USC Viterbi
School of Engineering

USC

Predicting the Position of the Sun, across Earth's Horizon, prior to Sunrise, using Image Processing

Omair A. Rahman **Daniel A. Erwin**

Division of Astronautical Engineering
University of Southern California, USA

Outline

USC Viterbi School of Engineering

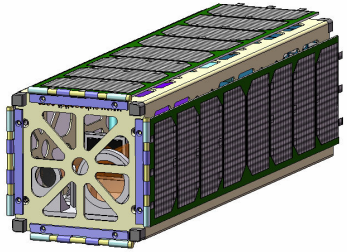
- Background
- Motivations
- Prior Work
- Our Solution
- Results

- **Background**
- Motivations
- Prior Work
- Our Solution
- Results

USC CUBESAT: OZ MISSION

USC Viterbi School of Engineering

- **Mission Objective**
  - The measurement of the Ozone Column in the Earth's atmosphere, OZMOSIS

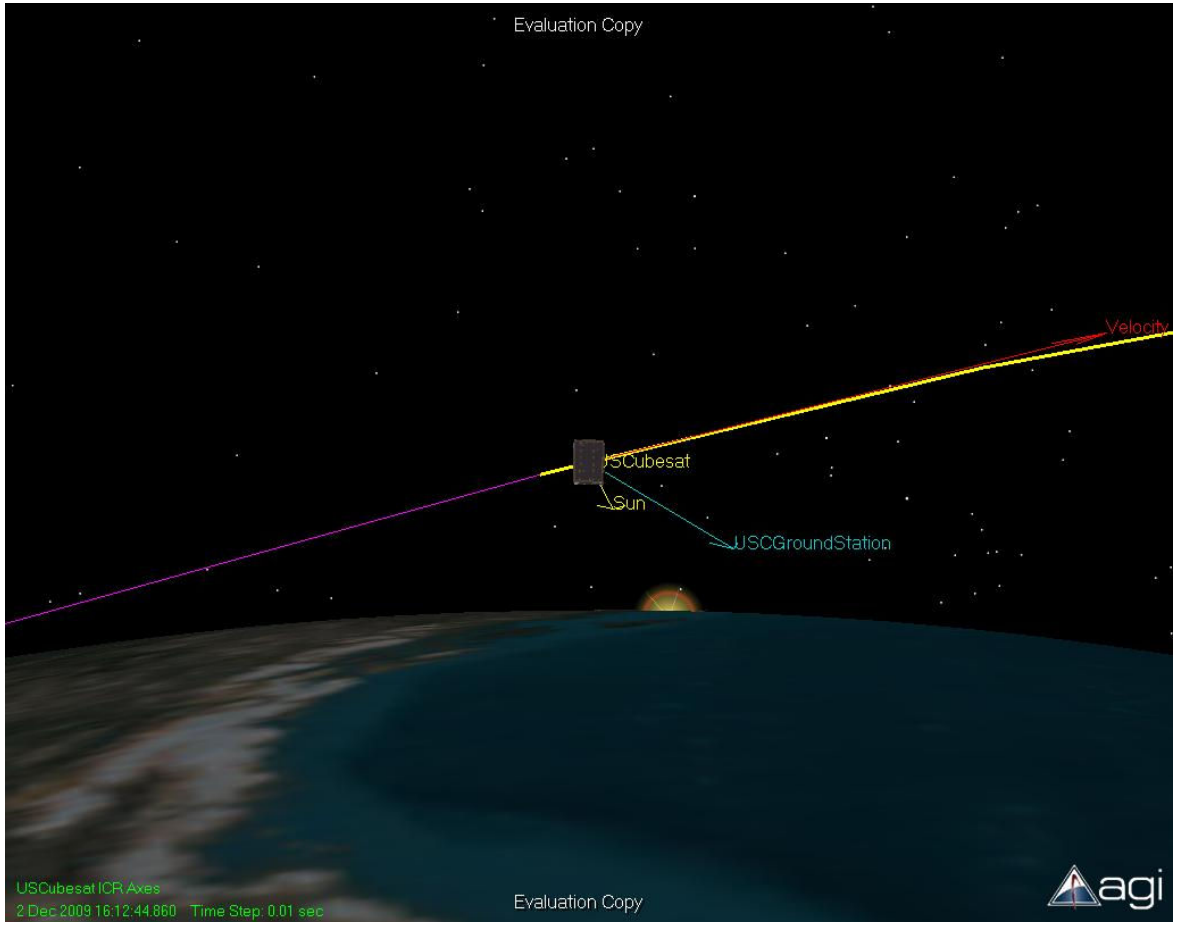


- **Method**
  - Measure the ratio of the intensity of light in the visible spectrum to the UV rays from the Sun – through the Earth's atmosphere – at Sunrise and Sunset
- **Platform**
  - Use a Three Axis Stabilized, sun pointing, 3U CubeSat

