Image Processing Chip for Planetary Navigation Applications

Steve Parkes, Martin Dunstan, Nick Rowell,
Iain Martin, Mark McCrum
Space Technology Centre, University of Dundee

Oliver Dubois-Matra
ESTEC, ESA

Introduction

- Purpose of Image Processing Chip
  - Feature Extraction (and Tracking) Integrated Circuit
  - FEIC
- FEIC Architecture
- Testing
- Test Results
- Problems and solutions
  - Wandering tracks
  - Short tracks
- Conclusions

Partners

- ESA
- Astrium SAS, Selex Galileo, CGS, INETI, Astrium UK, GMV, TAS-I
- Contracts:
  - PANGU
  - NPAL,
  - HARVD,
  - VisNav,
  - ExoMars,
  - NEO-GNC,
  - VisNav-EM

Space Technology Centre
University of Dundee

Purpose of FEIC

- Vision-based navigation
- Image co-processor
  - Offloads the image processing task from the host processor
- Applications:
  - Planetary landing
  - Asteroid station keeping, surveying and landing
  - Rover navigation

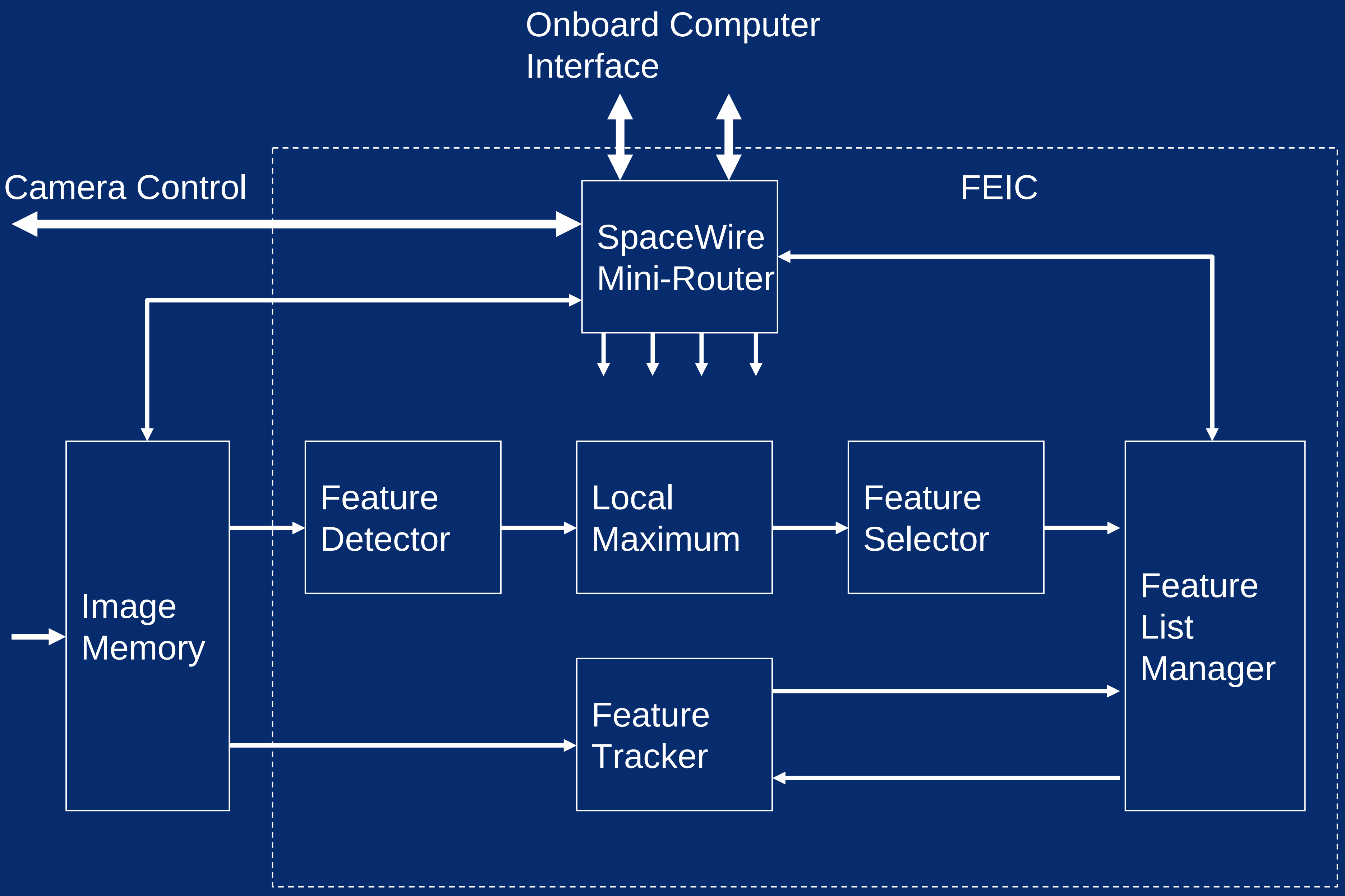
Space Technology Centre
University of Dundee

Feature Extraction and Tracking IC

- Image processing
  - Extracts feature points using Harris detector
  - Tracks them from frame to frame
  - Reports tracked feature points to GNC computer
  - For integration with IMU using a Kalman filter
- Image processing chip
  - Chip to do image processing (FEIC)
  - Developed for ESA
  - Navigation for Planetary Approach and Landing (NPAL)
  - Operates at 20 Hz frame rate

