Lens & Image Sensing System of Cube

Feature Overview

Integrated System:

- l Customized Ruggedized telephoto lens by customizing team of Schneider Optics
- l 28MP 20 bit color sensor
- l Capable data processing and storage board

Image Acquisition

- l 6644 (H) x 4452 (V) pixels (total)
- l 6576 (H) x 4384 (V) pixels (active)
- l 42.74x32.51mm sensor size

Lens Performance:

- l Ruggedized industrial lens
- l 35mm optical format 240mm focal length lens / F1.9
- l Meter pixel data charts attached below

Data Processing:

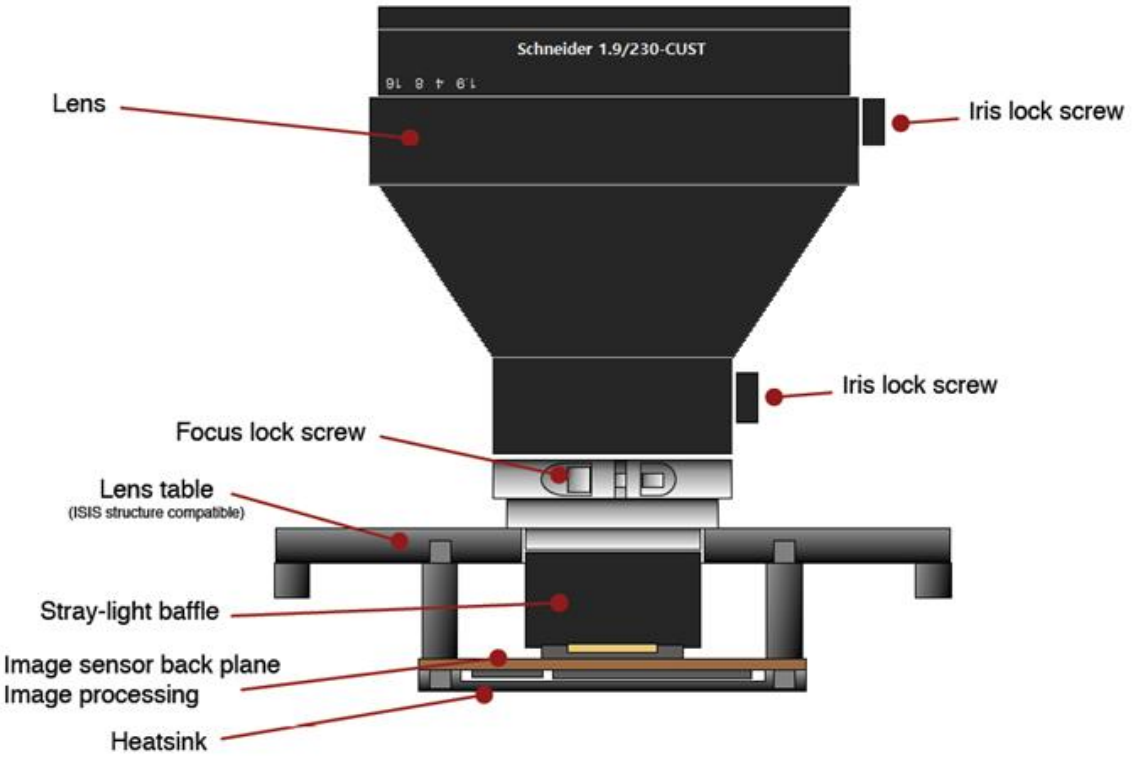
- l 4 GB on board RAM
- l 4.8 TB solid state storage
- l Demosaicing (Bayer to RGB conversion)
- l JPEG compression

Interface:

- l CSP-enabled I2C for command and data transfer
- l CSP-enabled serial port with text-based console

Mechanical:

- l Envelope: 96mm x 90mm x 100 mm (Manual Telephoto Lens)
- l Mass: 1660g



Imaging System Design Motivation

The goal of the lens and image sensing system of Cube Eyes satellites is to capture Earth ground images with a spatial resolution of at most 10 meters per pixel. Taking the NanoCam C1U base system and customizing it with the specified lens and sensor will result in a system with a resolution of 2.59 meter/pixel at 160Km of altitude to 24.3 meter/pixel at 1500Km of altitude. As CubeEyes aims to launch to an orbit of 450 km, this should provide a resolution of exactly 10 meters/pixel. result of law image size is over 30Mb and it reduced to 3.8Mb after image compression

Mechanical Description

As a result, our customized image system perfectly fits to 3 cube size. It has 8.148974 degree of horizontal field of view and 6.202887 degree of vertical field of view, with minimum 2.59meter/pixel to maximum 24.3meter/pixel.

Data/Calculation results (No compression)

Altitude [Km]	160	200	300	400	500	600
FOV [Km]	22.79	28.49	42.74	56.98	71.233	85.48
meter/pixel	3.466	4.33	6.499	8.6658	10.8323	12.9987

Picture area [Km^2]	141.29	176.16	264.92	353.23	441.5385	529.84
picture count - non consider overlap	3620200	2896160	1930773	1448080	1158464	965386
total size [TB] at RAW/CR2	103.78	82.86	55.24	41.43	33.13	27.62
total size [TB] after compression	13.11	10.49	6.997	5.2477	4.198	3.498