SYSTEM DESCRIPTION

USC Viterbi School of Engineering

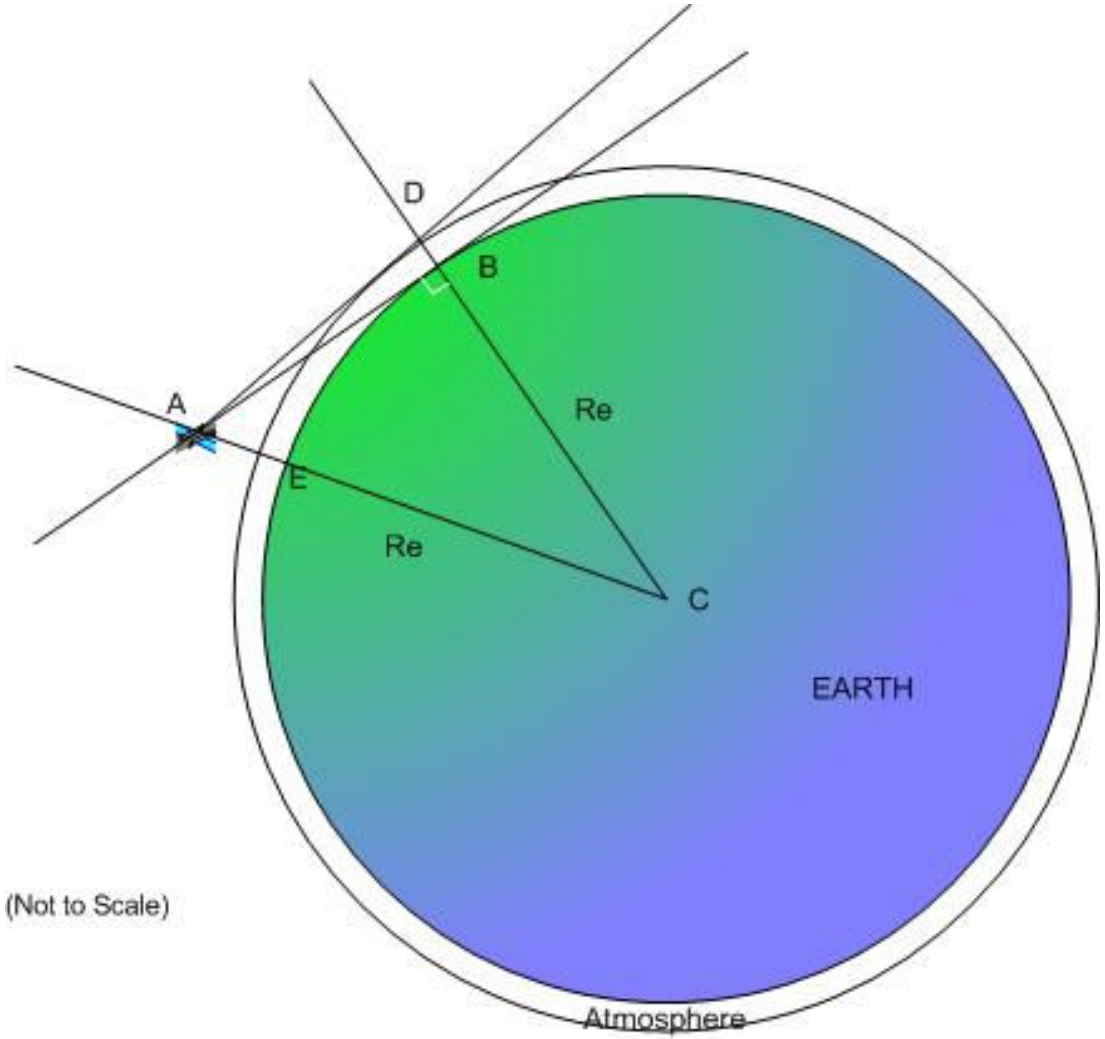
Mission Illustration



SYSTEM DESCRIPTION

USC Viterbi School of Engineering

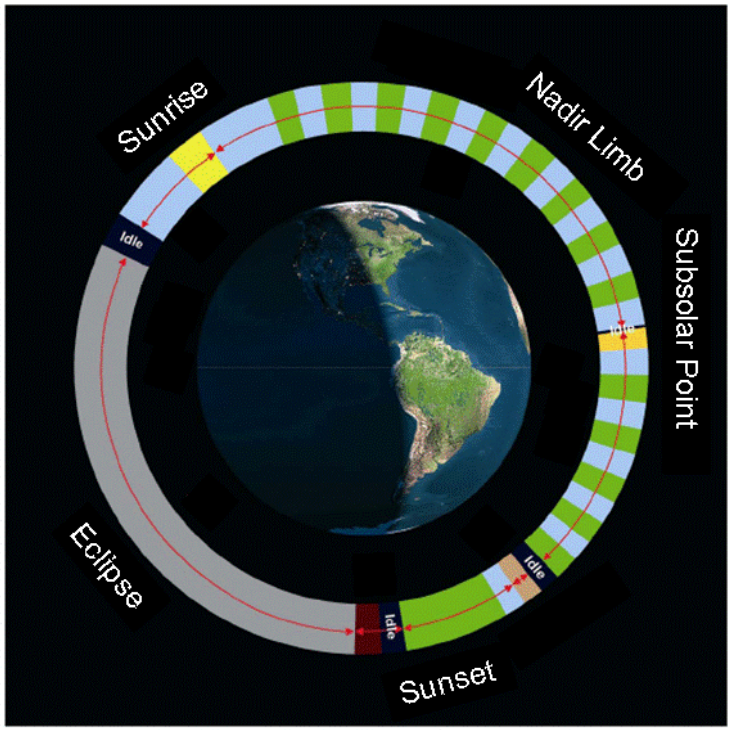
Mission Illustration

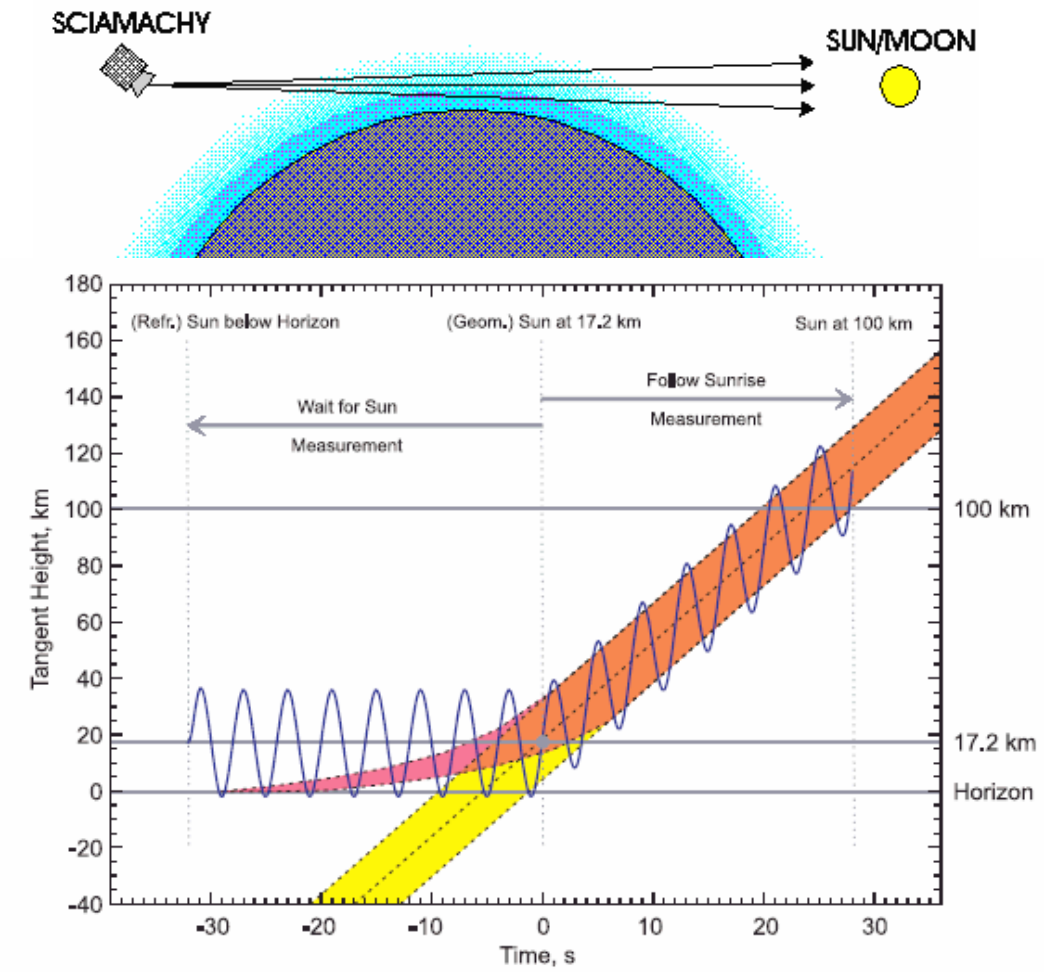


SYSTEM DESCRIPTION

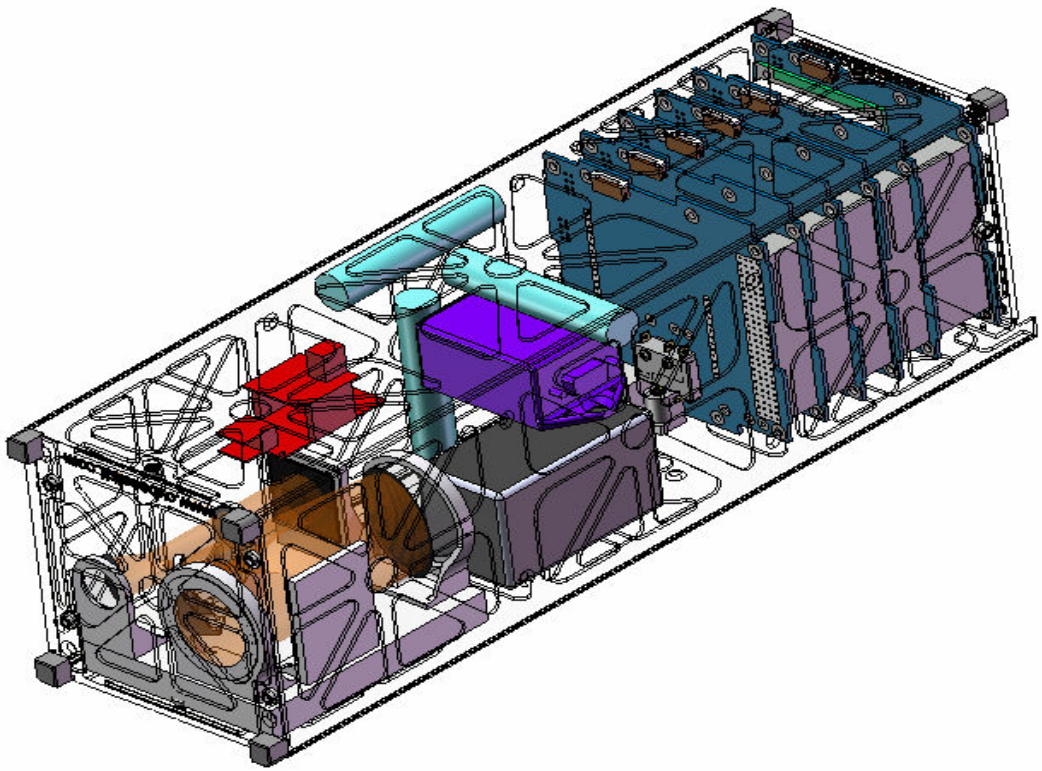
USC Viterbi School of Engineering

Mission Illustration

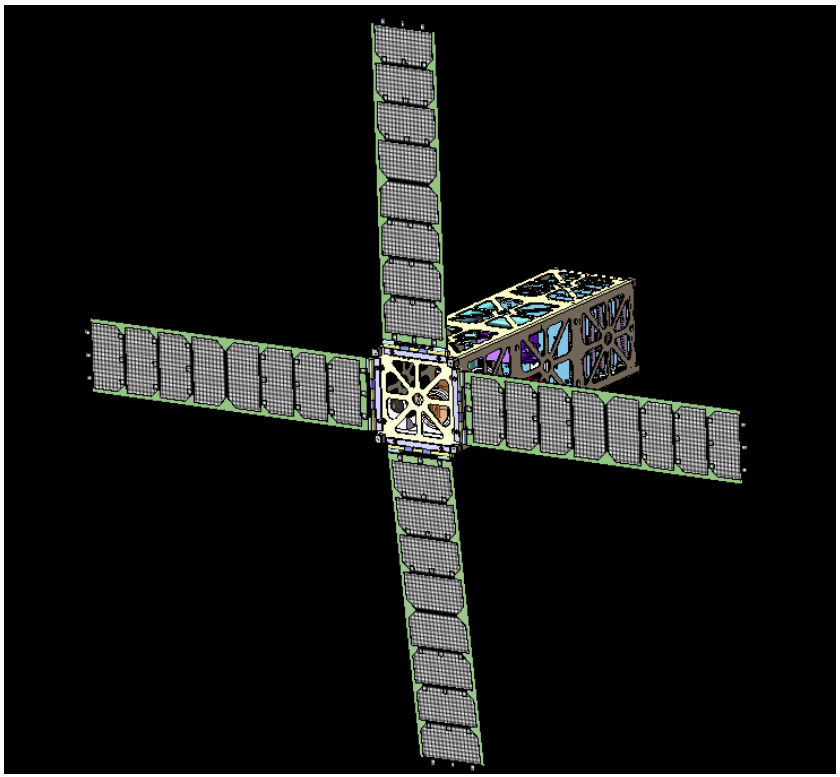




CubeSat Internal View



CubeSat Deployed View



- Background
- **Motivations**
- Prior Work
- Our Solution
- Results

MOTIVATIONS

USC Viterbi School of Engineering

- **Image Processing is a highly computation intensive task**
- **Serial Implementations of Image Processing are very slow due to large n*n Matrix manipulations**
- **Embedded Control Systems require faster data processing than the onboard microcontrollers can provide**
- **Nano(Pico)-Satellites are prohibitively small for legacy sensors, so innovative new ideas are required**
- **The idea of putting a system in space is awesome**
- **Complete Image acquisition, filtering and data processing in under 500ms at relatively low power**

- Background
- Motivations
- **Prior Work**
- Our Solution
- Results

Prior Work on Ozone Measurement

- **SCIAMACHY** (SCanning Imaging Absorption SpectroMeter for Atmospheric CartograpHY)
  - SCIAMACHY is an spectrometer instrument aboard ENVISAT launched by ESA in March 2002
  - SCIAMACHY has three different viewing geometries which yield total column values as well as distribution profiles in the stratosphere
- **POAM III** (Polar Ozone and Aerosol Measurement)
