	uint8	full size of a pixel including data quality flag (ex) 16 [bit]
	1-10		number_of_data_quality_flag_bits_per_pixel	uint8	Number of data quality flag (DQF) bits per pixel (ex) 2 [bit]
	1-11		number_of_valid_bits_per_pixel	uint8	number of valid pixel bits per pixel: max. 14 bits (ex) 14 or 13 or 12 or 11 [bit]
	1-12		data_quality_flag_meaning	text	Explanation of DQF meaning
	1-13		ground_sample_distance_ew	float64	Ground sample distance EW [radian] (ex) 0.000028
	1-14		ground_sample_distance_ns	float64	Ground sample
	2	brightness_object_avoidance		uint8 x [N]	brightness object avoidance swaths (format depends on the number of BOA swaths, N) (ex) [3,5,...,17,20]
Geometric	1	sc_position		None	
	1-1		sc_position_first_pixel	float64 x [3]	Spacecraft position in ECEF at first pixel [meter]
	1-2		sc_position_center_pixel	float64 x [3]	Spacecraft position in ECEF at center pixel [meter]
	1-3		sc_position_last_pixel	float64 x [3]	Spacecraft position in ECEF at last pixel [meter]
	1-4		sc_zenith_angle	float64	Spacecraft zenith angle at image center [radian]
	1-5		sc_azimuth_angle	float64	Spacecraft azimuth angle at image center [radian]
	2	sun_position		None	
	2-1		sun_position_ecef	float64 x [3]	Sun position in ECEF at center pixel [meter]
	2-2		sun_zenith_angle	float64	Sun zenith angle at image center [radian]
	2-3		sun_azimuth_angle	float64	Sun azimuth angle at image center [radian]
INR Performance	3	moon_position		None	
	3-1		moon_position_ecef	float64 x [3]	Moon position in ECEF at center pixel [meter]
	1	nav_measurement_type		uint8	measurement type for navigation filter (ex) 0:STAR or 1:LANDMARK
	2	nav_average_residual_ew		float64	Average residual EW of all second stars of back-to-back during this scene
	3	nav_average_residual_ns		float64	Average residual NS of all second stars of back-to-back during this scene
	4	number_of_inr_performance		uint32	Number of INR performance stars (second stars of back-to-back) during this scene
	5	nav_inr_performancetime		float64 x [N]	Measurement time of INR performance stars (second stars of back-to-back) [sec]
	6	nav_residual_ew		float64 x [N]	Residual EW for second stars of back-to-back (format depends on the number of second stars of back-to-back, N)
	7	nav_residual_ns		float64 x [N]	Residual NS for second stars of back-to-back (format depends on the number of second stars of back-to-back, N)
	8	nav_max_att_roll		float64 x [N]	Max roll attitude after INR performance star (format depends on the number of second stars of back-to-back, N)
	9	nav_max_att_pitch		float64 x [N]	Max pitch attitude after INR performance star (format depends on the number of second stars of back-to-back, N)

	10	nav_max_att_yaw		float64 x [N]	Max yaw attitude after INR performance star (format depends on the number of second stars of back-to-back, N)
	11	nav_min_att_roll		float64 x [N]	Min roll attitude after INR performance star (format depends on the number of second stars of back-to-back, N)
	12	nav_min_att_pitch		float64 x [N]	Min pitch attitude after INR performance star (format depends on the number of second stars of back-to-back, N)
	13	nav_min_att_yaw		float64 x [N]	Min yaw attitude after INR performance star (format depends on the number of second stars of back-to-back, N)
	14	nav_average_att_roll		float64 x [N]	Average roll attitude after INR performance star (format depends on the number of second stars of back-to-back, N)
	15	nav_average_att_pitch		float64 x [N]	Average pitch attitude after INR performance star (format depends on the number of second stars of back-to-back, N)
	16	nav_average_att_yaw		float64 x [N]	Average yaw attitude after INR performance star (format depends on the number of second stars of back-to-back, N)
	17	nav_stddev_att_roll		float64 x [N]	Standard deviation of roll attitude after INR performance star (format depends on the number of second stars of back-to-back, N)
	18	nav_stddev_att_pitch		float64 x [N]	Standard deviation of pitch attitude after INR performance star (format depends on number of second stars of back-to-back, N)
	19	nav_stddev_att_yaw		float64 x [N]	Standard deviation of yaw attitude after INR performance star (format depends on number of second stars of back-to-back, N)
Star	1	number_of_vis_stars		uint32	Number of visible stars during this scene.
	2	vis_star_measure_time		float64 x [N]	Visible star measurement time [sec] for second star of back-to-back star observation (format depends on the number of stars, N)
	3	vis_star_spectral_radiance		float64 x [N]	Visible star's spectral radiance [W/m ² /sr/μm] for second star of back-to-back star observation (format depends on the number of stars, N)
	4	vis_star_right_ascension		float64 x [N]	Right Ascension (EW) angle of measured VIS star in radians for second star of back-to-back star observation. (format depends on the number of stars, N)
	5	vis_star_declination		float64 x [N]	Declination (NS) angle of measured VIS star in radians for second star of back-to-back star observation. (format depends on the number of stars, N)
	6	vis_star_residual_ew		float64 x [N]	Azimuth(EW) Residuals for Visible Stars in radians for second star of back-to-back star observation. (format depends on the number of stars, N)
	7	vis_star_residual_ns		float64 x [N]	Elevation(NS) Residuals for Visible Stars in radians for second star of back-to-back star observation. (format depends on the number of stars, N)
	8	number_of_ir_stars		uint32	Number of IR stars during this scene.
	9	ir_star_measure_time		float64 x [N]	IR star measurement time [sec] (format depends on the number of stars, N)
	10	ir_star_spectral_radiance		float64 x [N]	IR star's spectral radiance [mW/m ² /sr/cm-1] (format depends on the number of stars, N)
	11	ir_star_right_ascension		float64 x [N]	Right Ascension (EW) angle of measured IR star in radians. (format depends on the number of stars, N)
	12	ir_star_declination		float64 x [N]	Declination (NS) angle of measured IR star in radians. (format depends on the number of stars, N)
	13	ir_star_residual_ew		float64 x [N]	Azimuth(EW) Angle of Measured IR Star in radians. (format depends on the number of stars, N)
	14	ir_star_residual_ns		float64 x [N]	Elevation(NS) Angle of Measured IR Star [radian]. (format depends on the number of stars, N)
Landmark	1	number_of_matched_lmk		uint32	Number of matched landmarks
	2	matched_lmk_id		uint32 x [N]	Landmark ID (format depends on the number of matched landmarks, N)
	3	matched_lmk_channel		uint8 x [N]	Channel Number [1~16] (format depends on the number of matched landmarks, N)
	4	matched_lmk_chip_type		uint8 x [N]	LMK Type (format depends on the number of matched landmarks, N) (ex: 0:GSHHS or 1:High-Res)
	5	matched_lmk_longitude		float64 x [N]	Longitude [radian] (format depends on the number of matched landmarks, N)
	6	matched_lmk_latitude		float64 x [N]	Latitude [radian] (format depends on the number of matched landmarks, N)
	7	matched_lmk_l1a_ew		float64 x [N]	EW position in L1A coordinate [radian] (format depends on the number of matched landmarks, N)
	8	matched_lmk_l1a_ns		float64 x [N]	NS position in L1A coordinate [radian] (format depends on the number of matched landmarks, N)

	9	matched_lmk_residual_ew		float64 x [N]	EW residual in L1A coordinate [radian] (format depends on the number of matched landmarks, N)
	10	matched_lmk_residual_ns		float64 x [N]	NS residual in L1A coordinate [radian] (format depends on the number of matched landmarks, N)
	11	matched_lmk_accuracy		float64 x [N]	Accuracy [radian] (format depends on the number of matched landmarks, N)
	12	matched_lmk_corr_coeff		float64 x [N]	Correlation coefficient [0.0~1.0] (format depends on the number of matched landmarks, N)
	13	matched_lmk_sharpness		float64 x [N]	Correlation sharpness [0.0~1.0] (format depends on the number of matched landmarks, N)
	14	matched_lmk_second_peak		float64 x [N]	Second peak ratio [0.0~1.0] (format depends on the number of matched landmarks, N)
	15	matched_lmk_cloud_coverage		float64 x [N]	Cloud coverage ratio [0.0~1.0] (format depends on the number of matched
	16	matched_lmk_chip_cloud		float64 x [N]	Chip cloud ratio [0.0~1.0] (format depends on the number of matched landmarks, N)
	17	matched_lmk_measure_time		float64 x [N]	Landmark measurement time [sec] (format depends on the number of matched landmarks, N)
Inter-Calibration	1	gsics_coeff_intercept		float64	TBD
	2	gsics_coeff_intercept_standard_error		float64	TBD
	3	gsics_coeff_slope		float64	TBD
	4	gsics_coeff_slope_standard_error		float64	TBD
	5	gsics_coeff_quadratic		float64	TBD
	6	gsics_coeff_quadratic_standard_error		float64	TBD
	7	gsics_coeff_valid_range_upper_limit		float64	TBD
	8	gsics_coeff_valid_range_lower_limit		float64	TBD
	9	gsics_start_time_of_validity_period		float64	TBD
	10	gsics_end_time_of_validity_period		float64	TBD

4. 관측데이터

4.1 image_pixel_values

Variables의 image_pixel_values는 $(m \times n)$ 크기의 행렬로, 관측 데이터를 저장한다. 행과 열의 개수인 m과 n은 각각 Global Attribute의 number_of_columns와 number_of_lines에 저장된다. image_pixel_values의 요소를 픽셀값(pixel value) 또는 화소값이라 한다. 픽셀값은 16비트 unsigned integer 형태로 표현된다.

