Feature Quality Assessment

Yang Cheng
JPL
ycheng@jpl.nasa.gov, 818 354 1857
adnan.ansar@jpl.nasa.gov

Outline of this Lecture

- Motivation and objective
- Parametric based feature quality assessment
- Non-parametric feature quality assessment
- Homework assignment

References/Reading Materials

- Richard Szeliski “Computer Vision: Algorithm and Application” Chapter 2.1.1 to 2.1.5 and 6.1.1 to 6.1.5, A.2 and A.3

- Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), 381–395.

Optional

- Rousseeuw, P. J. (1984). Least median of squares regresssion. Journal of the American Statistical Association,79, 871–880.

- Rousseeuw, P. J. and Leroy, A. M. (1987). Robust Regression and Outlier Detection. Wiley, New York.

- Torr, P. H. S. and Murray, D. W. (1997). The development and comparison of robust methods for estimating the fundamental matrix. International Journal of Computer Vision, 24(3), 271–300.

- Stewart, C. V. (1999). Robust parameter estimation in computer vision. SIAM Reviews, 41(3), 513–537.

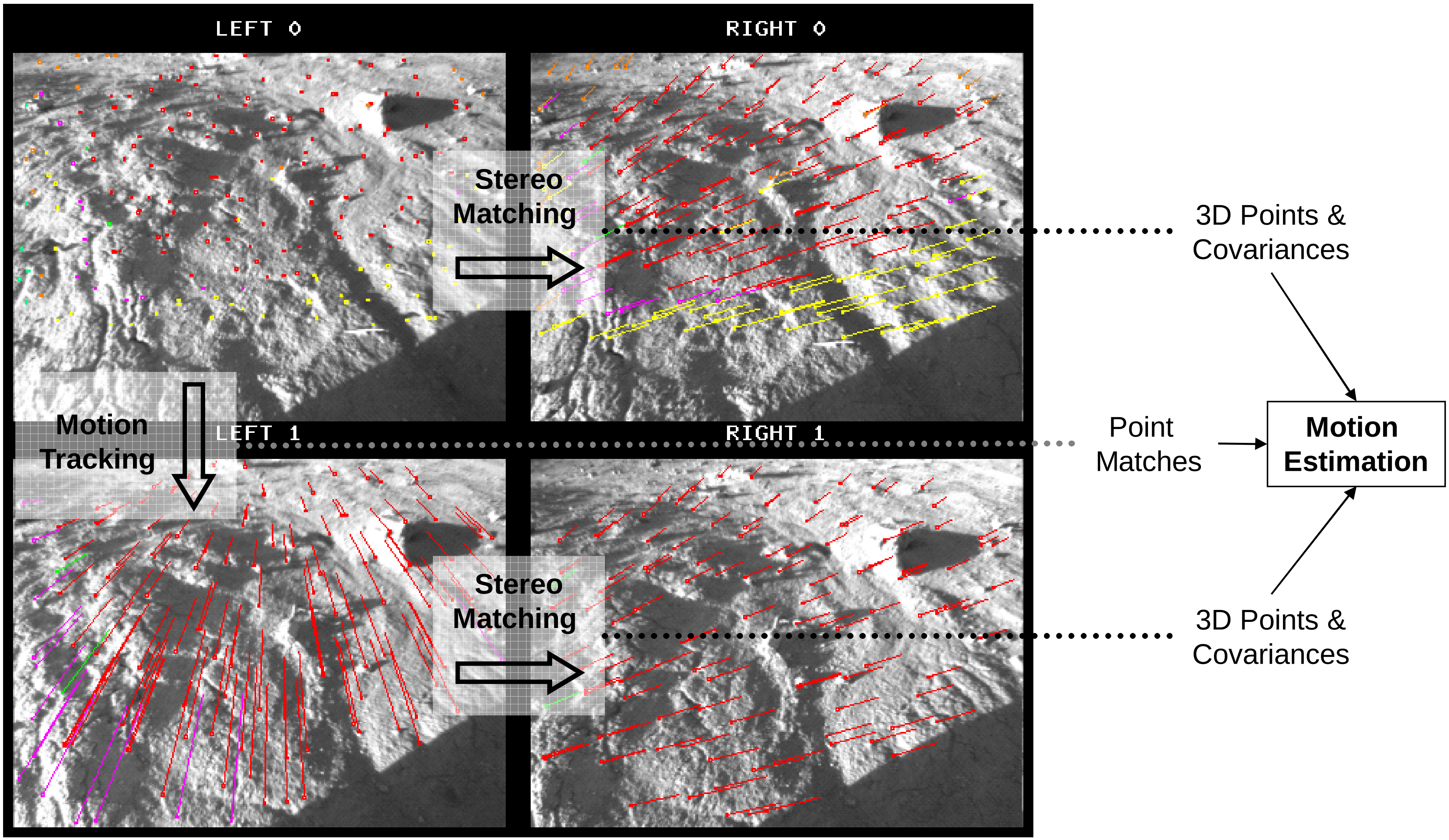
- Y. Cheng, A.Johnson, L. Matthies “MER-dimes: a planetary landing application of computer vision” CVPR 2005.

-

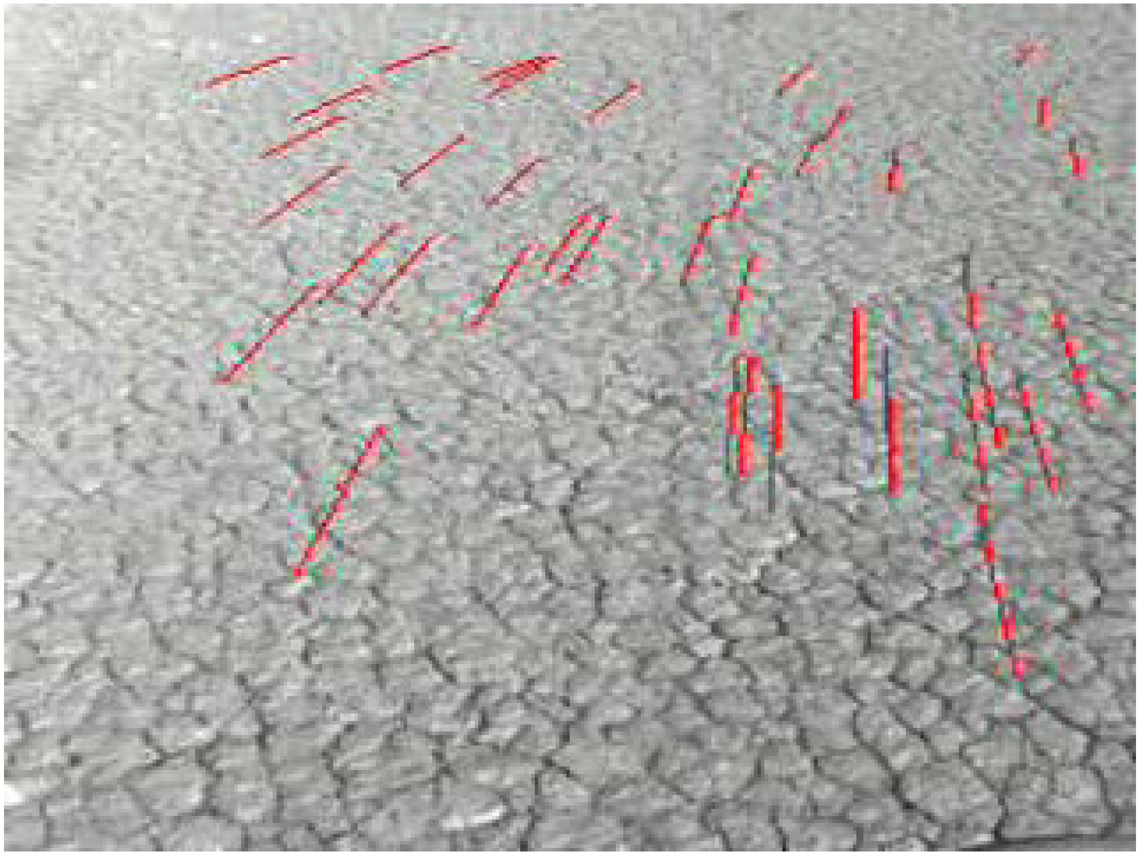
Motivation and Objective

- Motivation:
  - Feature selection/matching/tracking produce the correspondents (feature lists) in an image sequence. Due to deficiency of algorithms, sensor, or data, outliers (wrong) or bad features (less accurate) often exist in the feature list.
  - These outliers or bad features could cause the following algorithms (such as pose estimation, motion estimation etc) less accurate, less reliable, even failure
  - Reliable vision system for robotic operation requires a clean and precise feature (landmark) list.
- Objective
  - We need to find a set of feature that will produce a high-accuracy outputs for alignment, motion, pose, velocity, and/or attitude estimates.

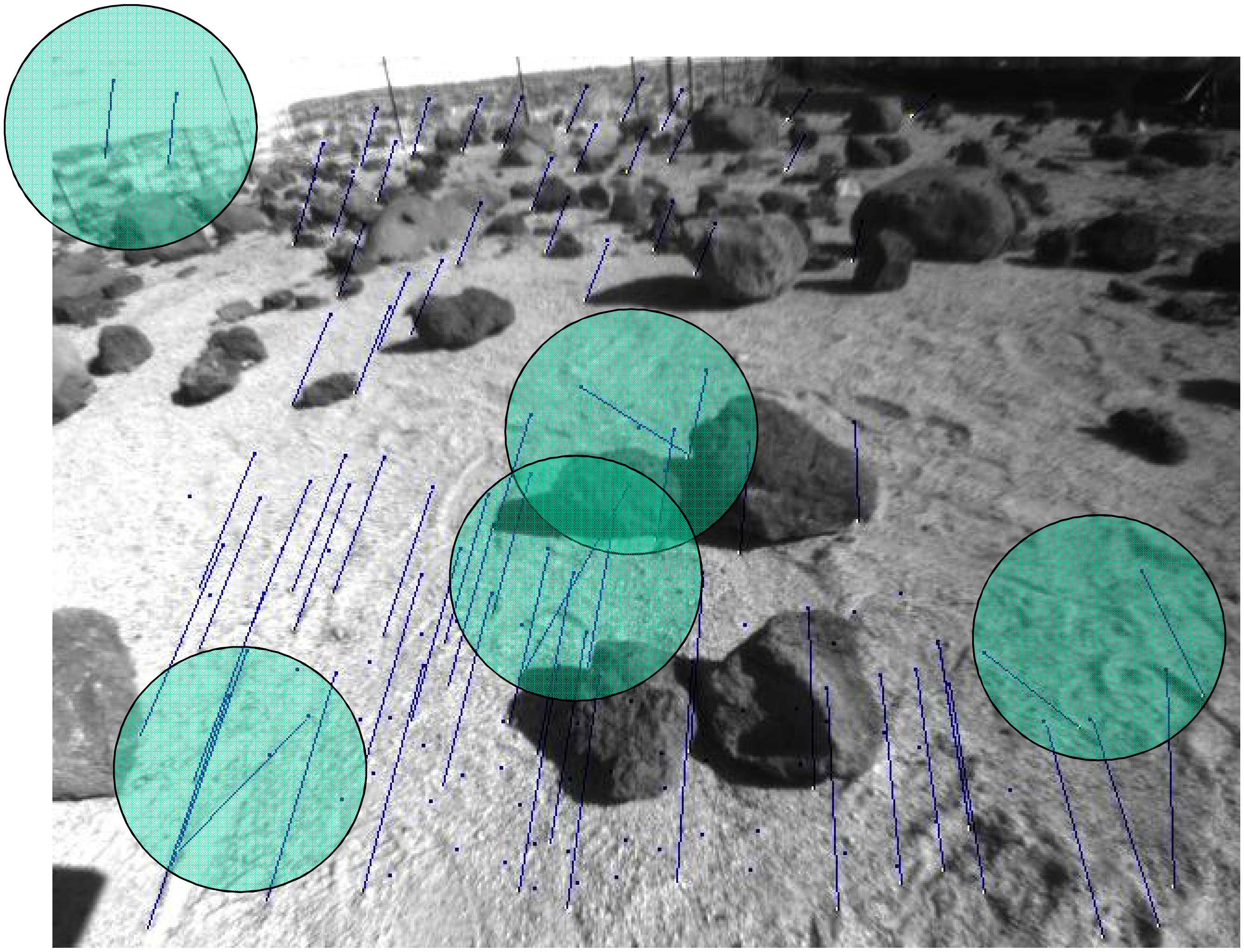
VO Feature Tracking Concept



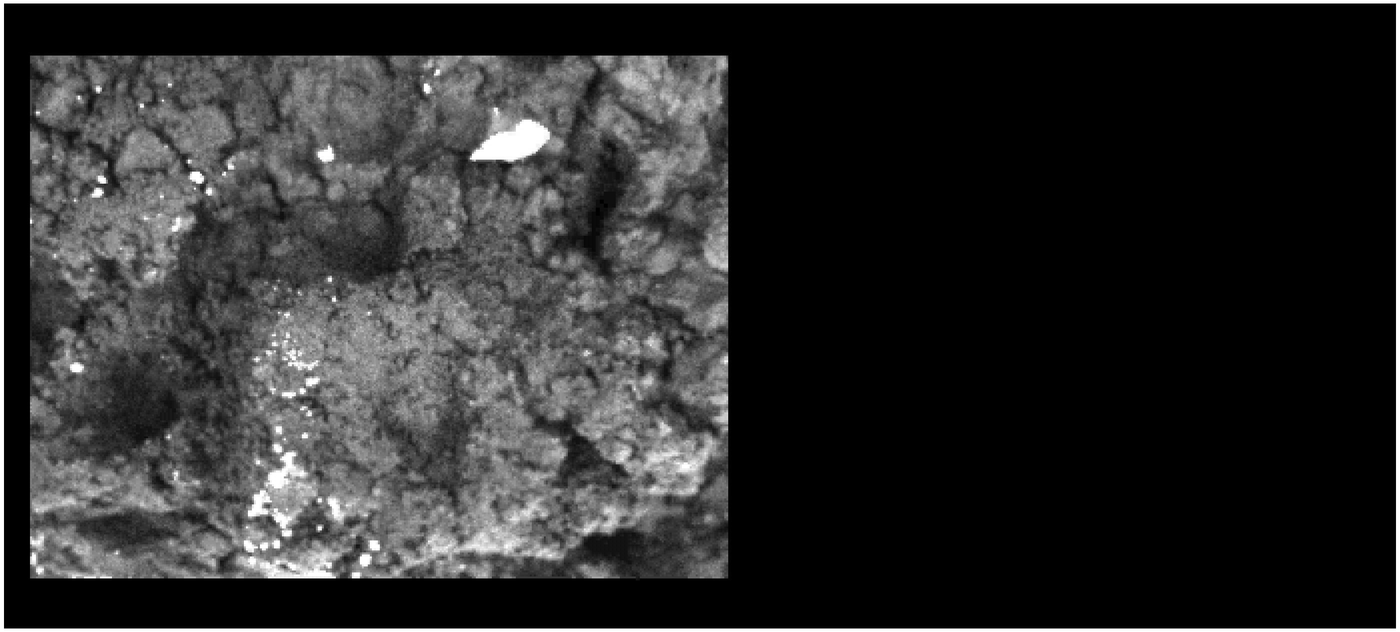
Vo Feature Tracker Video



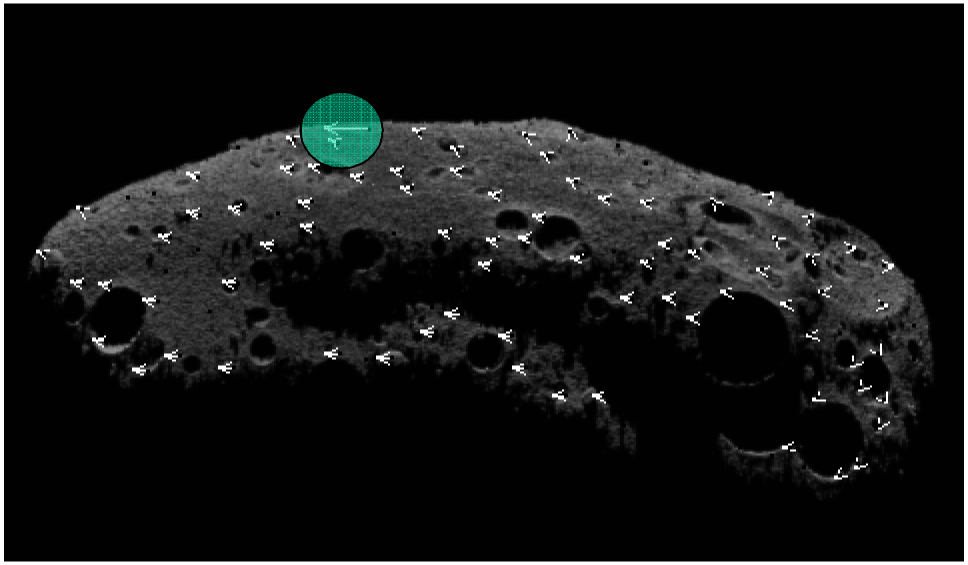
Visual Odometry Feature Outliers



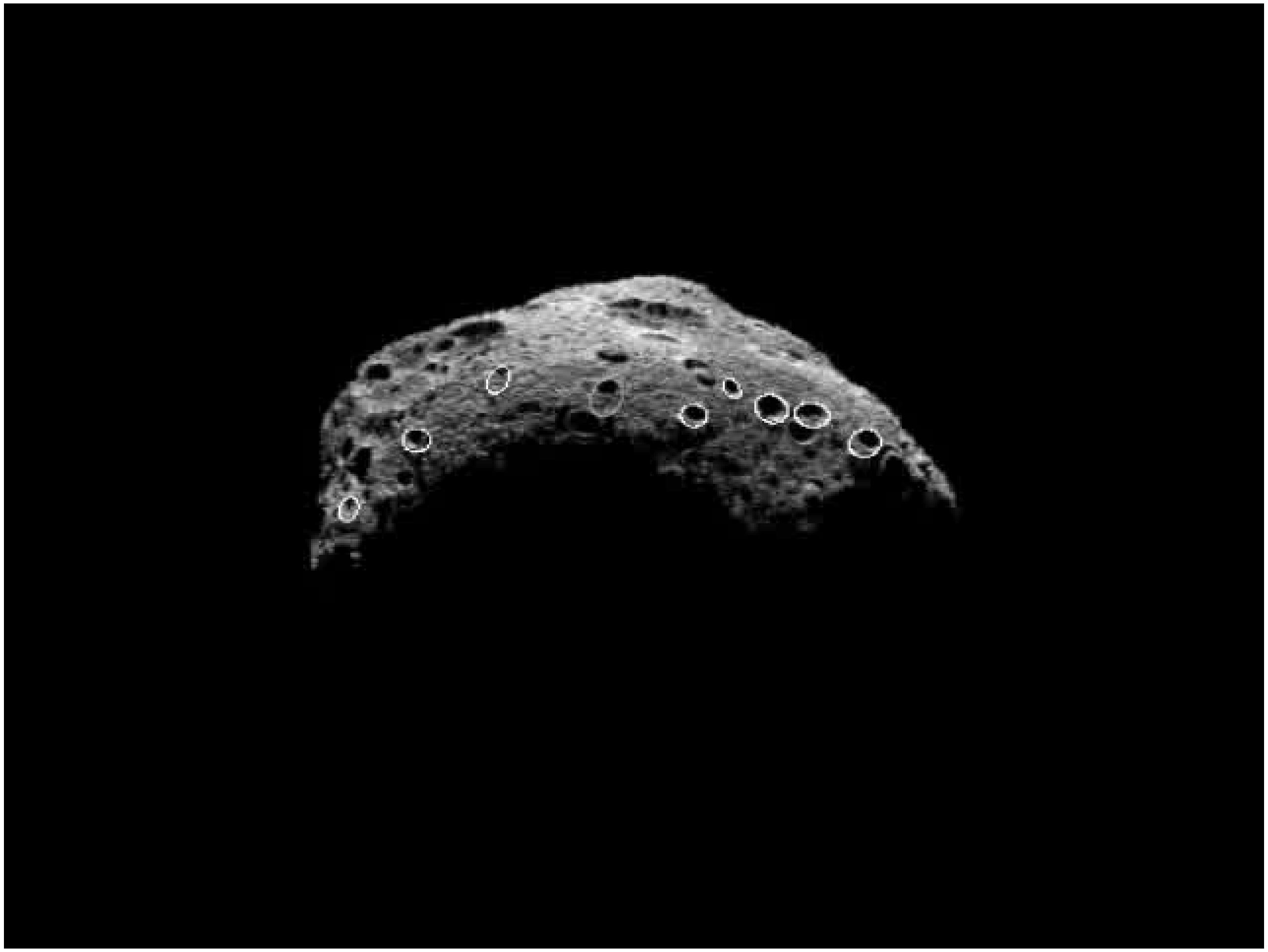
Small Body Monocular Motion Estimation and 3D Modeling



Small Body Navigation Feature Tracking Outliers



Crater Landmark Detection/Tracking



How to find and remove outliers and bad features

- Feature motion is not random and they follow certain geometric rules or constraints
- Factors to determine the geometric constraint
  - Camera attitude and translation motion, camera field of view
  - Scene geometry (flatness, 3D relief, distance to camera) and motion (rigid motion or none rigid motion)
  - Applications (structure from motion, pose estimation, attitude estimation (star tracker) etc)
  - Knowledge of robotic state (position, velocity, attitude)
- Once the geometry constraint is determined, how to obtain the best fitting
- How to guarantee the optimal solution is obtained

2D Transformations

- 2D points (pixel coordinates in an image) can be denoted using a pair of values

$$X = \begin{bmatrix} x \\ y \end{bmatrix} \quad or \quad \bar{X} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- **2D translations** can be written as X' = X + t or $X' = [I, t]\bar{X}$

- **Rotation + translation**. This transformation is also known as 2D rigid body motion or the 2D Euclidean transformation (since Euclidean distances are preserved). It can be written as

X' = RX + t or $X' = [R, t]\bar{X}, \quad R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$

2D to 2D Projection

- **Scaled rotation + translation**. Also known as the similarity transform, this transformation can be expressed

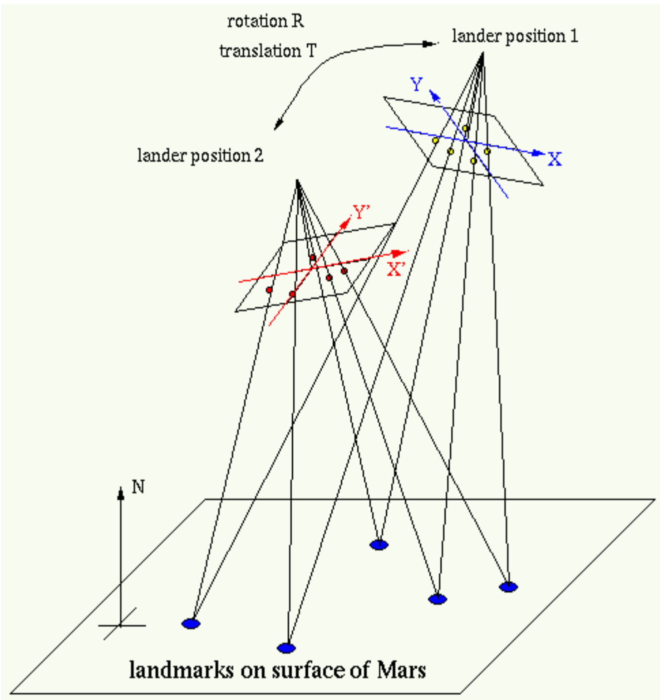
$$X' = [sR, t]\overline{X} = \begin{bmatrix} a & -b & t_x \\ b & a & t_y \end{bmatrix} \overline{X}$$

- **Affine Transformation:** Parallel lines remain parallel under affine transformations.

$$X' = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{bmatrix} \overline{X}$$

Notes: Spacecraft often carries very narrow FOV camera (1 or 2 degrees) and the transformation between image often follows affine transformation.

Homography transformation for Planar Surface



Plane surface:

$$N^T P = n_1 X + n_2 Y + n_3 Z = 1$$

Camera motion:

$$R = R(\kappa)R(\varphi)R(\omega) = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \qquad T = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

A Landmark in images

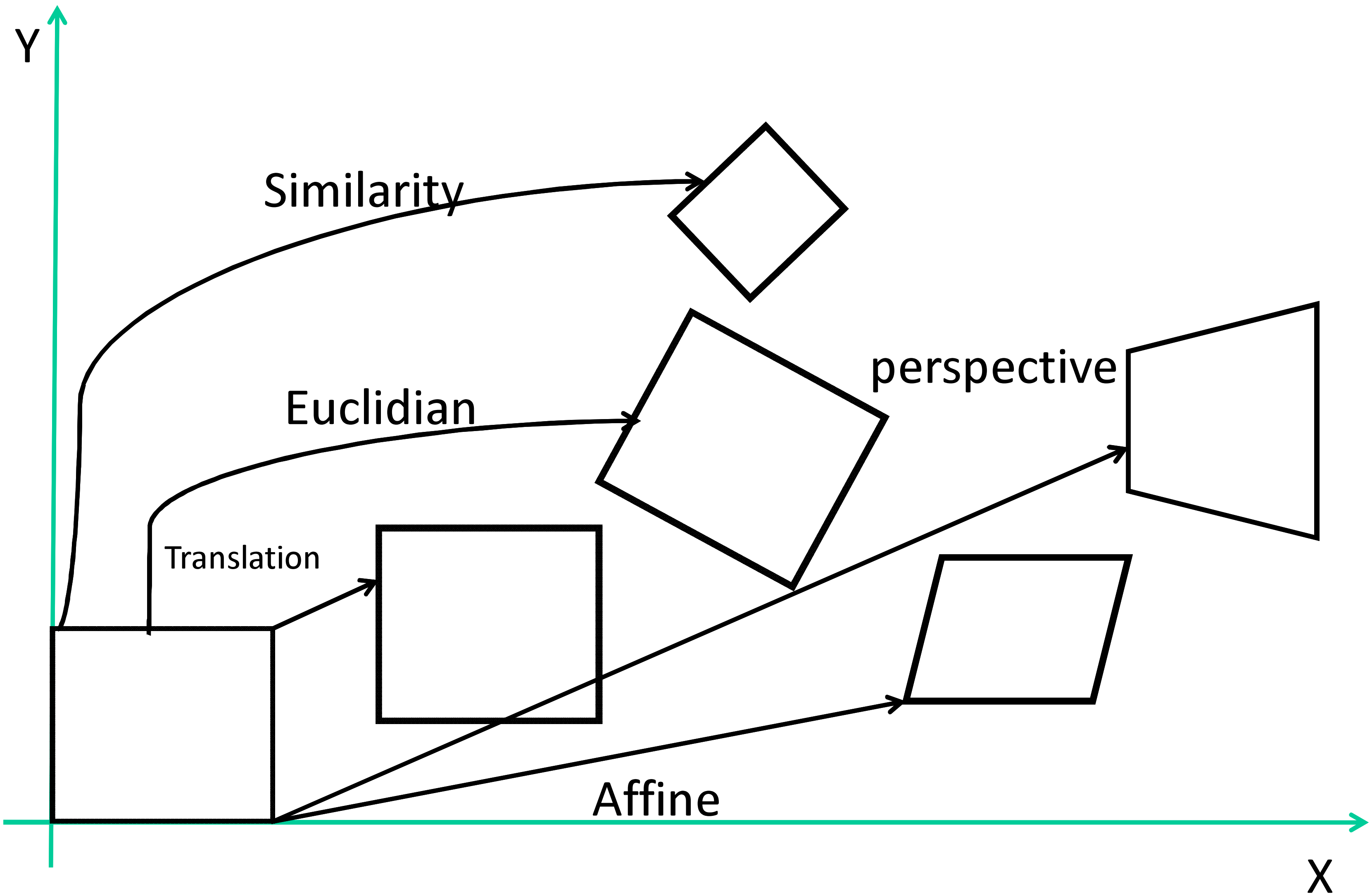
$$P = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad P' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