  - POAM III was launched on the SPOT 4 satellite in March 1998 and measures atmospheric transmission in nine wavelength bands
  - The POAM III experiment is a visible/near infrared solar occultation instrument designed to measure aerosols and trace constituents in the polar stratosphere

Prior Work on Sensor Selection

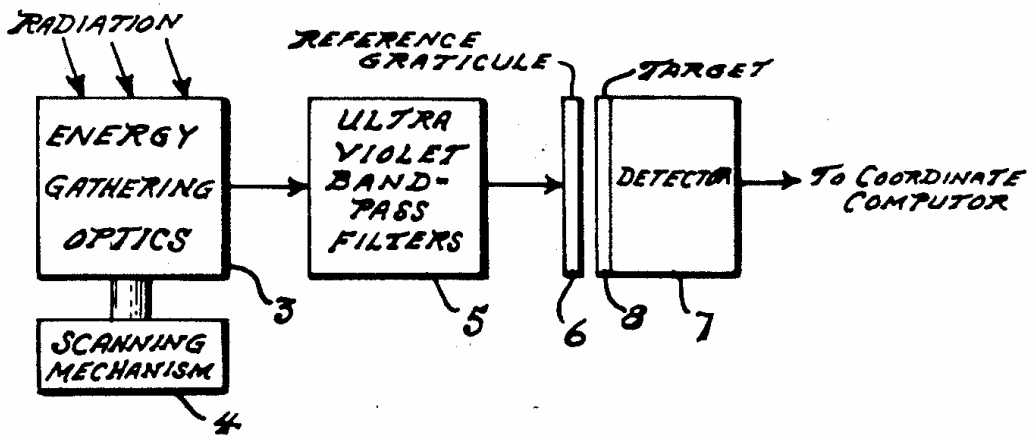
- **Star Sensors**
  - Determination of the Attitude by looking at the Stars and matching the data to a database for Navigation
  - Would not work for NanoSats, as most star-sensors have the same size as a nanosat
- **Magnetometers**
  - Gives a measurement of the magnetic field around them, which can be matched to the IGRF
  - Cannot work in Torque based Actuator systems
- **Inertial Measurement Units**
  - Inertial Measurement Systems provide rate of angular movements, which can be integrated to find attitude
  - The Drifts are too high, and without any other system, large errors would accumulate during the Eclipse time

Proposed Sensor Selection

- **Imaging Camera**
  - A low resolution camera with a wide angle lens and a mechanical aperture
  - Image Frames to be acquired and processed
  - Step 1: Find the Limb
  - Step 2: Find the most probable location of the expected sun-rise on the limb

Prior Work on Limb Pointing

- **"Space Vehicle System for determining Earth's Ultraviolet Radiation Limb", Drohan et al, United States Patent 3,715,594**
  - Abstract: An optical system is used to scan the Earth's horizon and project an image from which position information is derived for use in a space vehicle naviagation control system
  - The system utilizes the Earth's ultraviolet radiation limb as an earth-space boundary reference