표 9 image_pixel_values 행렬 (m 행 \times n 열)

Line Number	Column number							
	0	1	2	...	C	...	n-2	n-1
0	[0,0]	[0,1]	[0,2]	...	[0,C]	...	[0,n-2]	[0,n-1]
1	[1,0]	[1,1]	[1,2]	...	[1,C]	...	[1,n-2]	[1,n-1]
2	[2,0]	[2,1]	[2,2]	...	[2,C]	...	[2,n-2]	[2,n-1]
...
L	[L,0]	[L,1]	[L,2]	...	[L,C]	...	[L,n-2]	[L,n-1]
...
m-2	[m-2,0]	[m-2,1]	[m-2,2]	...	[m-2,C]	...	[m-2,n-2]	[m-2,n-1]
m-1	[m-1,0]	[m-1,1]	[m-1,2]	...	[m-1,C]	...	[m-1,n-2]	[m-1,n-1]

5. 데이터 품질 표식 (Quality Flag)

5.1 데이터 품질 표식

픽셀값은 데이터 품질 표식(data quality flag, 이하 DQF)을 포함한다. 픽셀값의 16개 비트 중에서 가장 좌측에 있는 2개 비트에 DQF가 할당되어 있다. DQF의 2개 비트(이진표기 0b0000000000000000)는 해당 픽셀의 품질을 표시한다. 나머지 14개의 비트(이진표기 0b0000000000000000)가 해당 픽셀의 실제 값을 의미한다.

표 10 데이터 품질 표식의 범위와 의미

이진표기	범위 (십진수)	의미
0b <u>00</u> 00000000000000	0~16383	좋은 품질
0b <u>01</u> 00000000000000	16384~32767	조건에 따라 사용가능
0b <u>10</u> 00000000000000	32768~49151	관측 영역 외부
0b <u>11</u> 00000000000000	49152~65535	오류 존재

예를 들어, 픽셀 값이 0b01000000011101 (십진수로 16413)인 경우, 해당 픽셀이 조건에 따라 사용가능conditionally usable)한 품질이고, 해당 픽셀의 실제 값이 십진수로 29임을 나타낸다.

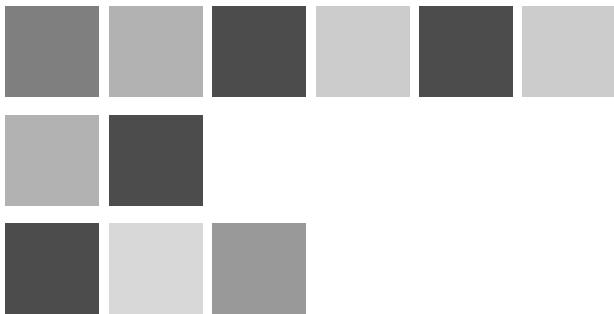
6. 픽셀당 데이터 비트 개수

6.1 픽셀당 데이터 비트 개수

픽셀값은 관측 채널에 따라 실제 값이 담긴 데이터 비트 개수(bit size per pixel)가 다르다. 표 11을 통해 16개 채널별 픽셀당 데이터 비트 개수를 확인할 수 있다.

표 11 채널별 픽셀당 데이터 비트 개수

Channel No.	Channel Name	Bit size per pixel	Channel No.	Channel Name	Bit size per pixel
1	VI004	11	9	WV069	13
2	VI005	11	10	WV073	13
3	VI006	12	11	IR087	13
4	VI008	13	12	IR096	13
5	NR013	12	13	IR105	13
6	NR016	11	14	IR112	13
7	SW038	14	15	IR123	13
8	WV063	12	16	IR133	13



IV

Level 1B 자료

값 추출 방법



1. 사용자 맞춤형 영상처리도구 (공개 소프트웨어)

국가기상위성센터에서 제공하는 사용자 맞춤형 영상처리도구를 이용해 GK-2A L1B 데이터의 영상처리가 가능하다. (국가기상위성센터 국/영문 홈페이지를 통해 소프트웨어 제공)

그림 5 사용자 맞춤형 영상처리도구의 특징



그림 6 사용자 맞춤형 영상처리도구 - image_pixel_values를 이미지로 표출

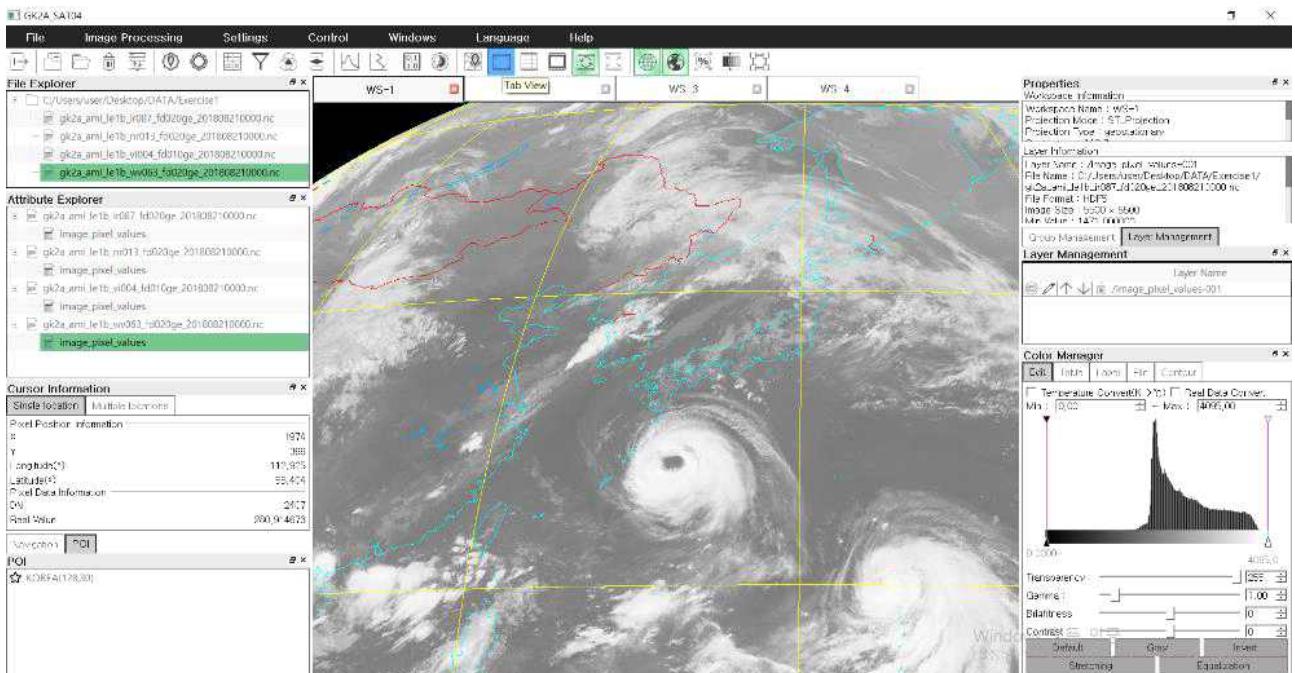
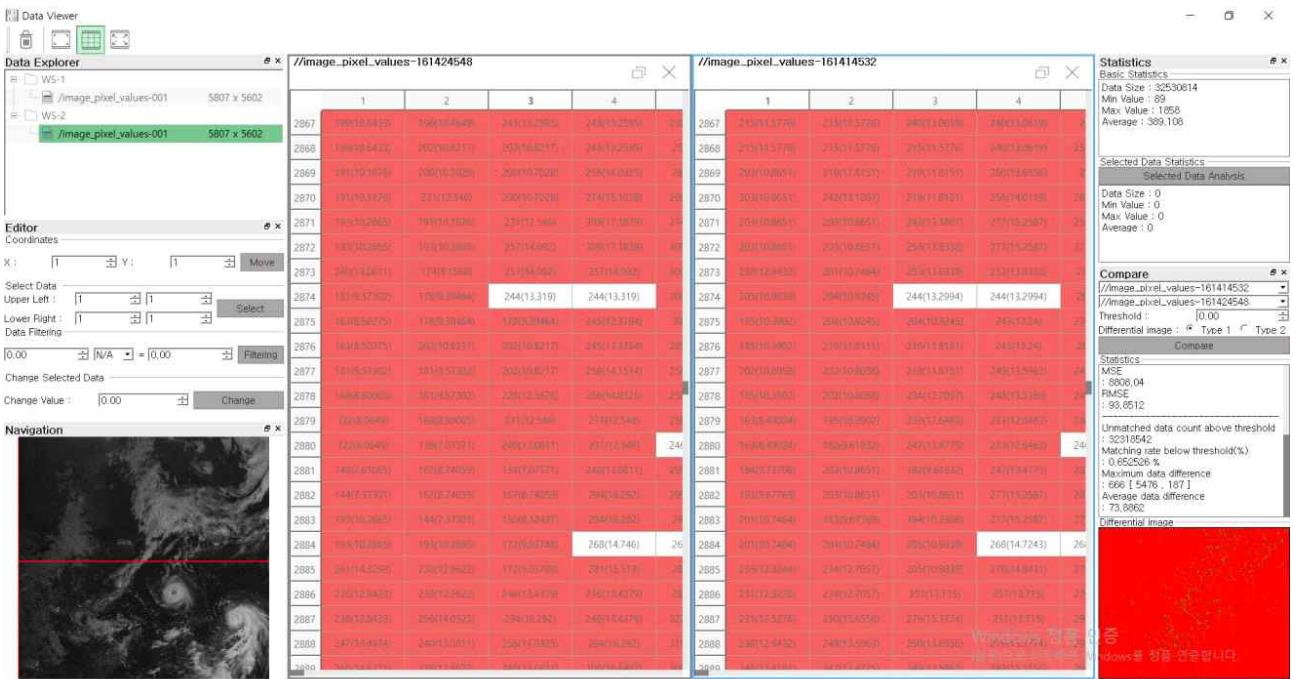
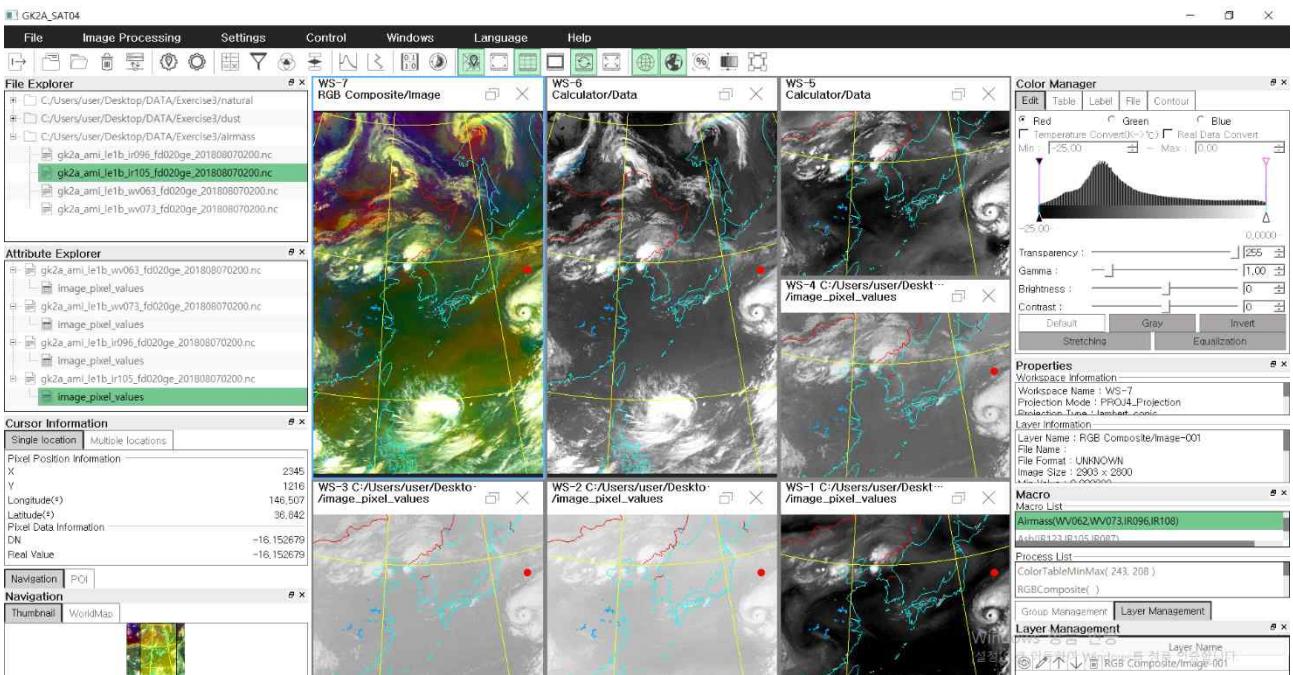