The relations between two images

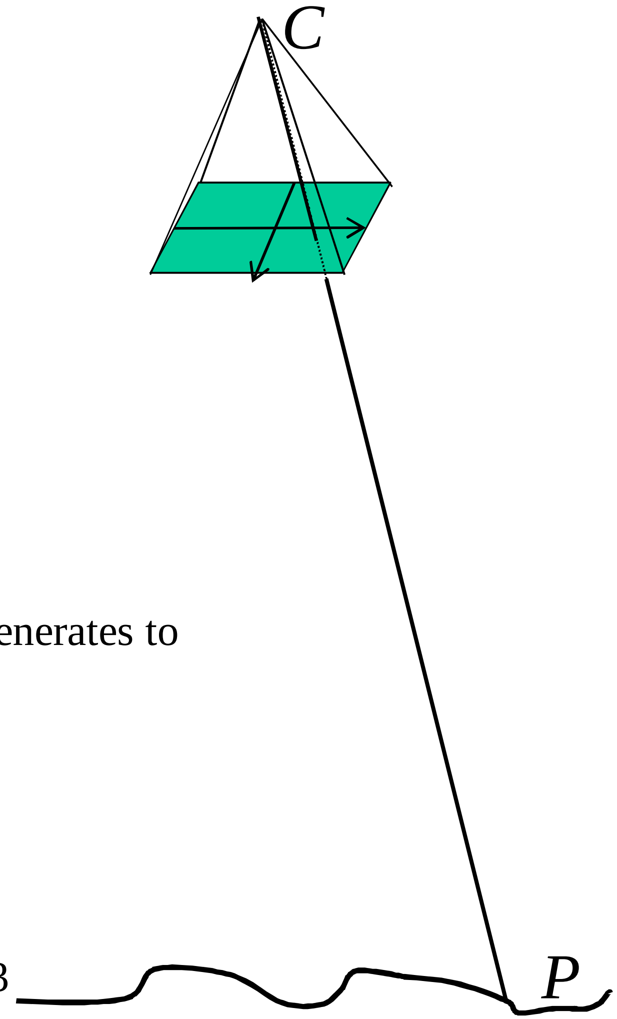
$$P' = {}_2R_1(P - T) = {}_2R_1(P - TN^T P) = {}_2R_1(I - TN^T)P$$

or $$x' = \frac{h_1 x + h_2 y + h_3}{h_7 x + h_8 y + 1} \qquad y' = \frac{h_4 x + h_5 y + h_6}{h_7 x + h_8 y + 1}$$

2D Planer Transformations



3D to 2D Perspective Projection



The 3D to 2D projection can be written as

$$\overline{X} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \cong \begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} {}_c R_w (P - C)$$

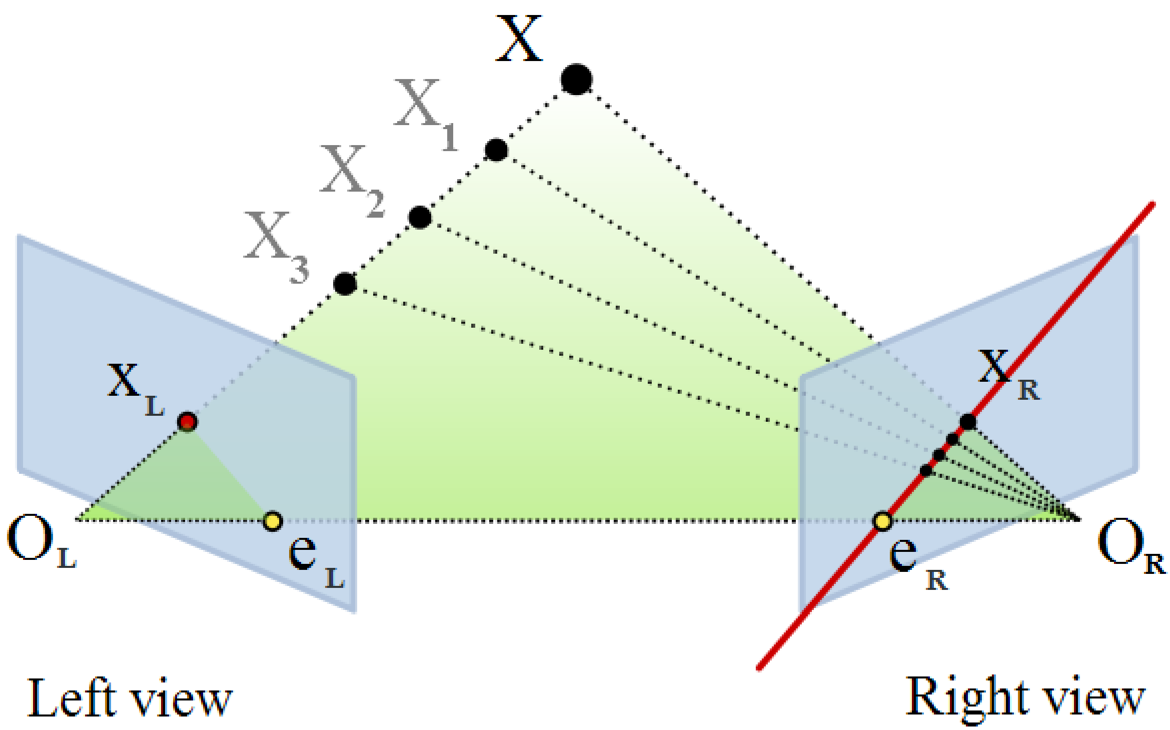
Its unknowns are R and C and its degree of freedom is 6

When the P at infinite distance away (stars), the projection degenerates to

$$\overline{X} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \cong \begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} {}_c R_w r$$

In this case, its unknown is R only and its degree of freedom is 3

Epipolar Constraint (Fundamental matrix)



$$\overline{X}'^{T} F \overline{X} = \overline{X}'^{T} \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & 1 \end{bmatrix} \overline{X} = 0$$

2D to 2D Projection Summery

Transform	Matrix	Parameters p	Jacobian J
translation	$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix}$	(t_x, t_y)	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
Euclidean	$\begin{bmatrix} c_\theta & -s_\theta & t_x \\ s_\theta & c_\theta & t_y \end{bmatrix}$	(t_x, t_y, θ)	$\begin{bmatrix} 1 & 0 & -s_\theta x - c_\theta y \\ 0 & 1 & c_\theta x - s_\theta y \end{bmatrix}$
similarity	$\begin{bmatrix} 1+a & -b & t_x \\ b & 1+a & t_y \end{bmatrix}$	(t_x, t_y, a, b)	$\begin{bmatrix} 1 & 0 & x & -y \\ 0 & 1 & y & x \end{bmatrix}$
affine	$\begin{bmatrix} 1+a_{00} & a_{01} & t_x \\ a_{10} & 1+a_{11} & t_y \end{bmatrix}$	$(t_x, t_y, a_{00}, a_{01}, a_{10}, a_{11})$	$\begin{bmatrix} 1 & 0 & x & y & 0 & 0 \\ 0 & 1 & 0 & 0 & x & y \end{bmatrix}$
projective	$\begin{bmatrix} 1+h_{00} & h_{01} & h_{02} \\ h_{10} & 1+h_{11} & h_{12} \\ h_{20} & h_{21} & 1 \end{bmatrix}$	$(h_{00}, h_{01}, \ldots, h_{21})$	(see Section 6.1.3)

Table 6.1 Jacobians of the 2D coordinate transformations $x' = f(x; p)$ shown in Table 2.1, where we have re-parameterized the motions so that they are identity for $p = 0$.

Linear Least Square

Many of the motion models presented in Section 2.1.2 and Table 2.1, i.e., translation, similarity, and affine, have a *linear* relationship between the amount of motion $\Delta \boldsymbol{x} = \boldsymbol{x}' - \boldsymbol{x}$ and the unknown parameters $\boldsymbol{p}$,

$$\Delta \boldsymbol{x} = \boldsymbol{x}' - \boldsymbol{x} = \boldsymbol{J}(\boldsymbol{x})\boldsymbol{p}, \tag{6.4}$$

where $\boldsymbol{J} = \partial \boldsymbol{f} / \partial \boldsymbol{p}$ is the *Jacobian* of the transformation $\boldsymbol{f}$ with respect to the motion parameters $\boldsymbol{p}$ (see Table 6.1). In this case, a simple *linear* regression (linear least squares problem) can be formulated as

$$\begin{aligned} E_{\mathrm{LLS}} &= \sum_i \|\boldsymbol{J}(\boldsymbol{x}_i)\boldsymbol{p} - \Delta \boldsymbol{x}_i\|^2 && (6.5) \\ &= \boldsymbol{p}^T \left[\sum_i \boldsymbol{J}^T(\boldsymbol{x}_i)\boldsymbol{J}(\boldsymbol{x}_i) \right] \boldsymbol{p} - 2\boldsymbol{p}^T \left[\sum_i \boldsymbol{J}^T(\boldsymbol{x}_i)\Delta \boldsymbol{x}_i \right] + \sum_i \|\Delta \boldsymbol{x}_i\|^2 && (6.6) \\ &= \boldsymbol{p}^T \boldsymbol{A} \boldsymbol{p} - 2\boldsymbol{p}^T \boldsymbol{b} + c. && (6.7) \end{aligned}$$

The minimum can be found by solving the symmetric positive definite (SPD) system of *normal equations*[2]

$$\boldsymbol{A}\boldsymbol{p} = \boldsymbol{b}, \tag{6.8}$$

where

$$\boldsymbol{A} = \sum_i \boldsymbol{J}^T(\boldsymbol{x}_i)\boldsymbol{J}(\boldsymbol{x}_i) \tag{6.9}$$

Nonlinear Least Square (brief)

To minimize the non-linear least squares problem, we iteratively find an update $\Delta \boldsymbol{p}$ to the current parameter estimate $\boldsymbol{p}$ by minimizing

$$\begin{aligned} E_{\mathrm{NLS}}(\Delta \boldsymbol{p}) &= \sum_i \|\boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{p} + \Delta \boldsymbol{p}) - \boldsymbol{x}'_i\|^2 && (6.13) \\ &\approx \sum_i \|\boldsymbol{J}(\boldsymbol{x}_i; \boldsymbol{p})\Delta \boldsymbol{p} - \boldsymbol{r}_i\|^2 && (6.14) \\ &= \Delta \boldsymbol{p}^T \left[\sum_i \boldsymbol{J}^T \boldsymbol{J}\right] \Delta \boldsymbol{p} - 2\Delta \boldsymbol{p}^T \left[\sum_i \boldsymbol{J}^T \boldsymbol{r}_i\right] + \sum_i \|\boldsymbol{r}_i\|^2 && (6.15) \\ &= \Delta \boldsymbol{p}^T \boldsymbol{A} \Delta \boldsymbol{p} - 2\Delta \boldsymbol{p}^T \boldsymbol{b} + c, && (6.16) \end{aligned}$$

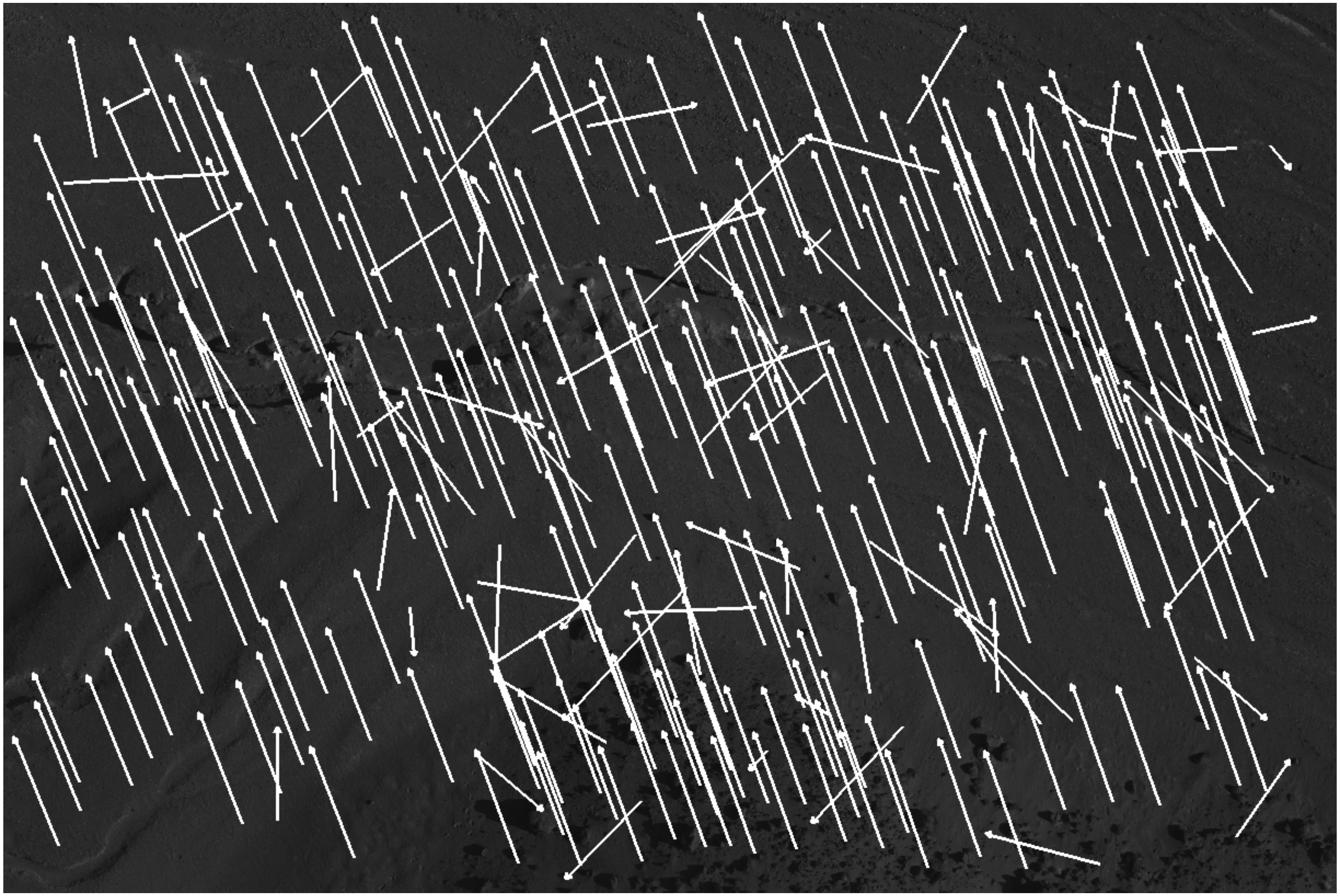
where the "Hessian"[5] $\boldsymbol{A}$ is the same as Equation (6.9) and the right hand side vector

$$\boldsymbol{b} = \sum_i \boldsymbol{J}^T(\boldsymbol{x}_i)\boldsymbol{r}_i \qquad (6.17)$$

$$(A + \lambda diag(A))\Delta P = b$$

$$P = P + \Delta P$$

Feature Inliers and Outliers



The RANSAC (RAndom SAmple Consensus) Algorithm

Assume:

- The parameters can be estimated from *N* data items.
- There are *M* data items in total.
- The probability of a randomly selected data item being part of a good model is .
- The probability that the algorithm will exit without finding a good fit if one exists is .

The algorithm:

1. selects *N* data items at random
2. estimates parameter (linear or nonlinear least square)
3. finds how many data items (of *M*) fit the model with parameter vector within a user given tolerance, *T*. Call this *K*.
4. if *K* is the largest (best fit) so far, accept it.
5. repeat 1. to.4 *S* times

Questions:

What is the tolerance?

How many trials, *S*, to ensure success

Some considerations for defining feature correspondence tolerance

- Methods used in finding correspondences
  - Feature matching ~ 1 to 2 pixels
  - Feature tracking ~0.3 to 0.5 pixel
  - Star centroid ~0.1 pixel
- Senior type, data quality and environment
- Consider least-median-of-squares method if the tolerance is hard to be determined.

$$\min median(r_i^2)$$

How many trials, *S*, to ensure success

- To ensure that RANSAC has high chance to find correct inliers, a sufficient number of trials much be executed. Let *p* be the probability of inliers of any given correspondence and *P* is a success probability after *S* trials. We have

$$(1-P)=(1-p^k)^S$$

- And

$$S=\frac{\log(1-P)}{\log(1-p^k)}$$

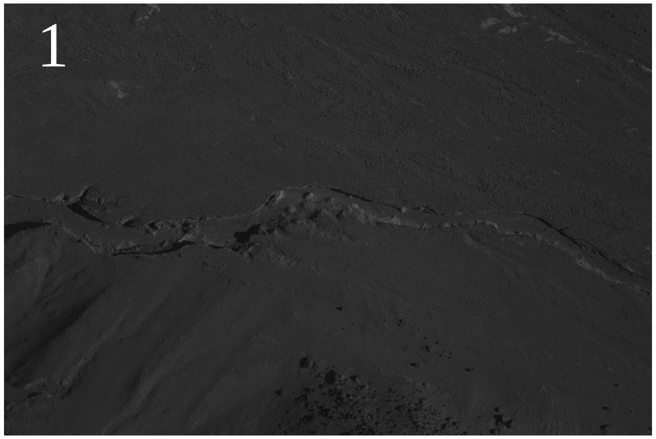
RANSAC Number of Trials

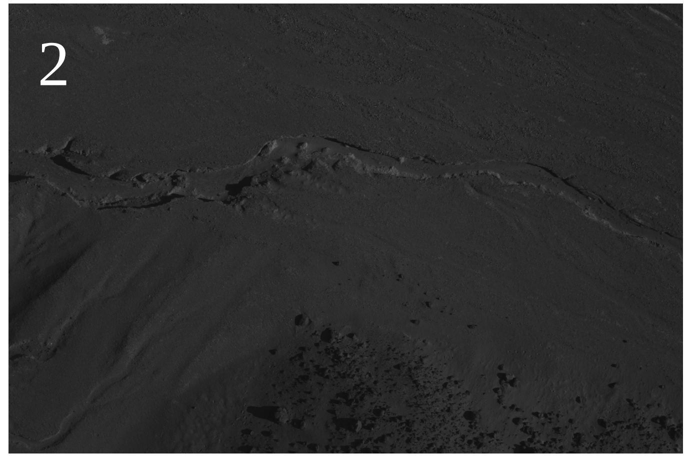
k	p	S
3	0.5	35
6	0.6	97
6	0.5	293

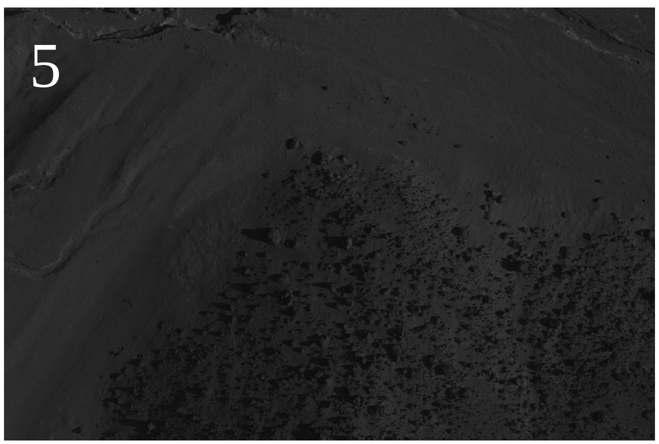
Table 6.2 Number of trials S to attain a 99% probability of success (Stewart 1999).