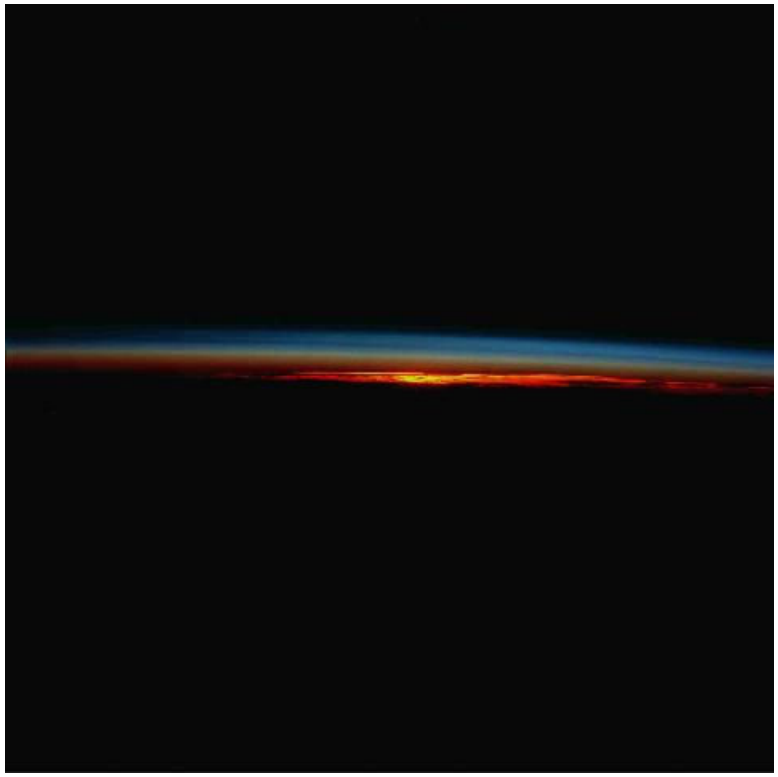


Prior Work on Image Processing

- **The Project would take some standard techniques used for image processing and have both Serial (standard software) and Parallel Implementations to incorporate into one algorithm.**
- **The following are some key previous research to be consulted for implementation of this project**
  - Parallel Image Erosion(Dilation): "Morphological Image Processing and its Parallel Implementation", He Sha, Chan Wah, ICSP 96.
  - Erosion Operations in segmented images: "Morphological Operations on Images Represented by QuadTrees", Reitsing Lin, Edward K. Wong, ICASSP 1996.
  - Original Sobel Edge Detection: "A 3x3 Isotropic Gradient Operator for Image Processing", Sobel, I., Feldman,G.
  - Parallel Sobel Edge Detection: "Performance Analysis of FPGA Based Sobel Edge Detection Operator", I. Yasri et al, ICED 2008.

Images for Processing

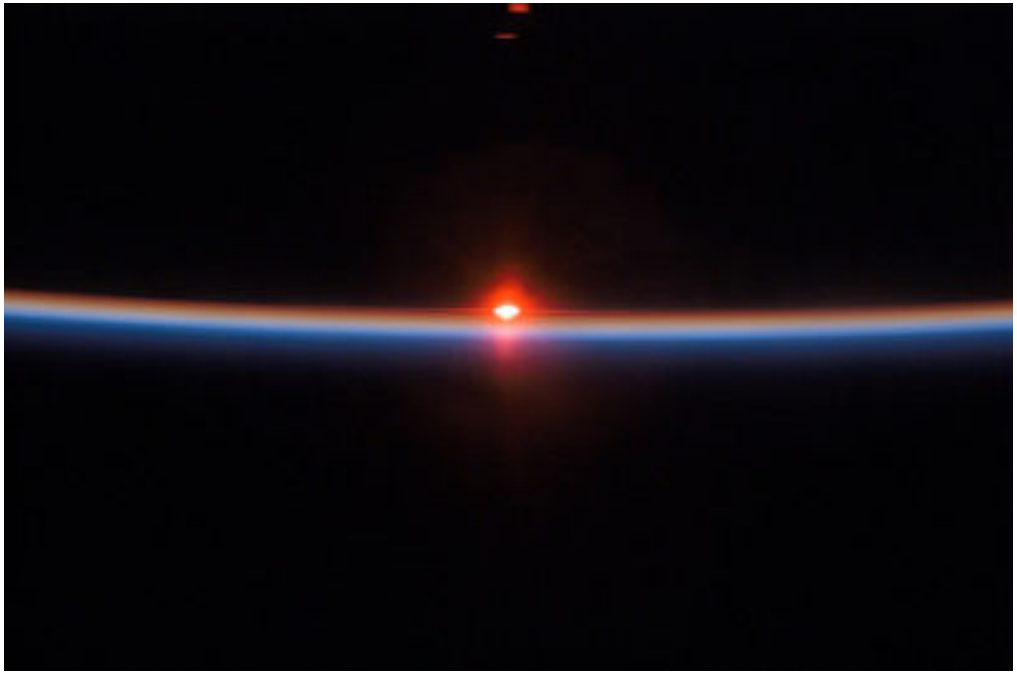
- ***Some of the images from previous space flights, used for testing the Algorithm***

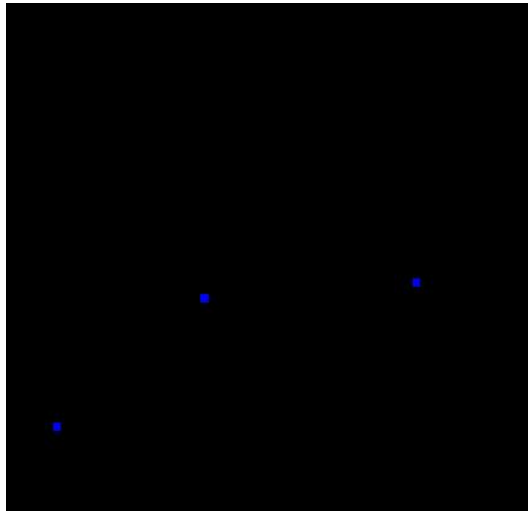


Images for Processing

Images for Processing

Images for Processing





- Background
- Motivations
- Prior Work
- **Our Solution**
- Results

SOLUTION

USC Viterbi School of Engineering

- **Prediction by Image Processing**
  - A low resolution camera with a wide angle lens and a mechanical aperture
  - Image Frames to be acquired and processed preferably every Attitude Correction Control Cycle
  - Image to be processed to find the Limb of the Earth before Sunrise, by looking for the refracted Coronal light across the Horizon
  - Once the Limb/Horizon has been found, then traversing on the edge, to find the point of highest probability of Sunrise.
  - The point of Highest Probability would be found, by looking for the highest intensity point on the coronal image
  - After each point allocation, the spacecraft would be moved so that the proposed point is in the middle of the image
  - If no limb is found then the GNC would move in an outward expanding circular helix to look for the horizon