그림 7 사용자 맞춤형 영상처리도구 - image_pixel_values 값 표출



사용자 맞춤형 영상처리도구를 통해 위성자료의 편집, 합성 등의 작업을 수행할 수 있으며, NetCDF4 형태의 자료를 HDF5, Binary, 이미지 형태로 출력하는 것도 가능하다.

그림 8 사용자 맞춤형 영상처리도구 – RGB Composite 이미지 생성



2. Panoply : NetCDF, HDF, GRIB Viewer

Panoply(<https://www.giss.nasa.gov/tools/panoply/>)는 NetCDF, HDF, GRIB 파일을 열 수 있는 프로그램이다. 이 프로그램을 활용하여 GK-2A L1B 데이터의 정보 확인이 가능하다.

그림 9 Panoply – 천리안위성 2A호 샘플 데이터 불러오기

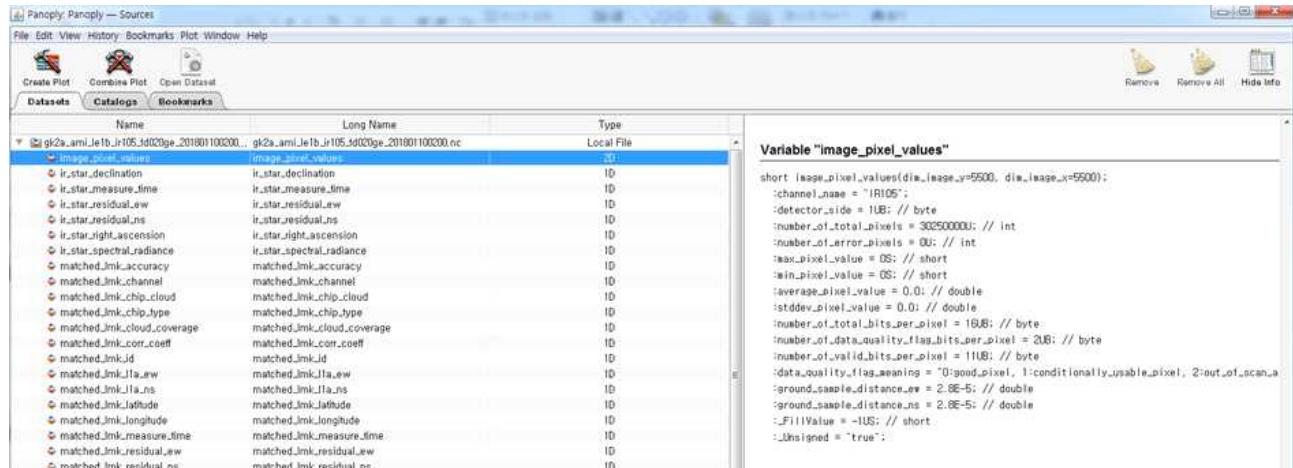


그림 10 Panoply - image_pixel_values를 이미지로 표출

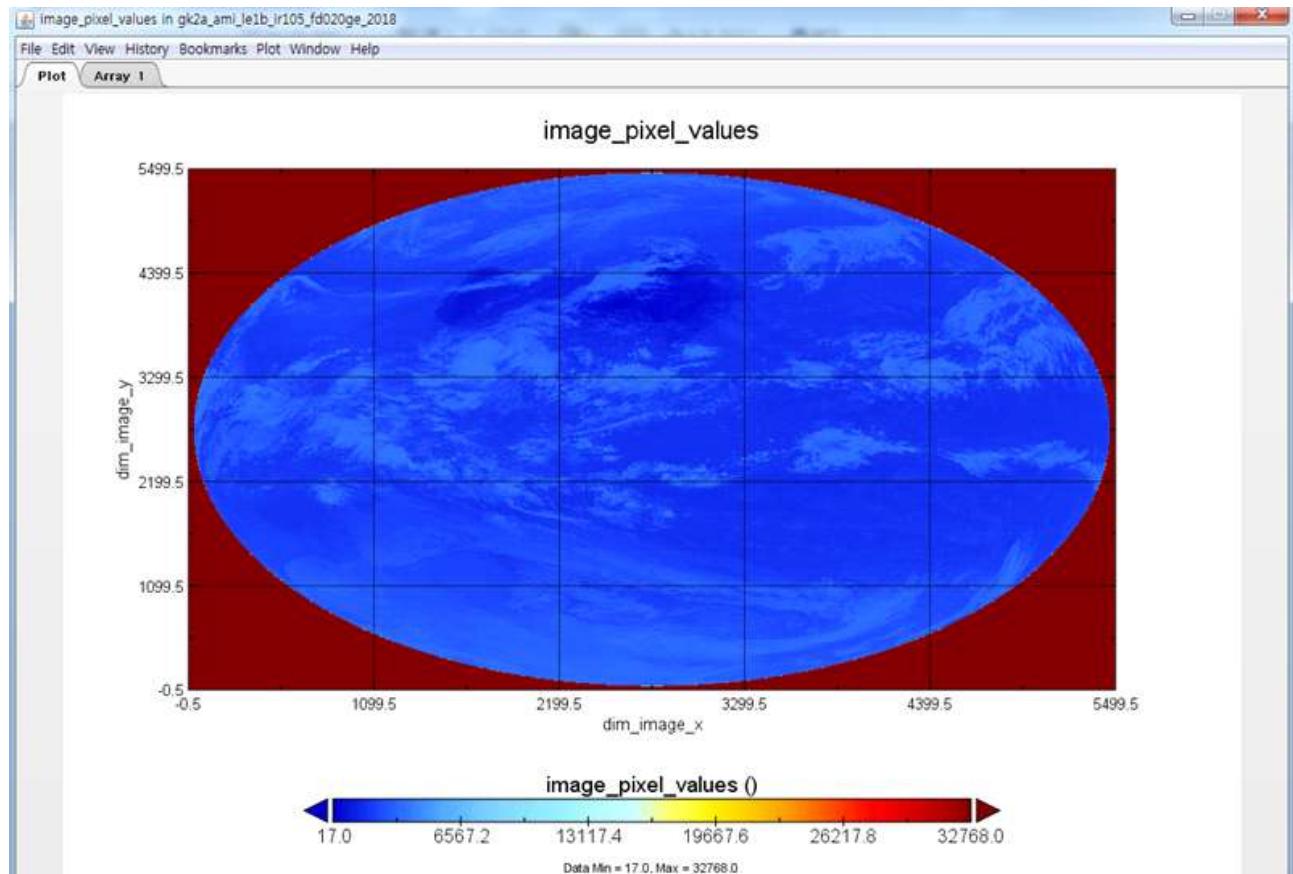


그림 11 Panoply - image_pixel_values 값 표출

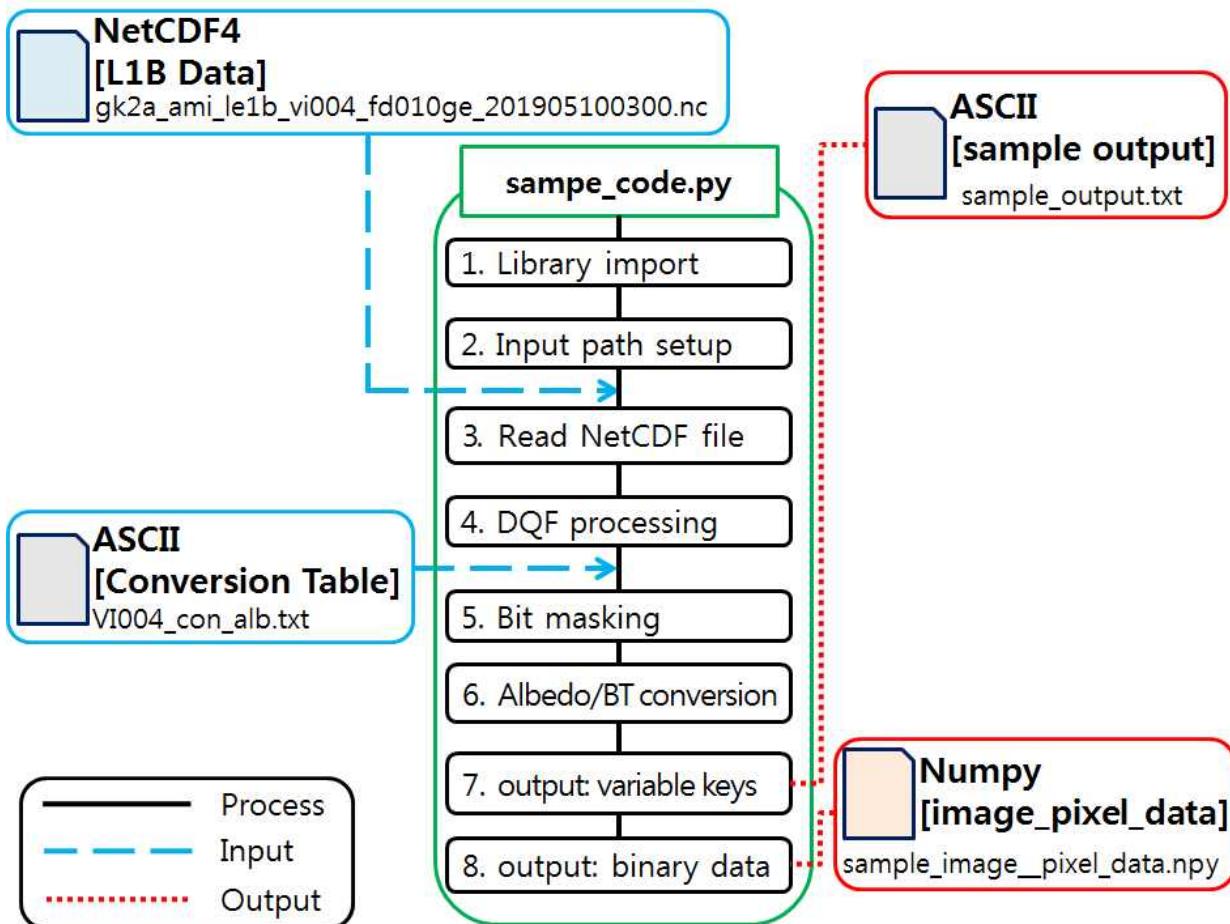
The screenshot shows the Panoply software interface with the title bar "Image_pixel_values in gk2a_ami_le1b_ir105_fd020ge_201801100200.nc". The menu bar includes File, Edit, View, History, Bookmarks, Plot, Window, and Help. The tabs "Plot" and "Array 1" are selected. Below the tabs, it says "Dataset: gk2a_ami_le1b_ir105_fd020ge_201801100200.nc" and "Variable: image_pixel_values, image_pixel_values". The main area displays a 2D array of numerical values. The columns are labeled "X-Axis: dim_image_x" and the rows are labeled "Y-Axis: dim_image_y". The data values range from 2416.0 to 2994.0. At the bottom, there are buttons for "Format: %1f", "Flip Table B/T", "Flip Table L/R", and "Show cell indices".

	2632.0	2633.0	2634.0	2635.0	2636.0	2637.0	2638.0	2639.0	2640.0	2641.0	2642.0	2643.0	2644.0	
4954.0	2440.0	2451.0	2444.0	2446.0	2440.0	2420.0	2406.0	2373.0	2398.0	2416.0	2442.0	2432.0	249	
4953.0	2418.0	2417.0	2415.0	2414.0	2416.0	2420.0	2406.0	2373.0	2398.0	2412.0	2411.0	2432.0	247	
4952.0	2414.0	2413.0	2407.0	2400.0	2403.0	2404.0	2412.0	2423.0	2433.0	2423.0	2408.0	2458.0	254	
4951.0	2410.0	2411.0	2403.0	2402.0	2403.0	2397.0	2404.0	2425.0	2437.0	2408.0	2399.0	2468.0	253	
4950.0	2420.0	2410.0	2405.0	2407.0	2401.0	2396.0	2403.0	2415.0	2418.0	2401.0	2430.0	2487.0	251	
4949.0	2455.0	2431.0	2410.0	2412.0	2404.0	2403.0	2410.0	2412.0	2402.0	2395.0	2404.0	2422.0	248	
4948.0	2486.0	2503.0	2499.0	2456.0	2427.0	2422.0	2427.0	2432.0	2421.0	2401.0	2396.0	2403.0	242	
4947.0	2494.0	2512.0	2515.0	2522.0	2531.0	2535.0	2529.0	2495.0	2477.0	2474.0	2456.0	2438.0	2432.0	243
4946.0	2517.0	2511.0	2515.0	2515.0	2511.0	2519.0	2523.0	2503.0	2482.0	2470.0	2456.0	2447.0	243	
4945.0	2569.0	2535.0	2521.0	2510.0	2499.0	2493.0	2492.0	2494.0	2490.0	2486.0	2465.0	2449.0	243	
4944.0	2606.0	2599.0	2565.0	2540.0	2507.0	2491.0	2480.0	2474.0	2478.0	2485.0	2474.0	2451.0	243	