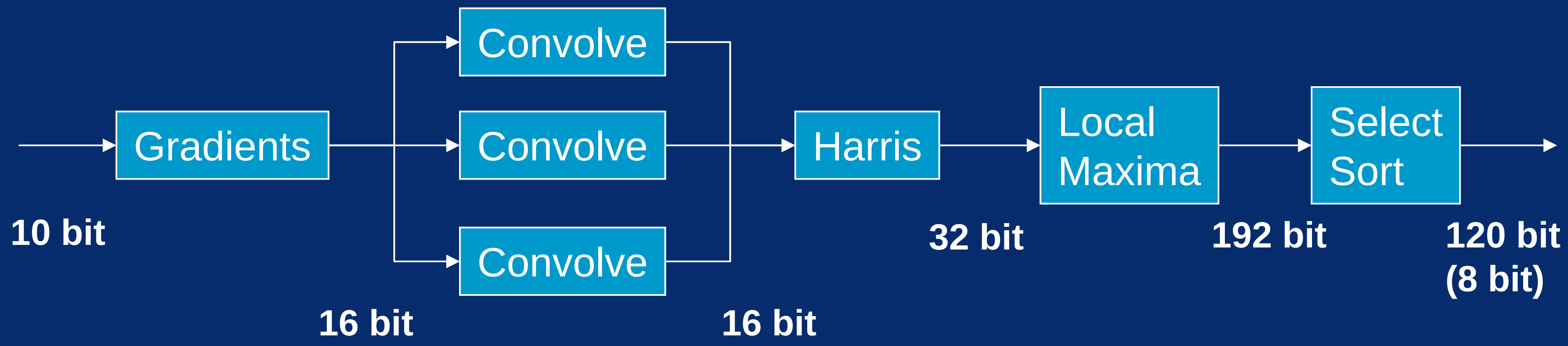
Space Technology Centre
University of Dundee

FEIC Architecture



Feature Extraction

- Compute gradient products:
- Convolve with Gaussian-like kernel
- Apply the Harris corner detector
- Apply a 7×7 local maxima filter
- Sort and select the strongest features
- Uses fixed-point saturating arithmetic



Harris Corner Detector

$$W = 7 \times 7 \text{ convolution kernel}$$

$$M = \begin{bmatrix} \left(\frac{\partial}{\partial x}\frac{\partial}{\partial x}\right) \otimes W & \left(\frac{\partial}{\partial x}\frac{\partial}{\partial y}\right) \otimes W \\ \left(\frac{\partial}{\partial x}\frac{\partial}{\partial y}\right) \otimes W & \left(\frac{\partial}{\partial y}\frac{\partial}{\partial y}\right) \otimes W \end{bmatrix}$$

$$H = \det(M) - k \cdot \mathrm{trace}(M)^2$$

$$k = \frac{5}{128} \approx 0.04$$

Space Technology Centre
University of Dundee

Local Maxima Unit

- 7×7 local maxima unit:
  - – pass feature if stronger than all others …
  - – … within a 7×7 neighbourhood
  - – otherwise suppress the feature
  - – all features output will be >3 pixels apart

0	2	0
3	9	3
1	1	4

PASS

2	0	5
9	3	6
1	4	1

FAIL

0	5	6
3	6	3
4	1	0

FAIL

Select/Sort Unit

- Purpose:
  - stores feature points sorted by strength …
  - … but only retains strongest N points (e.g. N=200)
  - 8 fields (120 bits) of data per point
- Performance constraint:
  - feature points arrive every 8 cycles at 40 MHz
  - must sort and store features within this time
  - we use a single-cycle insertion sort
  - uses lots of chip area so N<100 for V6000

Space Technology Centre
University of Dundee

Space Technology Centre
University of Dundee

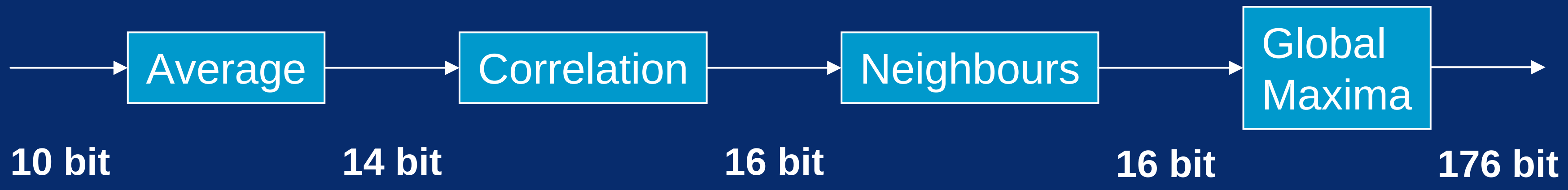
Feature Tracking

- Using intensity insensitive correlation:
  - correlate 7×7 texture with N×M window
  - OBC suggests window origin (guidance hint)
  - FEIC returns correlation strength(s)
  - implemented using 32-bit fixed point arithmetic

$$c(x,y)=\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\left[\left(I(x+i,y+j)-\bar{I}_{xy}\right)\left(T(i,j)-\bar{T}\right)\right]}{\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N}\left[I(x+i,y+j)-\bar{I}_{xy}\right]^2\sum_{i=1}^{N}\sum_{j=1}^{N}\left[T(i,j)-\bar{T}\right]^2}}$$

Feature Tracking

- Invoked for each tracked point:
  - load correlator with feature point texture
  - stream N×M pixel window into 7×7 average
  - compute correlation measure (pipelined)
    - numerator/denominator computed on alternate cycles
  - recover 8-neighbours from correlator stream
  - filter to identify the global maximum



List Manager

- Provides storage and the interface to OBC:
  - maintains the LN (new) and LT (tracked) points lists
  - responsible for all OBC feature point operations
  - drives the tracker and (partly) the extractor
- Provides interface between FEIC and OBC

Development of FEIC by UoD

- Prototyped in C++:
  - – started with “standard” algorithms
  - – replaced with H/W data-flow algorithms
  - – defined FEIC architecture
  - – implemented FEIC in VHDL
- Validation using PANGU image sequences:
  - – validated simulated VHDL against C++
  - – FPGA driver can validate output against C++
- Validation using Astrium acceptance tests:
  - – compares performance against floating-point