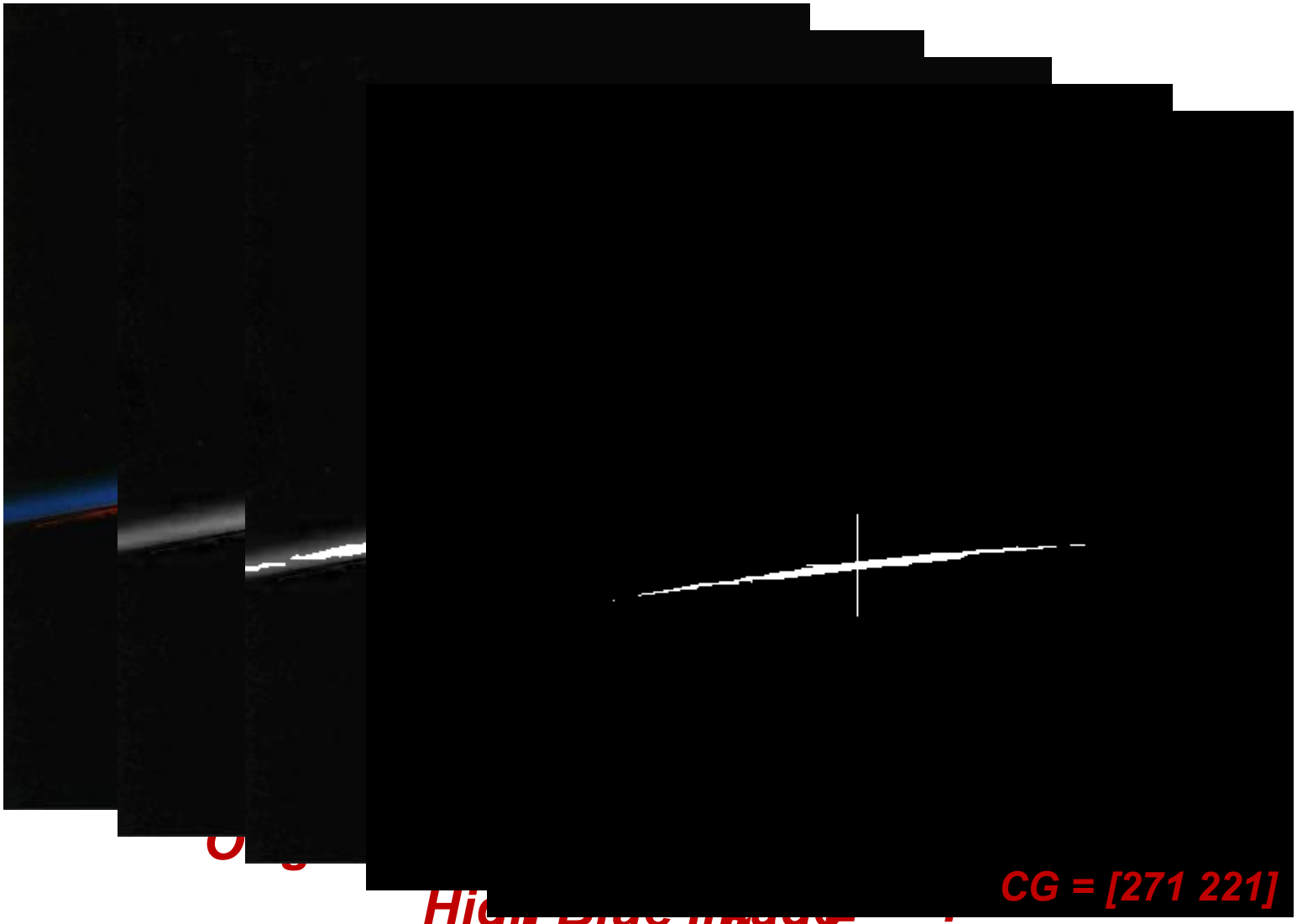
SOLUTION

USC Viterbi School of Engineering

- **The Primary Assumptions**
  - The scope of the project assumes that there is a camera that has already been interfaced in some way to the Hardware
  - The camera has an RGB interface, and we have the ability to acquire a single coloured portion of the image
  - The image is a 100x100 pixel 2-D matrix
  - Each pixel is standard 8 bit entity

- **The Proposed Algorithm**
  1. Acquire the Original Image
  2. Obtain the Blue Image from the Original Image
  3. Apply a threshold filter on the Blue Image to convert it into a monochromatic image for all pixels > threshold A
  4. Apply multiple (three) iterations of 3*3 Rectangular Element Image Erosion to fade out noise and stars
  5. Return with the Message/Flag "Not Found", and CG=Previous CG, if
     - The number of nonzero pixels in the eroded image is less than threshold B
     - The number of non-zero pixels is greater than threshold C
  6. Find the Center of Gravity of the Pixels in the Eroded Image

MATLAB RESULTS



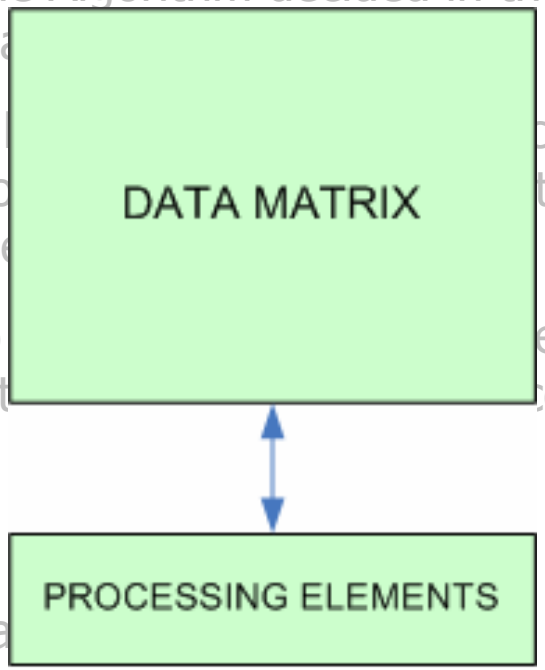
SOLUTION

USC Viterbi School of Engineering

- **Matlab Implementation**
  - Implementation done iteratively for multiple processing steps to create one algorithm for the problem set
  - A single simple m-file function named WhereWouldTheSunBe() written
  - The Image Processing Toolset used
  - Final Algorithm selected on the basis of simplicity, accuracy of results and the ability to be implemented in parallel processing environment
  - Implementation Complete and results will be presented

- **Hardware Implementation**
  - Implementation of the Algorithm decided in the Matlab prototyping on a n-row and n-column matrix of 8 bit elements
  - All operations would be circular so the architecture would not proceed forwards; actually would rewrite the matrix with the new values, thus reducing the area needed to implement
  - Addition of Non-Zero Pixels: A binary tree adder implementation, which adds a one to the Sum, if the Pixel is non-zero. Since there are n^2 pixels, then the final sum can have a max value of n^2
  - Image Erosion:
    - Since Erosion is a neighbouring operation, it can be applied in parallel on the complete matrix, therefore the step can be completed in constant time

- **Hardware Implementation**
  - Implementation of the Algorithm decided in the Matlab prototyping on a n-row and n-column ma
  - All operations would l cture would not proceed forwards; actually wo th the new values, thus reducing the area nee
  - Addition of Non-Zero er implementation, which adds a one to the Sum, if t e there are n^2 pixels, then the final sum can have a
  - Image Erosion:
    - Since Erosion is a it can be applied in parallel on the complete mat n be completed in constant time

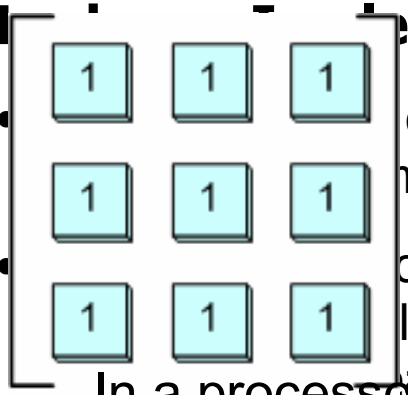


Compact Architecture

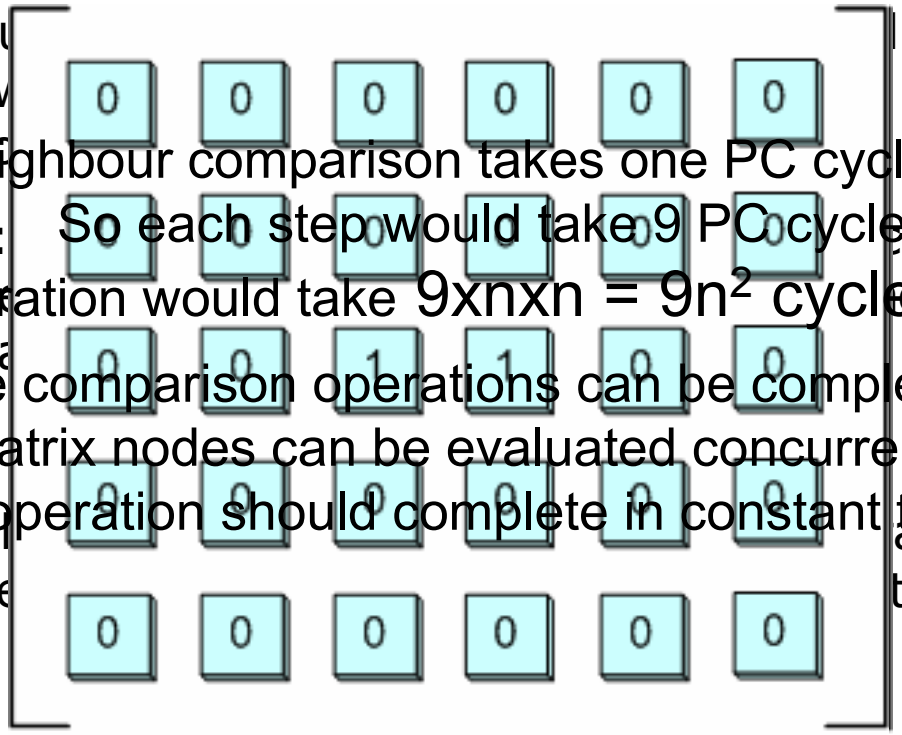
SOLUTION

USC Viterbi School of Engineering

- **H[illegible]mentation**
  - [illegible] of the Algorithm decided in the Matlab prototyping on a n-[illegible]nn matrix of 8 bit elements
  - [illegible]ould be circu[illegible]ly would rev[illegible]a needed to
  - Addition of Non-Zero Pixels: [illegible]h adds [illegible]e the final sum can have a max va[illegible]
  - Image Erosion:
    - Since Erosion is a neighb[illegible]allel on the complete matrix, the [illegible]tant time