4943.0	2614.0	2581.0	2594.0	2598.0	2548.0	2511.0	2497.0	2492.0	2486.0	2473.0	2466.0	2459.0	245	
4942.0	2600.0	2609.0	2614.0	2611.0	2573.0	2545.0	2544.0	2545.0	2536.0	2510.0	2498.0	2481.0	247	
4941.0	2615.0	2665.0	2672.0	2672.0	2650.0	2602.0	2574.0	2575.0	2551.0	2533.0	2528.0	2523.0	250	
4940.0	2667.0	2713.0	2771.0	2821.0	2813.0	2730.0	2661.0	2614.0	2592.0	2566.0	2544.0	2533.0	252	
4939.0	2637.0	2664.0	2754.0	2852.0	2854.0	2837.0	2766.0	2710.0	2690.0	2663.0	2639.0	2611.0	259	
4938.0	2586.0	2609.0	2642.0	2729.0	2780.0	2801.0	2843.0	2808.0	2776.0	2748.0	2733.0	2738.0	273	
4937.0	2609.0	2582.0	2592.0	2608.0	2628.0	2662.0	2740.0	2759.0	2783.0	2811.0	2791.0	2781.0	282	
4936.0	2582.0	2564.0	2566.0	2570.0	2602.0	2643.0	2699.0	2706.0	2752.0	2793.0	2787.0	2777.0	277	
4935.0	2598.0	2591.0	2559.0	2557.0	2584.0	2633.0	2721.0	2791.0	2752.0	2691.0	2702.0	2728.0	278	
4934.0	2670.0	2691.0	2688.0	2645.0	2620.0	2624.0	2681.0	2818.0	2856.0	2747.0	2709.0	2680.0	270	
4933.0	2739.0	2774.0	2854.0	2845.0	2823.0	2775.0	2714.0	2699.0	2768.0	2827.0	2802.0	2719.0	268	
4932.0	2792.0	2794.0	2812.0	2862.0	2892.0	2888.0	2840.0	2760.0	2735.0	2811.0	2878.0	2800.0	265	
4931.0	2795.0	2797.0	2809.0	2844.0	2913.0	2945.0	2901.0	2852.0	2772.0	2753.0	2852.0	2799.0	264	
4930.0	2776.0	2796.0	2810.0	2826.0	2865.0	2940.0	2967.0	2942.0	2874.0	2796.0	2770.0	2757.0	264	
4929.0	2723.0	2757.0	2767.0	2792.0	2816.0	2833.0	2876.0	2951.0	2979.0	2899.0	2836.0	2771.0	269	
4928.0	2616.0	2660.0	2699.0	2727.0	2788.0	2805.0	2797.0	2847.0	2962.0	2970.0	2891.0	2825.0	277	
4927.0	2550.0	2557.0	2578.0	2625.0	2660.0	2688.0	2729.0	2774.0	2848.0	2972.0	3045.0	2940.0	288	
4926.0	2581.0	2587.0	2569.0	2588.0	2637.0	2650.0	2706.0	2806.0	2808.0	2876.0	3055.0	3101.0	299	
4925.0	2707.0	2673.0	2686.0	2674.0	2687.0	2690.0	2780.0	2891.0	2913.0	2934.0	3027.0	3102.0	310	
4924.0	2886.0	2814.0	2756.0	2726.0	2727.0	2693.0	2686.0	2788.0	2903.0	2997.0	3057.0	3088.0	310	
4923.0	2221.0	2221.0	2221.0	2221.0	2221.0	2221.0	2221.0	2221.0	2221.0	2221.0	2221.0	2221.0	2221	

3. 샘플코드1 처리 순서도

본 매뉴얼에는 GK-2A L1B 자료 처리를 위한 Python 샘플코드 2종류가 기재되어 있으며, 첫 번째 샘플코드의 순서도를 그림 12에서 확인할 수 있다. 실제 코드는 IV-5에 기재되어 있다. 샘플코드의 테스트 환경은 붙임 2를 참조하면 된다.

그림 12 샘플코드1 처리 순서도



3.1 입력파일영역

흐름도 좌측의 파란색 선으로 둘러싸인 개체는 샘플코드의 작동을 위한 입력 파일이다. 하나는 GK-2A 샘플데이터이고, 다른 하나는 Conversion Table이다. Conversion Table에 대한 설명은 IV-5 항목을 참조하면 된다..

3.2 내부처리영역

중앙의 초록색 선으로 둘러싸인 영역 내부는 샘플코드 내부에서 처리되는 작업이다. L1B자료의 입력, 참조, 가공, 출력이 수행된다. 내부처리 영역의 번호는 샘플코드1의 섹션 번호와 일치한다.