Space Technology Centre
University of Dundee

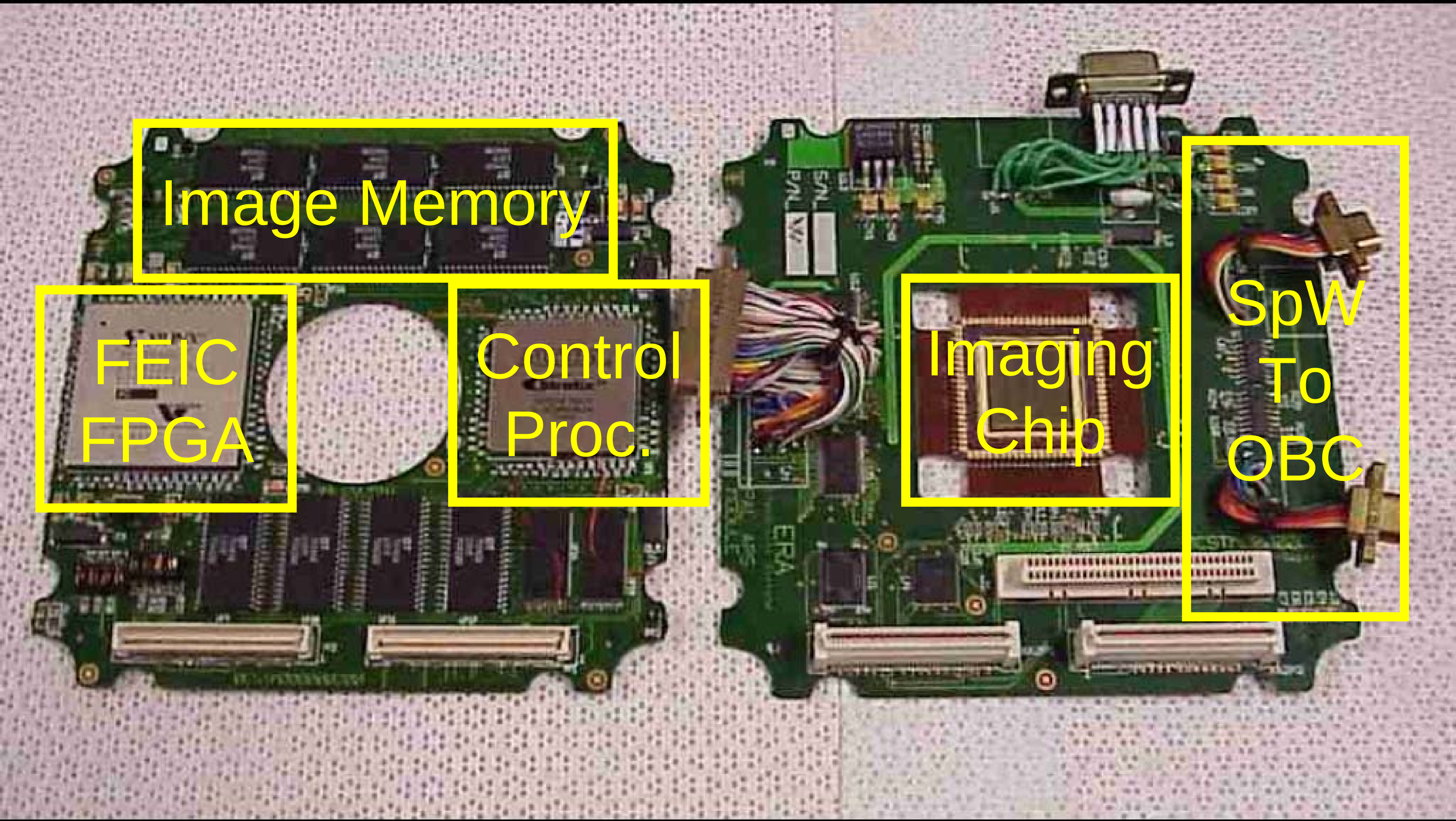


Image Memory
FEIC FPGA
Control Proc.
Imaging Chip
SpW To OBC

NPAL Camera
EADS Astrium (France)
Selex Galileo (Italy)
University of Dundee

FEIC Test Environment

- PANGU
  – Planet and Asteroid Natural-scene Generation Utility
  – Software tool
    - Simulates planets and asteroids
    - Simulates cameras and other sensors viewing those bodies
    - Developed specifically to test vision based GNC algorithms
    - Realistic and high performance
    - Extensively vested and validated so that it can be used for testing flight systems
  – Being used for lander, rover, orbiter simulation

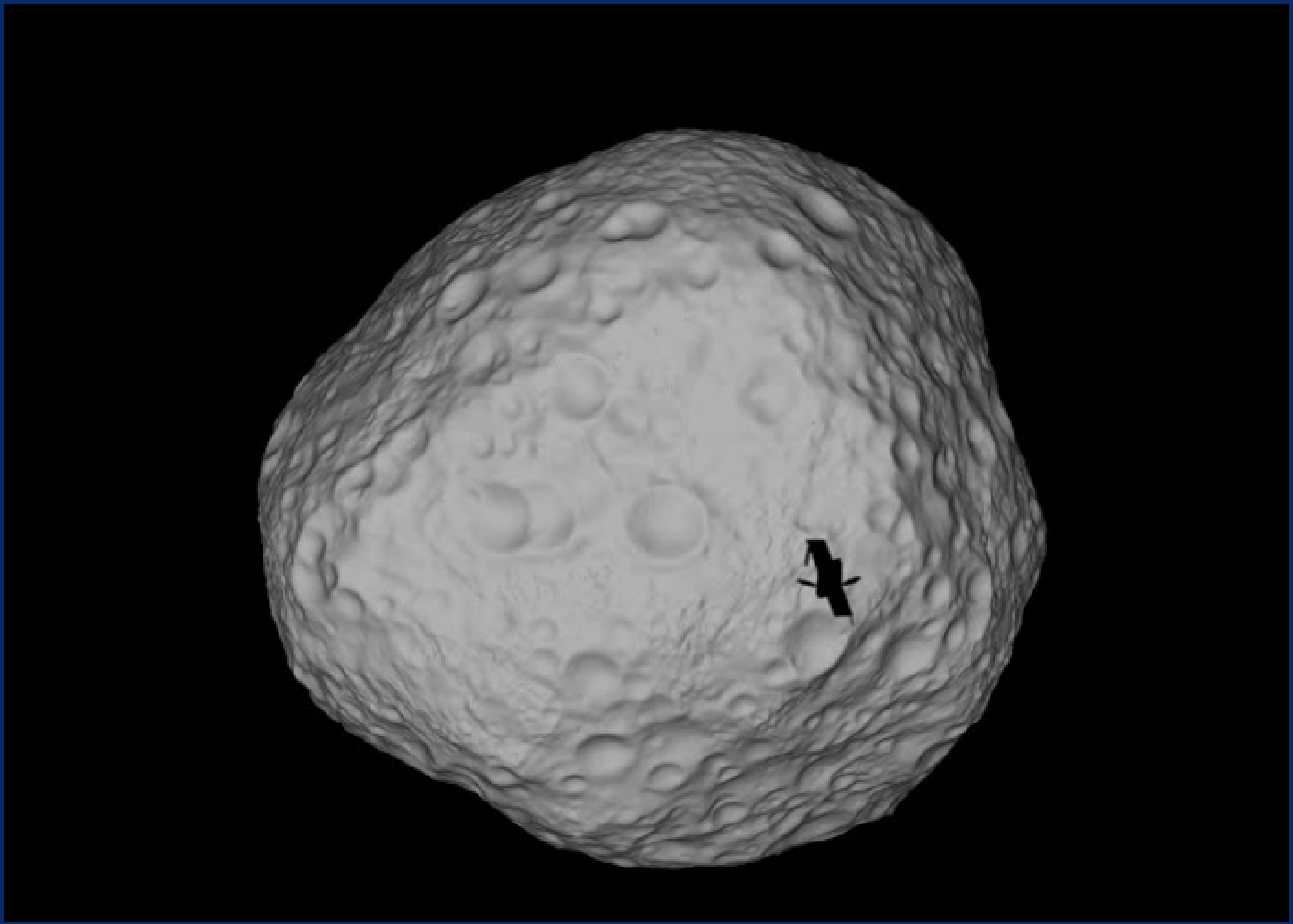
Space Technology Centre
University of Dundee