3x3 Square Matrix



In a processor each neighbour comparison takes one PC cycle,
So each step would take 9 PC cycles,
therefore the matrix operation would take $9xnxn = 9n^2$ cycles
On a parallel hardware the comparison operations can be completed in parallel, and all the matrix nodes can be evaluated concurrently, so the operation should complete in constant time

- **Hardware Implementation**
  - Implementation of the Algorithm decided in the Matlab prototyping on a n-row and n-column matrix of 8 bit elements
  - All operations would be circular so the architecture would not proceed forwards; actually would rewrite the matrix with the new values, thus reducing the area needed to implement
  - Addition of Non-Zero Pixels: A binary tree adder implementation, which adds a one to the Sum, if the Pixel is non-zero. Since there are n^2 pixels, then the final sum can have a max value of n^2
  - Image Erosion:
    - Since Erosion is a neighbouring operation, it can be applied in parallel on the complete matrix, therefore the step can be completed in constant time

- **Hardware Implementation (contd.)**
  - Center of Gravity:
    - The first part of the calculation for the Center of Gravity would again be an adder implementation, once for rows and once for columns
    - The second part is classically division, to achieve the final X and Y CG values
  - Edge Detection (prospective):
    - An implementation of the Sobel Edge Detection using two sets of masks for Horizontal and Vertical Edges
    - Combination of the gradients for each point using approximation
    - Each application of the mask requires 9 shifts/assigns and 6 additions