3.3 출력파일영역

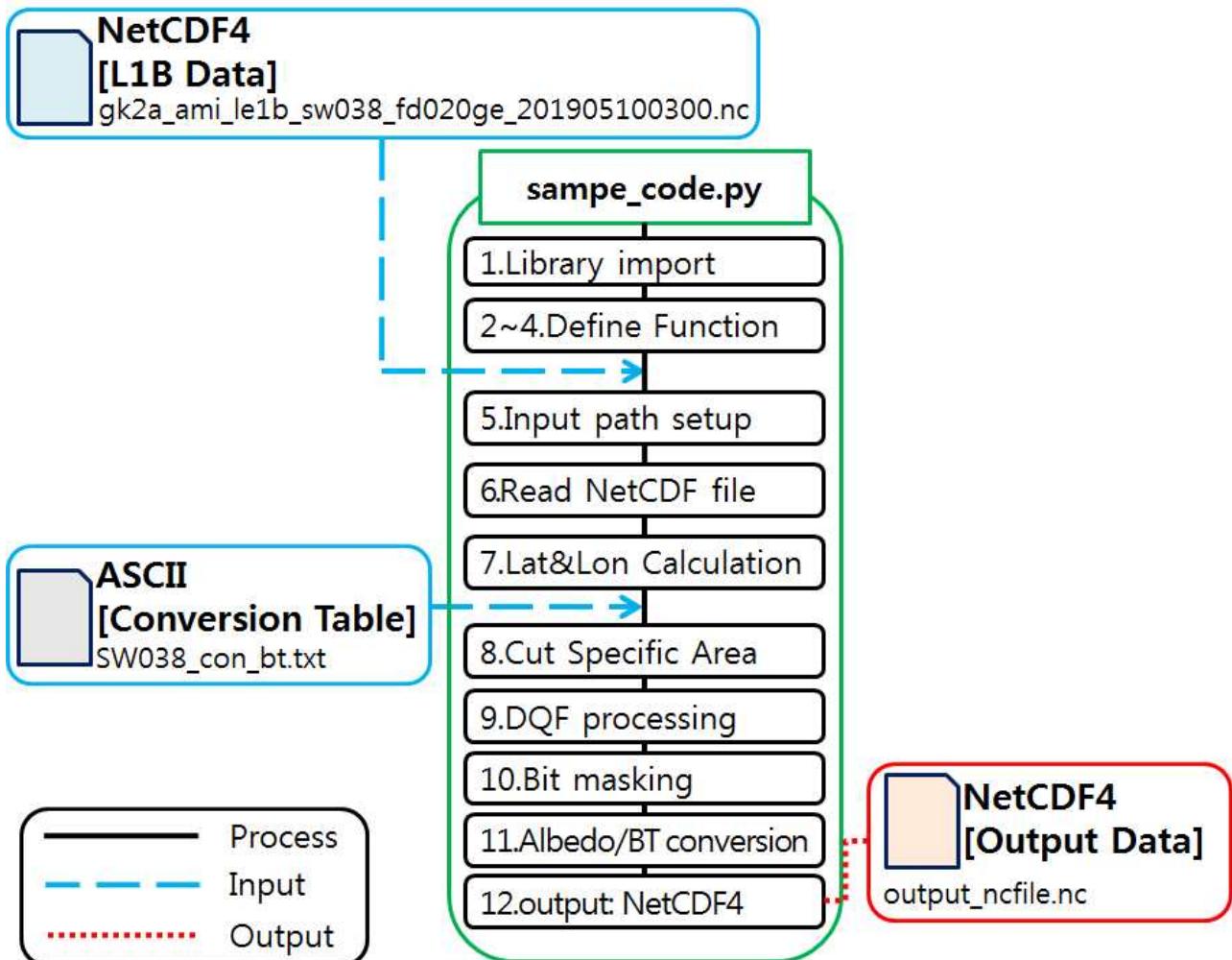
우측의 빨간색 선으로 둘러싸인 개체는 작업이 정상적으로 종료되면 생산되는 출력물이다. 하나는 npy 형식으로, 이미지 데이터(image_pixel_values)의 물리값을 담고 있다. 해당 파일은 python의 numpy library를 통해 입출력이 가능하다. 다른 하나는 텍스트(ASCII) 형식으로, Variables의 정보가 담겨 있다.

4. 샘플코드2 처리 순서도

샘플코드2는 사용자가 설정한 특정 영역을 L1B 파일에서 잘라내어 해당 영역의 물리값 및 위경도, 메타데이터를 NetCDF4 형태로 저장하는 작업을 수행한다. 실제 코드는 IV-7에 기재되어 있다.

샘플코드의 테스트 환경은 붙임 2를 참조하면 된다.

그림 13 샘플코드2 처리 순서도



4.1 입력파일영역

좌측의 파란색 선으로 둘러싸인 개체는 샘플코드의 작동을 위한 입력 파일이다. 하나는 GK-2A 샘플 자료 파일이고, 다른 하나는 Conversion Table이다. Conversion Table에 대한 설명은 IV-5 항목을 참조하면 된다.

4.2 내부처리영역

중앙의 초록색 선으로 둘러싸인 영역 내부는 샘플코드가 내부의 작업이다. L1B자료의 입력, 참조, 가공, 출력을 수행한다. 내부처리영역의 번호는 샘플코드2의 섹션 번호와 일치한다.

4.3 출력파일영역

우측의 빨간색 선으로 둘러싸인 개체는 작업이 정상적으로 종료되면 생산되는 출력물이다.
출력물은 NetCDF4 형식이며, 사용자 정의 영역의 물리 값(Albedo/BT) 및 메타데이터를 포함한다.

5. 물리값 변환

5.1 Conversion table을 이용한 방법

Conversion table을 이용하여 image_pixel_values의 픽셀값을 반사율(Albedo)이나 밝기온도(Brightness Temperature) 등으로 변환한다. 샘플코드의 “Albedo/BT conversion”에서 이러한 작업을 수행한다. Conversion table에 대한 자세한 설명은 붙임4를 참조하면 된다..

그림 14 Conversion Table을 이용한 image_pixel_values의 변환

The diagram illustrates the conversion process using three tables:

- image_pixel_values [16bit integer]**: A 6x3 grid of integers. The columns represent the input values: 0, 3, 1; 12, 1, 2; 0, 0, 1; 0, 0, 5; and two rows of 0s.
- Conversion Table**: A table mapping input values to Albedo values. It shows a linear decrease from -0.010001 at value 0 to -0.007501 at value 5, with intermediate values and an ellipsis.
- Albedo [16bit float]**: A 6x3 grid of floating-point numbers. The first five rows correspond to the values in the conversion table, and the last row is all zeros.

Red arrows point downwards from the first table to the second, and from the second to the third, indicating the flow of the conversion process.

image_pixel_values [16bit integer]		
0	0	1
1	3	0
12	1	2
0	0	1
0	0	5
0	0	0

↓

Conversion Table		
Value	→	Albedo
0	→	-0.010001
1	→	-0.009501
2	→	-0.009001
3	→	-0.008501
4	→	-0.008001
5	→	-0.007501
...		...

↓

Albedo [16bit float]		
-0.010001	-0.010001	-0.009501
-0.009501	-0.008501	-0.010001
-0.004000	-0.009501	-0.009001
-0.010001	-0.010001	-0.009501
-0.010001	-0.010001	-0.007501
-0.010001	-0.010001	-0.010001

6. 샘플코드1

범례	코드내용	필수 사항	선택 사항	설명	변경 가능
	색상	code	code	code	code
#	코드				주석
0	<pre>#!/usr/bin/env python ##### # # This program extract variables information and # image_pixel_values data(Albedo) of # GK2A Visible Ch.4 Level 1B data in netCDF4 format # # Input : GK2A L1B file [sample file : VI004/fd010ge] (netCDF4) # GK2A conversion table(netCDF4) # # Output : sample_output.txt (ASCII) # sample_image_pixel_data.npy (numpy format) # ##### </pre>				프로그램 헤더부 프로그램 기능 및 입/출력 파일 설명
1	<pre>import netCDF4 import numpy</pre>				- 파일 라이브러리 사용 준비 단계
2	<pre>input_ncfile_path = 'gk2a_ami_le1b_vi004_fd010ge_201905100300.nc' CT_path='./conversion_table/'</pre>				- 입력 자료 설정 >L1B 자료 파일 이름 >Conversion table 디렉토리
	<pre>output_txt_path='sample_output.txt' output_npy_path='sample_image_pixel_data.npy'</pre>				- 출력 자료 설정 >텍스트(ASCII)파일 이름 >이진(npy)파일 이름
3	<pre>ncfile = netCDF4.Dataset(input_ncfile_path,'r',format='netcdf4') ipixel=ncfile.variables['image_pixel_values'] ipixel_process = ipixel[:]</pre>				- image_pixel_values 추출 >netCDF4 파일 불러오기 >image_pixel_values 변수 불러오기 ->ipixel >변수에서 행렬 추출 ->ipixel_process
4	<pre>number_of_error_pixels = ipixel.getncattr('number_of_error_pixels') if (number_of_error_pixels > 0): ipixel_process[ipixel_process>49151] = 0</pre>				- DQF 처리 Error pixel(III-5참조) 값을 0으로 필터링
5	<pre>channel=ipixel.getncattr('channel_name') if ((channel == 'VI004') or (channel == 'NR016') or (channel == 'VI005')): mask = 0b0000011111111111 #11bit mask elif ((channel == 'VI006') or (channel == 'NR013') or (channel == 'WV063')): mask = 0b0000111111111111 #12bit mask elif (channel == 'SW038'): mask = 0b0011111111111111 #14bit mask else: mask = 0b0001111111111111 #13bit mask ipixel_process_masked=numpy.bitwise_and(ipixel_process,mask)</pre>				-Bit Masking pixel value의 Bit size (III-6참조)에 따라 bit mask 생성 및 마스크 처리

6	<pre> AL_postfix='_con_alb.txt' BT_postfix='_con_bt.txt' if (channel[0:2] == 'VI') or (channel[0:2] == 'NR'): conversion_table=numpy.loadtxt(cal_table_path+channel+AL_postfix,'float64') else: conversion_table=numpy.loadtxt(CT_path+channel+BT_postfix,'float64') ipixel_process_masked_converted=conversion_table[ipixel_process_masked] </pre>	<ul style="list-style-type: none"> - Albedo/BT 변환 픽셀 값을 알베도 혹은 밝기온도로 변환 (IV-4참조)
7	<pre> output_txt=open(output_txt_path,'wt') var_keys = ncfile.variables.keys() output_txt.write("number of variable keys : %sn" %len(var_keys)) for i in range(len(var_keys)): output_txt.write("%s%s%s%s%s%s%s\n" %(i,var_keys[i],ncfile.variables[var_keys[i]].dimensions,ncfile.variables[var_keys[i]].dtype)) output_txt.close() print "variable list wrote to sample_output.txt" </pre>	ASCII 포맷 출력 내용은 불임-3참조
8	<pre> numpy.save(output_npy_path,ipixel_process_masked_converted,False,False) </pre>	비압축 이진 파일 Numpy 포맷 출력파일용량: 924MB
9	<pre> ##### #End of Program ##### </pre>	프로그램 종료