Open and Closed Loop Testing

- Open loop testing
  - PANGU image sequence from known positions
  - Fed to vision based navigation system
- Closed loop testing
  - Spacecraft dynamics simulation
  - Gives PANGU camera position and orientation
  - PANGU provides image of surface from that position
  - Position and orientation of target object
    - Can also be specified
    - Useful for asteroids
- Testing of real hardware
  - Using monitor and camera

Space Technology Centre
University of Dundee

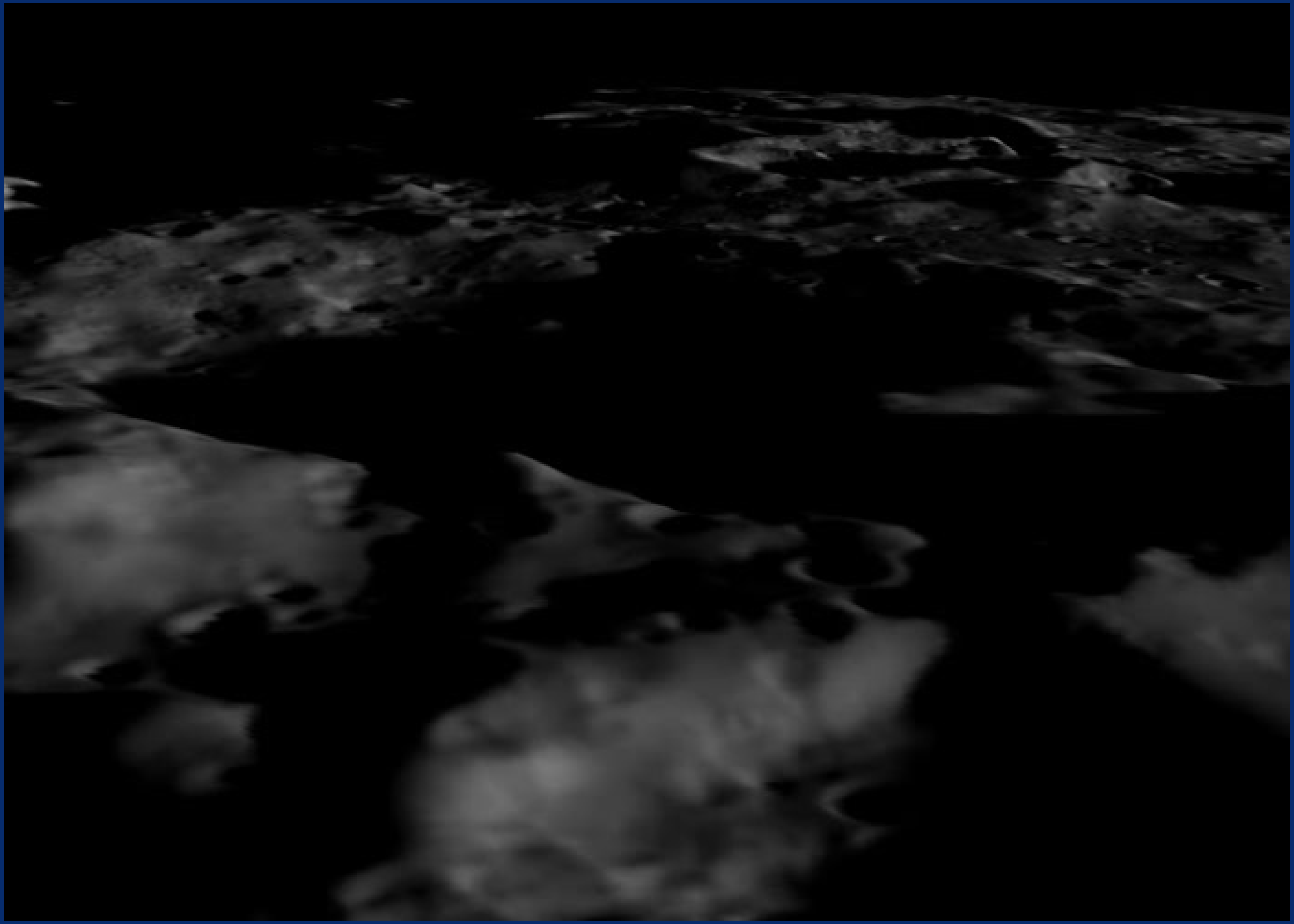
PANGU Cratered Asteroid + Spacecraft Shadow