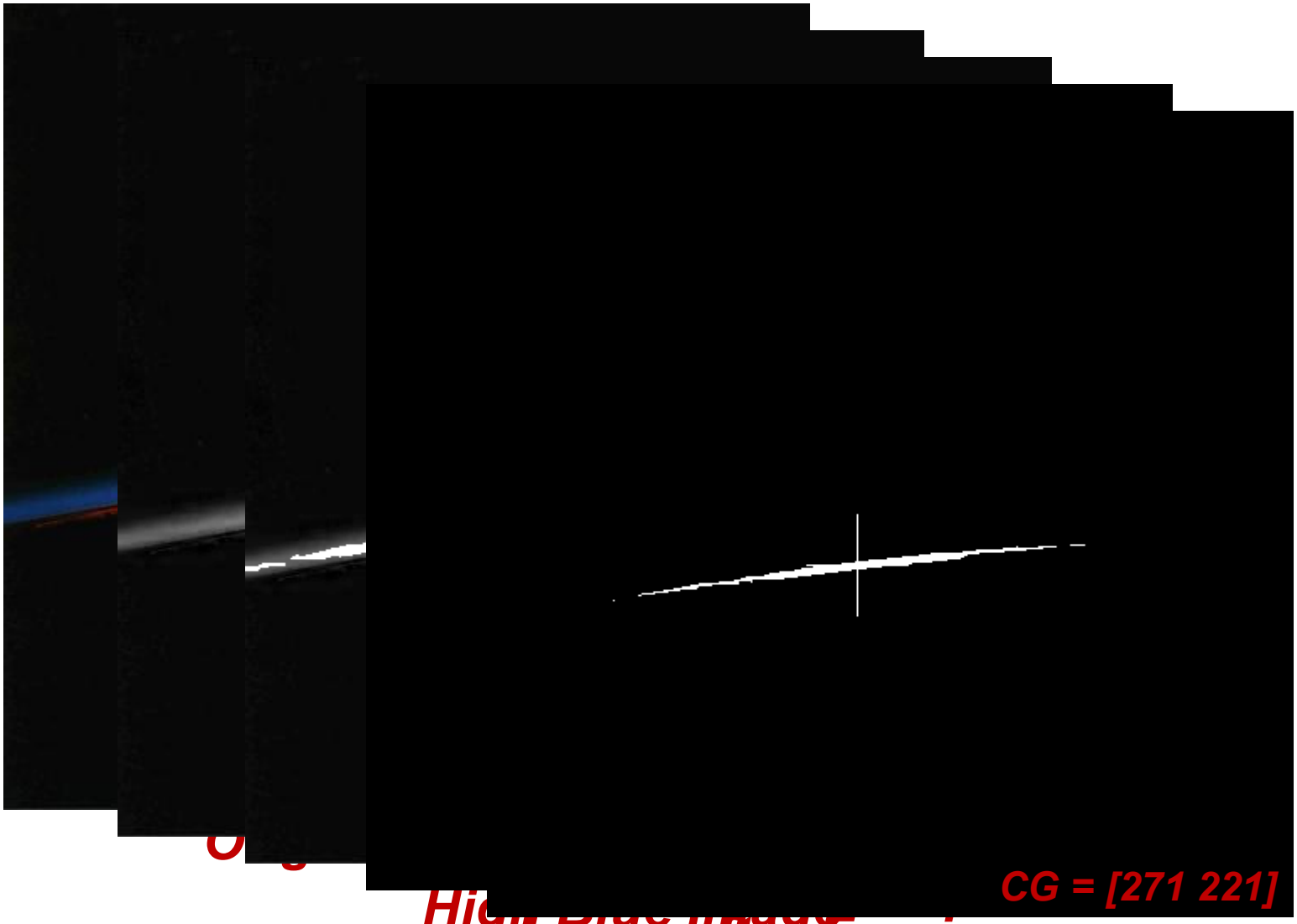
Implementation Issues

USC Viterbi School of Engineering

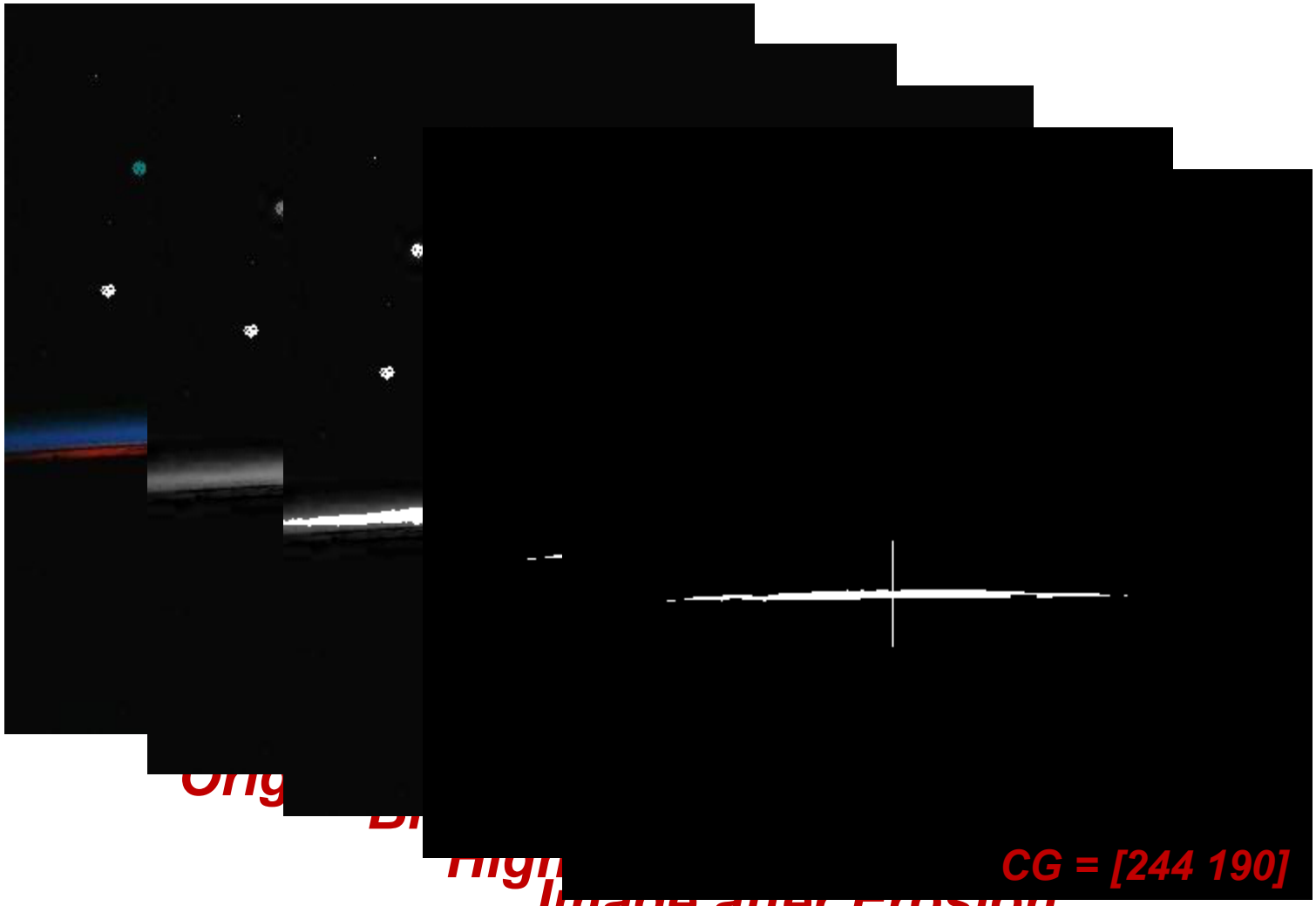
- **The memory allocation and handling**
  - How the data would reside and how would it be moved
- **Verification of the correct outputs based on simulation**
  - How can we be sure if the results match the expected results
- **Fixed I/O speed of Camera, conflicting with the variable time for algorithm completion**
  - Clock Mismatch requires time fixes in the UCF file and use of derived/buffered clock divisions

- Background
- Motivations
- Prior Work
- Our Solution
- **Results**

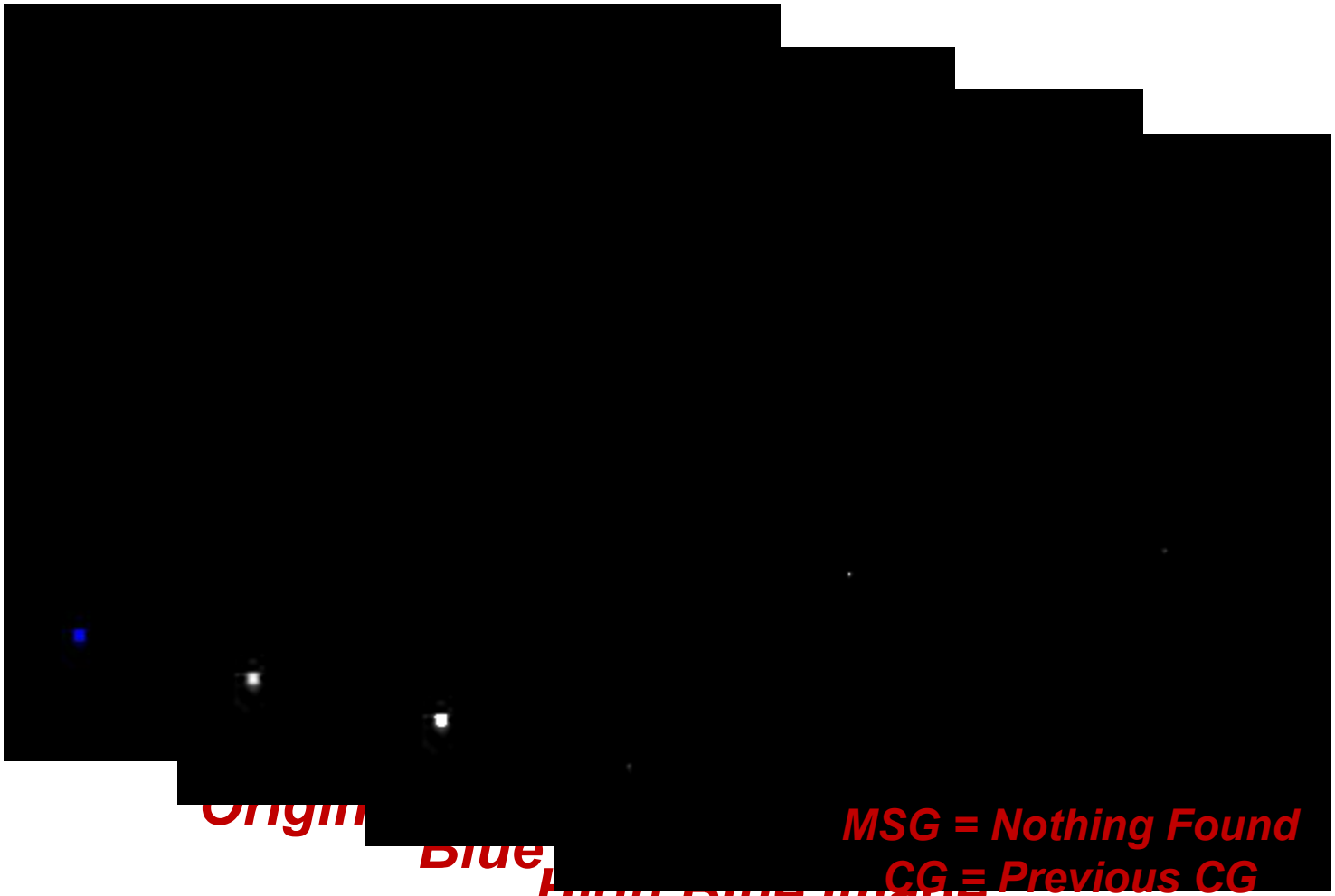
MATLAB RESULTS



MATLAB RESULTS



MATLAB RESULTS



THANK YOU

USC Viterbi School of Engineering

Predicting the Position of the Sun, across Earth's Horizon, prior to Sunrise, using Image Processing