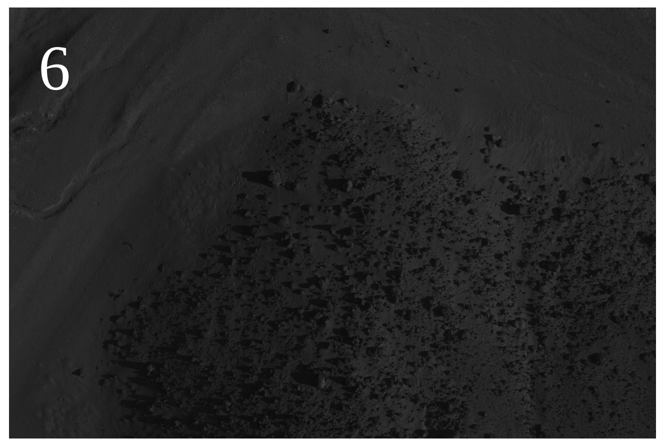
RANSAC Applications Example

Mars Hill Image Sequence

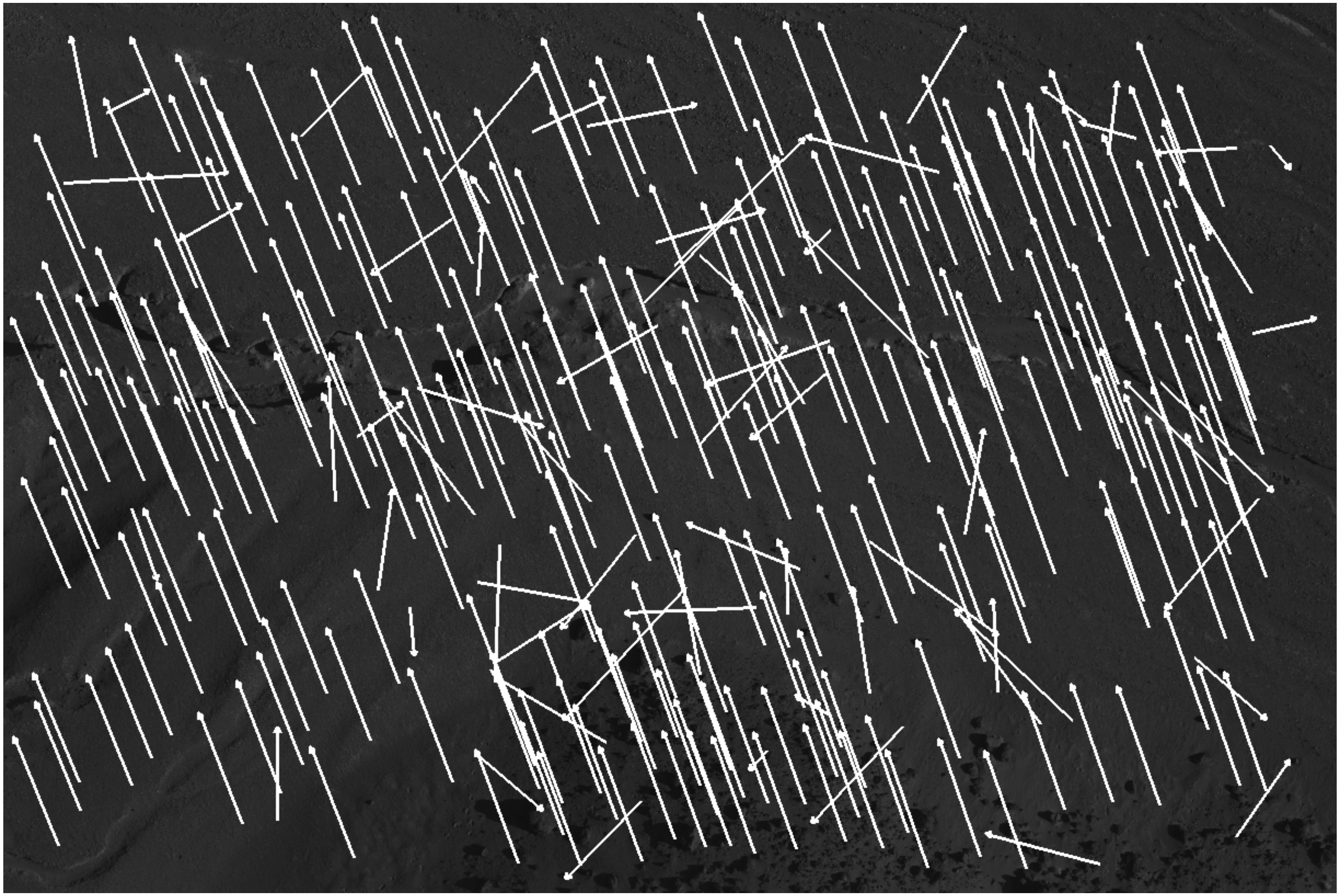




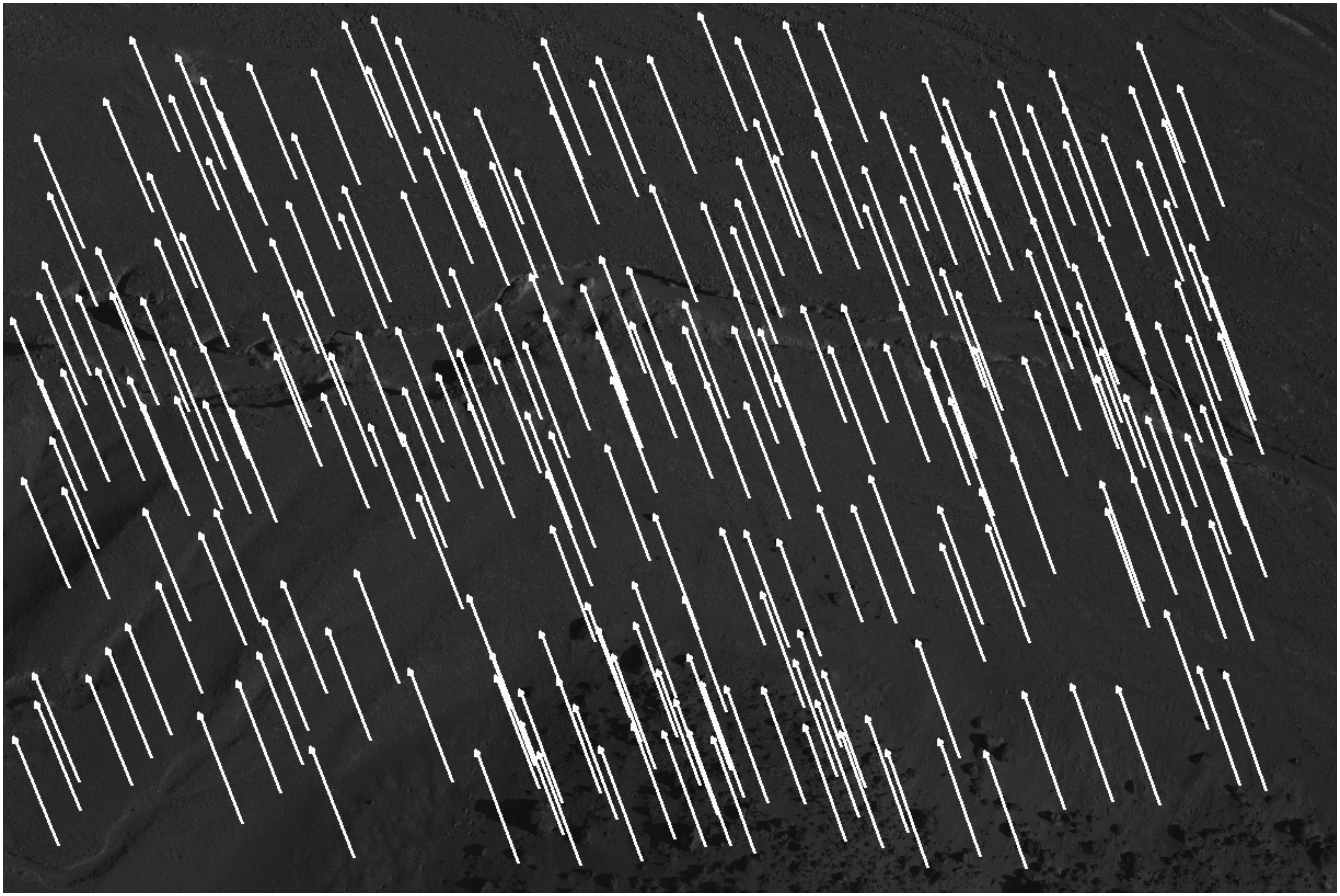


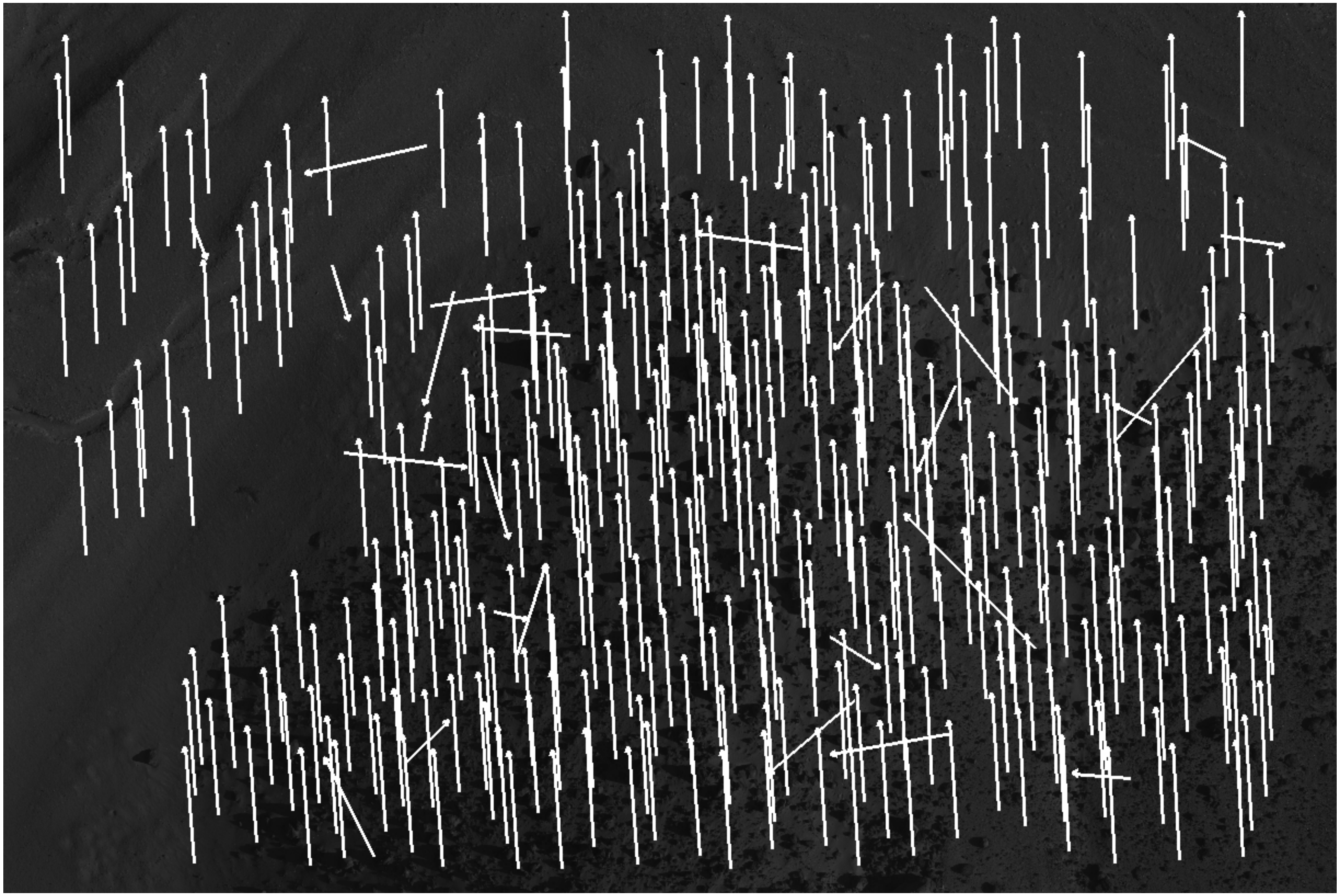


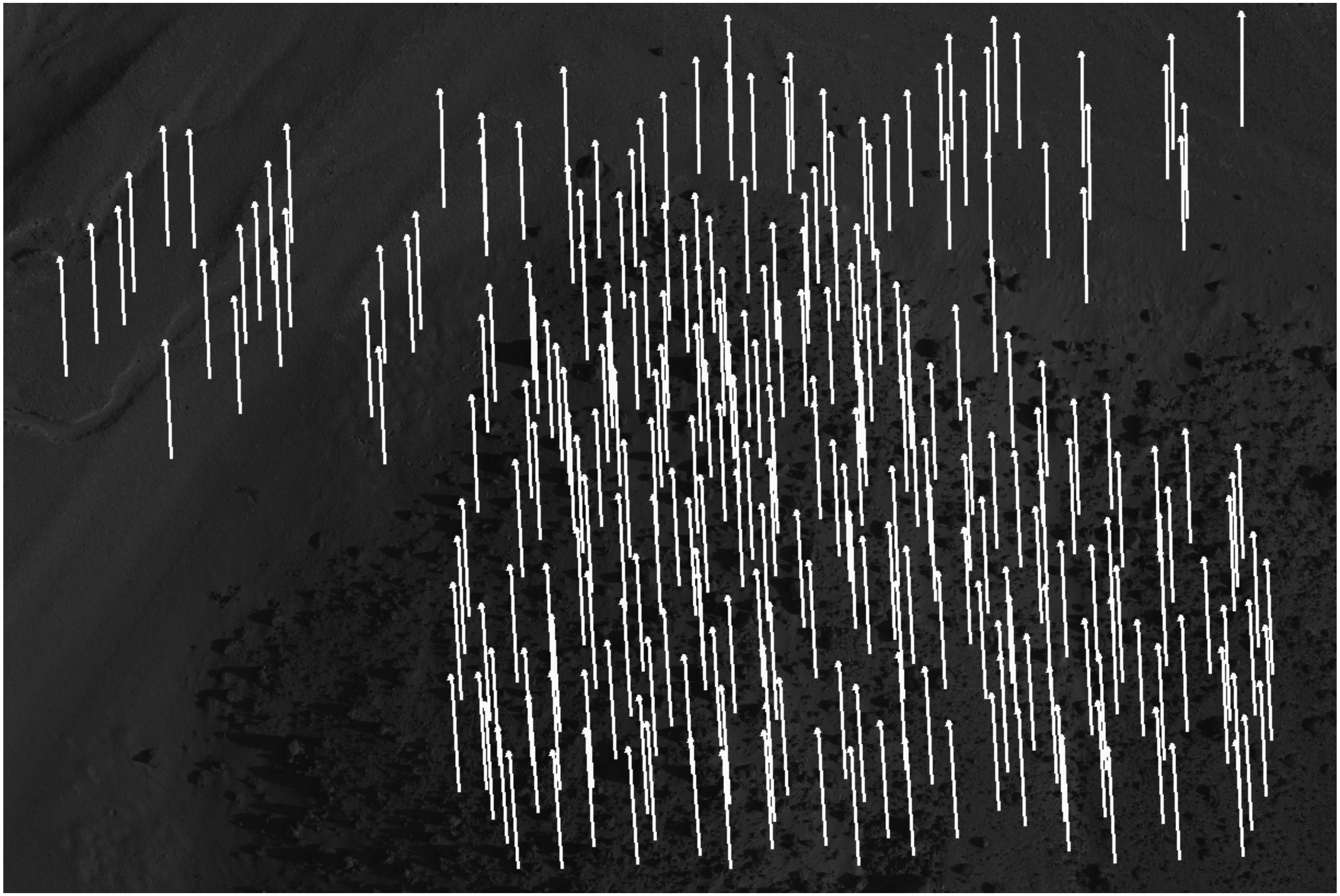
Feature Outlier Detection



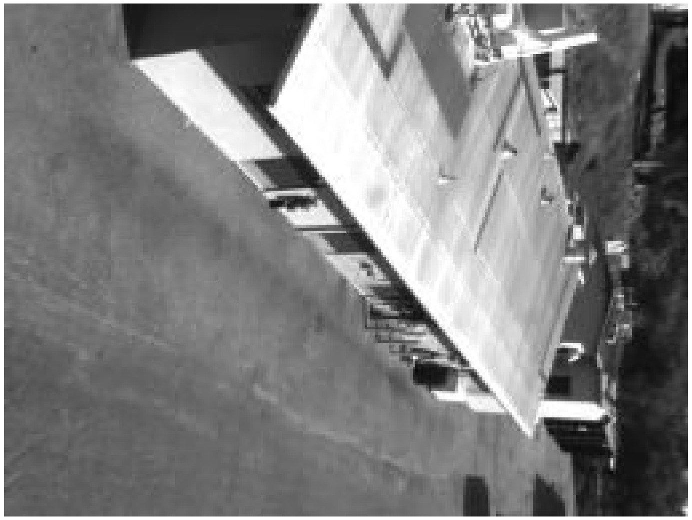
Inliners

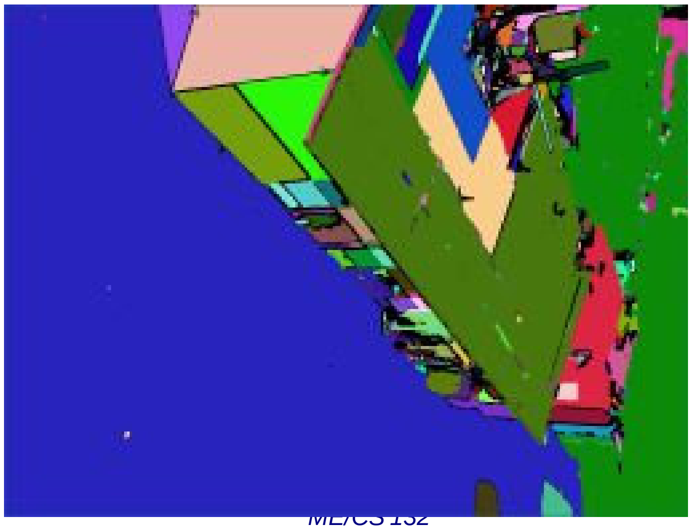




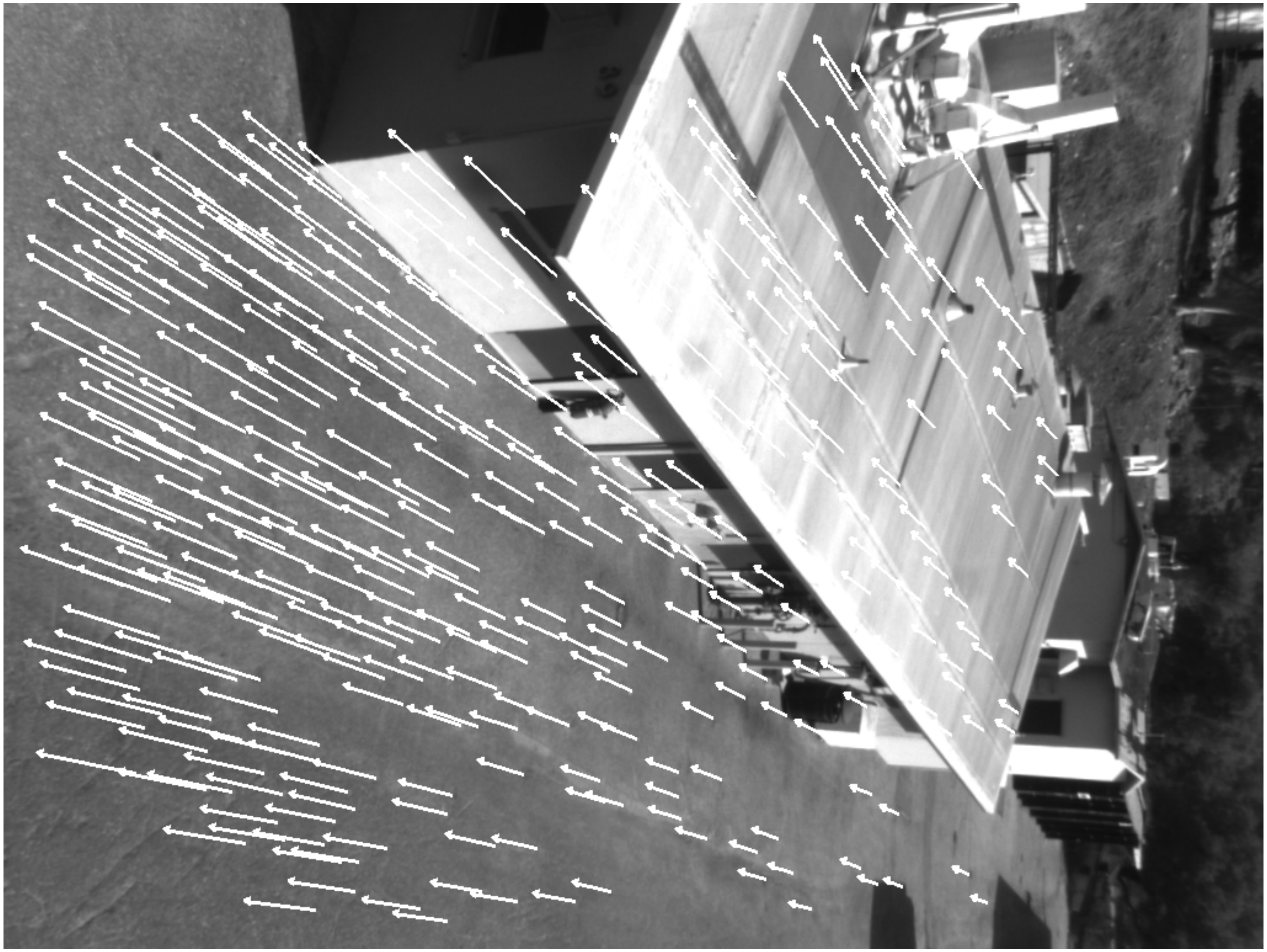


Feature Tracking & Object Detection Example

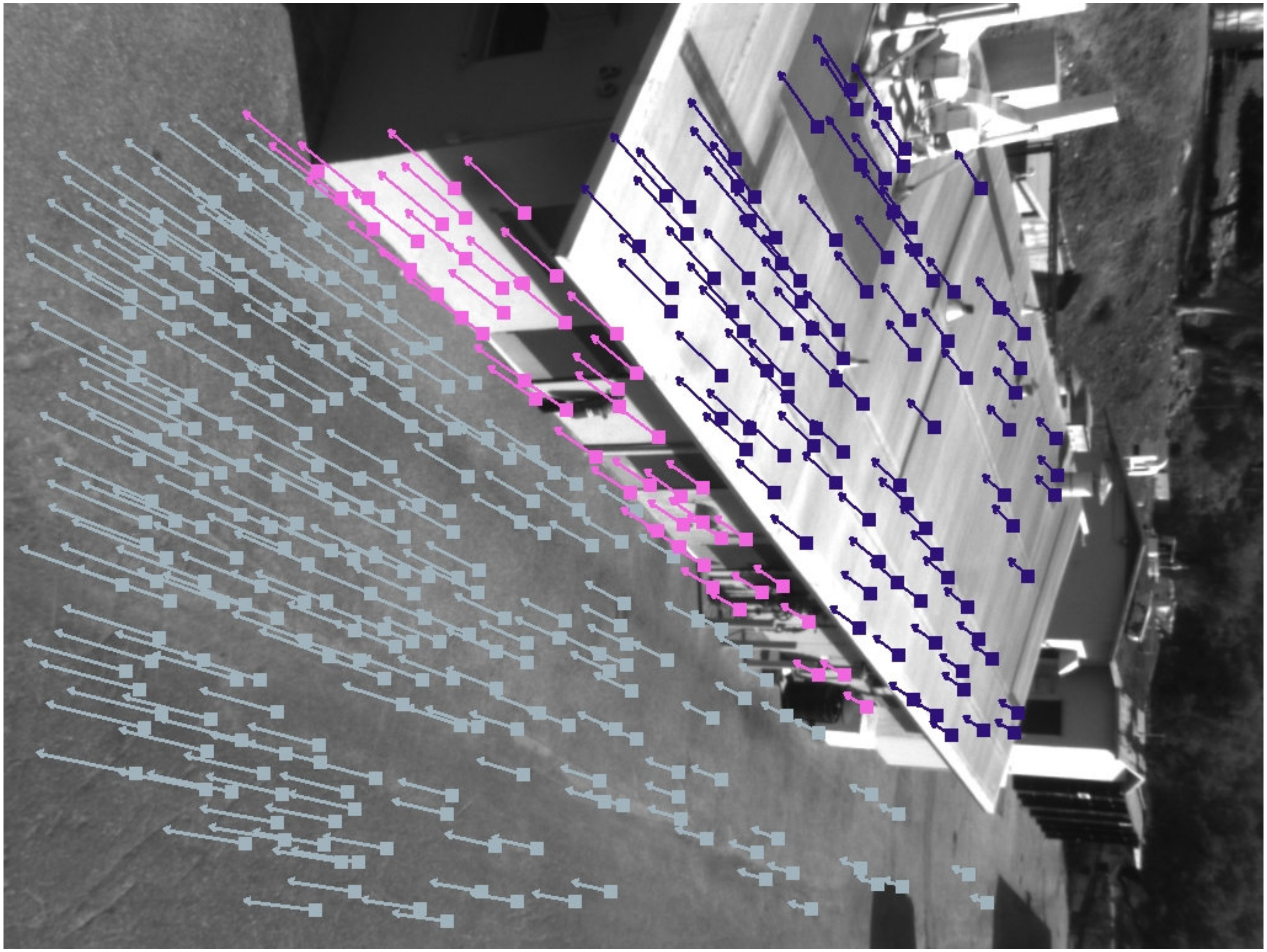




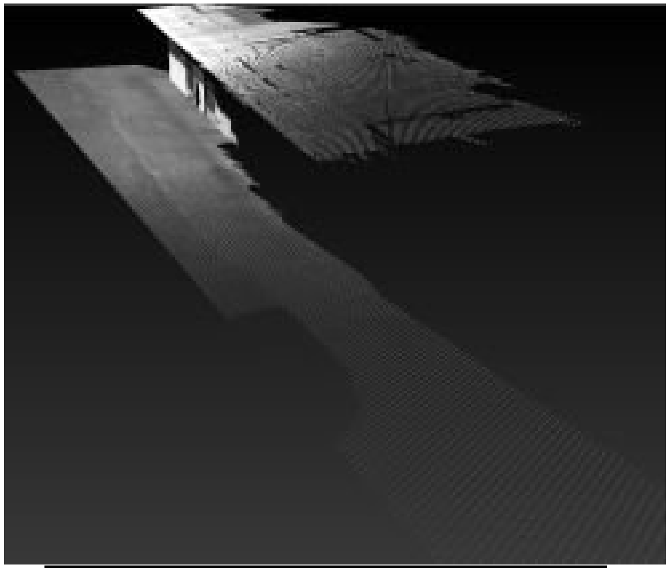
Tracked Features

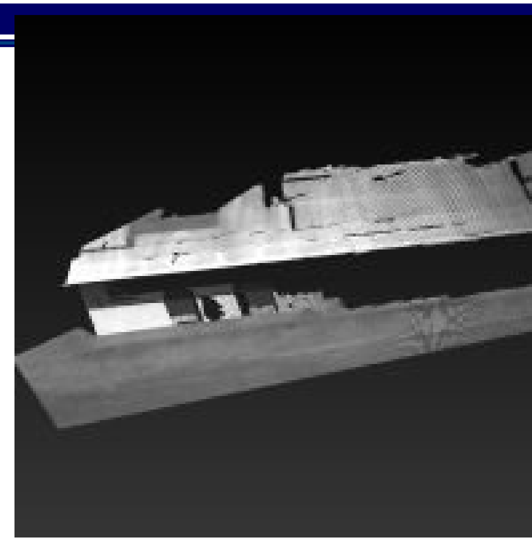


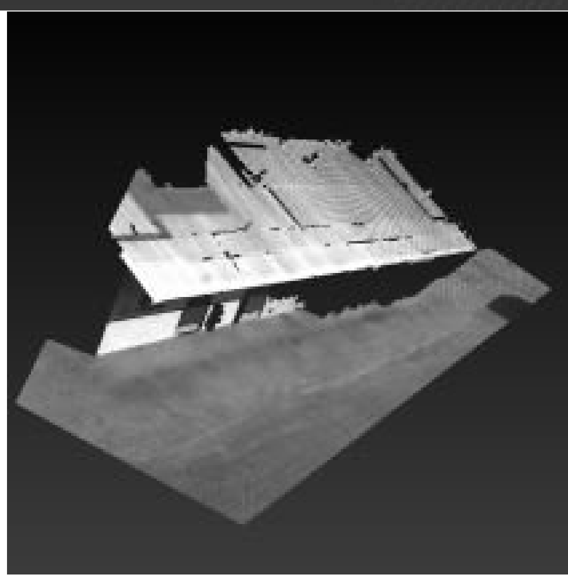
RANSAC Finds Three Major Planes in this Scence

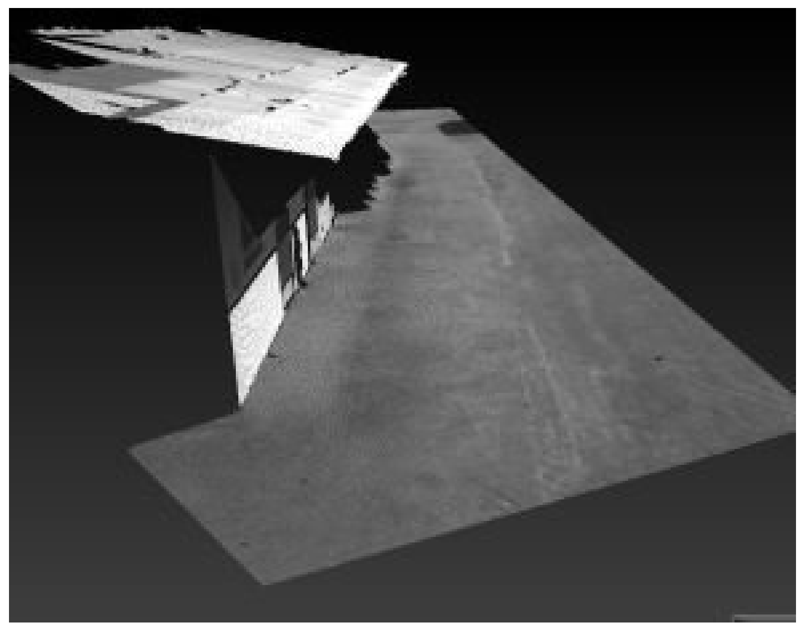


Reconstructed 3D model



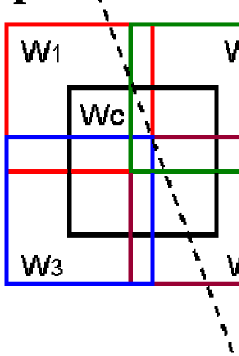






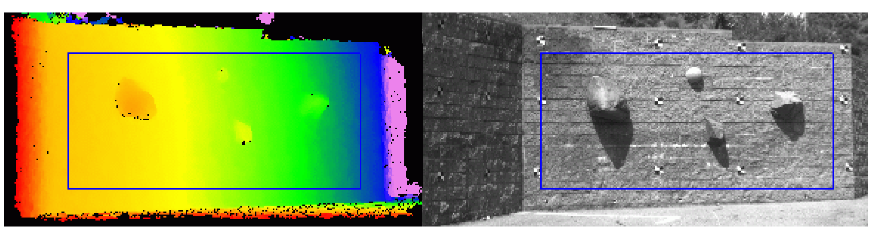
Slope Estimation Algorithm

Step 1: Stereo to produce elevation map

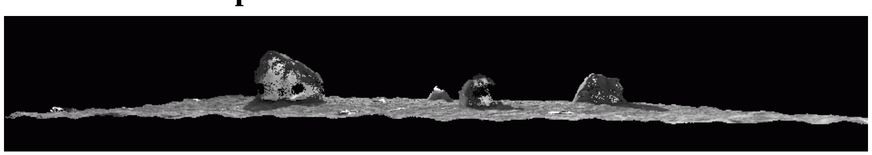


SAD5 correlator performs well at range discontinuities; improves overall SNR

Stereo sub-pixel disparity

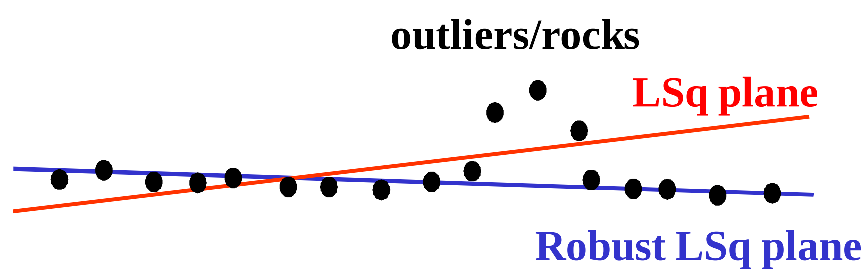


3d elevation map



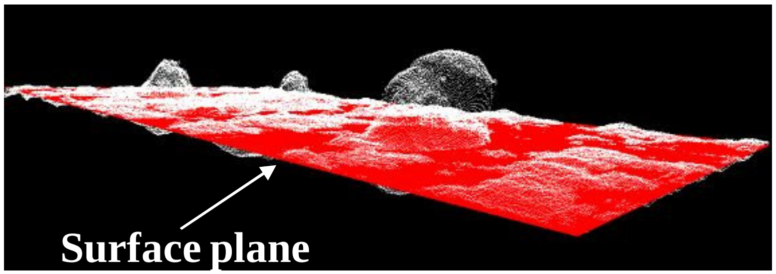
Step2: Robust plane fitting to get slope

- underlying surface is desired for slope computation
- rocks are outliers
- repeat for multiple triples
  - **select random triple of points and fit plane**
  - **if median of square plane error is minimum keep as best**
- throw out points that are far from plane
- fit Least Square (LSq) plane to remaining points

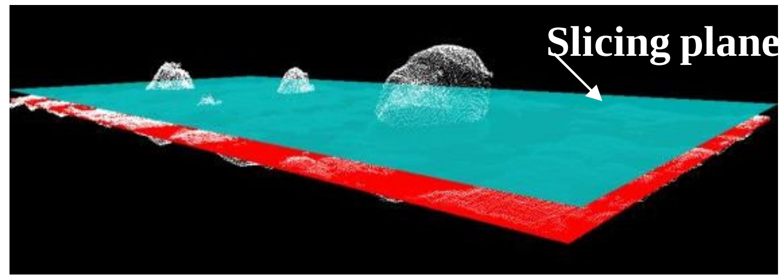


Rock Detection Algorithm

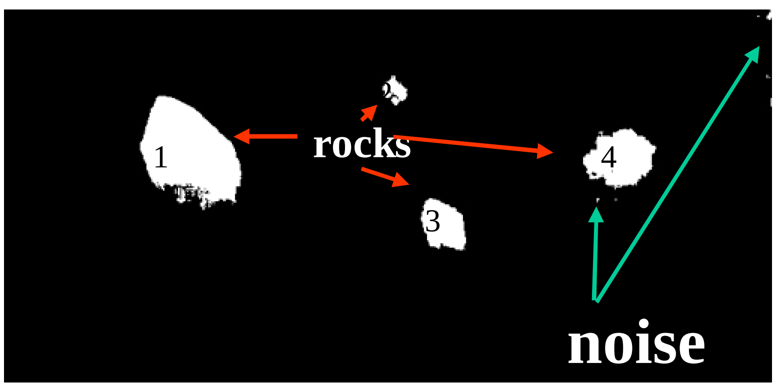
1. Robust plane fit over point cloud



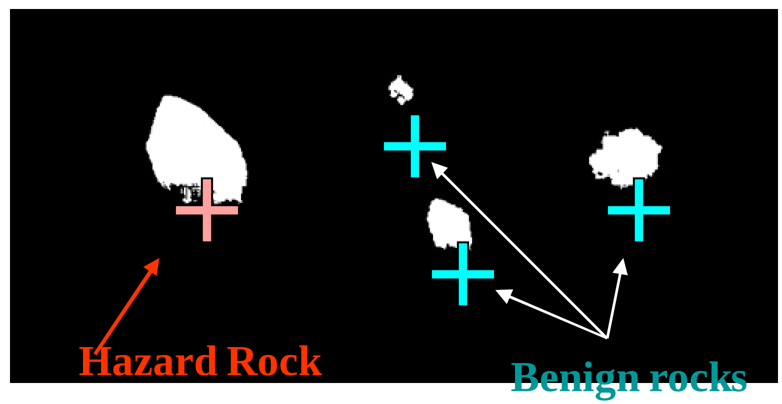
2. Slice point cloud 1σ above the plane



3. Extract connected regions and remove very small regions as noise



4. Estimate rock heights and positions by averaging the top *n* highest points



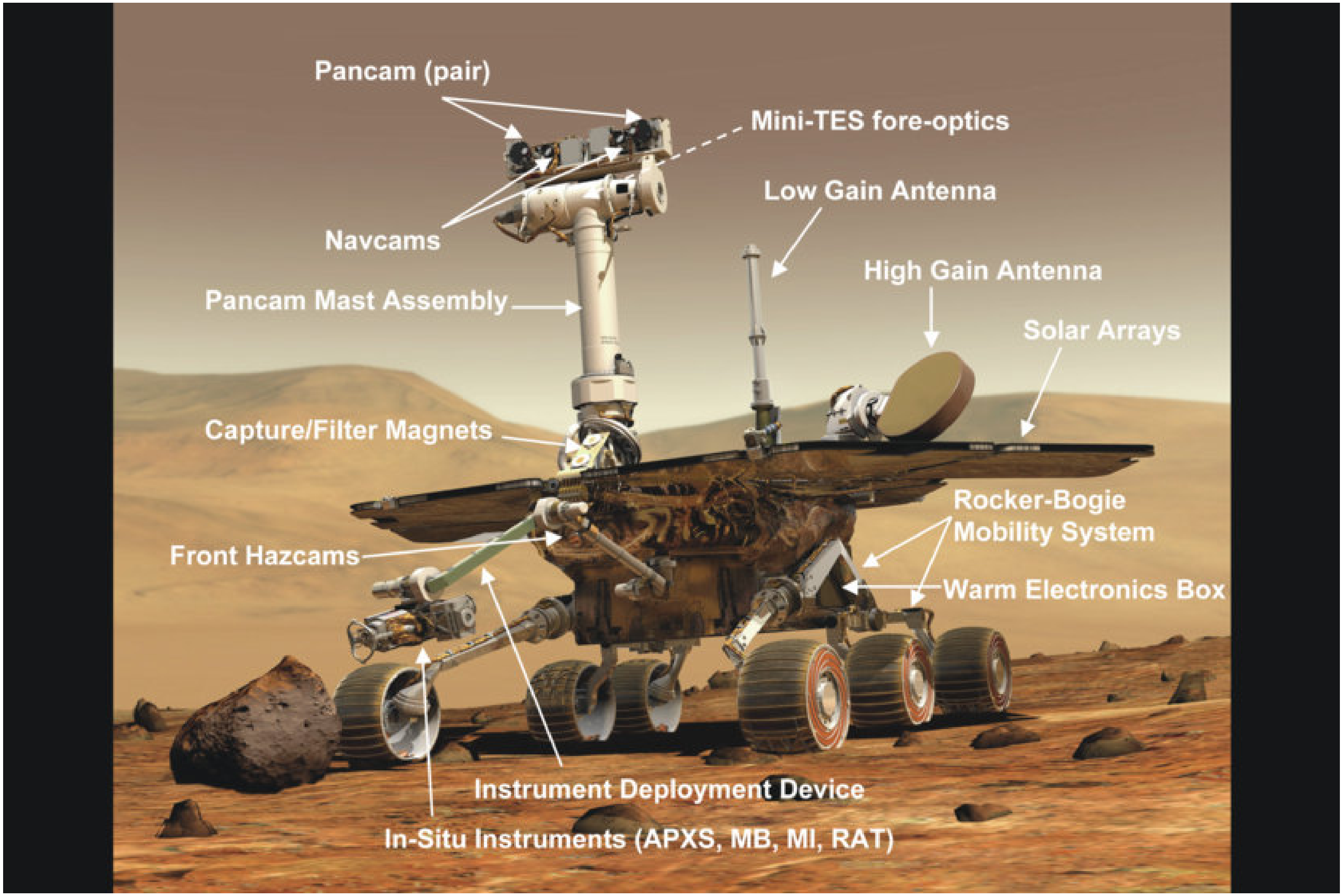
Summary

- RANSAC is a very powerful and commonly used method for outlier removal
- In order to reduce the number of trials, to reduce the DOF of a problem can dramatically reduce the number of trials
- Some time, best use of known knowledge, such as attitude, translation etc also can reduce the computation
-