Space Technology Centre
University of Dundee

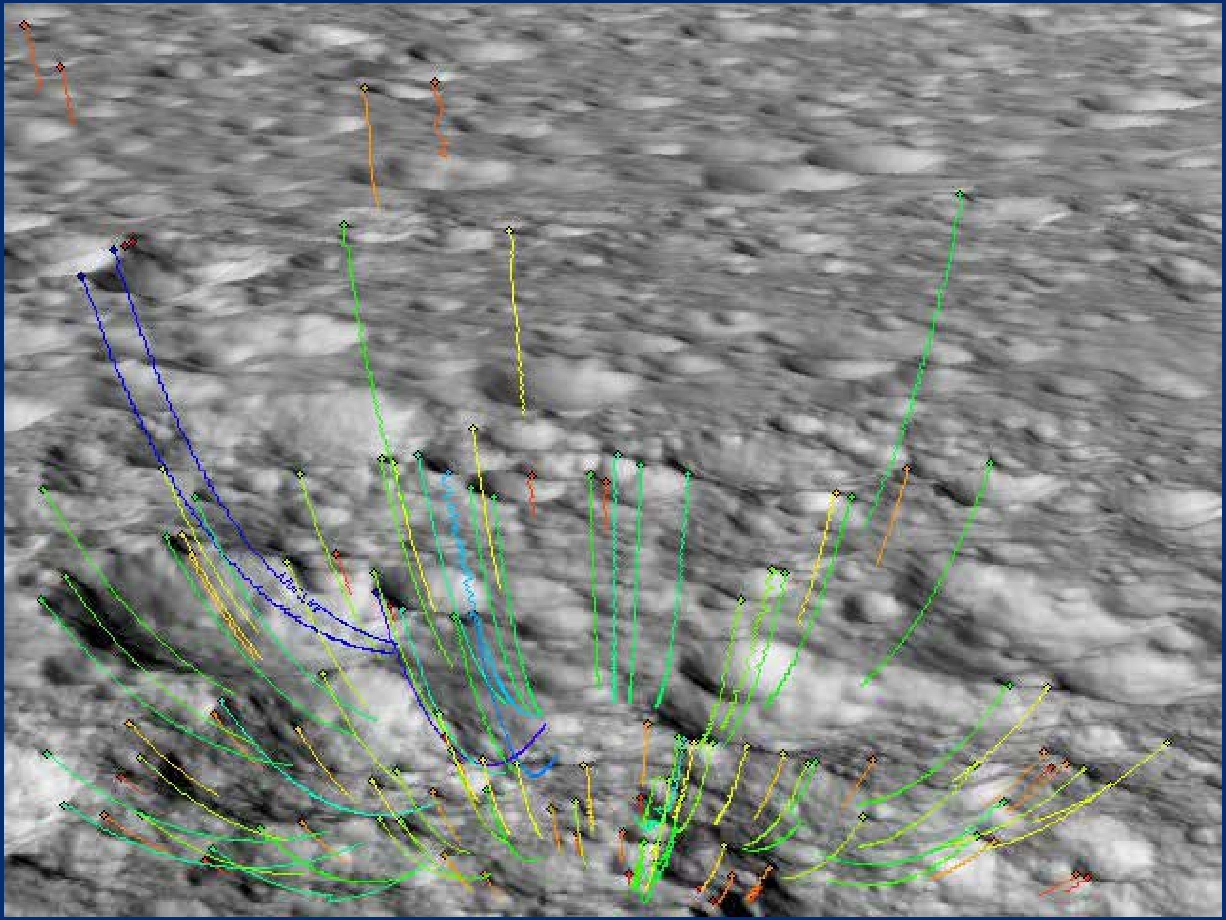
PANGU Martian Surface and Rover

PANGU Lunar South Pole



Space
Technology
Centre
University of Dundee

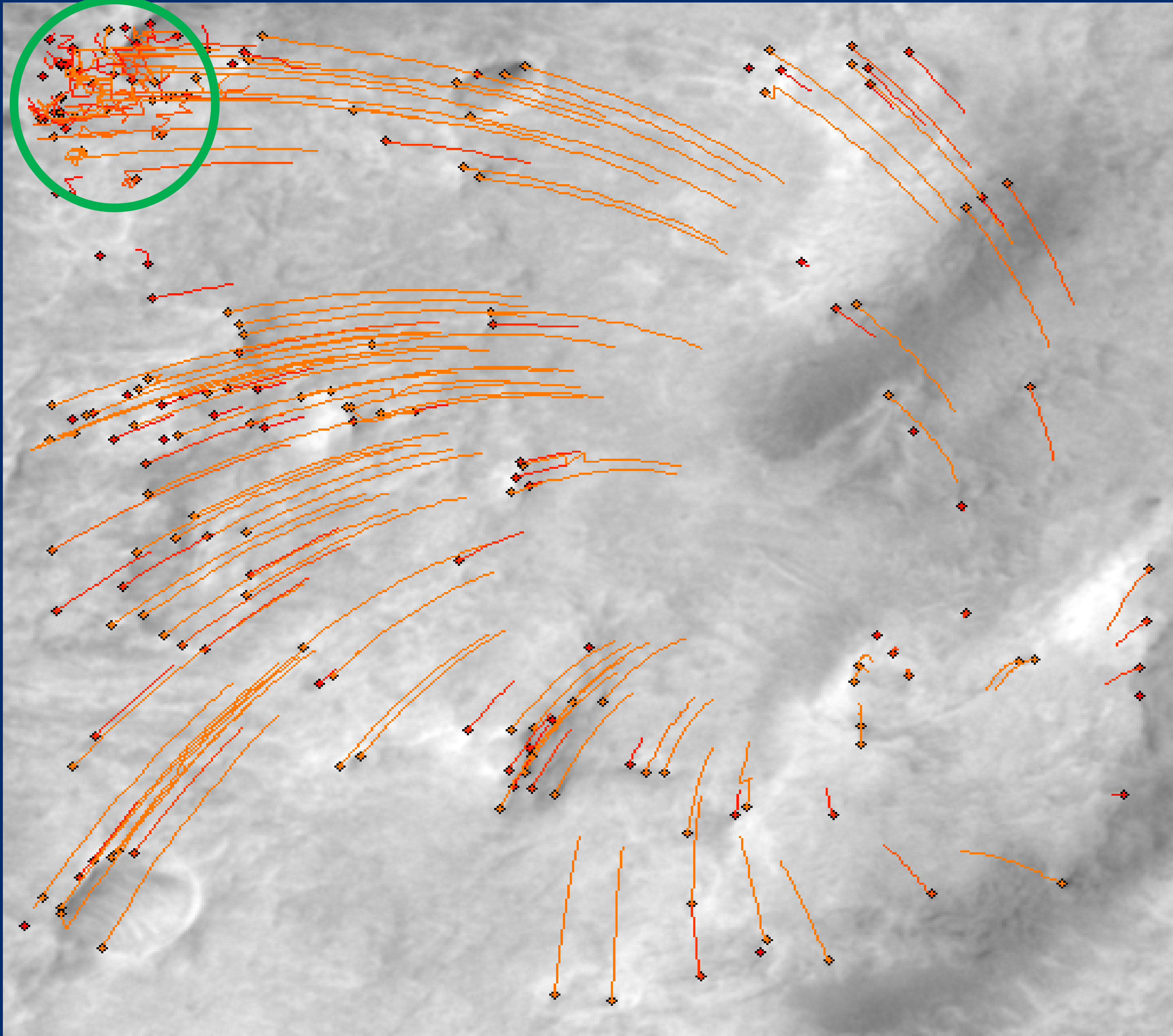
Empirical Results



Tracking using the FPGA implementation.