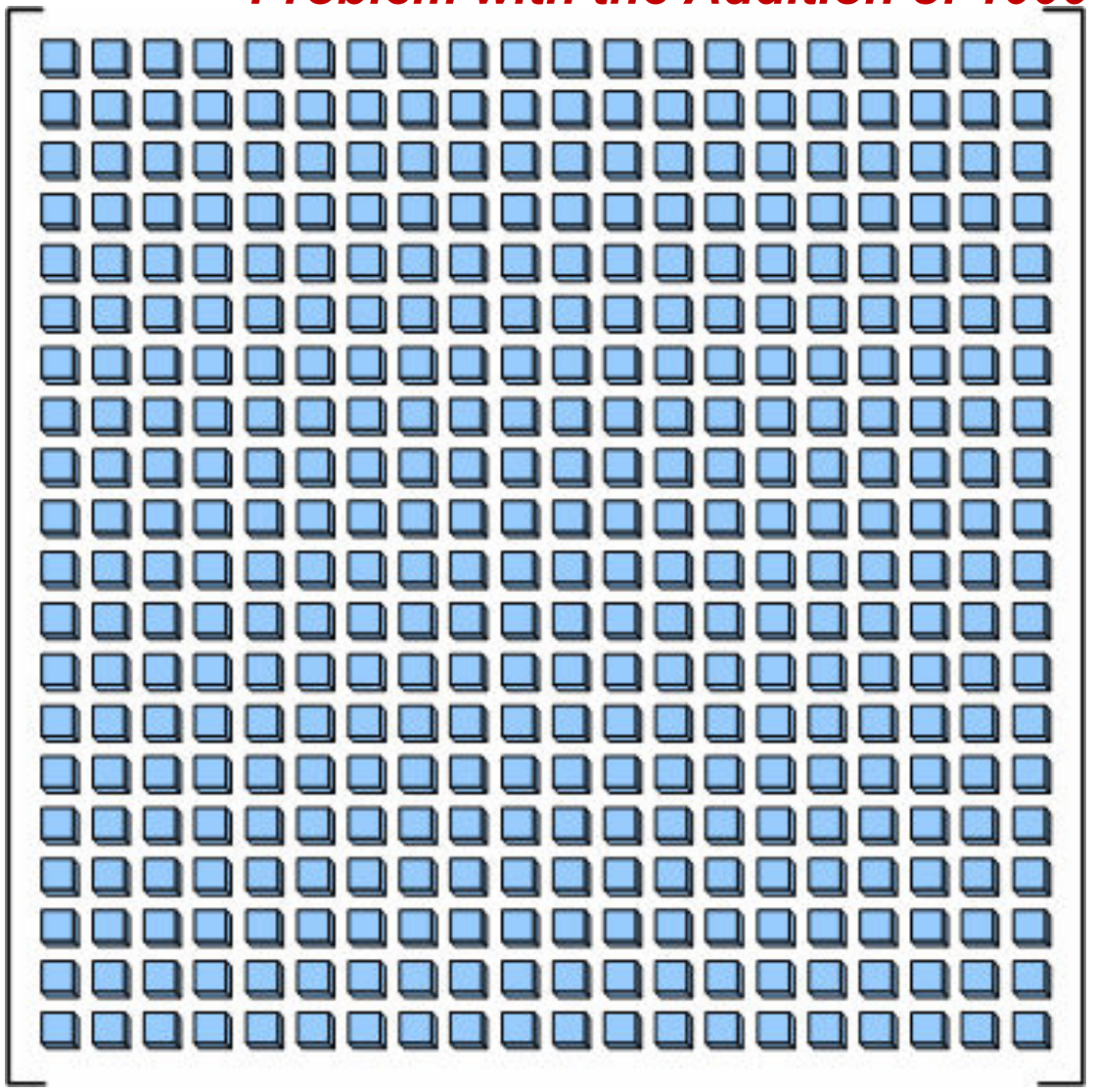
QUESTIONS ?

SOLUTION

USC Viterbi School of Engineering

Problem with the Addition of 10000 data points

- **Level 1**
- **Level 2**
- **Level 3**
- **Level 4**
- **Level 5**
- **Level 6**
- **… .**



SOLUTION

USC Viterbi School of Engineering

Problem with the Addition of 10000 data points

- **Generalize**
  - Number of levels: $\log_2 (n^2)$
  - Number of bits in the result: $\log_2 (n^2)$
- **Specifically**
  - Number of levels: 14
  - If implementing with 4 bit CLAs
    - 50 cycles of addition stages
    - Very Huge Architecture of CLA Adders

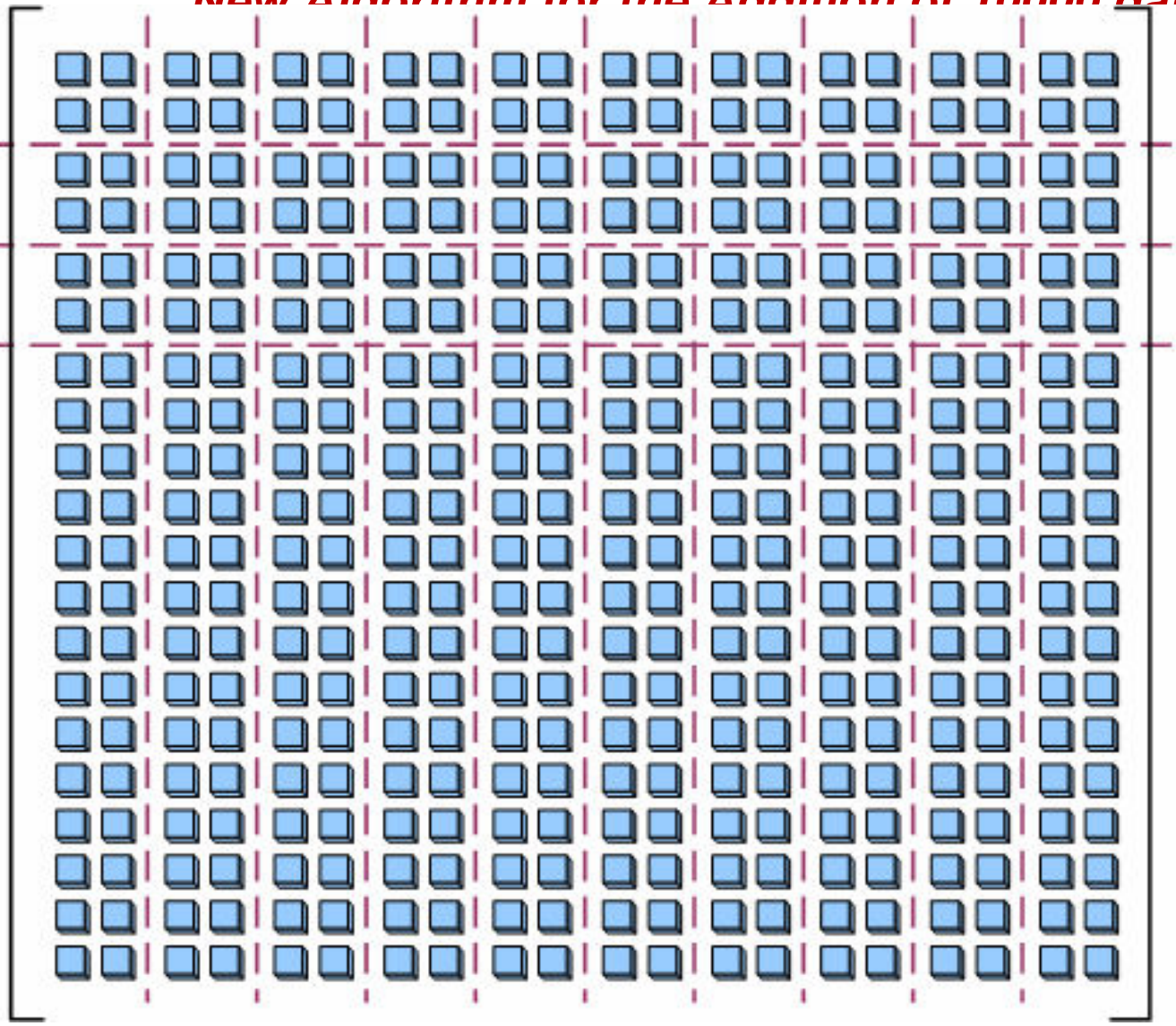
Level 1	5000 additions	1 bit each: 1CLA+1FA
Level 2	2500 additions	2 bit each: 1CLA+1FA
Level 3	1250 additions	3 bit each: 1CLA+1FA
Level 4	625 additions	4 bit each: 1CLA+1FA
Level 5	312 additions	5 bit each: 2CLA+1FA
Level 6	156 additions	6 bit each: 2CLA+1FA
… .		

SOLUTION

USC Viterbi School of Engineering

New Algorithm for the Addition of 10000 data points

- **Divide**
  - Ste
    -
  - Ste
    -
    -
  - Ste
    (gre
  - Ste
    -
    -
  - **Har**
  - **The**



bit is 1
s 1.