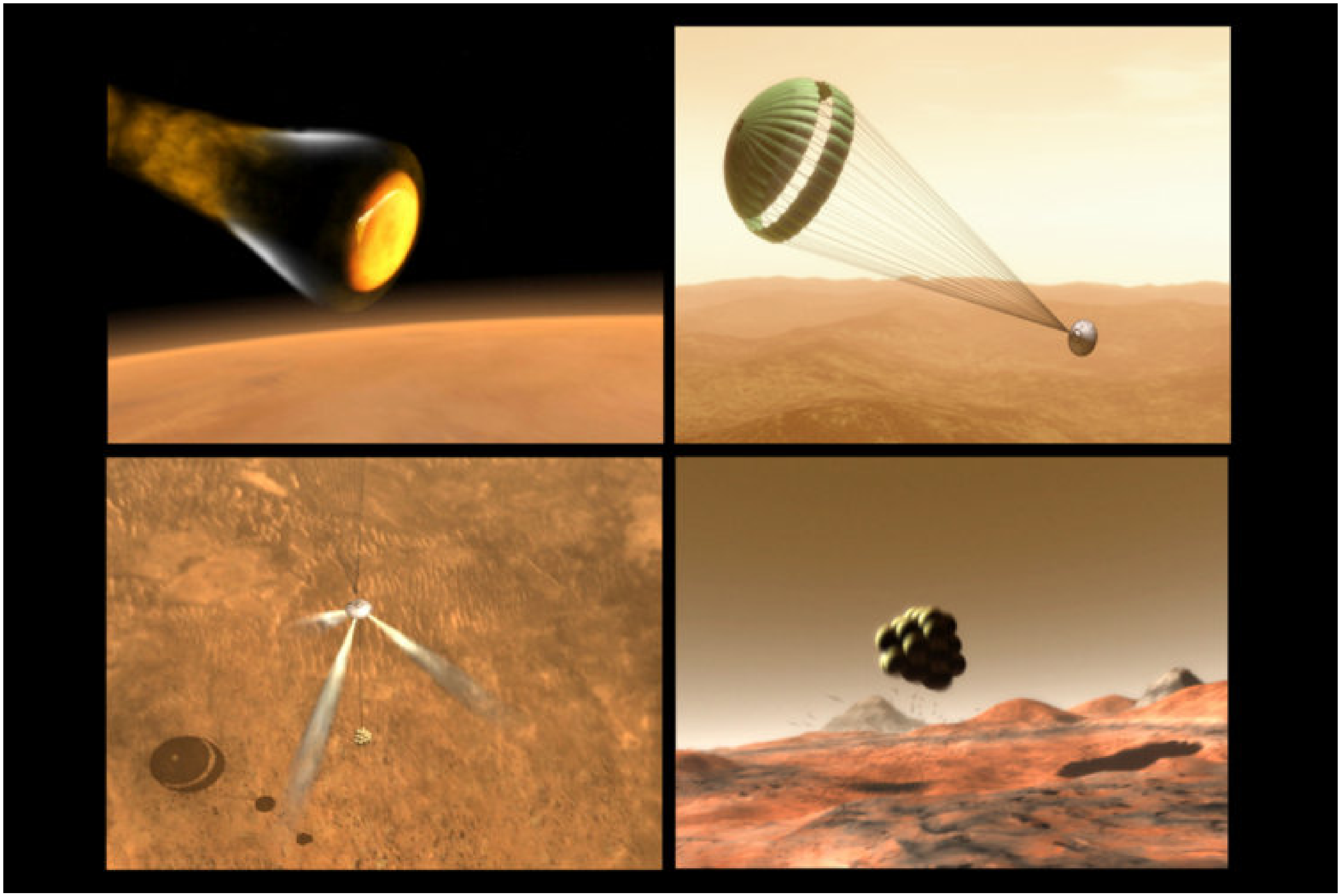
Nonparametric Methods (Case Study)

- The parametric approach above assumes a large number of features are available and majority of them are inliers. However, in extreme simulations, such as spacecraft landing, due to very fast descending speed, low on board computer power, we can not offer to tracker many feature as we hope. In many case, only one or a few features are available

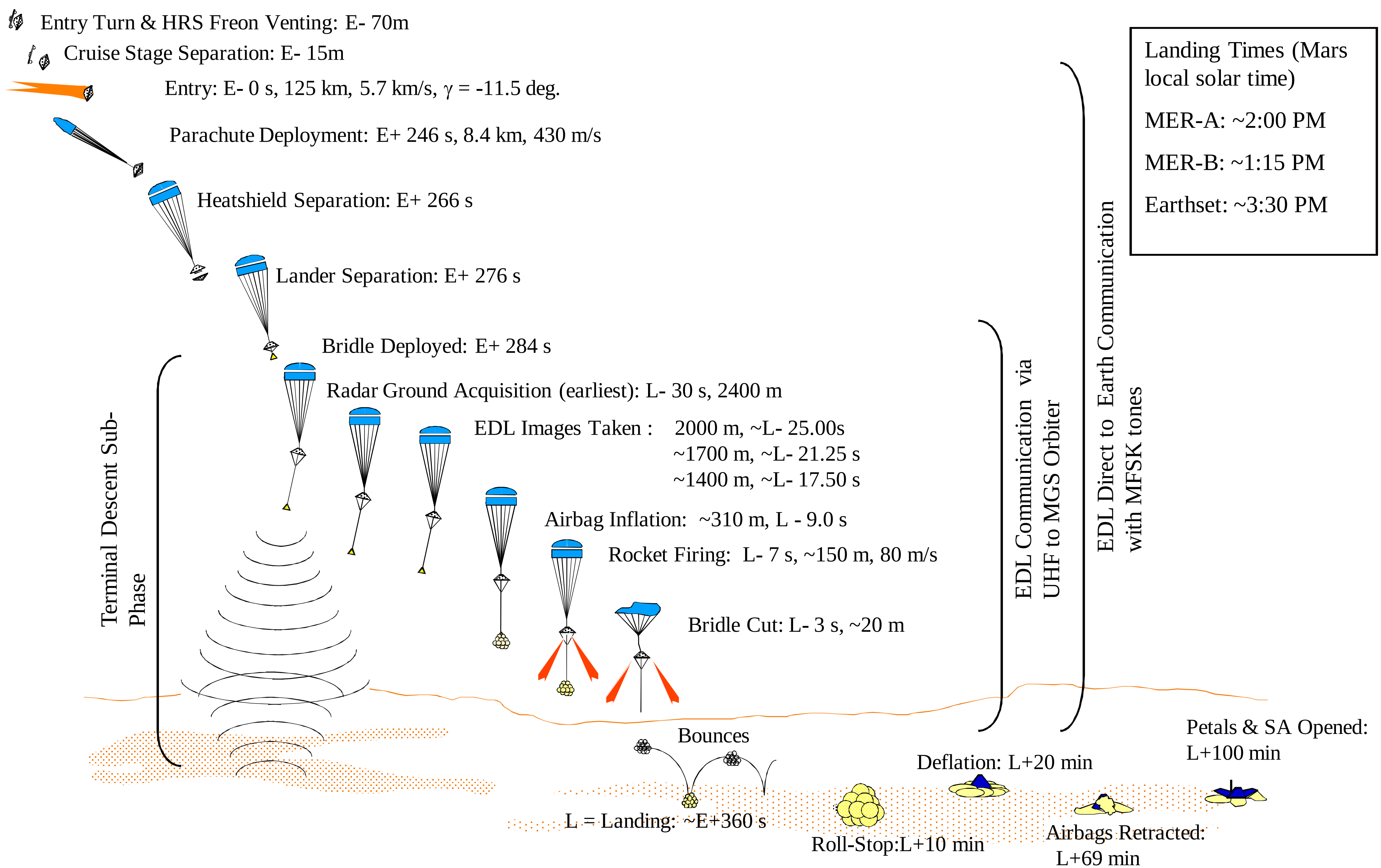
Mars Exploration Rovers (MER)
Descent Image Motion Estimation System (DIMES)



Pancam (pair)
Mini-TES fore-optics
Low Gain Antenna
Navcams
High Gain Antenna
Pancam Mast Assembly
Solar Arrays
Capture/Filter Magnets
Rocker-Bogie
Mobility System
Front Hazcams
Warm Electronics Box
Instrument Deployment Device
In-Situ Instruments (APXS, MB, MI, RAT)



MER Entry, Descent & Landing Scenario



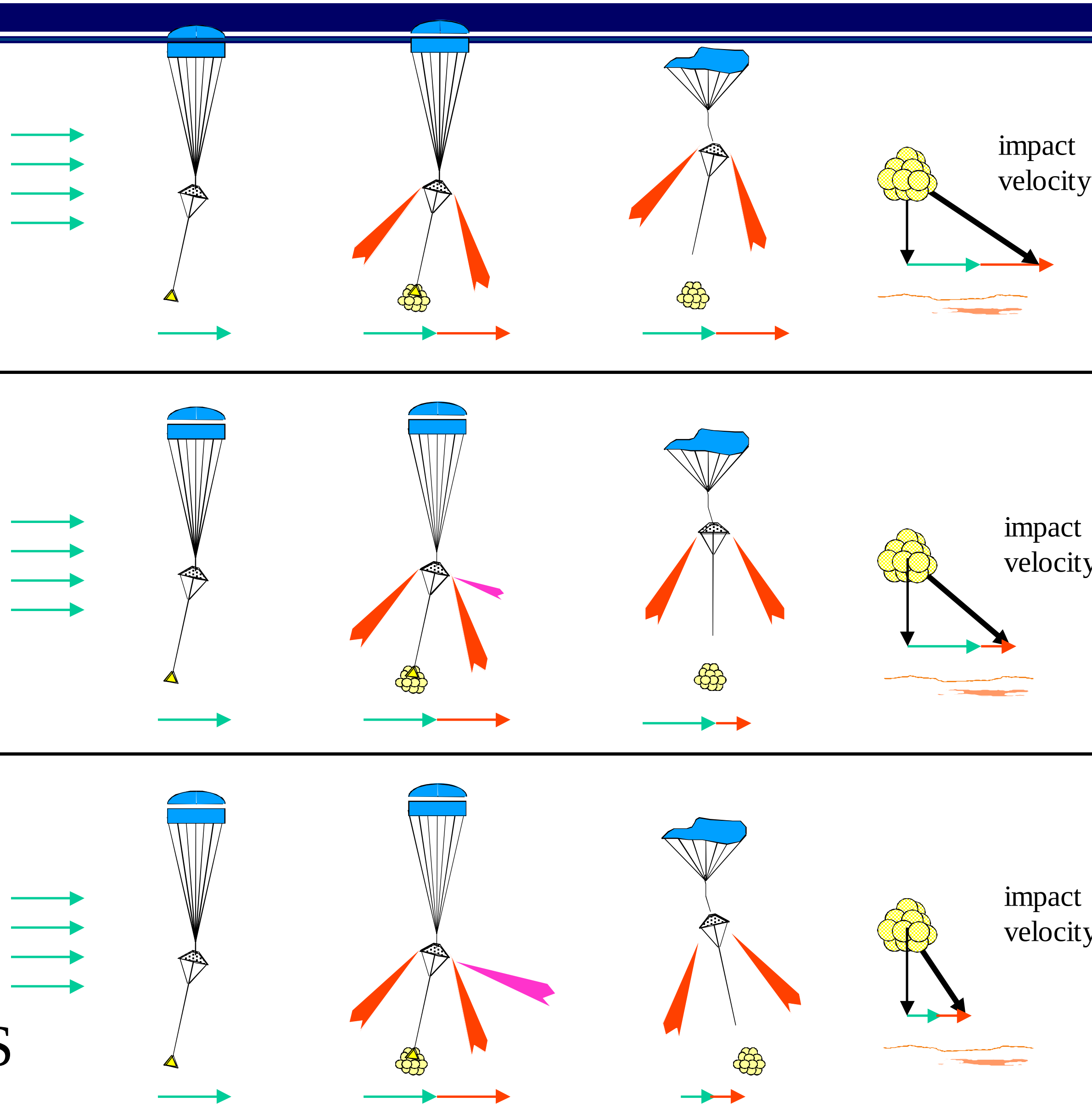
Airbag is vulnerable to sharp & pointing rocks

Effect and Mitigation of Winds

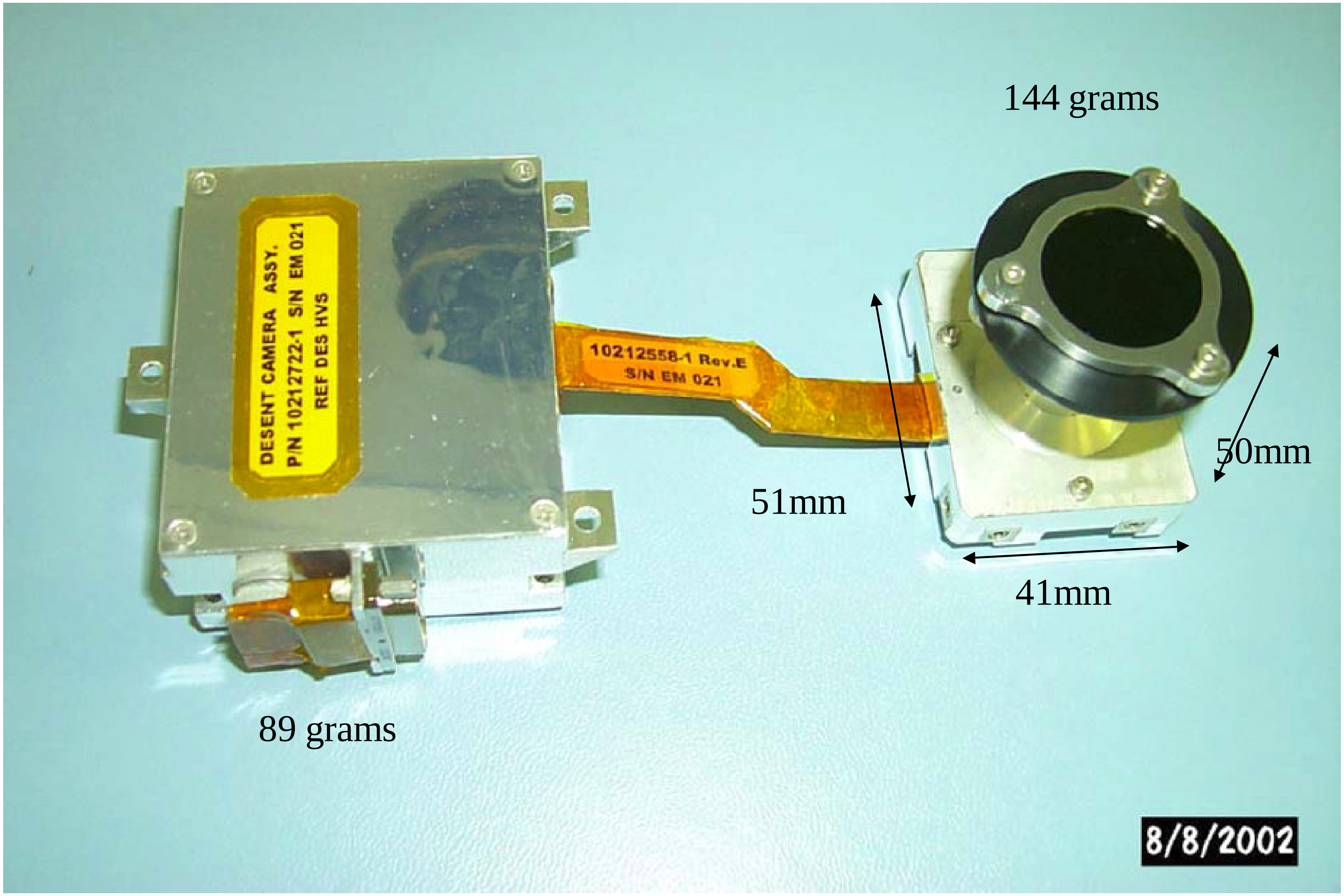
Pathfinder

Pathfinder with TIRS

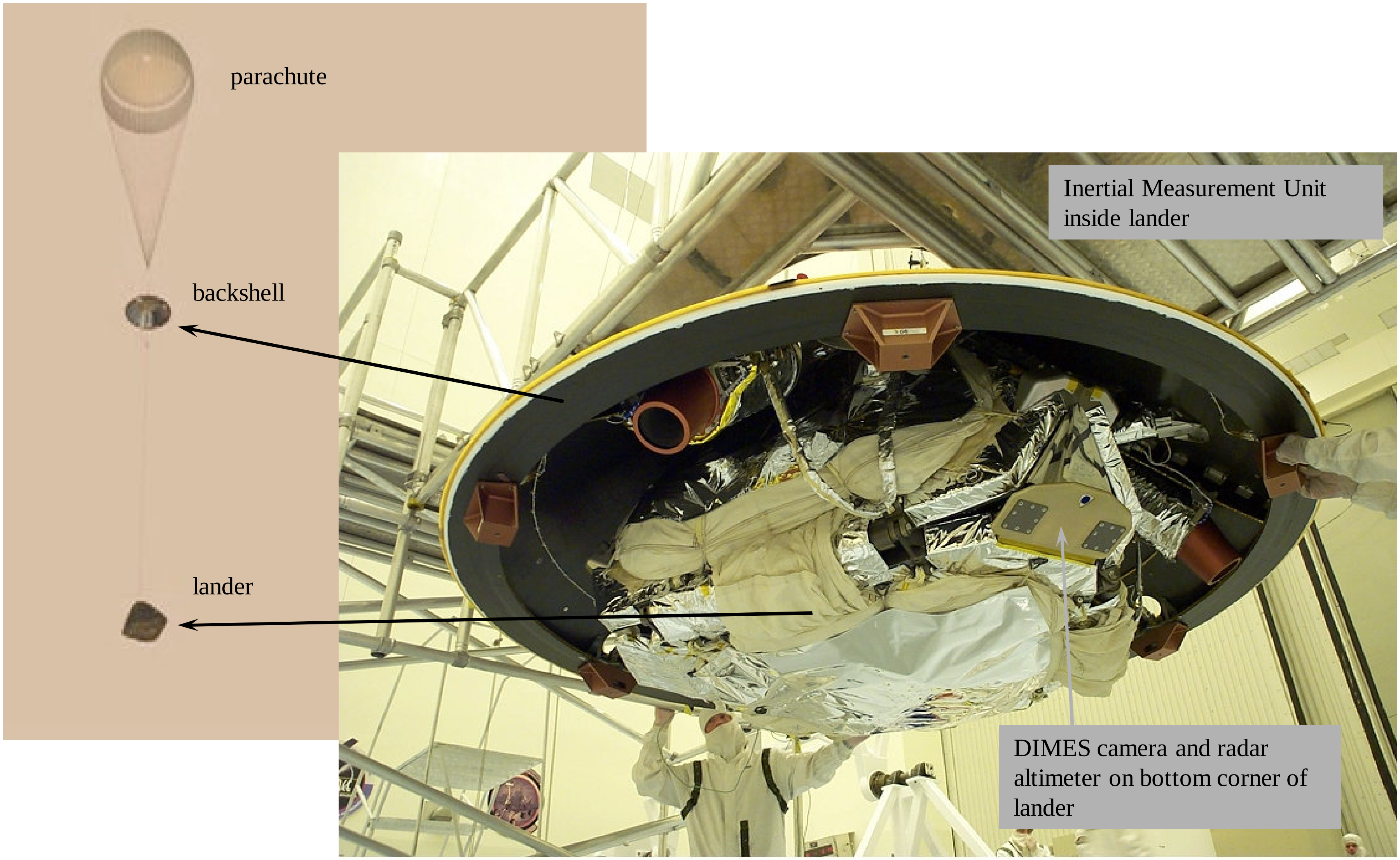
Pathfinder with TIRS and DIMES



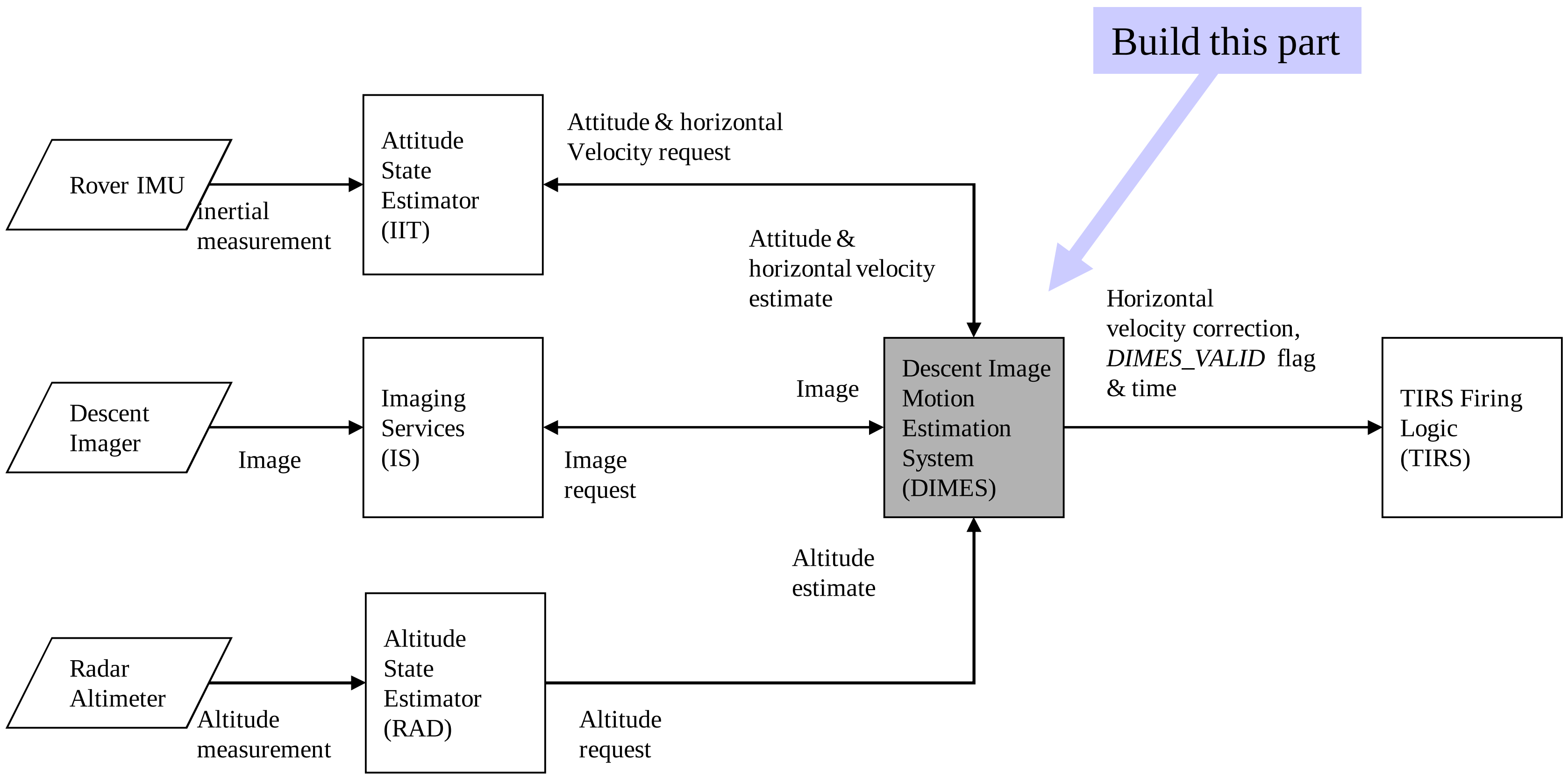
DIMES Camera



DIMES Hardware



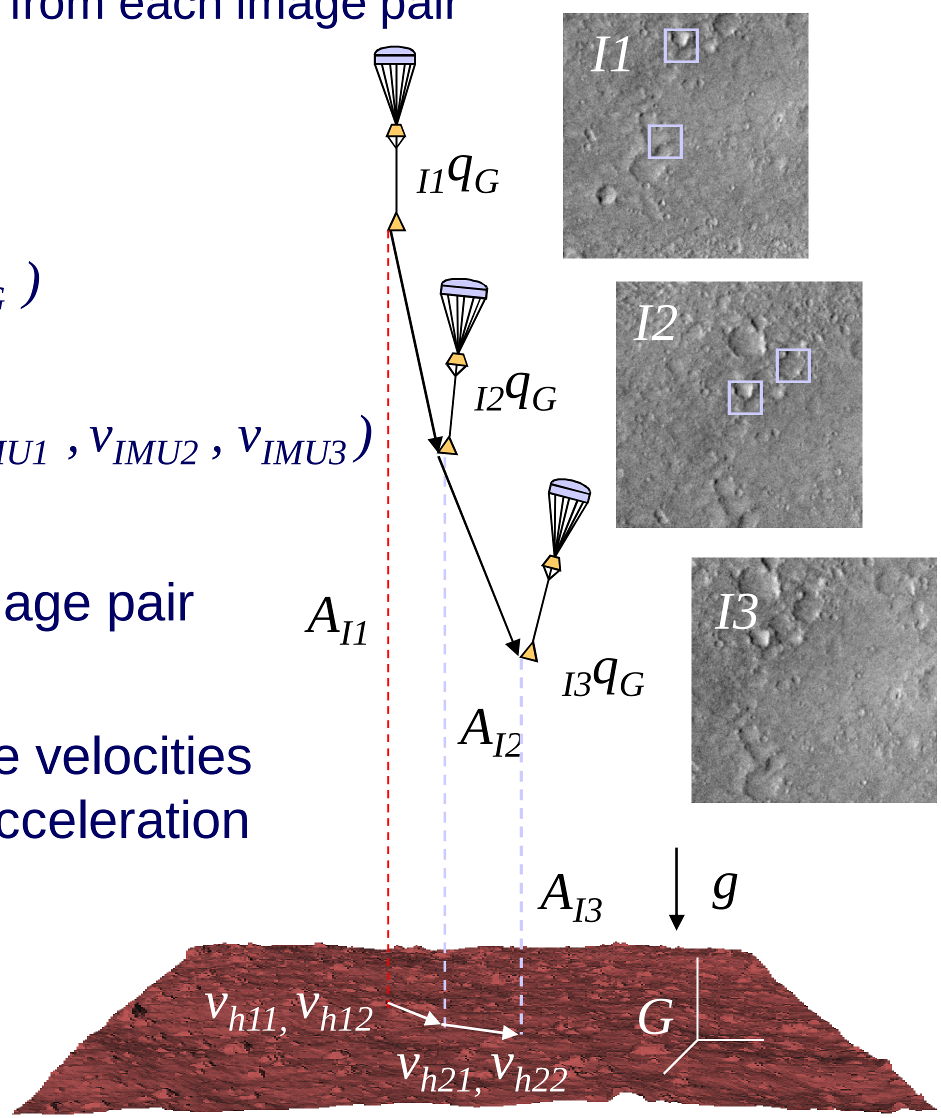
DIMES Functional Block Diagram



DIMES Algorithm

Using three images and two templates from each image pair improves overall DIMES robustness