Tracks start red then change colour depending on length of time tracked

Only need to track six points successfully to recover motion of spacecraft



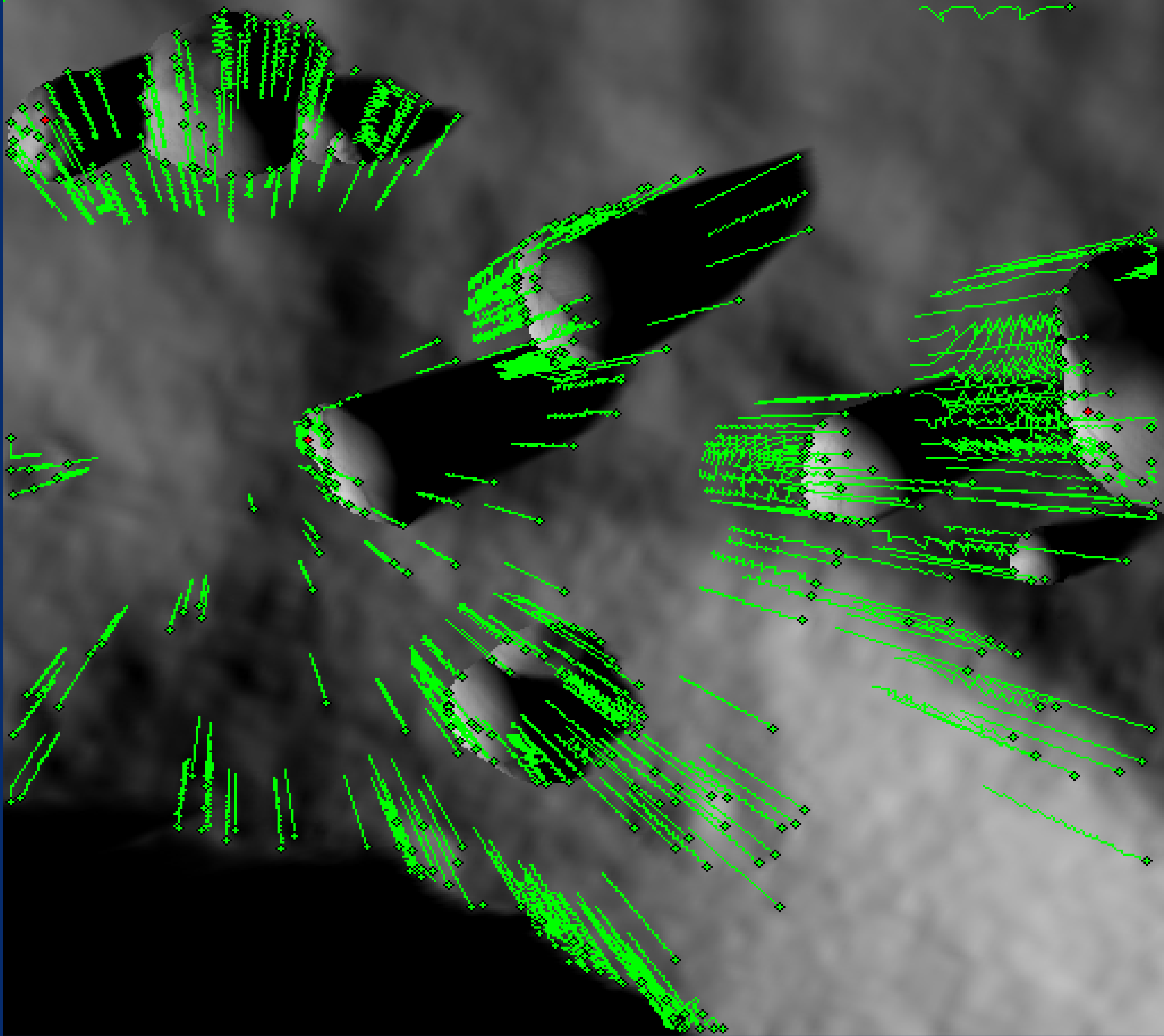
Space
Technology
Centre
University of Dundee