7. 샘플코드2

범례	코드내용	필수 사항	선택 사항	설명	변경 가능
	색상/서식	code	code	code	code
#	코드			주석	
0	<pre> #!/usr/bin/env python ##### # GK2A L1B data Processing sample code2 # # This program extracts user defined area's pixel value/latitude/longitude value from GK2A NetCDF4 file # and converts digital count number to Albedo/Brightness Temperature. # After that it saves converted data to new NetCDF4 file with geographic coordinates. # # Input: GK2A L1B file [sample file: SW038/fd020ge] (netCDF4) # GK2A conversion table(ASCII) # # process: read input files -> cut user defined area from input # -> convert digital count number to Albedo/Brightness Temperature # -> save data with netCDF4 form # # Output : Albedo/Brightness Temperature from cut area (netCDF4) # # The output netCDF4 file includes next data # -user defined area's line & column size # -user defined area's image_pixel_values # -latitude & longitude of every pixel in cut area # -user defined area's line/column number of left upper point in original GEOS image array(in global attribute) # -user defined area's line/column number of right lower point in original GEOS image array(in global attribute) ##### # Library import & Function define ##### </pre>			프로그램 헤더부	프로그램 기능 및 입/출력 파일 설명
					라이브러리 준비 및 함수정의구간 시작

```

1 import netCDF4
import numpy
2 def latlon_from_lincol_geos(Resolution, Line ,Column):
    degtorad=3.14159265358979 / 180.0
    if(Resolution == 0.5):
        COFF=11000.5
        CFAC=8.170135561335742e7
        LOFF=11000.5
        LFAC=8.170135561335742e7
    elif(Resolution == 1.0):
        COFF=5500.5
        CFAC=4.0850677806678705e7
        LOFF=5500.5
        LFAC=4.0850677806678705e7
    else:
        COFF=2750.5
        CFAC=2.0425338903339352e7
        LOFF=2750.5
        LFAC=2.0425338903339352e7
    sub_lon=128.2
    sub_lon=sub_lon*degtorad
    x= degtorad *( (Column - COFF)*2**16 / CFAC )
    y= degtorad *( (Line - LOFF)*2**16 / LFAC )
    Sd = np.sqrt( (42164.0*np.cos(x)*np.cos(y))**2 - (np.cos(y)**2 + 1.006739501*np.sin(y)**2)*1737122264)
    Sn = (42164.0*np.cos(x)*np.cos(y)-Sd) / (np.cos(y)**2 + 1.006739501*np.sin(y)**2)
    S1 = 42164.0 - ( Sn * np.cos(x) * np.cos(y) )
    S2 = Sn * ( np.sin(x) * np.cos(y) )
    S3 = -Sn * np.sin(y)
    Sxy = np.sqrt( ((S1*S1)+(S2*S2)) )
    nlon=(np.arctan(S2/S1)+sub_lon)/degtorad
    nlat=np.arctan( ( 1.006739501 *S3)/Sxy)/degtorad
    return (nlat, nlon)

```

- 파일 라이브러리 사용 준비 단계

- 함수 정의:
전구 영역 GEOS 도법 자료의 행과 열
번호를 입력받아 위경도를 출력하는
함수

```

3 def lincol_from_latlon_geos(Resolution, Latitude, Longitude):
    degtorad=3.14159265358979 / 180.0
    if(Resolution == 0.5):
        COFF=11000.5
        CFAC=8.170135561335742e7
        LOFF=11000.5
        LFAC=8.170135561335742e7
    elif(Resolution == 1.0):
        COFF=5500.5
        CFAC=4.0850677806678705e7
        LOFF=5500.5
        LFAC=4.0850677806678705e7
    else:
        COFF=2750.5
        CFAC=2.0425338903339352e7
        LOFF=2750.5
        LFAC=2.0425338903339352e7
        sub_lon=128.2
    sub_lon=sub_lon*degtorad
    Latitude=Latitude*degtorad
    Longitude=Longitude*degtorad
    c_lat = np.arctan(0.993305616*np.tan(Latitude))
    RL = 6356.7523 / np.sqrt( 1.0 - 0.00669438444*np.cos(c_lat)**2.0 )
    R1 = 42164.0 - RL *np.cos(c_lat)*np.cos(Longitude - sub_lon)
    R2 = -RL* np.cos(c_lat) *np.sin(Longitude - sub_lon)
    R3 = RL* np.sin(c_lat)
    Rn = np.sqrt(R1**2.0 + R2**2.0 + R3**2.0 )
    x = np.arctan(-R2 / R1) / degtorad
    y = np.arcsin(-R3 / Rn) / degtorad
    ncol=COFF + (x* 2.0**(-16) * CFAC)
    nlin=LOFF + (y* 2.0**(-16) * LFAC)
    return (nlin,ncol)

```

- 함수 정의:
 위경도를 입력받고 전구 영역 GEOS
 도법 자료의 해당 행과 열의 번호를
 출력하는 함수

```

4 def cut_with_latlon_geos(Array, Resolution, Latitude1, Longitude1, Latitude2, Longitude2):
    Array=np.array(Array)
    if(Resolution == 0.5):
        Index_max=22000
    elif(Resolution == 1.0):
        Index_max=11000
    else:
        Index_max=5500
    (Lin1,Col1) = lincol_from_latlon_geos(Resolution, Latitude1, Longitude1)
    (Lin2,Col2) = lincol_from_latlon_geos(Resolution, Latitude2, Longitude2)
    Col1=int(np.floor(Col1))
    Lin1=int(np.floor(Lin1))
    Col2=int(np.ceil(Col2))
    Lin2=int(np.ceil(Lin2))
    cut=np.zeros((Index_max,Index_max))
    if( (Col1 <= Col2) and (Lin1 <= Lin2) and (0 <= Col1) and (Col2 < Index_max) and (0 <= Lin1) and
       (Lin2 < Index_max) ):
        cut=Array[Lin1:Lin2,Col1:Col2]
    return cut

```

- 함수 정의 :
 전구 영역 GEOS 도법 자료의 위도, 경도, 픽셀값 행렬을 사용자 정의 영역대로 잘라내는 함수

입력 변수
 -Array: 전구 영역 GEOS 도법 자료의 image_pixel_values/위경도 행렬 [array/numpy array]

-Resolution: 전구 영역 GEOS 도법 자료의 해상도(km) [float]

-Latitude1: 사용자 정의영역의 좌상단에 해당하는 위도 (degree) [float]
 -Longitude1: 사용자 정의영역의 좌상단에 해당하는 경도 (degree) [float]

-Latitude2: 사용자 정의영역의 우하단에 해당하는 위도 (degree) [float]
 -Longitude2: 사용자 정의영역의 우하단에 해당하는 경도 (degree) [float]

출력
 사용자 정의 영역에 해당하는 image_pixel_values/위경도 행렬 [numpy array]

입력 변수는 항상 아래의 부등식을 만족해야한다.

Latitude1 >= Latitude2
 Longitude1 <= Longitude2

	##### #Main Program Start ##### 5 input_ncfile_path = 'gk2a_ami_le1b_sw038_fd020ge_201905100300.nc' CT_path='./conversion_table/' left_upper_lat=45.728965 left_upper_lon=113.996417 right_lower_lat=29.312252 right_lower_lon=135.246740 output_ncfile_path='output_ncfile.nc'	샘플코드 주요 처리영역 시작
6	input_ncfile = nc.Dataset(input_ncfile_path,'r',format='netcdf4') ipixel=input_ncfile.variables['image_pixel_values']	- 입력 자료 설정 >L1B 자료 파일 이름 >Calibration Table 디렉토리 >사용자정의영역 좌상단, 우하단 위경도 - 출력 자료 설정 >이진(nc)파일 이름
7	i = np.arange(0,input_ncfile.getncattr('number_of_columns'),dtype='f') j = np.arange(0,input_ncfile.getncattr('number_of_lines'),dtype='f') i,j = np.meshgrid(i,j) (geos_lat,geos_lon) = latlon_from_lincol_geos(2.0,j,i)	- image_pixel_values 추출 >netCDF4 파일 불러오기 >image_pixel_values 불러오기 ->ipixel - 위경도 계산 전구 영역 GEOS 도법 자료에서 각 픽셀별 위경도를 계산
8	cut_pixel=cut_with_latlon_geos(ipixel[:,2.0,left_upper_lat,left_upper_lon,right_lower_lat,right_lower_lon]) cut_lat=cut_with_latlon_geos(geos_lat,2.0,left_upper_lat,left_upper_lon,right_lower_lat,right_lower_lon) cut_lon=cut_with_latlon_geos(geos_lon,2.0,left_upper_lat,left_upper_lon,right_lower_lat,right_lower_lon) (ulc_lin,ulc_col)=lincol_from_latlon_geos(2.0,left_upper_lat,left_upper_lon) (lrc_lin,lrc_col)=lincol_from_latlon_geos(2.0,right_lower_lat,right_lower_lon) cut_pixel[cut_pixel>49151] = 0 #set error pixel's value to 0	- 사용자 정의 영역 추출 전구 영역 GEOS 도법 자료에서 사용자 정의 영역을 잘라냄.
9		- DQF 처리 Error pixel(III-5참조) 값을 0으로 필터링
10	channel=ipixel.getncattr('channel_name') if ((channel == 'VI004') or (channel == 'VI005') or (channel == 'NR016')): mask = 0b0000011111111111 #11bit mask elif ((channel == 'VI006') or (channel == 'NR013') or (channel == 'WV063')): mask = 0b0000111111111111 #12bit mask elif (channel == 'SW038'): mask = 0b0011111111111111 #14bit mask else: mask = 0b0001111111111111 #13bit mask cut_pixel_masked=np.bitwise_and(cut_pixel,mask)	- Bit Masking > 채널별 Bit size per pixel(III-6참조)에 따라 bit mask 생성 및 마스크 처리

```

11 AL_postfix='_con_alb.txt'
BT_postfix='_con_bt.txt'
if (channel[0:2] == 'VI') or (channel[0:2] == 'NR'):
    conversion_table=np.loadtxt(CT_path+channel+AL_postfix,'float64')
    convert_data='albedo'
else:
    conversion_table=np.loadtxt(CT_path+channel+BT_postfix,'float64')
    convert_data='brightness_temperature'
cut_pixel_masked_converted=conversion_table[cut_pixel_masked] # pixel data : table value / 1:1 matching
input_ncfile.close()

12 output_ncfile=nc.Dataset(output_ncfile_path,'w',format='NETCDF4')
data_lin_max=output_ncfile.createDimension("data_lin_max",cut_pixel.shape[0])
data_col_max=output_ncfile.createDimension("data_col_max",cut_pixel.shape[1])
output_ncfile.createVariable(convert_data,np.float32,"data_lin_max","data_col_max",)
output_ncfile.createVariable('latitude',np.float64,"data_lin_max","data_col_max",)
output_ncfile.createVariable('longitude',np.float64,"data_lin_max","data_col_max",)
output_ncfile.variables[convert_data][:]=cut_pixel_masked_converted
output_ncfile.variables['latitude'][:]=cut_lat
output_ncfile.variables['longitude'][:]=cut_lon
output_ncfile.left_upper_lin_col_from_geos="line number(start from 0):"+str(int(np.floor(ulc_lin)))+ " column number(start from 0):"+str(int(np.floor(ulc_col)))
output_ncfile.right_lower_lin_col_from_geos="line number(start from 0):"+str(int(np.ceil(lrc_lin)))+ " column number(start from 0):"+str(int(np.ceil(lrc_col)))
output_ncfile.close()

#####
#End of Program
#####