Altitude [Km]	700	800	900	1000	1100	1200
FOV [Km]	99.7266	113.9733	128.22	142.466	156.713	170.96
meter/pixel	15.165	17.3317	19.498	21.6646	23.8311	25.9975
Picture area [Km^2]	618.15	706.46	794.769	883.077	971.384	1059.69
picture count - non consider overlap	827474	724040	643591	579232	526574	482693
total size [TB] at RAW/CR2	23.67	20.71	18.41	16.57	15.07	13.81
total size [TB] after compression	2.998	2.623	2.3323	2.099	1.908	1.749

Matlab Codes

```
clear all
close all

%% basic Camera
% Gomspace - NANOCAM C1U
% 3MP
% 2048 pixels x 1536 pixel
% 9.22 IFOV
% 35mm Focal length (Fw)

%% LEO
% 160~1500km

%% Customized sensor - KAI-29050 image sensor
% 35mm optical format
% 36.17 mm (H) x 24.11 mm (V)
% Total Number of Pixels 6644 (H) x 4452 (V)
```

```
% Number of Active Pixels 6576 (H) x 4384 (V) = 28.829Mpixel
% Pixel Size 5.5 um (H) x 5.5 um (V)
% sensor size ¡æ 42.74x32.51 mm

h_pixel=6576;
v_pixel=4383;

h_size=42.74;    % [mm]
v_size=32.51;    % [mm]

fl=230;          % [mm]
                 % customized Lens Focal Length for LOW angle of FOV.
increased fl then lower FOVs
raw_size=30;     %[mb]
pic_size=3.8;    %[mb]
%% function - IFOV value [Degree] / H value [Km] needed for fov_cal
funct.

%%%%%%%%%%
h=1200; %[Km]
fov_cal(h,h_size,v_size,h_pixel,v_pixel,fl,pic_size)




function fov_cal(h,h_size,v_size,h_pixel,v_pixel,fl,pic_size)
%This is Km based%
%h_pixel - holizon pixel
%v_pixel - vertical pixel
%ifov - Instantaneous field of view

d_sensor=sqrt(h_size^2+v_size^2);
fprintf('\n Sensor Diagonal Size= %f mm',d_sensor);


h_fov=2*(atan((h_size)/(2*fl)))*180/pi;
v_fov=2*(atan((v_size)/(2*fl)))*180/pi;
fprintf('\n Horizontal AOV= %f degree',h_fov);
fprintf('\n Vertical AOV= %f degree',v_fov);
fprintf('\n');
%AOV=GFIOV --> same meaning. some of article use FOV for AOV...

ifov=h_fov; % Only will use h_fov cuz h_fov > v_fov. then reduce
satellite by using bigger(10 degree max) fov
            % if h_fov is bigger than 10 degree, then use v_fov

ifov_rad=ifov*pi/180;
```

```
scan_angle=0;          %%%%%%%%%%%%%%%%%%%%%%%%default
%gifov= h*tan(ifov);
fov=2*h*tan(scan_angle+ifov_rad/2);


fprintf('\n Distance from Satellite to Earth surface = %d Km',h);
%fprintf('\n IFOV = %d degree',ifov);
%printf('\n GIFOV = %f Km',gifov);
fprintf('\n FOV = %f Km',fov);
%fprintf('\n');

%fprintf('\n Customized sensor/lens has %dx%d
[pixels]',h_pixel,v_pixel);
w=fov/h_pixel; % wide -
l=fov/v_pixel; % length -
%fprintf('\n Horizontal Pixel= %f m',w*1000);
%fprintf('\n Vertical Pixel= %f m\n',l*1000);

fprintf('\n Customized sensor/lens has %f m/pixel',w*1000);
fprintf('\n');
e_radius=6380; % [km]
e_surface=4*pi*6380^2;

vfov_rad=v_fov*pi/180;
vfov=2*h*tan(scan_angle+vfov_rad/2);

d_pic=ifov*vfov;
n_pic=e_surface/(d_pic);
fprintf('\n one picture shows %f sqKm area',d_pic);
fprintf('\n %f pics needed for whole earth surface',n_pic);
totalrawsize=raw_size*n_pic;
fprintf('\n total RAW/CR2 picture size per day = %f
Tb',totalrawsize/1024^2);
totalsize=pic_size*n_pic;
fprintf('\n total compressed picture size per day = %f
Tb',totalsize/1024^2);
fprintf('\n')

orbital_speed = sqrt(398594433600000/(6380000+h));  % Km/sec
orbital_period =
2*pi*(6371000+h)/(sqrt(398594433600000/(6380000+h)))/3600; % hour
fprintf('\n satellite orbital speed = %d Km/sec',orbital_speed);
fprintf('\n satellite orbital period = %d Hour',orbital_period);
fprintf('\n ==================================================
\n');

end
```

Reference

1). http://www.onsemi.com/pub_link/Collateral/KAI-29050-D.PDF
2).http://www.onsemi.com/PowerSolutions/product.do?id=KAI-29050
3).https://www.schneideroptics.com/Ecommerce/CatalogItemDetail.aspx?CID=1758&IID=9168
4).http://www.clyde-space.com/cubesat_shop/structures/1u_structures/115_1u-chassis-walls
5).https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CCEQFjAAahUKEwjBmPX_1_DGAhUEVj4KHe0IAHg&url=http%3A%2F%2Fnativesat.com%2Fwp-content%2Fuploads%2F2014%2F07%2FNANOCAM-6.1.pdf&ei=A5KwVYGLHISs-QHtkYDABw&usg=AFQjCNFDOslHpinfP4GXh6jP3EjcwiyuEg&sig2=mJv1PAV2R3amg12OEOziuw&bvm=bv.98476267,d.cWw