- Input
  - 3 images $(I1 , I2, I3)$
  - 3 IMU attitudes $({}_{I1}q_G , {}_{I2}q_G , {}_{I3}q_G)$
  - 3 radar altitudes $(A_{I1} , A_{I2} , A_{I3})$
  - 3 IMU horizontal velocities $(v_{IMU1} , v_{IMU2} , v_{IMU3})$
- Algorithm
  - track two templates in each image pair
  - verify correlation of templates
  - compare difference of template velocities between image pairs to IMU acceleration

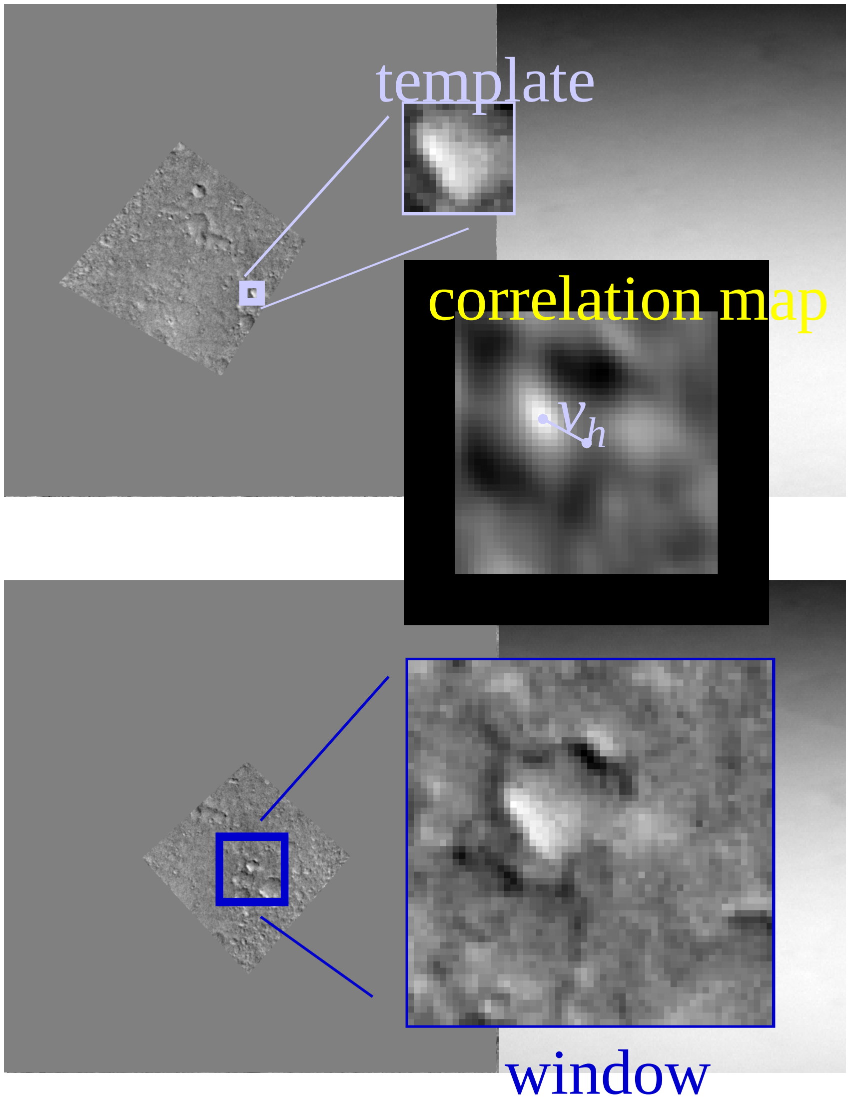


Challenges to Development of DIMES

- Development from concept to flight qualified system in 26 months before landing
  - accommodate camera and software into mature EDL system with minimal impact
- Numerous non-ideal imaging effects possible during EDL
  - bland landing sites, dust, cosmic rays, heatshield
- Algorithm must never generate an incorrect velocity
  - algorithm must be self checking
- Algorithm must run with minimal processing resources
  - 20% of 20 MHz RAD6000 for 20 seconds
- MER cameras not designed for descent imaging
  - motion blur, frame transfer smear, long readout time
- Imaging in the loop never used during EDL
  - skeptics wanted to kill the development
- Development must be low cost

DIMES Motion Estimation Concept

(not the actual optimized order of operations)



Correct Images

- Bin each image
- Radiometric correction of each image.
- Rectify each image to ground plane using IMU attitude and radar altitude.

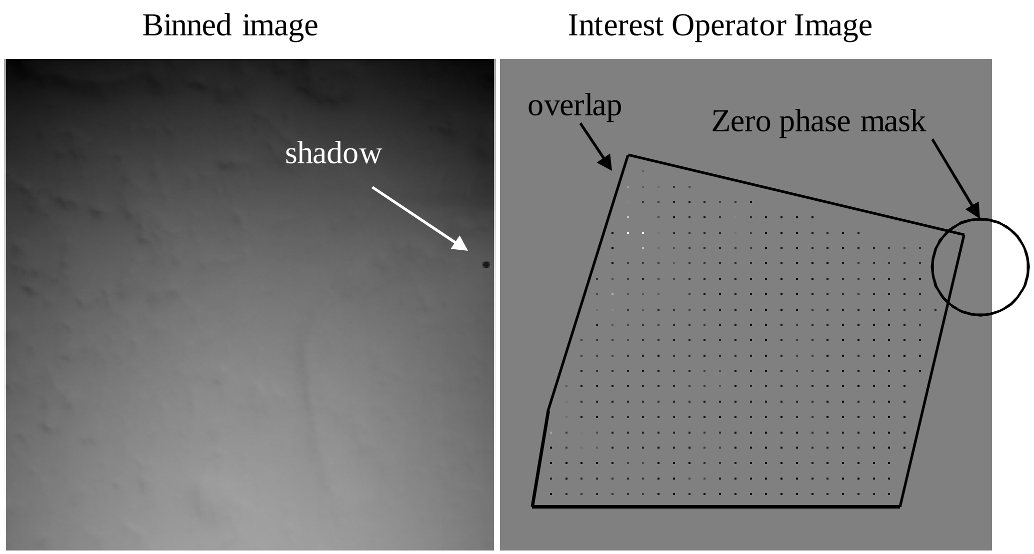
Correlate Images

- Apply Interest Operator to first image.
- Select high contrast template in image overlap that avoids zero phase spot.
- Slide template over window in second image and at each pixel compute linear correlation coefficient between template and window DN.
- Find maximum correlation and compute correlation performance metrics.
- Compute horizontal velocity from template shift and VALID measurement.

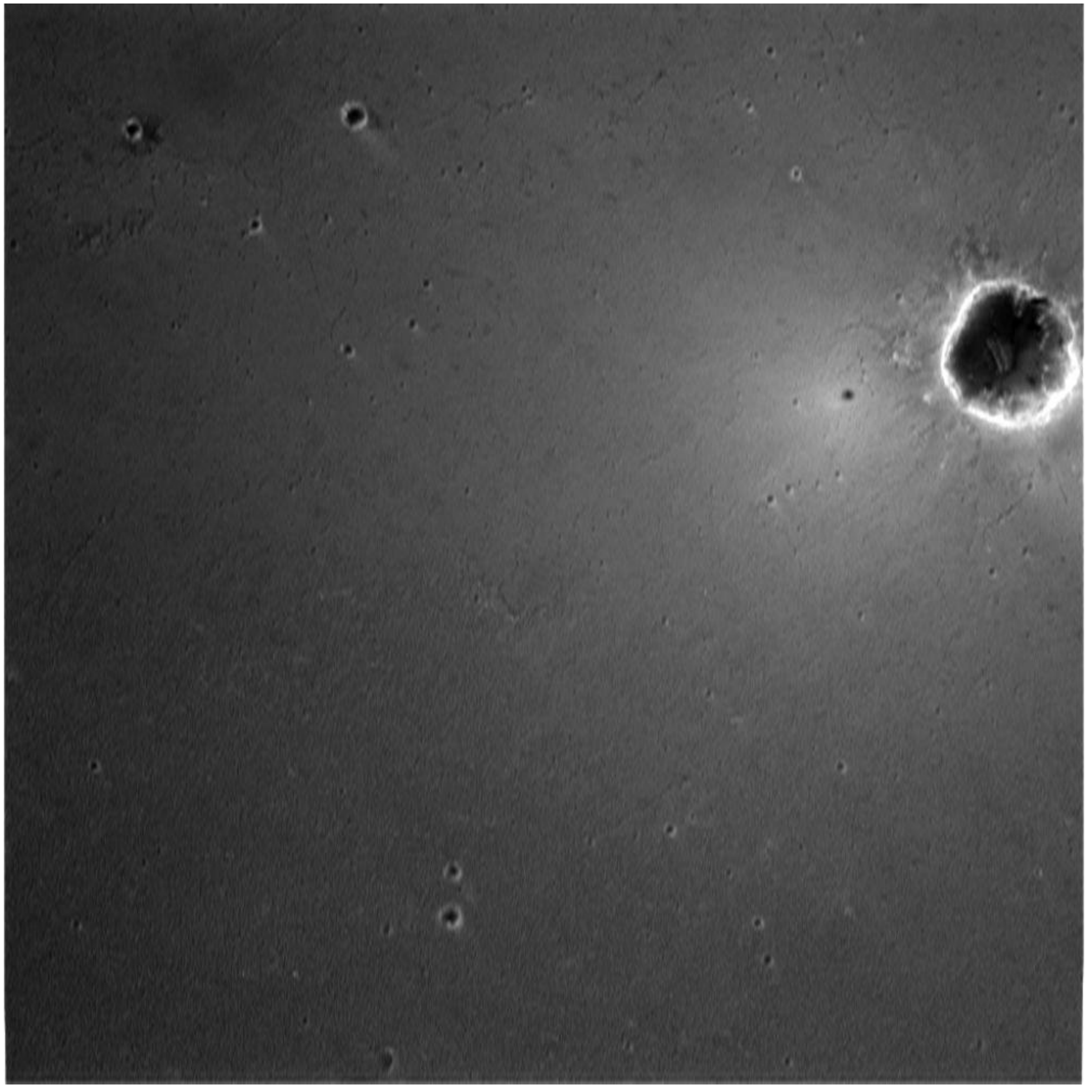
DIMES Algorithm Details

Template Selection

- Standard Interest operator
  - smallest eigenvalue of template autocorrelation
- Efficient Implementation
  - Interest operator computed before rectification and image flattening
    - Only template and window need to be rectified and flattened
  - Computed on a coarse grid
    - Width of template broadens interest operator peaks, so skip pixels
- Application Region
  - Only computed in overlap region of images
  - Sun direction parameter is used to mask out region around zero phase
    - Zero phase brightening
    - Parachute shadow



Opposition Effect



Mask off Opposition Effect

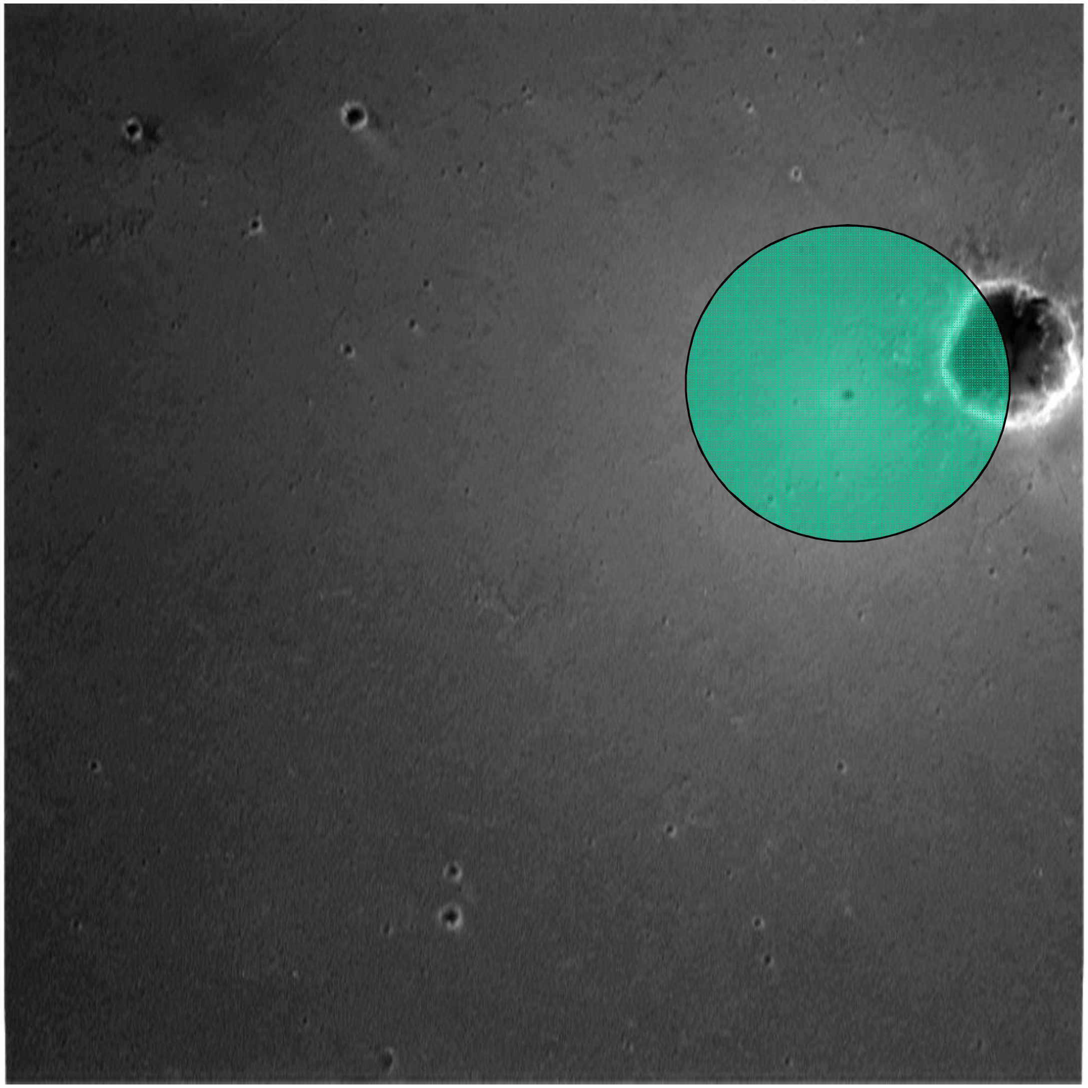
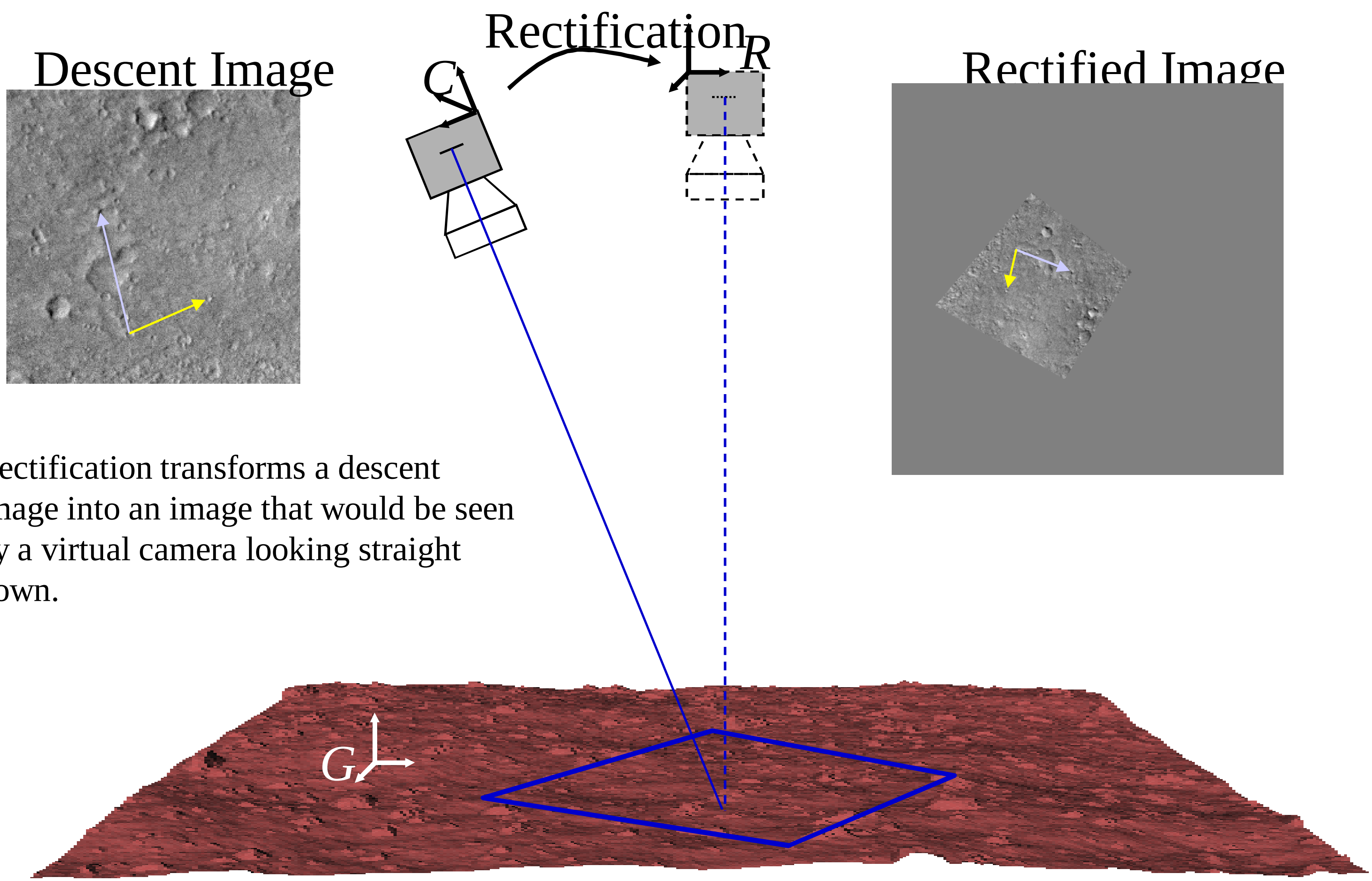


Image Rectification Concept



Rectification transforms a descent image into an image that would be seen by a virtual camera looking straight down.

Image Rectification

- Requires position and attitude of camera in ground frame
  - Ground relative attitude comes from IMU
  - Use altitude from radar altimeter for vertical position
  - Assume lander is dropping straight down (i.e., horizontal position is zero)
- Based on flat surface approximation
  - Surface slope and terrain relief introduce minor errors in rectification
    - high altitude
    - landing site safety requires small slopes and terrain relief
- Fast implementation
  - only rectify templates and window pixels
  - Approximate rectification with local homography
    - more efficient than reprojection using camera model with radial distortion

$$X = \frac{a_1 x + a_2 y + a_3}{a_7 x + a_8 y + 1} \qquad Y = \frac{a_4 x + a_5 y + a_6}{a_7 x + a_8 y + 1}$$

Two Stage Correlation

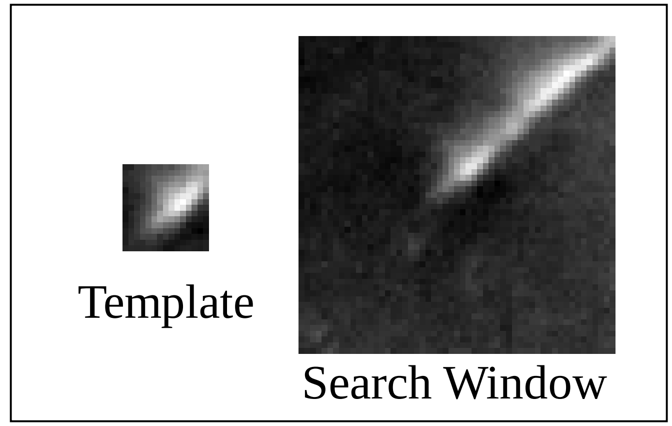
- Psuedo-normalized correlation

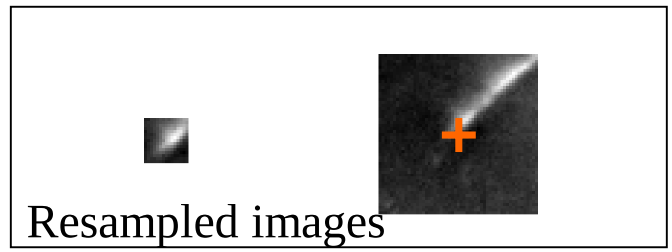
$$R = (2\sum_T \tilde{I}_1 * \tilde{I}_2)/(\sum_T \tilde{I}_1^2 + \sum_T \tilde{I}_2^2)$$

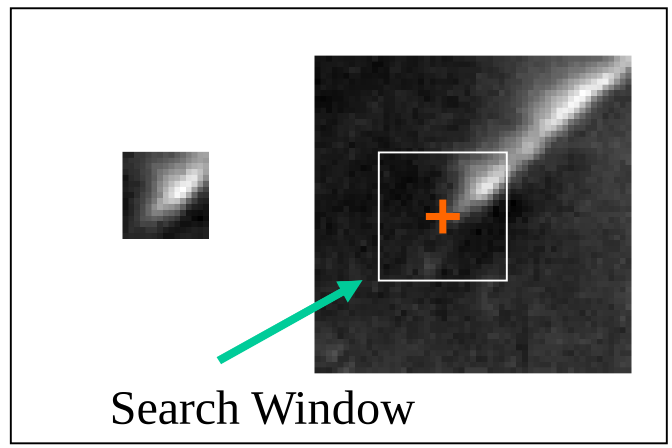
$$\tilde{I}_i = I_i - \sum_T I_i / N$$

- Speed up correlation by applying it at a coarse and then fine image resolution.
  1. Generate coarse data by binning template and search window to 2x2 resolution
  2. Correlate coarse template and window to get best and second best match locations
  3. Project the best correlation locations into the 1x1 resolution window.
  4. Find the best correlations in a smaller window around the projected point.

- Results in 2x speed improvement over single stage correlation

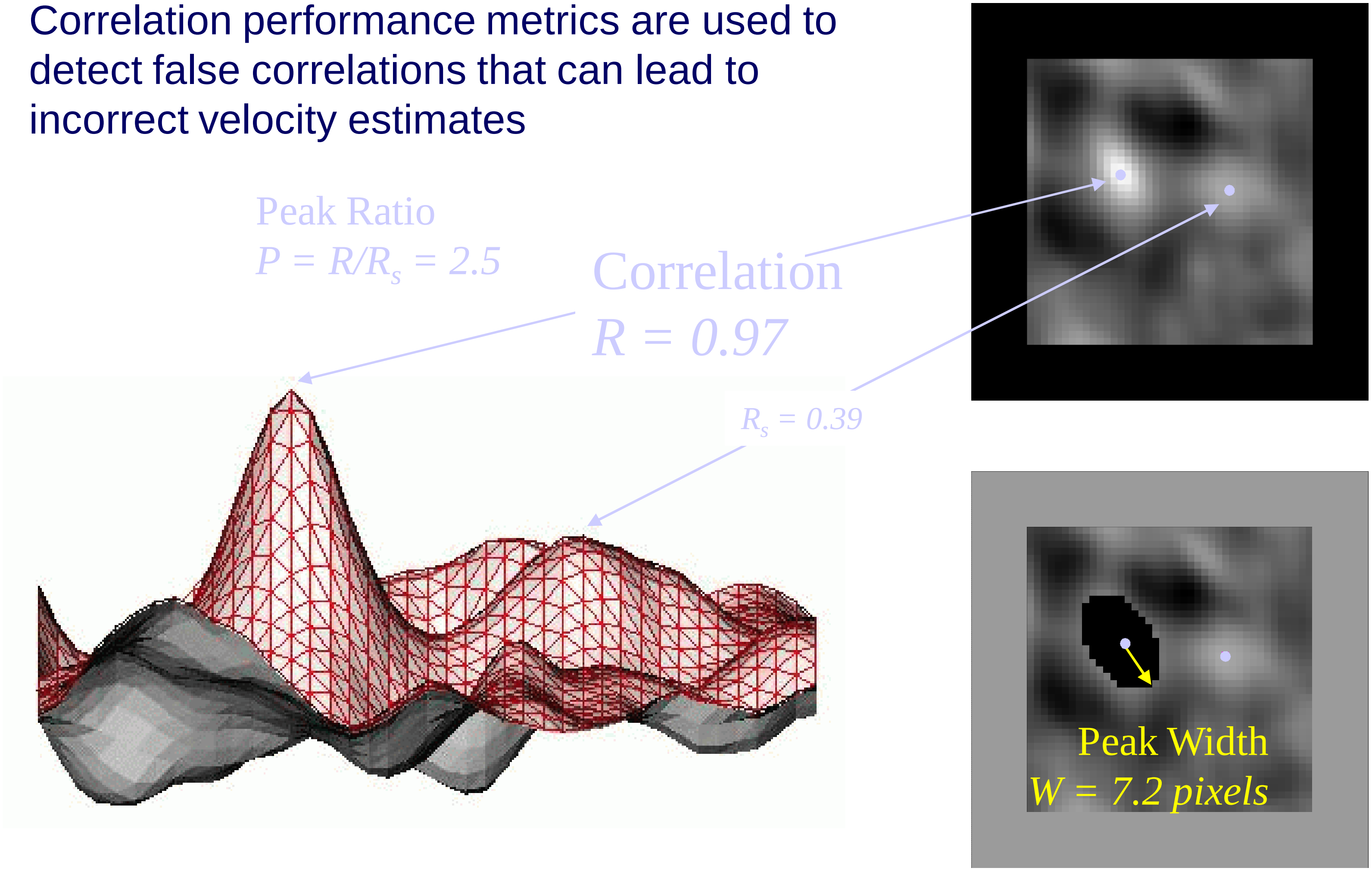






Correlation Performance Metrics

Correlation performance metrics are used to detect false correlations that can lead to incorrect velocity estimates



$$V_{err} = (0.6,0.8)\ m/s$$

IMU Check on Image Velocities

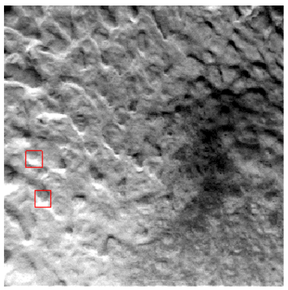


Image Pair 1 Feature Velocity - Image Pair 2 Feature Velocity = Image Delta Velocity