Improvements to image processing

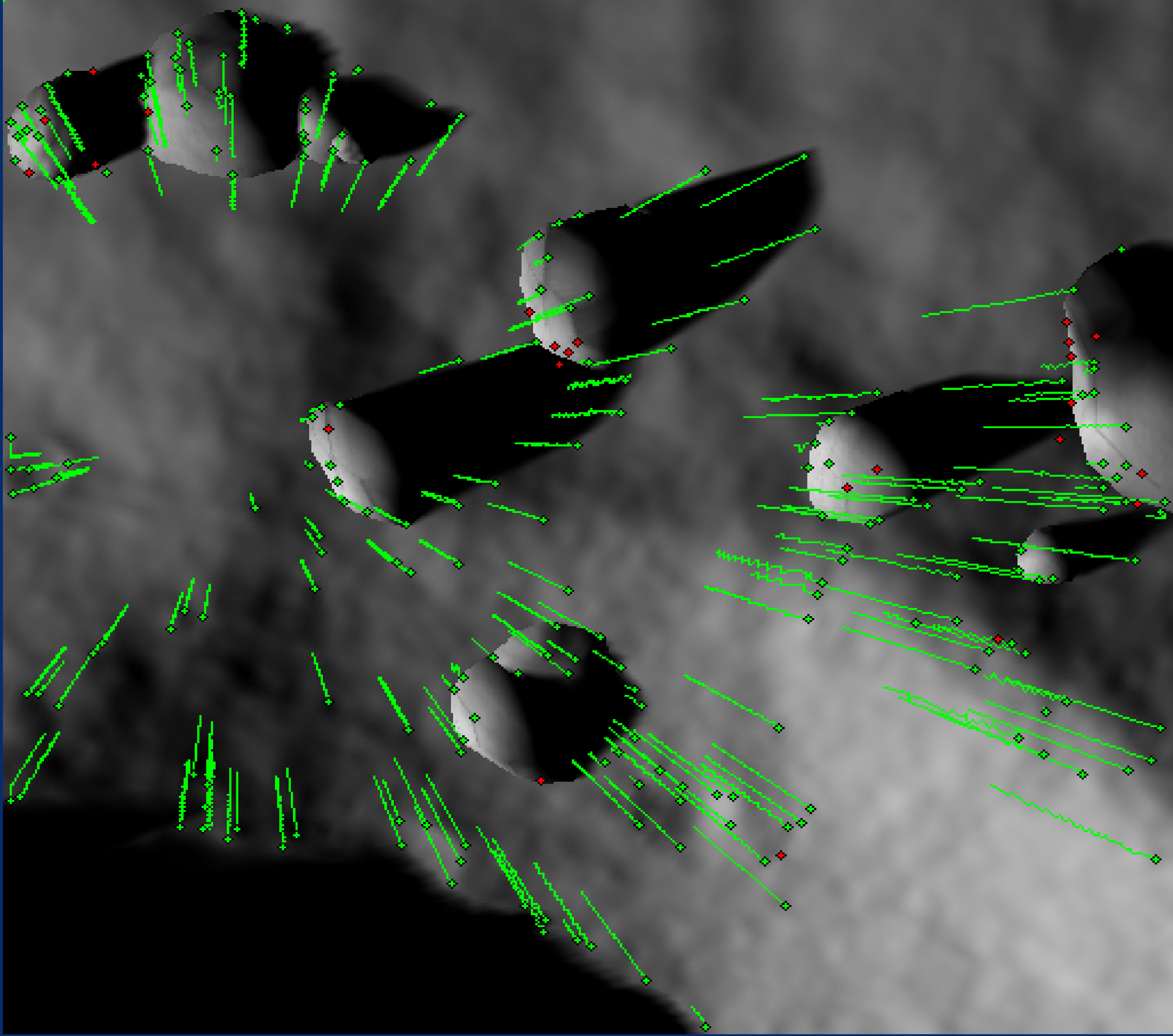
- Wandering Tracks
  - Possible for feature tracker to lock on to a similar feature close to the one being tracked.
  - Causes wandering features
  - Example in top left hand corner of image
  - True point is no longer in the image
  - Rather than being discarded a similar point close by has been picked up
- Proposed improvement
  - When a feature is extracted
  - Check that there are no similar features close by
  - That could cause confusion
  - If there are, reject the feature point

Space Technology Centre
University of Dundee

Space Technology Centre
University of Dundee



Space Technology Centre
University of Dundee



Problem with rotation

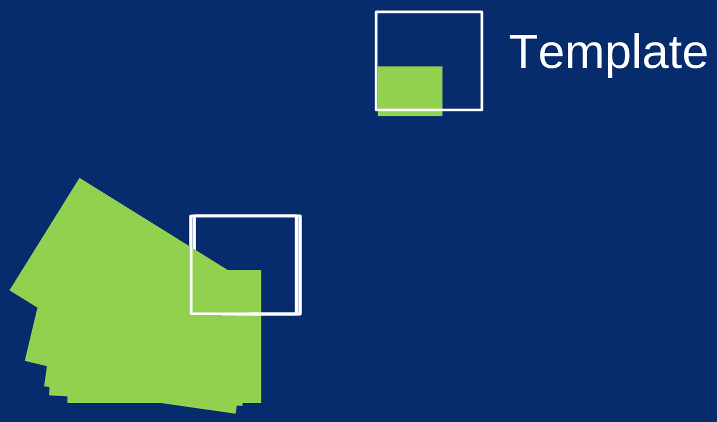
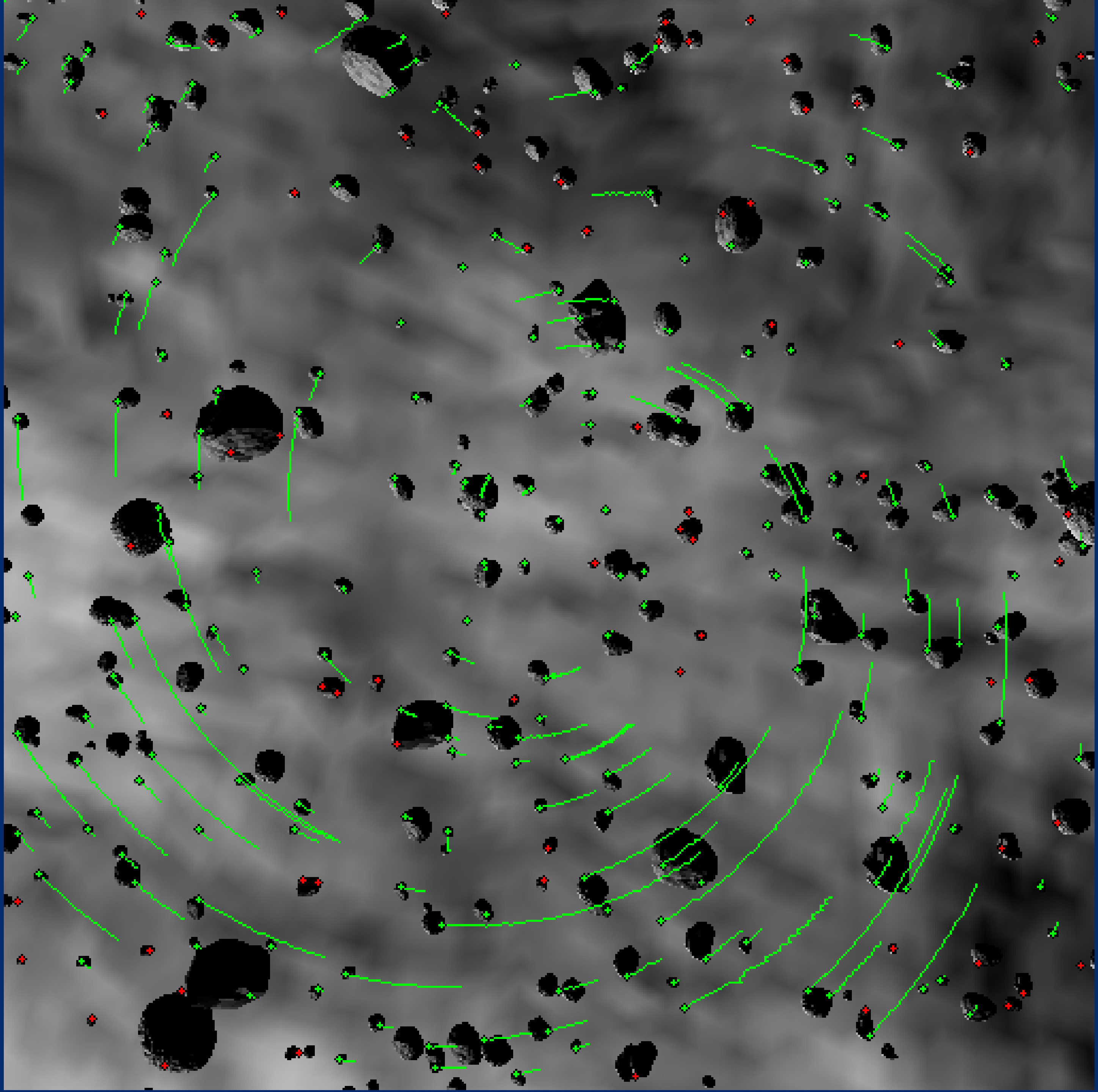