```

- Albedo/BT 변환
> 픽셀 값을 알베도/밝기온도로 변환
(IV-5참조)

- NetCDF4 형태로 출력
>output_ncfile.nc

프로그램 종료



붙임

붙임 1. NetCDF4

붙임 2. 샘플코드 구동환경

붙임 3. 샘플코드1 출력 파일(ASCII)

붙임 4. Conversion Table



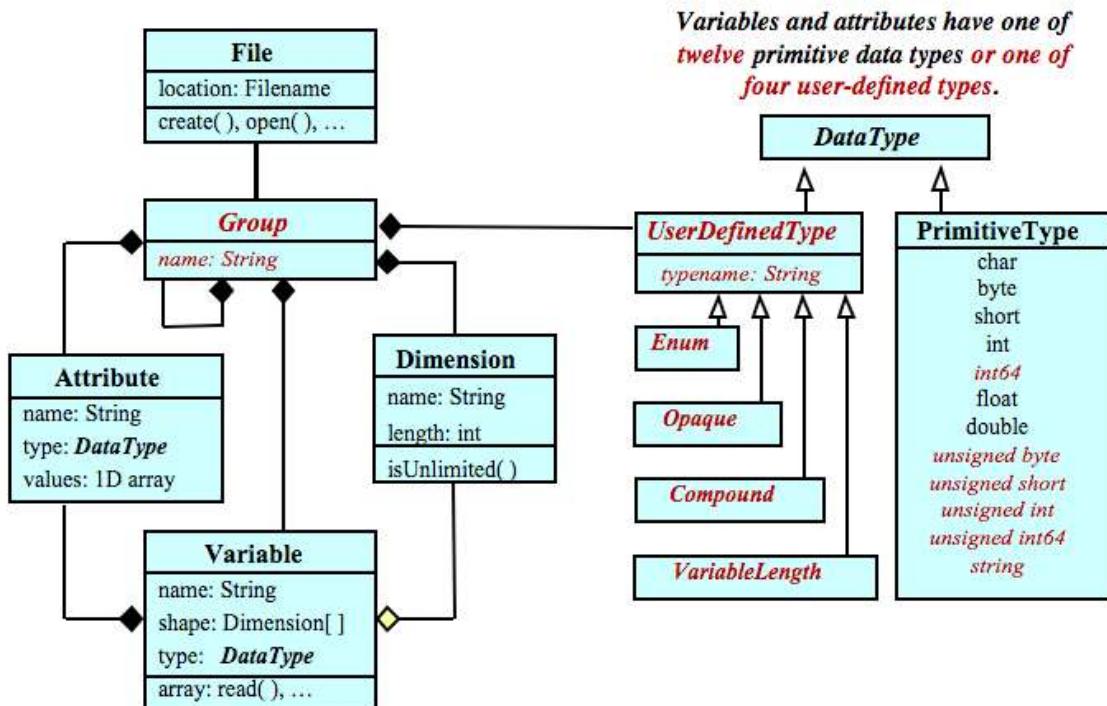
1. NetCDF4

NetCDF4는 자기기술성을 지닌 데이터 포맷의 한 종류이다. HDF5를 기반으로 하고 있기 때문에 HDF5와 일부 호환성을 지닌다.

NetCDF4 형식은 C, Fortran, Python 등 다양한 프로그래밍 언어를 지원한다.

NetCDF4/HDF5의 데이터 모델을 그림 15에서 확인할 수 있다.

그림 15 NetCDF4/HDF5 Data Model (www.unidata.ucar.edu)



A file has a top-level unnamed group. Each group may contain one or more named subgroups, user-defined types, variables, dimensions, and attributes. Variables also have attributes. Variables may share dimensions, indicating a common grid. One or more dimensions may be of unlimited length.

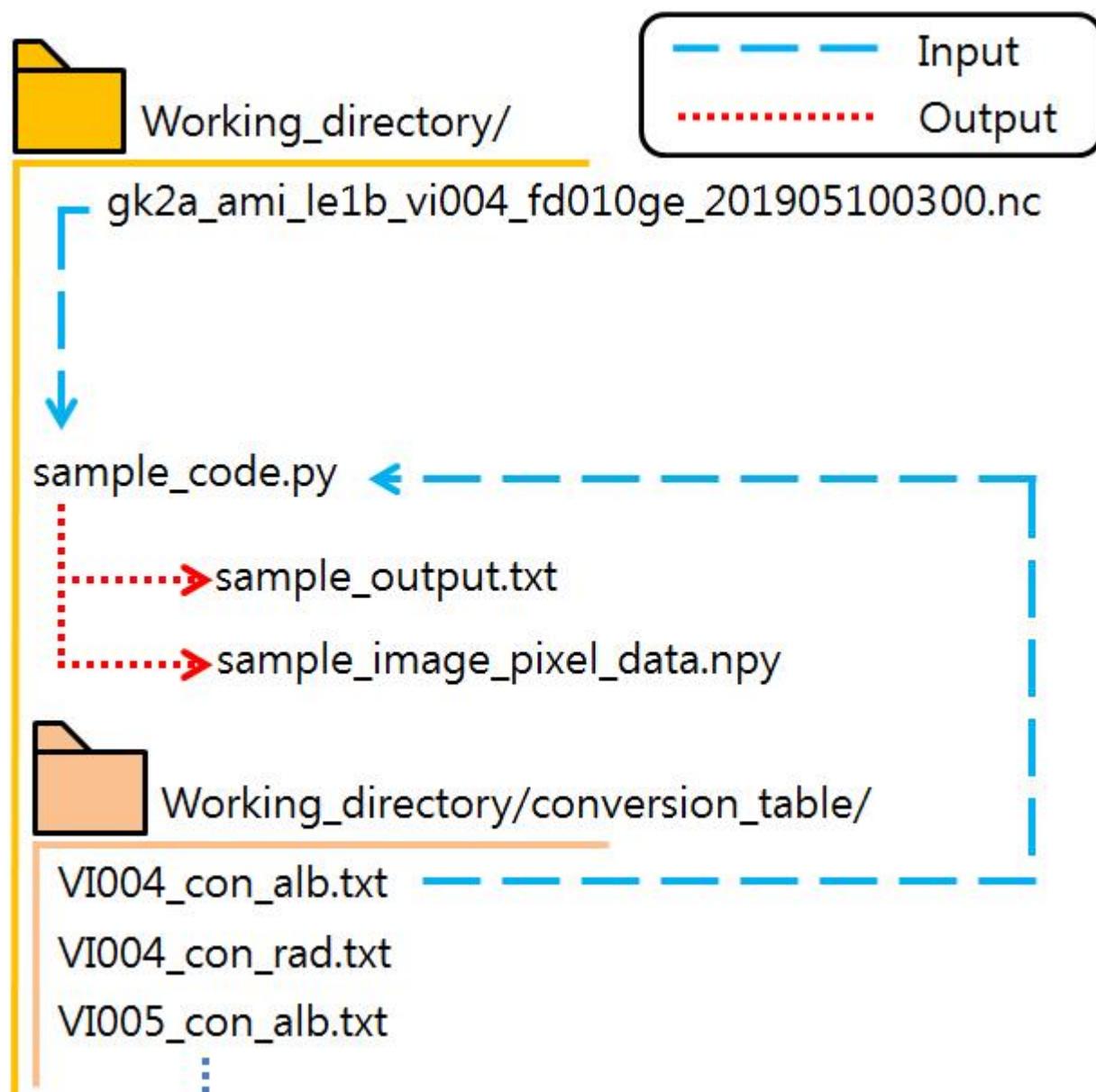
2. 샘플코드 구동 환경

- 샘플코드1 구동환경

표 12 샘플코드1 구동 환경 및 입출력 파일 정보

구동 환경			
OS			CentOS Linux 7.3.1611 (Core)
			kernel-3.10.0-514.e17.x86_64
작성언어	Python 2.7.13	라이브러리	netCDF4 1.2.4 numpy 1.14.2
입력 / 출력 정보			
입력파일	gk2a_ami_le1b_vi004_fd010ge_201905100300.nc	conversion_table/VI004_con_alb.txt	
출력파일	sample_output.txt		sample_image_pixel_data.npy