If equal then IMU confirms image velocities

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

IMU Pair 1 Biased Velocity - IMU Pair 2 Biased Velocity = IMU Delta Velocity

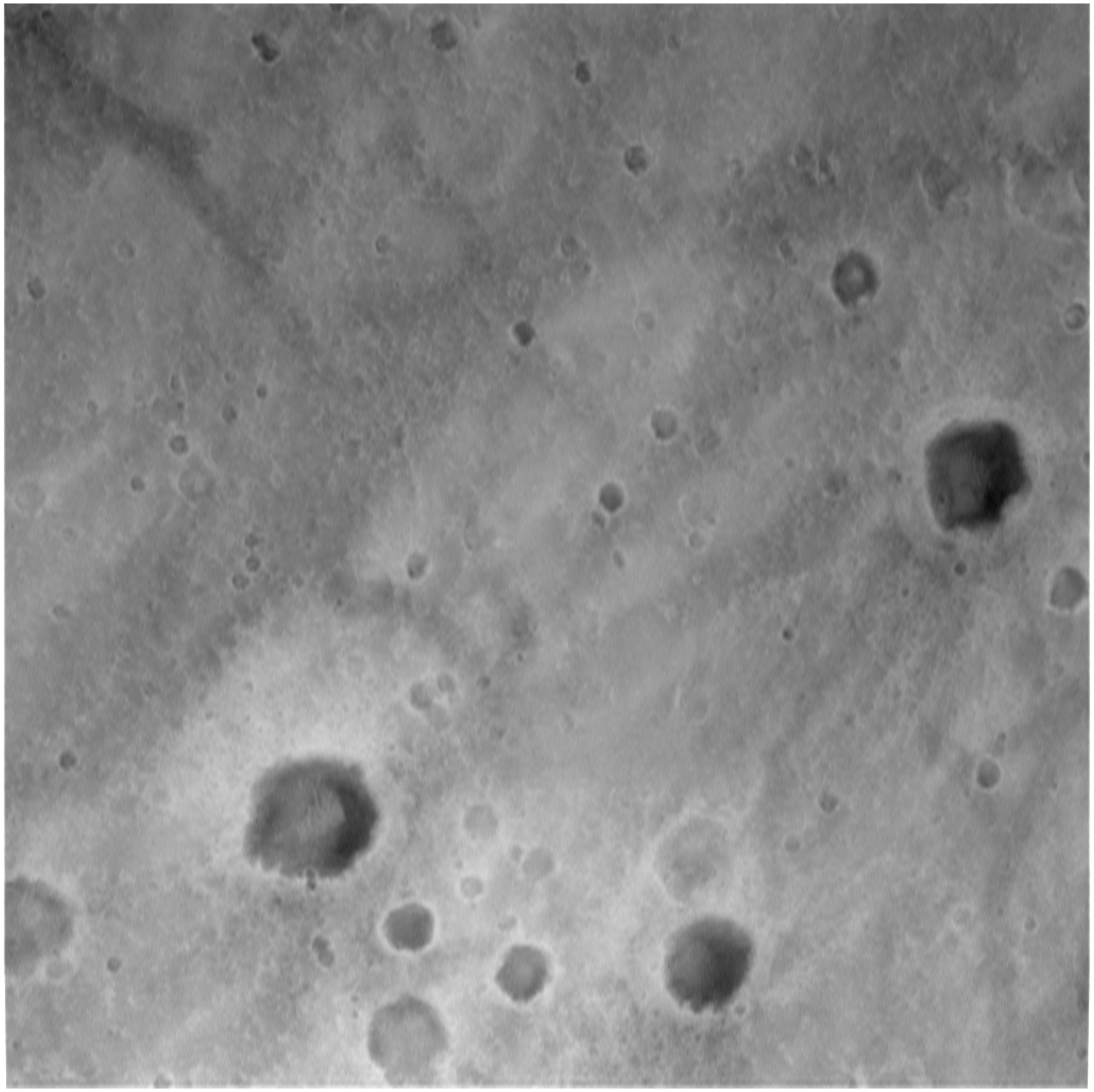
Spirit Performance

MER-A

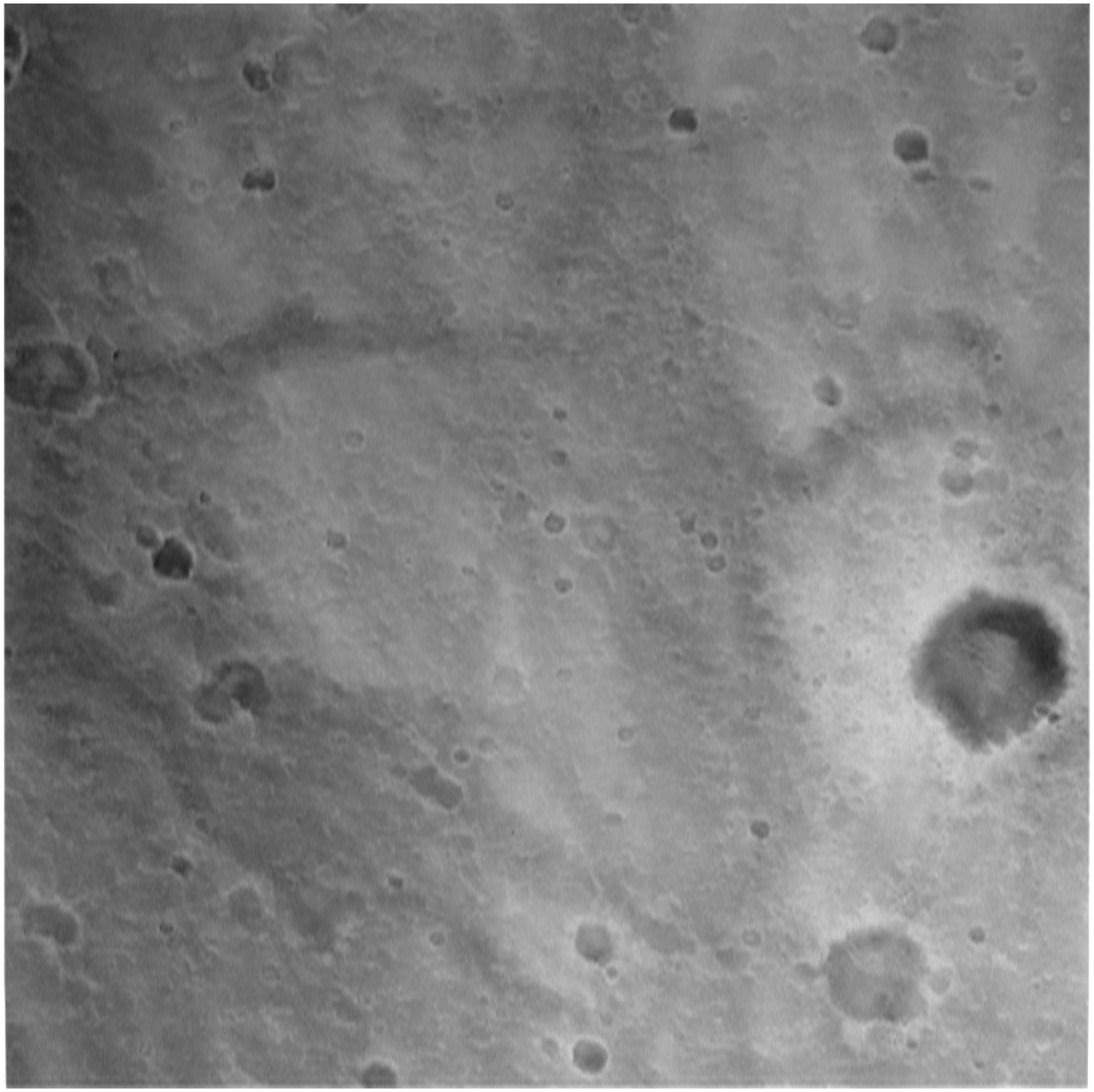
Gusev Crater

January 3rd, 2004

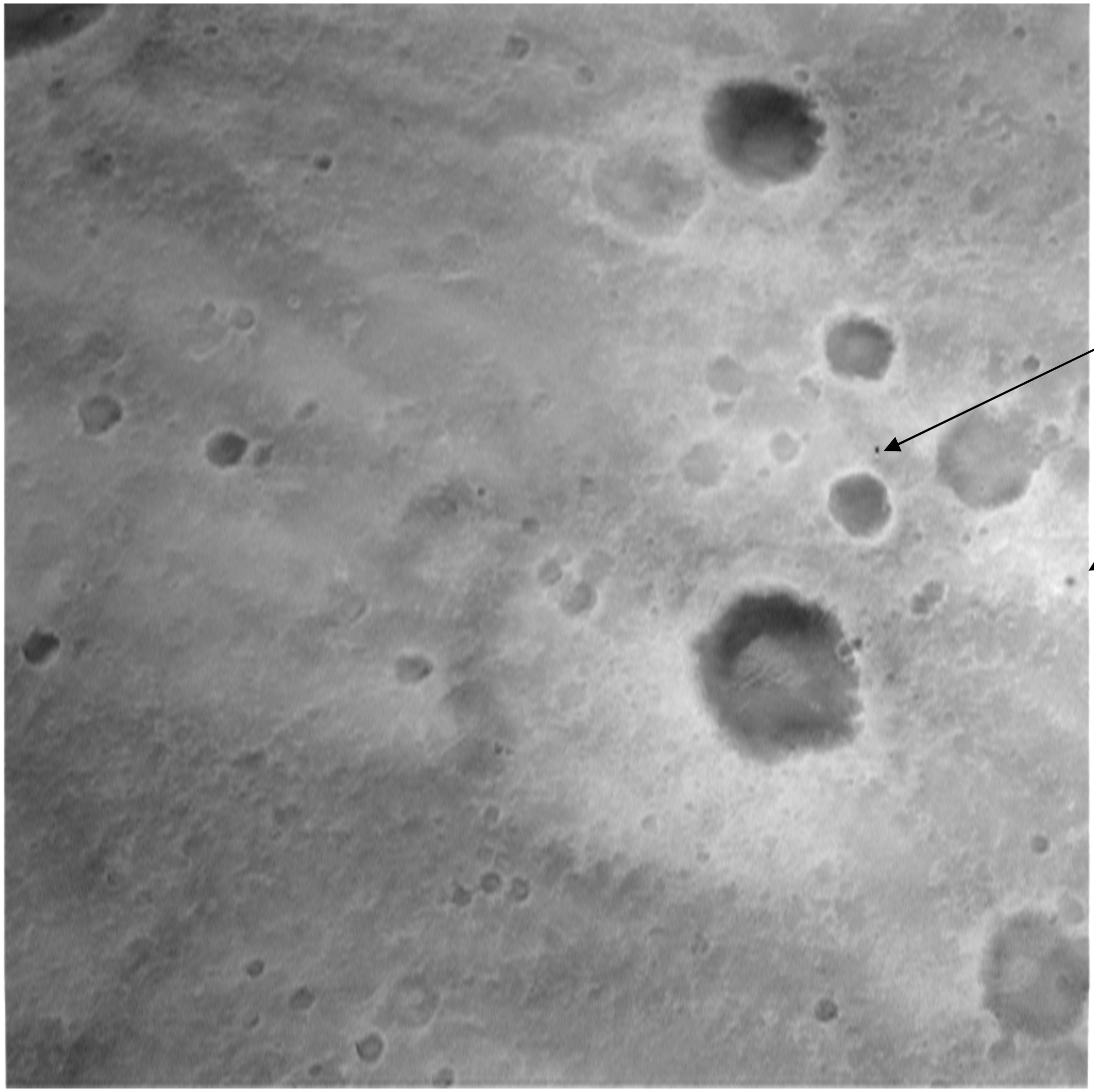
Spirit First Image (1983 m)



Spirit Second Image (1706 m)



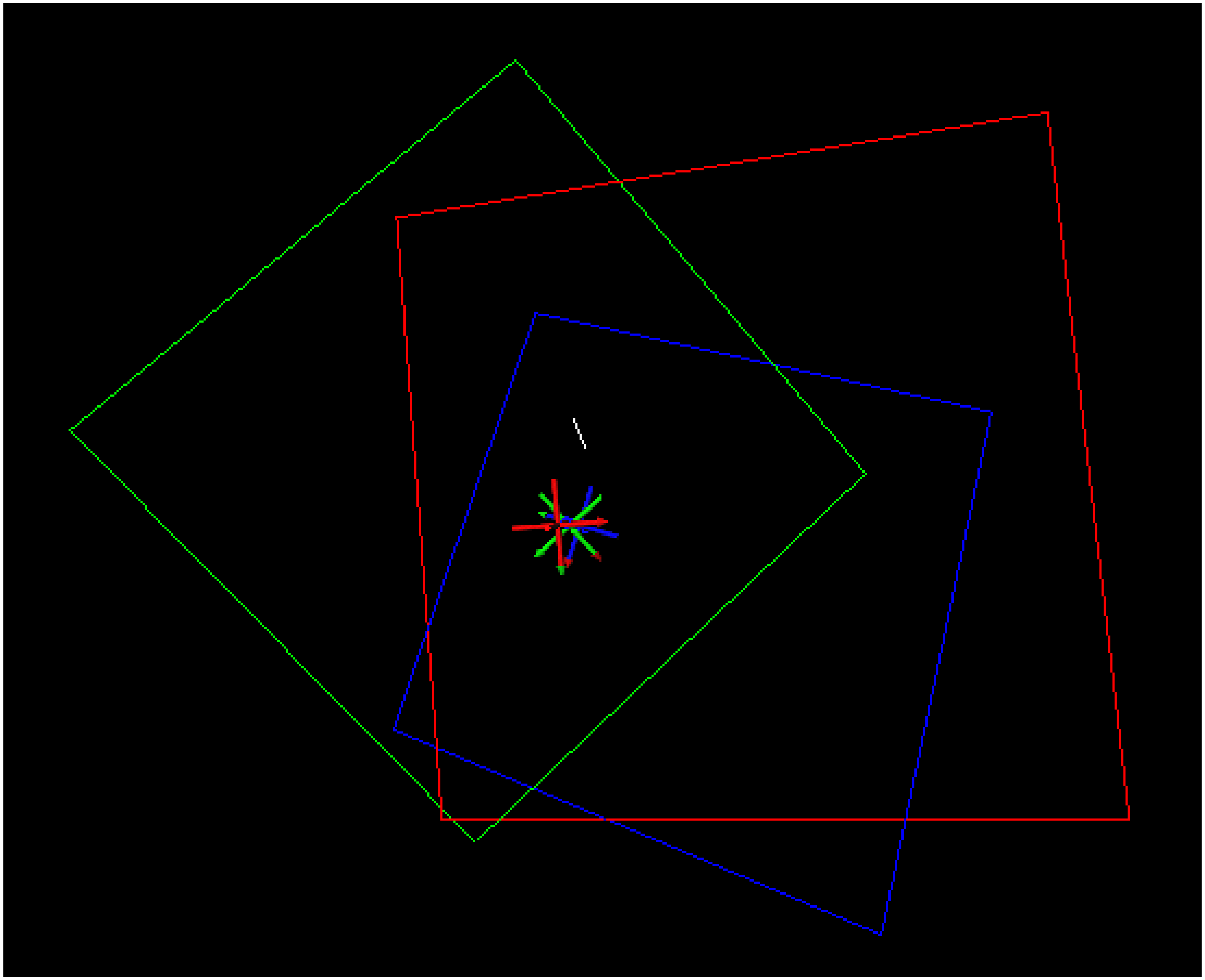
Spirit Third Image (1433 m)



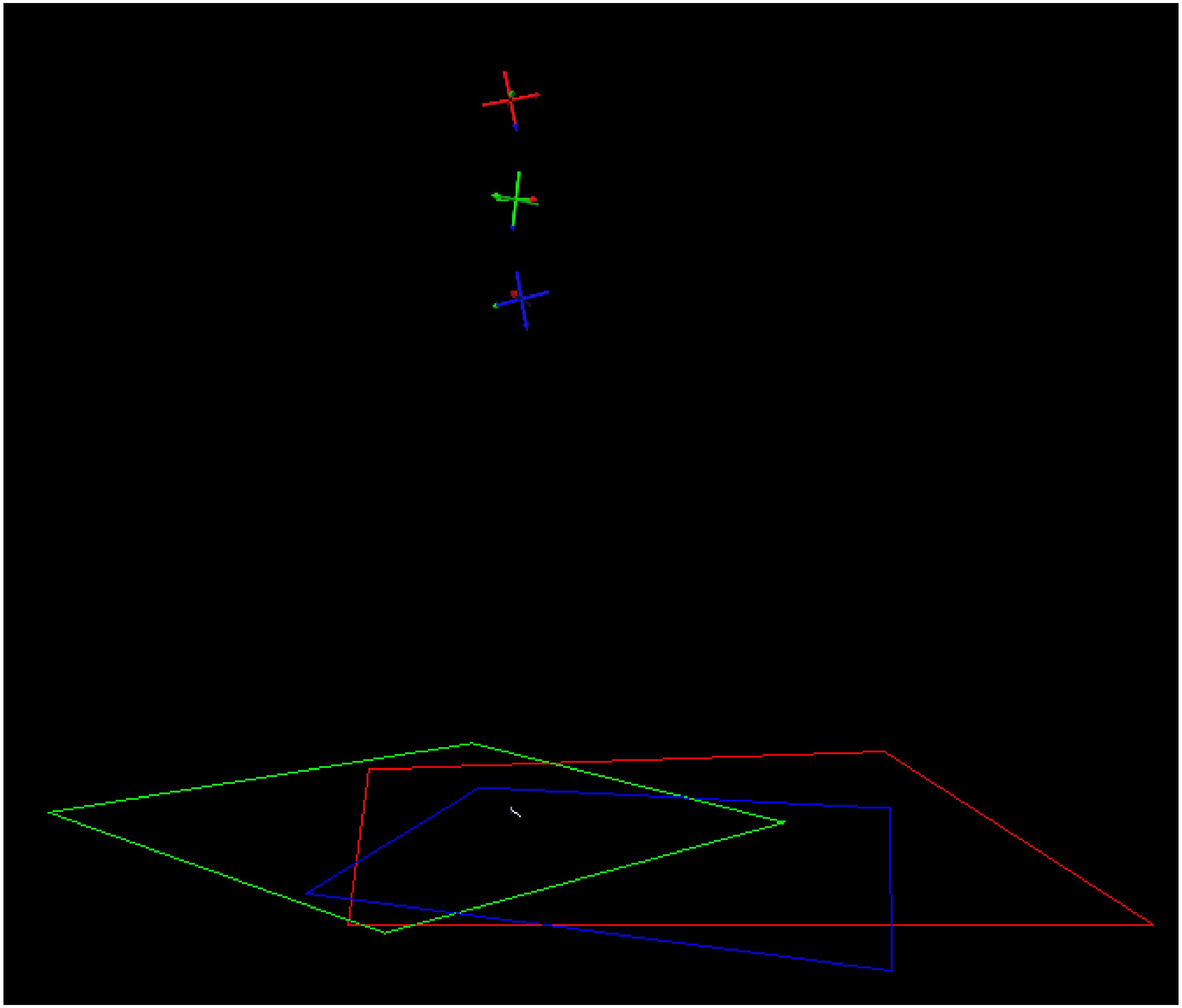
heatshield at
700 m altitude

parachute
shadow

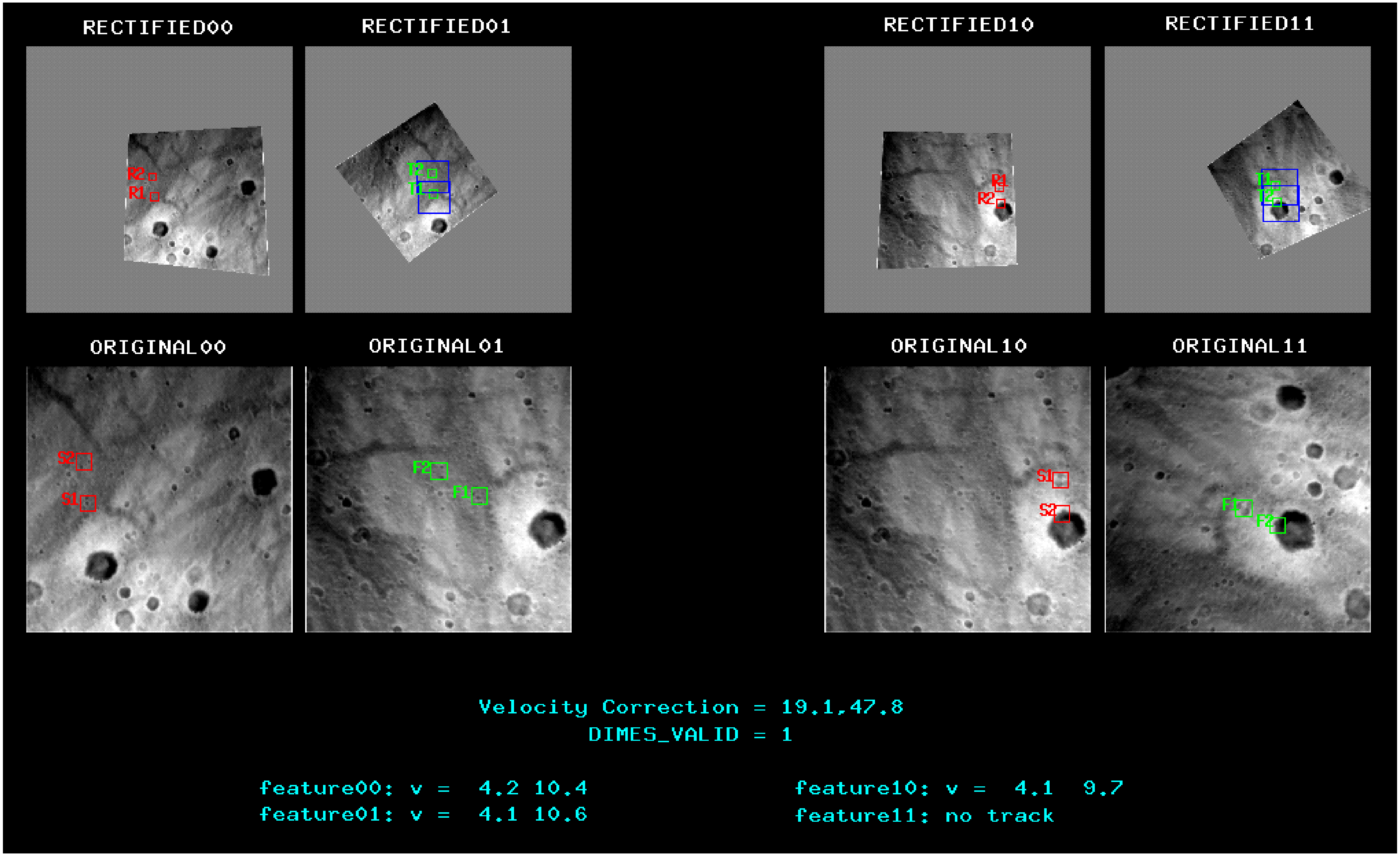
Spirit State Top View



Spirit State Side View



Spirit Velocity Result

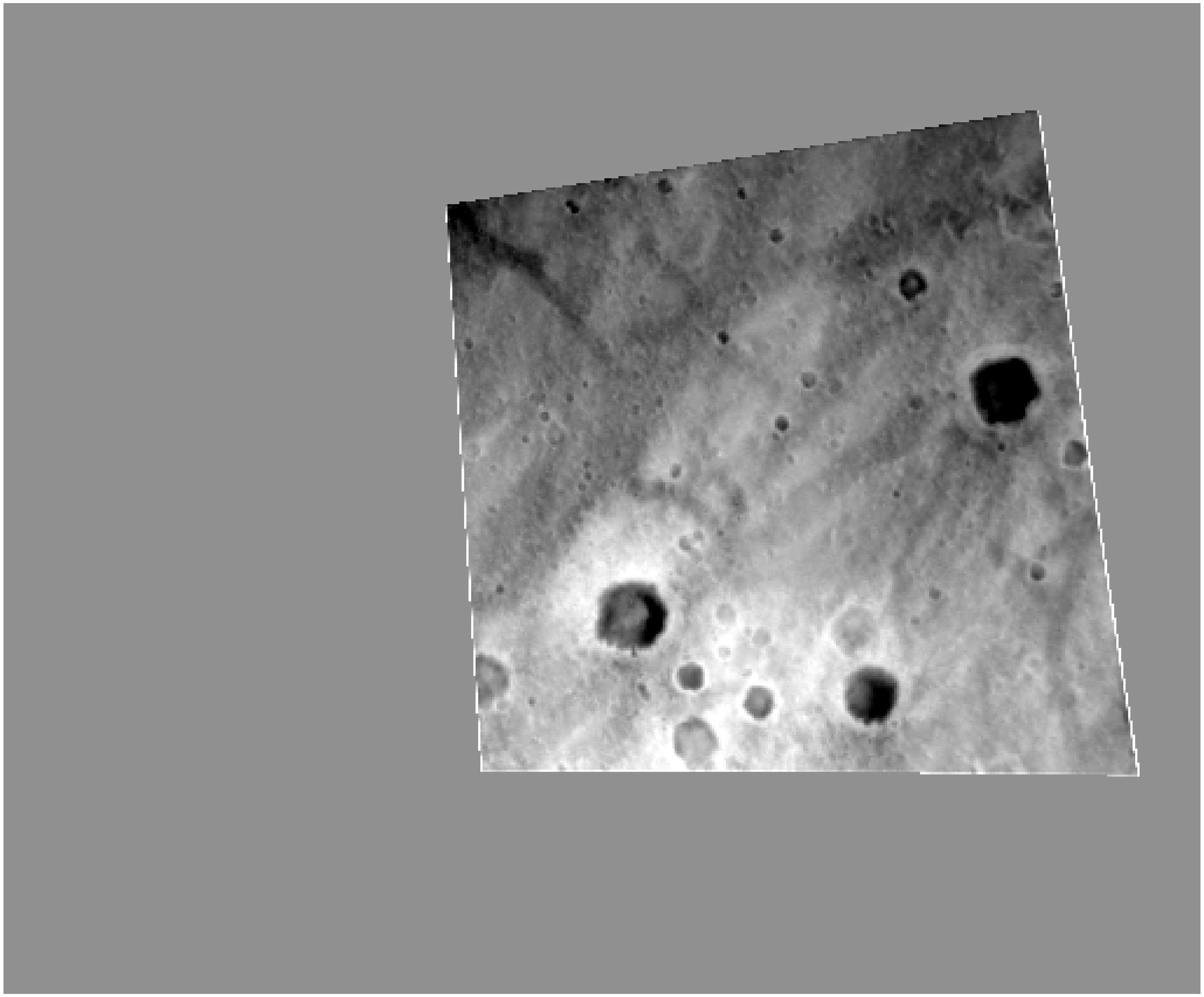


DIMES Verification

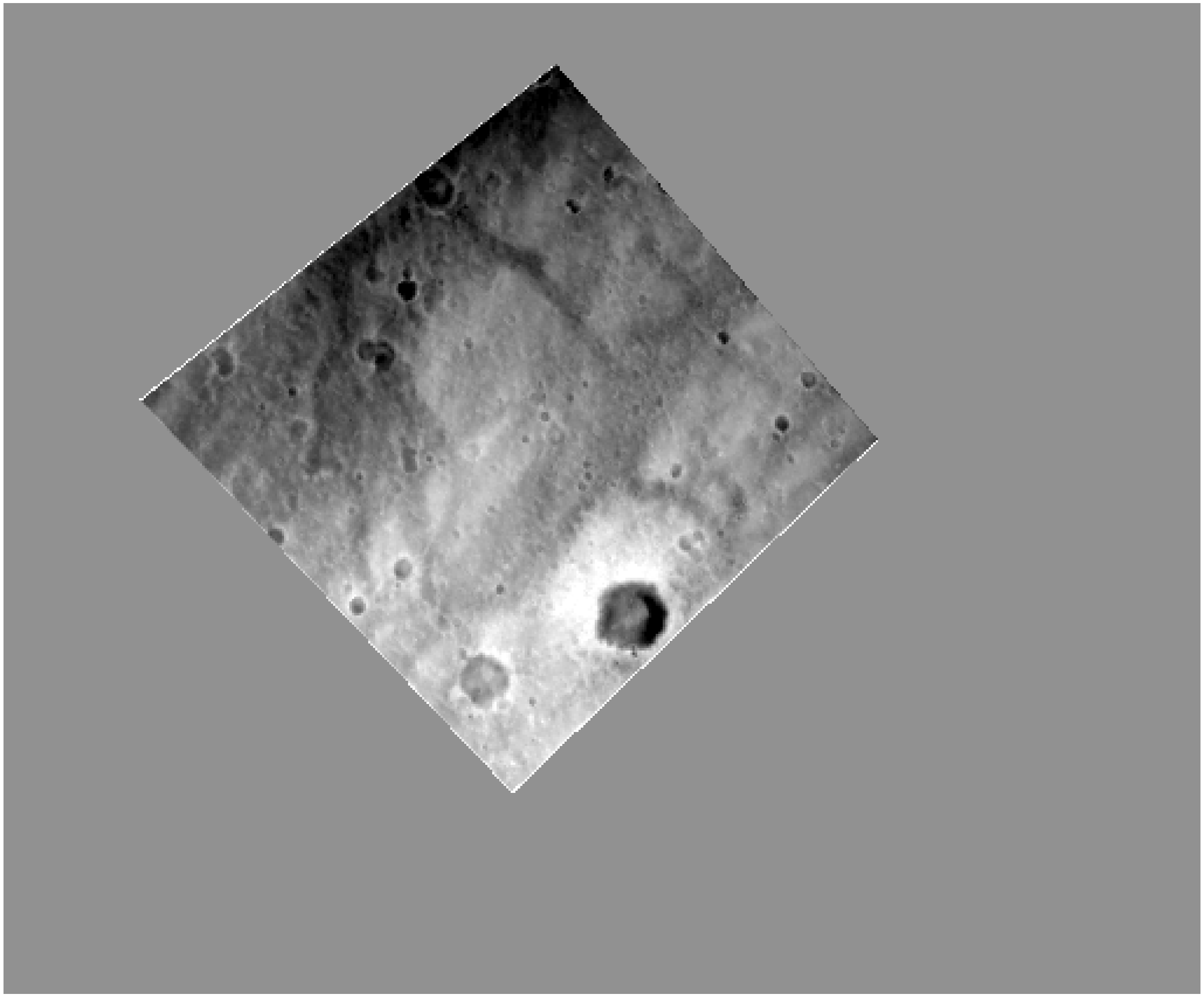
- The 3 images on the following pages are the DIMES images rectified to the local level frame. The position and attitude for rectification come from onboard measurements: attitude from IIT, altitude from RAS, horizontal motion from DIMES. For the images, North is left and East is down.