Image Processing Improvements

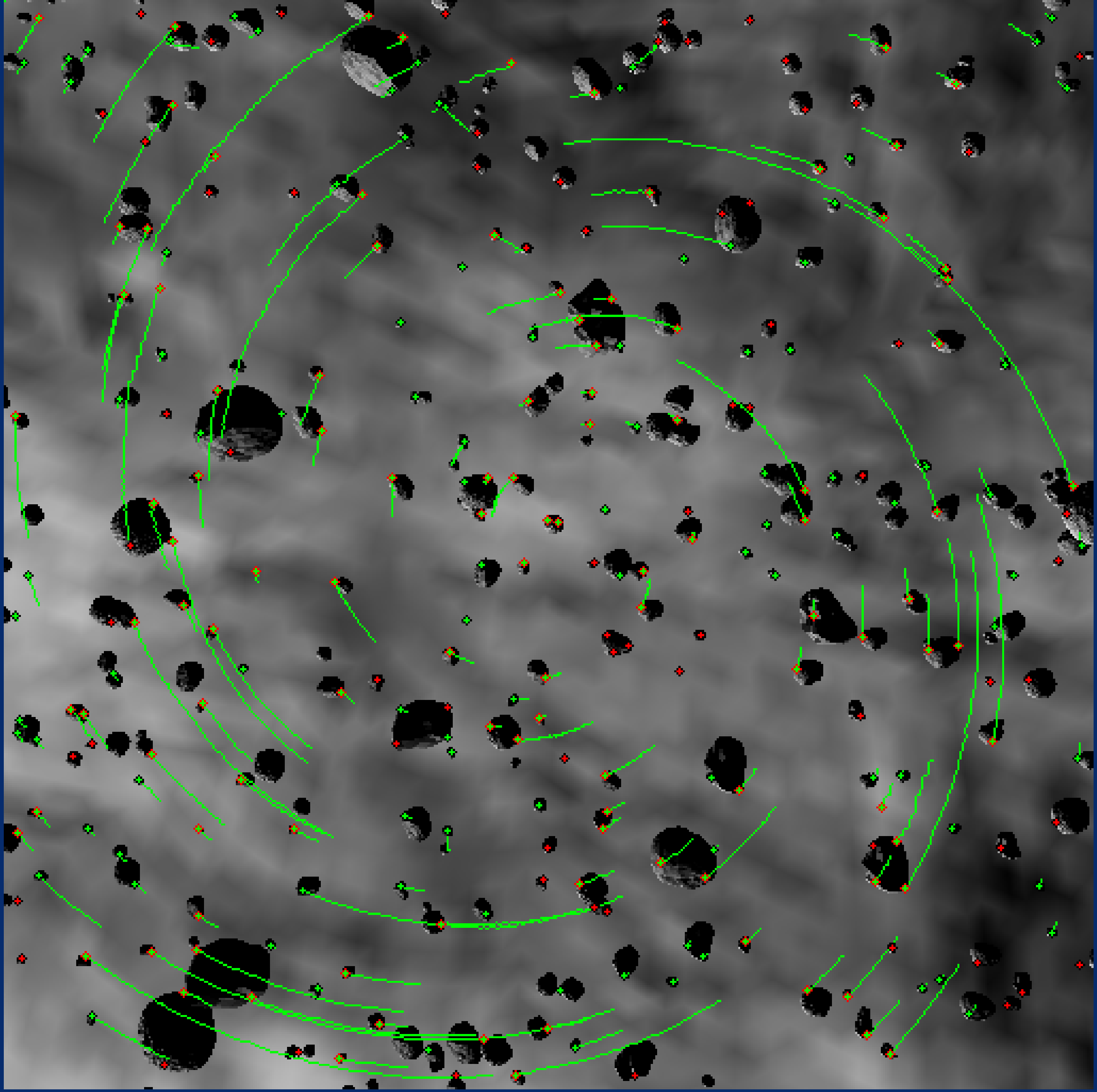
- Short tracks
  - With changes in the image perspective the feature template can become out of date
- Proposed improvement 1
  - Update feature template regularly
  - This causes feature wandering
  - Good tracking on a slightly incorrect feature template position propagated across many frames
- Proposed improvement 2
  - Update feature template regularly
  - Track with both old and new template
  - If feature positions are close over many frames
  - Accept new point and discard old one

Space Technology Centre
University of Dundee

Space Technology Centre
University of Dundee



Space Technology Centre
University of Dundee



Currrent work

- Results shown are without aiding

- Assessment of robustness
- Known landmark tracking
- Dense correlation for stereovision
- Surface elevation determination

- Design of a generic image processing device for vision-based navigation applications

Space Technology Centre
University of Dundee

Conclusions

- Image processing co-processor
  - Offloads computationally intense work from OBC
  - Interfaced using SpaceWire
- Simulation using PANGU
  - very useful for developing and testing image processing algorithms
- Potential improvements to FEIC
- Shown, using PANGU, to give significant benefit
  - Avoid wandering tracks
  - Longer tracks
- Eases job of OBC

Space Technology Centre
University of Dundee