그림 16 샘플코드1 구동 디렉토리 정보

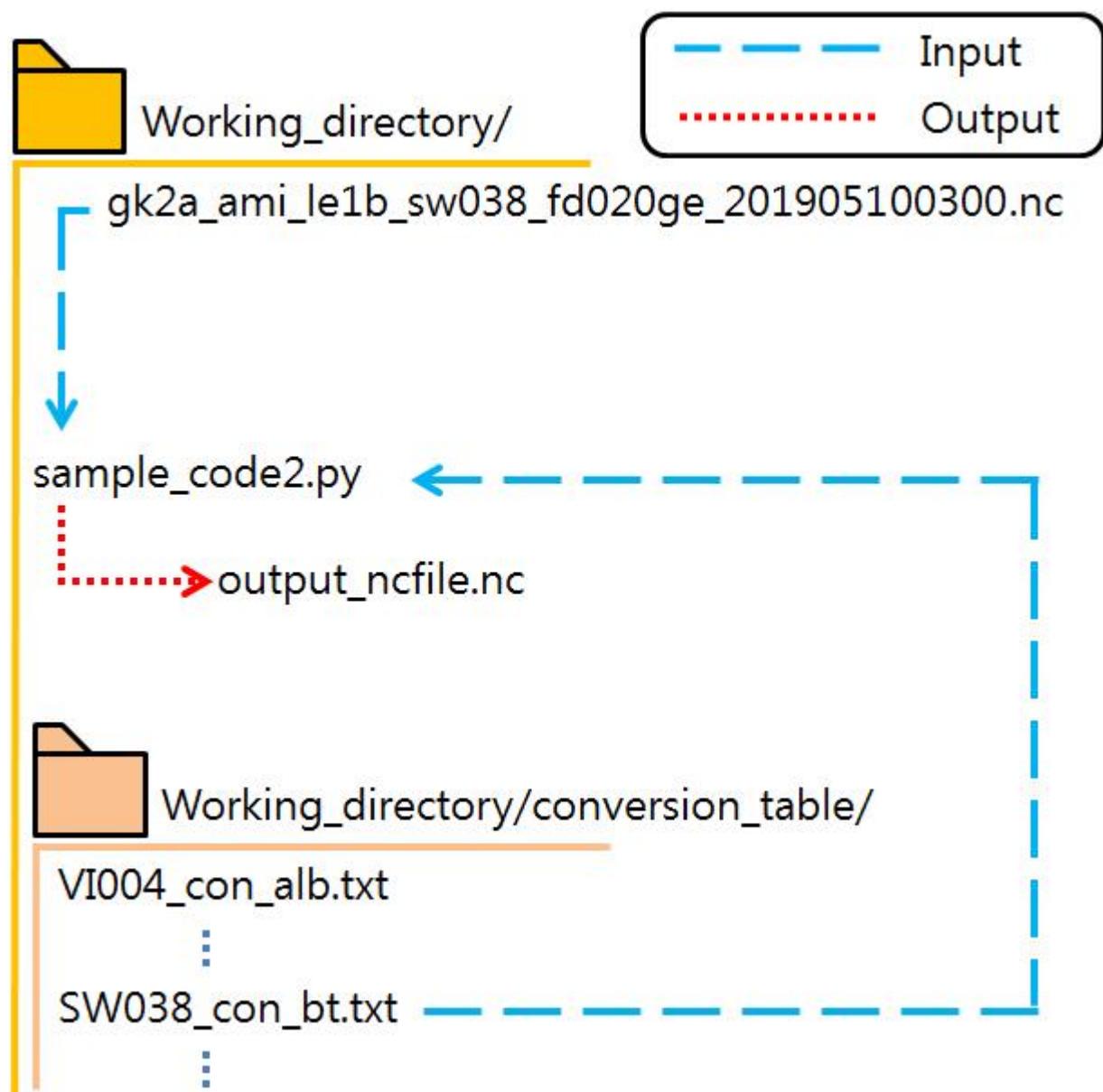


- 샘플코드2 구동환경

표 13 샘플코드2 구동 환경 및 입출력 파일 정보

구동 환경			
OS	CentOS Linux 7.3.1611 (Core) kernel-3.10.0-514.e17.x86_64		
작성언어	Python 2.7.13	라이브러리	netCDF4 1.2.4 numpy 1.14.2
입력 / 출력 정보			
입력파일	gk2a_ami_le1b_sw038_fd020ge_201905100300.nc		conversion_table/SW038_con_bt.txt
출력파일	output_ncfile.nc		

그림 17 샘플코드2 구동 디렉토리 정보



3. 샘플코드1 출력 파일(ASCII) : sample_output.txt

output file 형식

번호	<tab>	variable 명칭(=key)	<tab>	(x 차원, y 차원)	<tab>	variable 자료형
----	-------	-------------------	-------	----------------	-------	--------------

sample_output.txt

number of variable keys : 65

0	image_pixel_values	(u'dim_image_y', u'dim_image_x')	uint16
1	brightness_object_avoidance	(u'dim_boa_swaths',)	uint8
2	sc_position	(u'dim_1',)	uint8
3	sun_position	(u'dim_1',)	uint8
63	gsics_coeff_start_time_of_validity_period	(u'dim_1',)	float64
64	gsics_coeff_end_time_of_validity_period	(u'dim_1',)	float64

<중략>

4. Conversion Table

샘플코드에서 사용하는 conversion_table 폴더의 파일들은 Conversion Table 원본 파일에서 추출한 데이터이다. 표 14에서 보정 계수(Calibration Coefficients)와 방정식(Calibration Equation)을 확인할 수 있다.

파일별로 각 줄마다 번호(Count)에 해당하는 Radiance/Albedo/Effective Brightness Temperature/Brightness Temperature 값이 저장되어 있으며, 이는 'image_pixel_values'의 값과 일대일로 대응된다.

VI004_con_alb.txt (v3.0 / 2048개 값 존재)

```
-0.011329851
-0.010763357
-0.010196864
-0.009630370
-0.009063877
-0.008497383
-0.007930890
<중략>
1.144883247
1.145449740
1.146016233
1.146582727
1.147149220
1.147715714
```

그림 18 Conversion Table 원본 캡처(v3.0 / 일부)

Count	VIS 0.4		VIS 0.5		VIS 0.6		VIS 0.8	
	Radiance [W/m ² srμm]	Albedo [%]						
0	-7.270905	-0.011330	-6.872498	-0.011405	-6.194244	-0.011921	-3.657928	-0.011970
1	-6.907359	-0.010763	-6.528872	-0.010835	-6.039388	-0.011623	-3.612204	-0.011821
2	-6.543813	-0.010197	-6.185247	-0.010265	-5.884532	-0.011325	-3.566480	-0.011671
3	-6.180267	-0.009630	-5.841621	-0.009695	-5.729676	-0.011027	-3.520756	-0.011521
4	-5.816721	-0.009064	-5.497996	-0.009124	-5.574819	-0.010729	-3.475032	-0.011372
5	-5.453176	-0.008497	-5.154370	-0.008554	-5.419963	-0.010431	-3.429308	-0.011222
6	-5.089630	-0.007931	-4.810745	-0.007984	-5.265107	-0.010133	-3.383583	-0.011072
7	-4.726084	-0.007364	-4.467119	-0.007414	-5.110250	-0.009835	-3.337859	-0.010923
8	-4.362538	-0.006798	-4.123494	-0.006843	-4.955394	-0.009537	-3.292135	-0.010773
9	-3.998992	-0.006231	-3.779868	-0.006273	-4.800538	-0.009239	-3.246411	-0.010624
10	-3.635446	-0.005665	-3.436243	-0.005703	-4.645681	-0.008941	-3.200687	-0.010474
11	-3.271901	-0.005098	-3.092617	-0.005132	-4.490825	-0.008643	-3.154963	-0.010324
12	-2.908355	-0.004532	-2.748992	-0.004562	-4.335969	-0.008345	-3.109238	-0.010175
13	-2.544809	-0.003965	-2.405366	-0.003992	-4.181113	-0.008046	-3.063514	-0.010025

표 14 GK-2A AMI Calibration Coefficients and Calibration Equation (v3.0/20190415)

GK-2A AMI Calibration Coefficients							
Channel	Center Wavelength(um)	DN to Rad Gain	Rad to DN Gain	DN to Rad Offset	Rad to DN Offset	Rad. to Alb(c')	
VIS0.4	0.4702	0.363545805215835	2.750685018649320	-7.270904541015620	19.999968193001000	0.0015582450	
VIS0.5	0.5086	0.343625485897064	2.910145030102800	-6.872497558593750	19.999964614535200	0.0016595767	
VIS0.6	0.6394	0.154856294393539	6.457599956890890	-6.194244384765620	39.999952272034100	0.0019244840	
VIS0.8	0.8630	0.045724172145128	21.870270211257300	-3.657928466796870	79.999883982297800	0.0032723873	
NIR1.3	1.3740	0.034687809646130	28.828571483803000	-1.387512207031250	39.999994845049700	0.0087081313	
NIR1.6	1.6092	0.049800798296928	20.079999401569400	-0.996017456054687	20.000029921530800	0.0129512876	
Channel	Center Wavenumber (cm-1)(v)	DN to Rad Gain	Rad to DN Gain	DN to Rad Offset	Rad to DN Offset	Teff to Tbb	
IR3.8	2612.677373521110	-0.001082965172 82724000	-923.3907286135 09000	17.699987411499 00000000	16344.004272354 000000	-0.447843939824 124	1.000655680903 890
IR6.3	1617.609242531340	-0.010891467332 83990000	-91.81499328238 4000	44.177703857421 80000000	4056.1755829003 30000	-1.762794940111 470	1.004149105622 780
IR6.9	1441.575428760170	-0.008187798783 18309000	-122.1329476310 41000	66.748077392578 10000000	8152.1394406604 00000	-0.334311414359 106	1.000973598744 680
IR7.3	1365.249992024440	-0.009698271751 40380000	-103.1111548153 16000	79.060852050781 20000000	8152.0557556388 70000	-0.061312485969 660	1.000190087229 410
IR8.7	1164.949392856340	-0.014480655081 57010000	-69.05764928222 9300	118.05090332031 200000000	8152.3178789444 70000	-0.141418528203 155	1.000522329068 850
IR9.6	1039.960216776110	-0.017843546345 82990000	-56.04267114948 8400	145.46487426757 800000000	8152.2401123795 50000	-0.114017728158 198	1.000473805854 020
IR10.5	966.153383926055	-0.019819695502 51960000	-50.45486192625 2800	161.58013916015 600000000	8152.5036113503 80000	-0.142866448475 177	1.000640695720 490
IR11.2	891.713057301260	-0.021674485877 15620000	-46.13719585634 7800	176.71343994140 600000000	8153.0625890256 00000	-0.249111718496 148	1.001211668737 560
IR12.3	810.609007871230	-0.023379972204 56600000	-42.77165050712 5300	190.64962768554 600000000	8154.3992446797 30000	-0.458113885722 738	1.002455209755 350
IR13.3	753.590621482278	-0.024303756654 26250000	-41.14590243087 4400	198.22436523437 500000000	8156.1203913556 10000	-0.093852156852 766	1.000539821129 660
Calibration Equation							
VNIR	Radiance = Gain * Count + Offset				Radiance		[W/(m ² srμm)]
	Radiance to Albedo = Radiance * c'				Albedo		[%]
IR	Radiance = Gain * Count + Offset				Radiance		[mW/m ² /sr/cm ⁻¹]
	Te = (hc/k*(v*100)) / { ln((2hc ²) * (v*100) ³ / (R*(10 ⁻⁵)) +1) }				Effective Brightness Temperature		[K]
	Tb = c0 + c1*Te + c2*Te ²				Brightness Temperature		[K]
light speed				=	2.9979245800E+08		[m/s]
Planck's constant (h)				=	6.6260695700E-34		[m ² kg/s]
Boltzmann constant (k)				=	1.3806488000E-23		[m ² kg/s ² /K]
c1 (2hc ²)				=	1.1910428681E-16		
c2 (hc/k)				=	1.4387769600E-02		