- As you flip through the pages, you will see that in the overlap there is very little shift in image data. Qualitatively, this indicates that all of the measurements are consistent and specifically that the horizontal velocity computed by DIMES was correct.

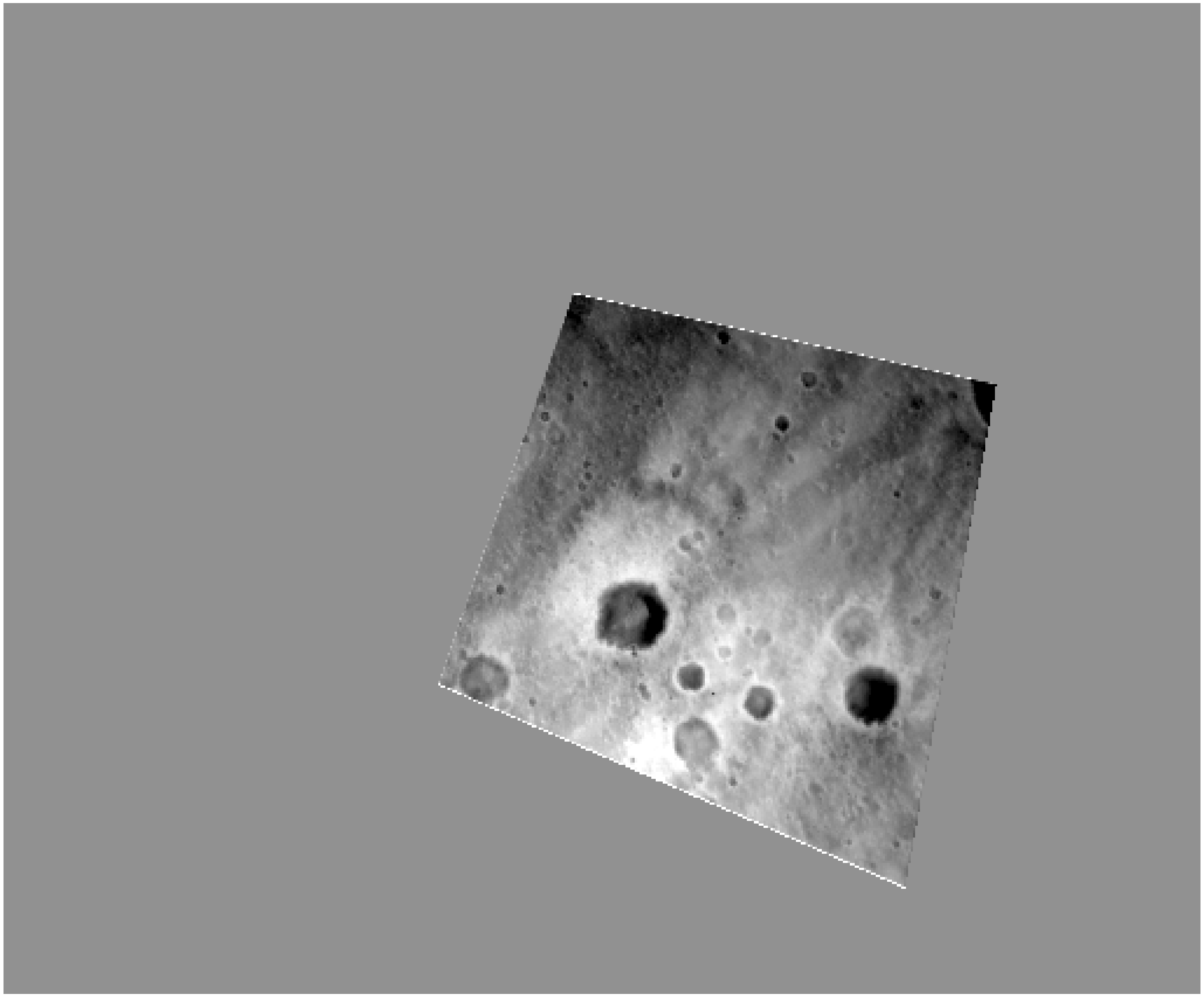
First Spirit Image
Mapped to Local Level



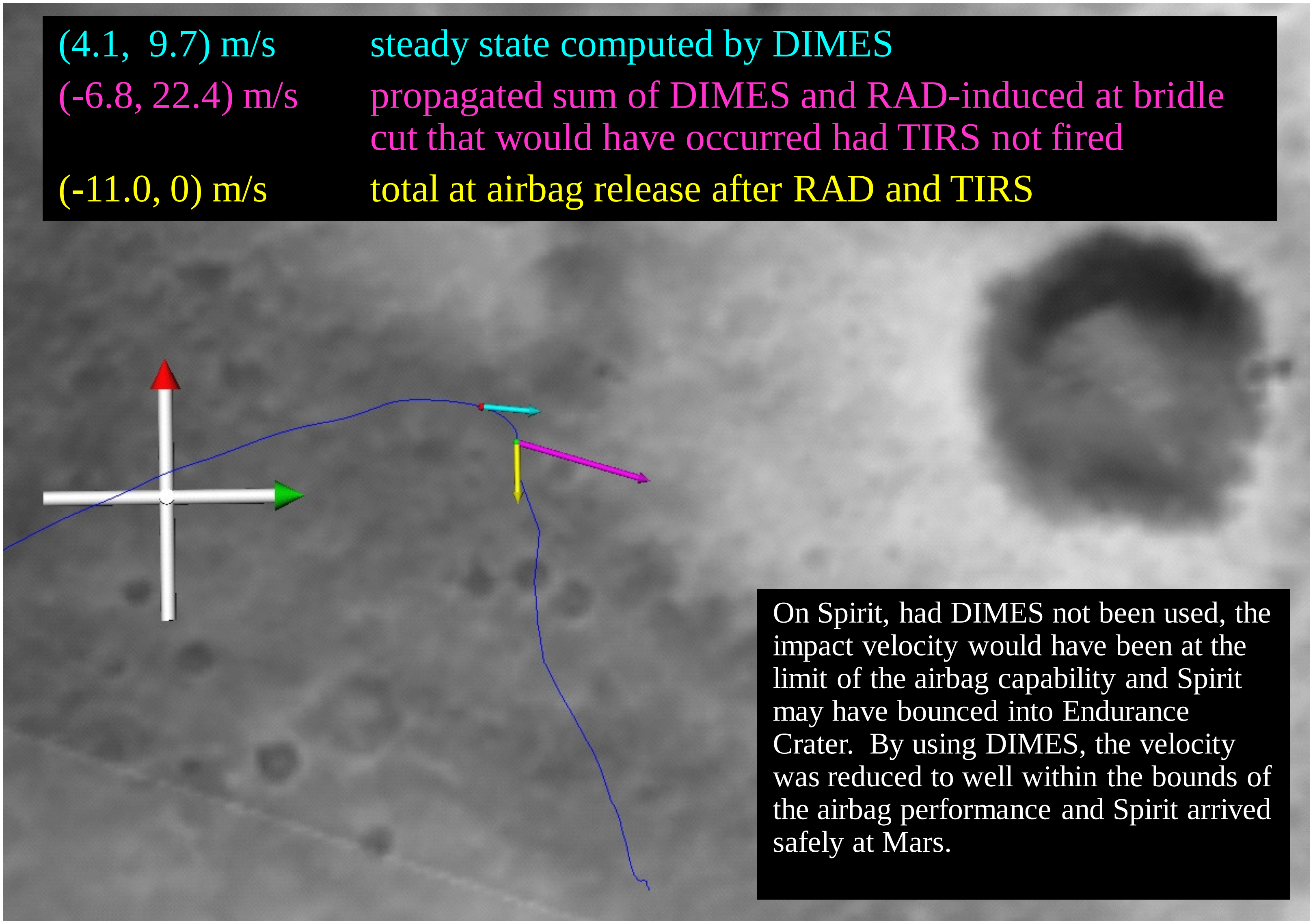
Second Spirit Image
Mapped to Local Level



Third Spirit Image
Mapped to Local Level



Spirit DIMES/TIRS Vector Diagram



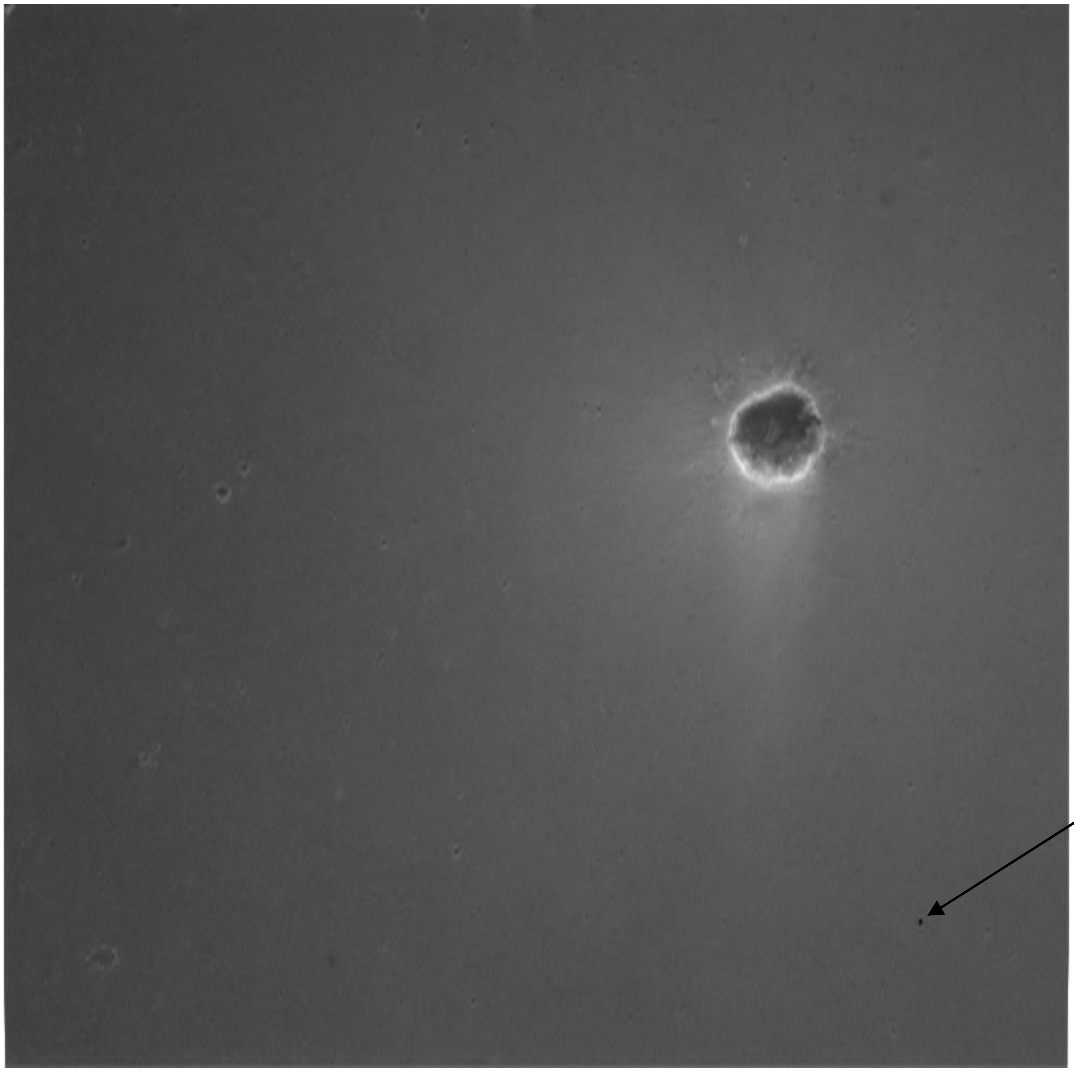
Opportunity Performance

MER-B

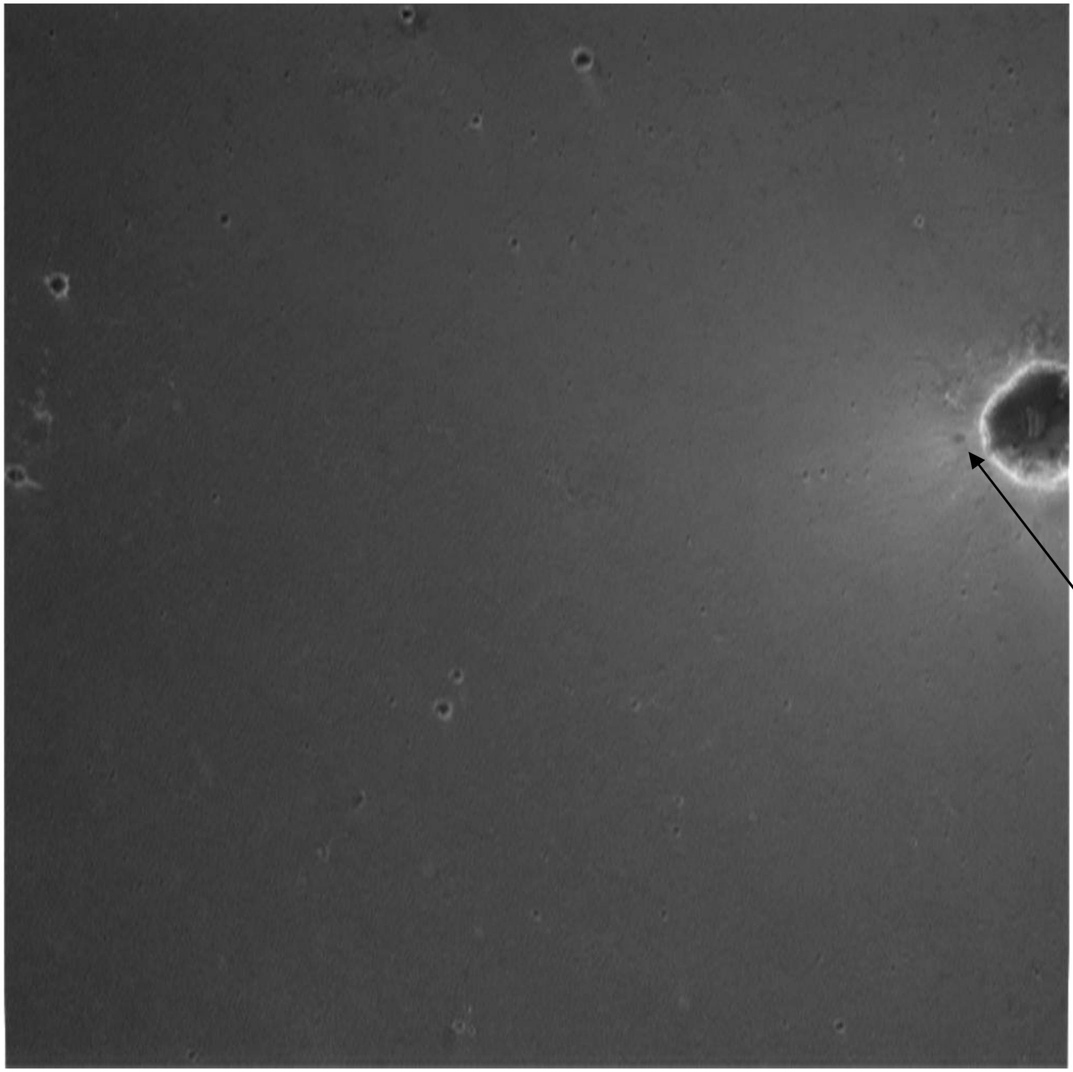
Meridiani Planum

January 23th, 2004

First Opportunity Image (1986 m)

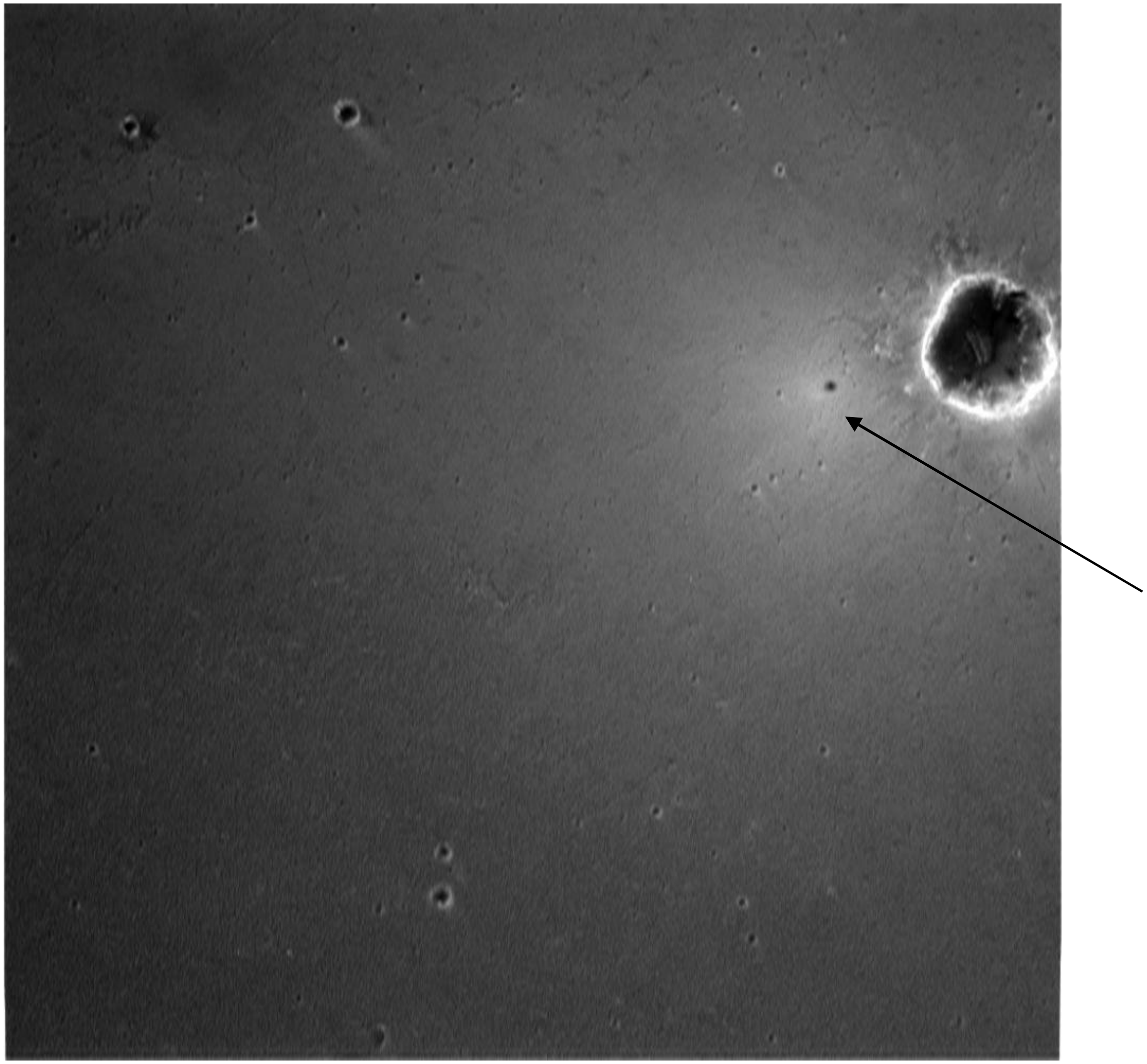


Second Opportunity Image (1690 m)

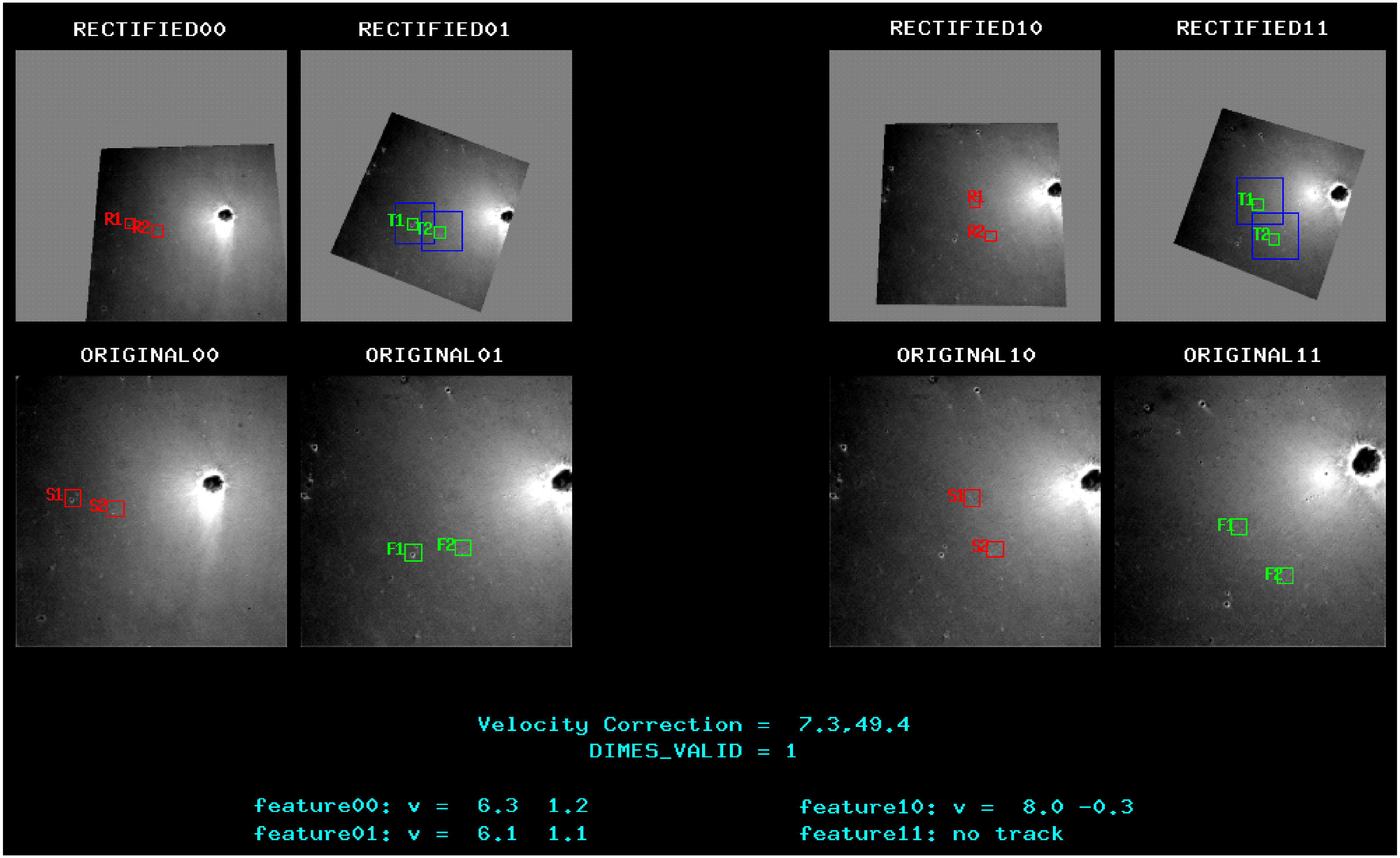


parachute shadow and opposition effect

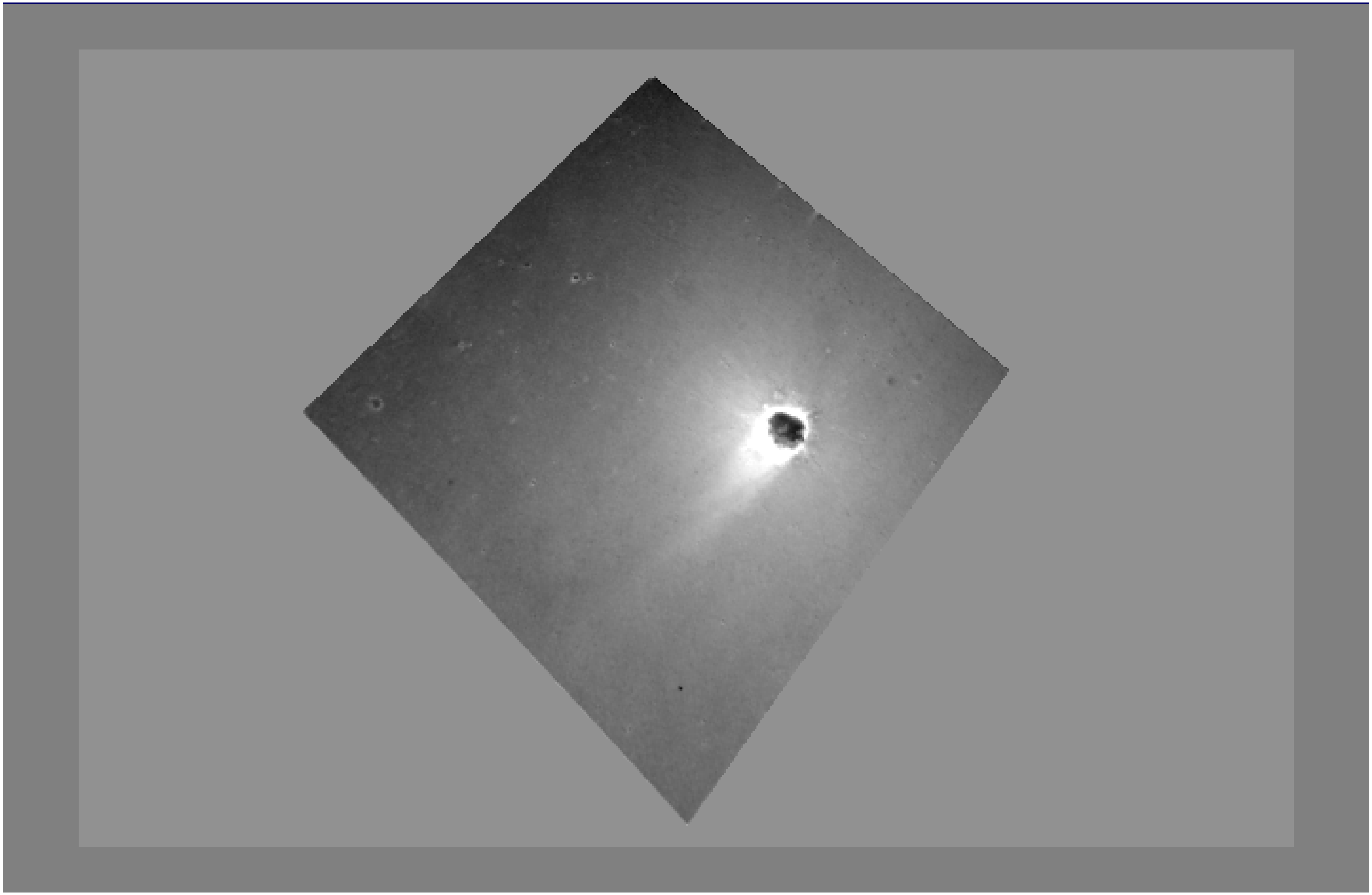
Third Opportunity Image (1404 m)



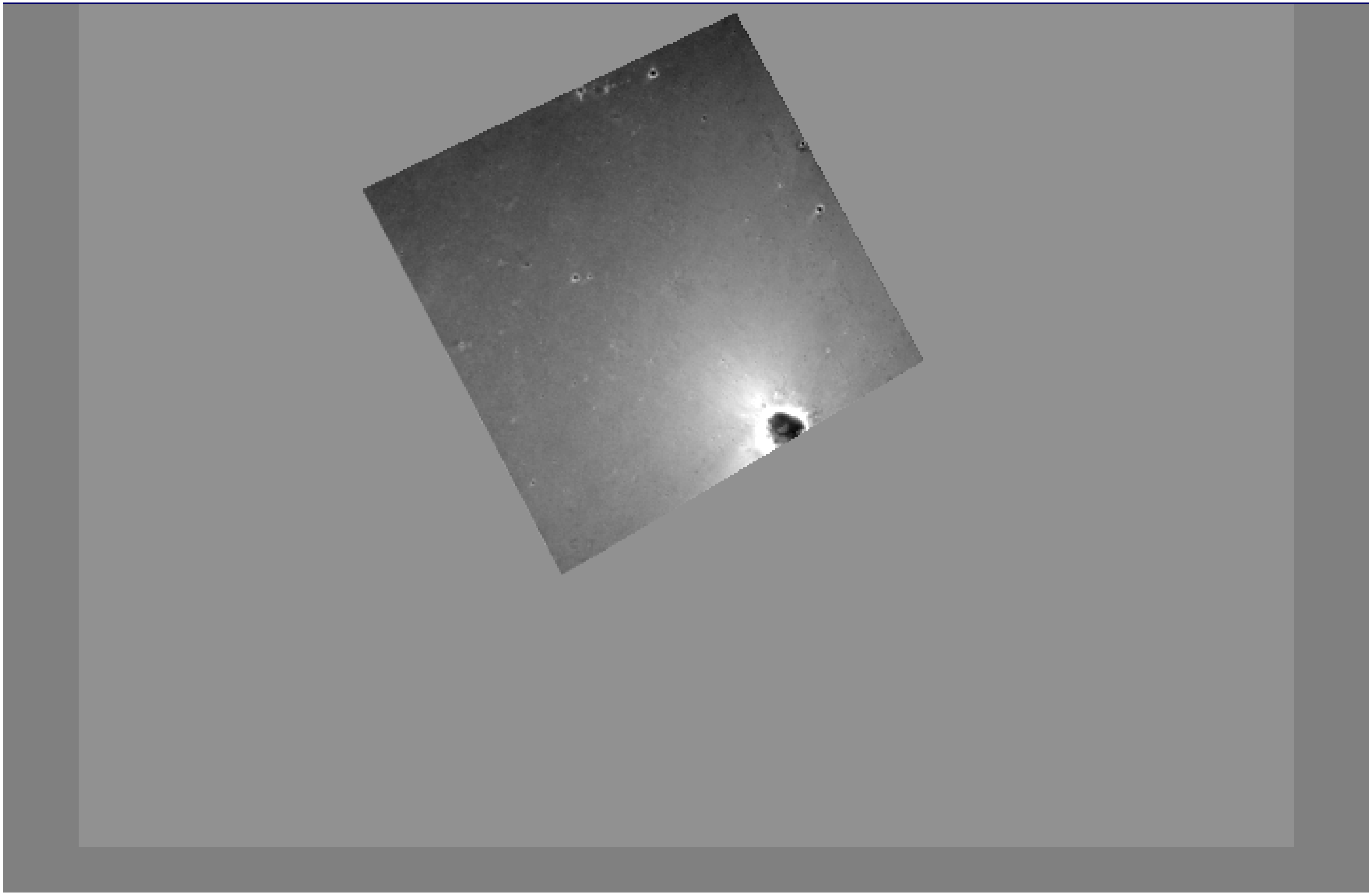
Opportunity Velocity Result



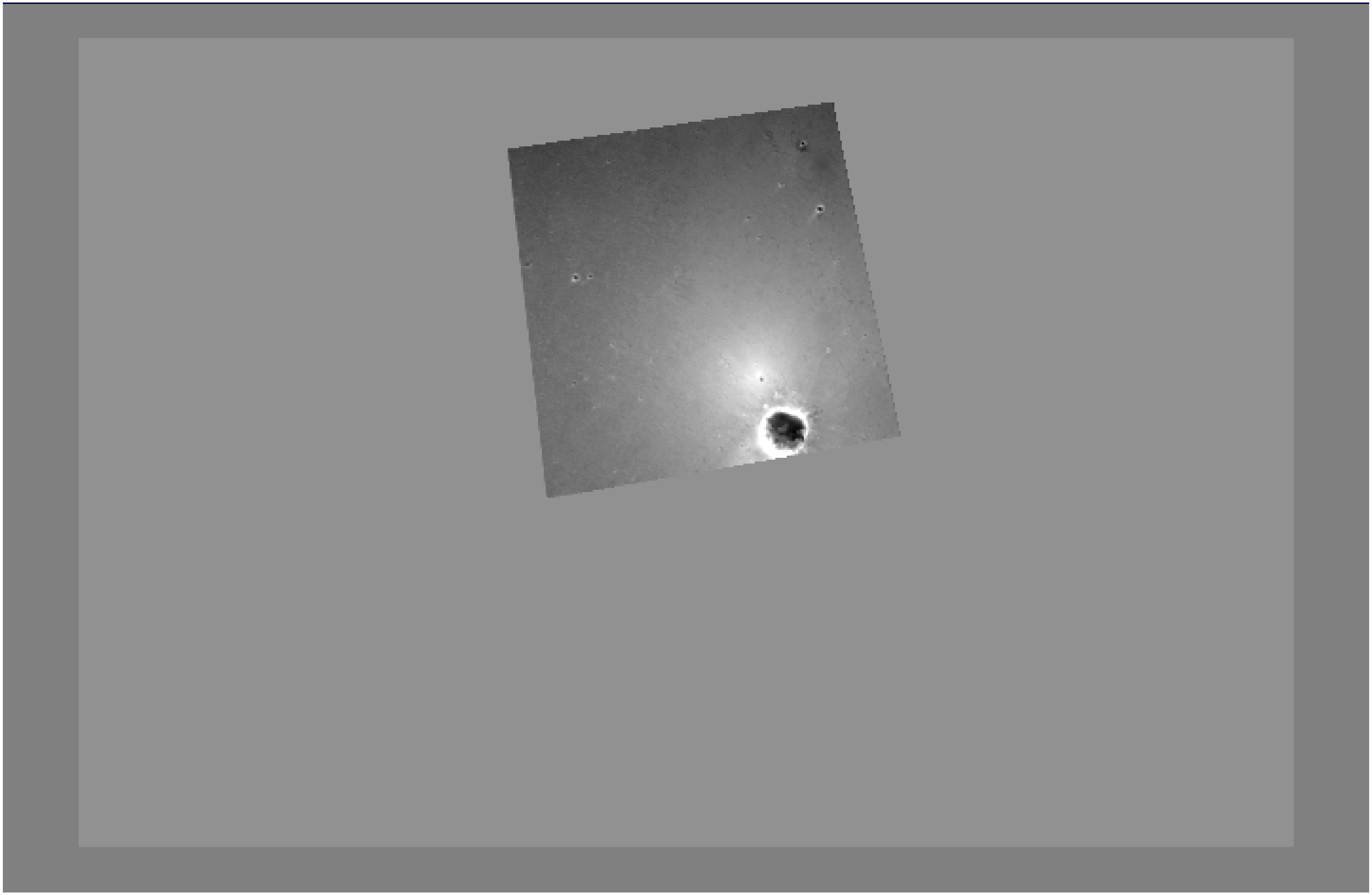
First Opportunity Image Mapped to Local Level



Second Opportunity Image Mapped to Local Level

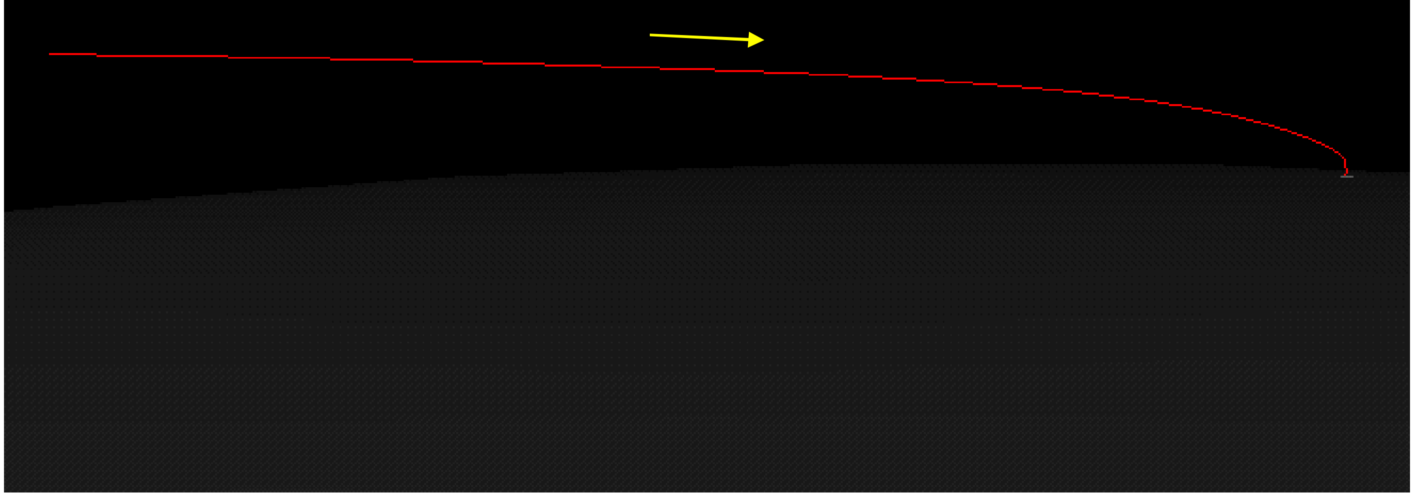


Third Opportunity Image Mapped to Local Level

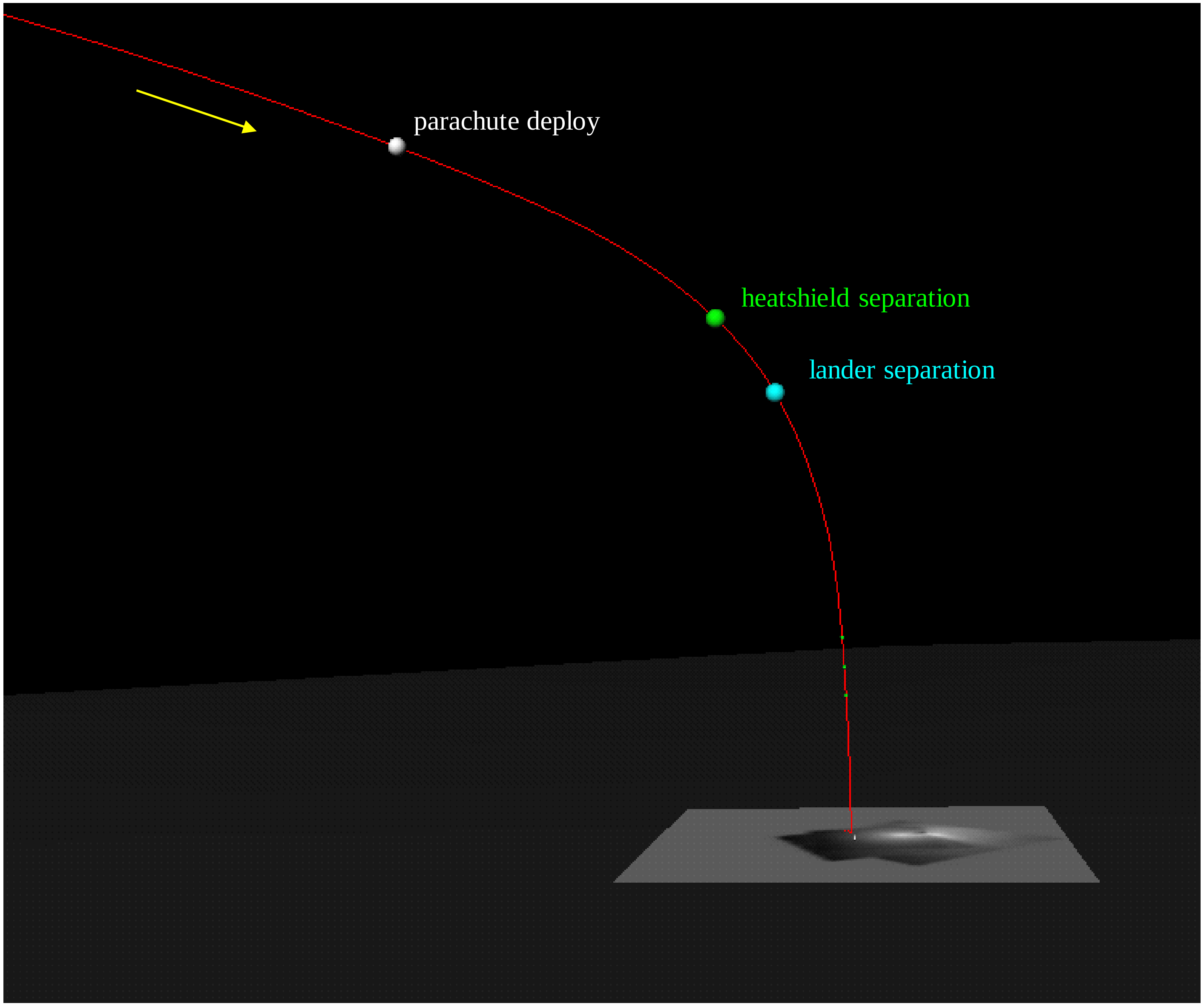


View of Entry Trajectory

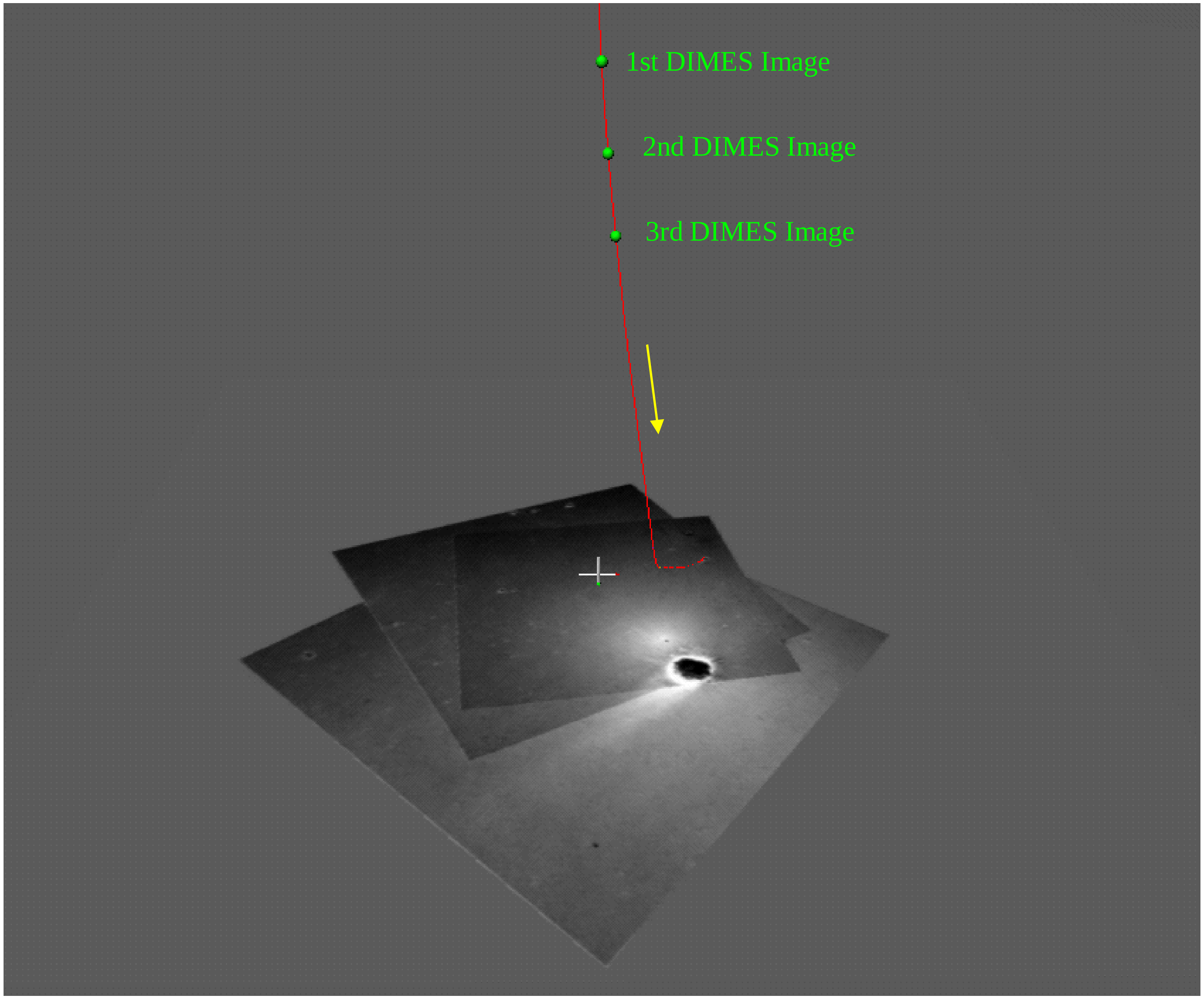
- The next 6 slides show different views of the EDL trajectory superimposed on a mosaic of the DIMES images.
  - This visualization is one of several pieces of information used to determine the location of Opportunity on the surface.
  - The EDL trajectory is a straight integration of acceleration using the DIMES velocity as an initial condition.
  - The DIMES images are mosaiced using IIT attitude, RAS altitude and DIMES horizontal motion estimates.



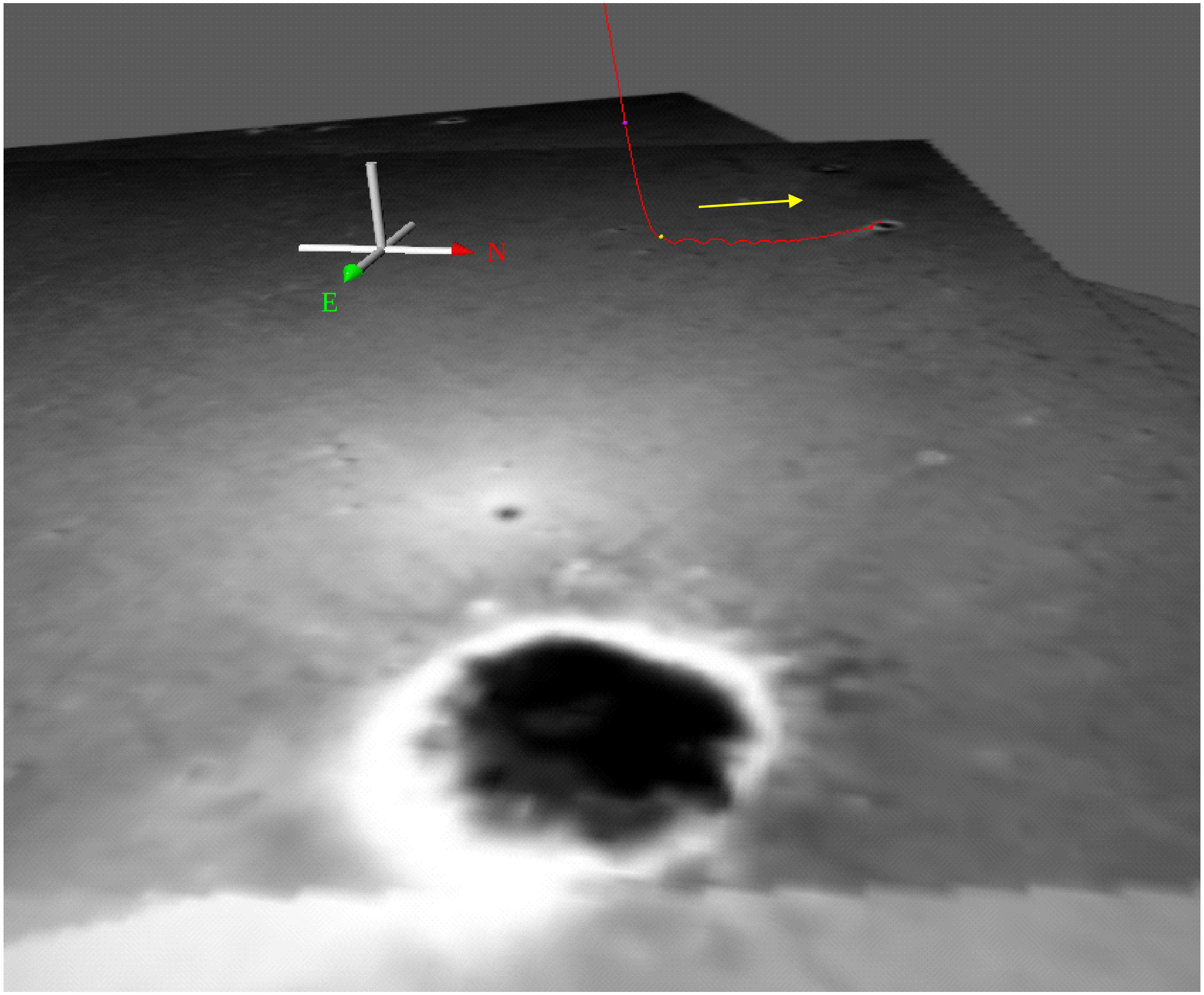
A Closer View of Entry



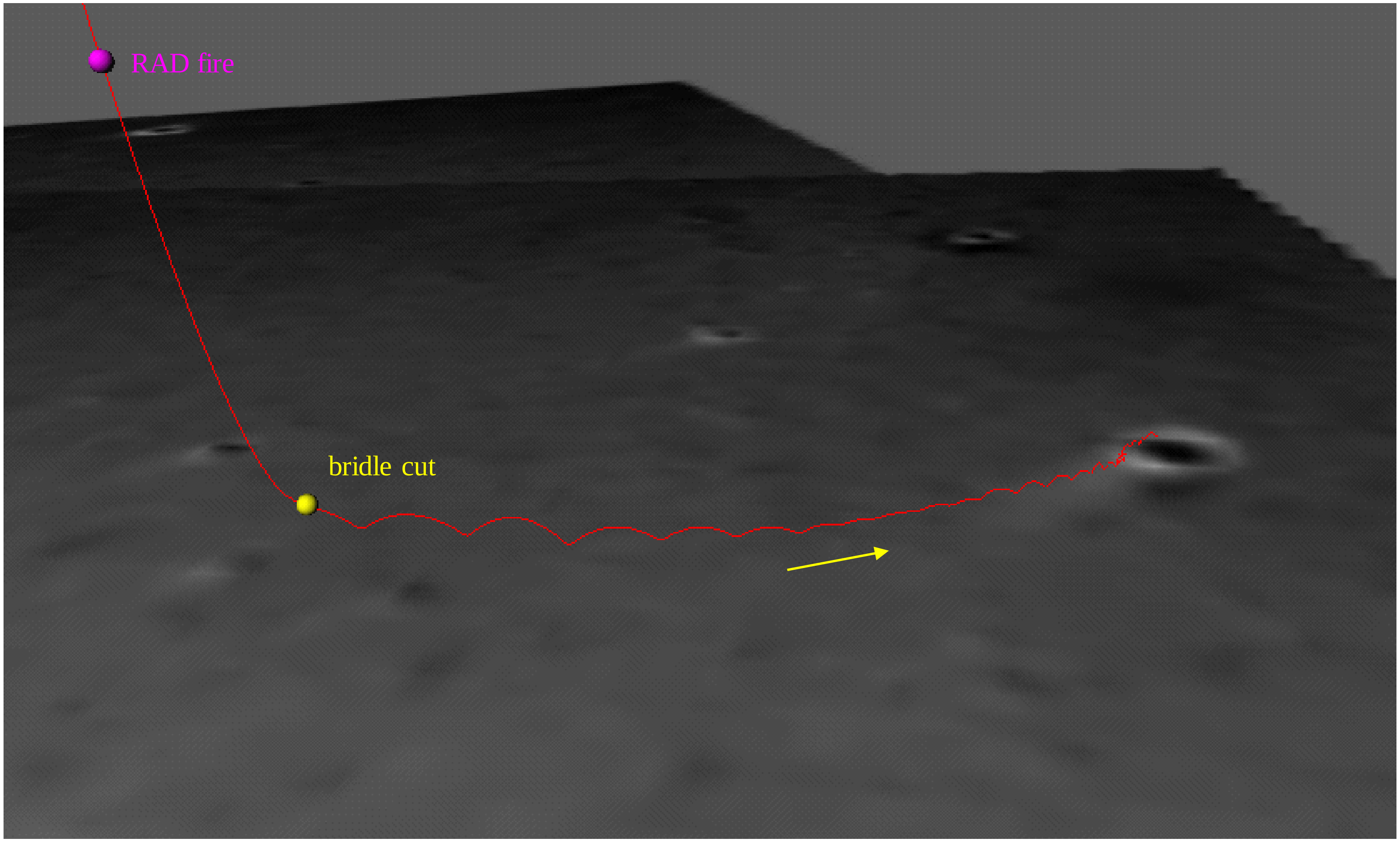
Descent and DIMES Images



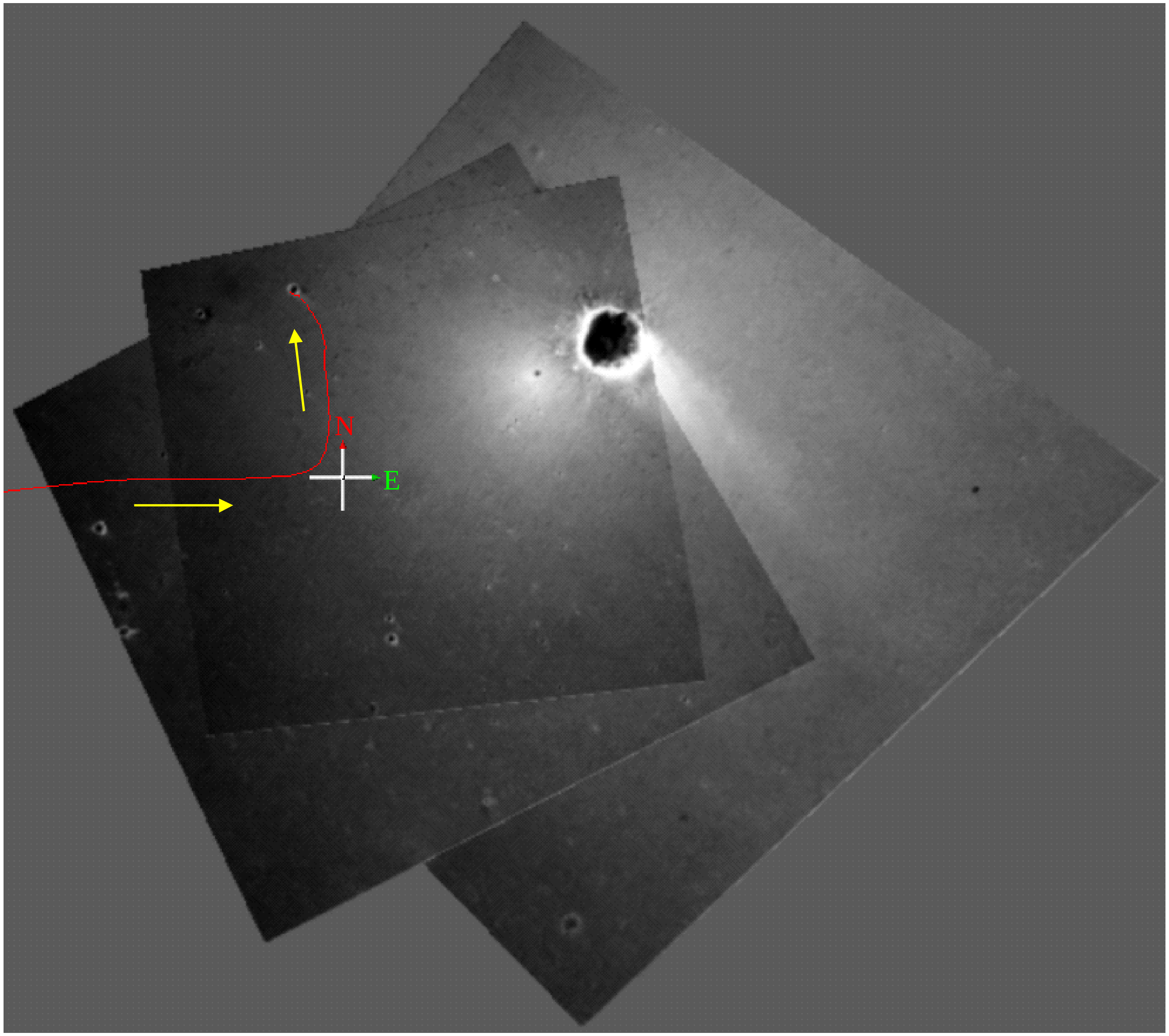
Direction of Bouncing



Side View of Bouncing



Overhead View of Trajectory



Home Work Assignment

- You will design and implement a linear RANSAC outlier detection algorithm to detect and remove any outliers of a feature list generated from feature detection/tracking lecture.
  - The input of your algorithm
  - Feature list
  - Feature tracking error tolerance
  - Outliner probability
  - Probability of success
- Your code will do
  - Determine the number of trials to obtain the input probability of success
  - Execute a RANSAC
  - Compute a reprojection error of each valid feature and statistics (mean and standard deviation)
  - Plot the valid feature vectors and invalid feature vectors in different color in single plot
- Your code will output
  - A new list of filtered feature list
  - A transformation parameter file
  - Final valid feature statistics
- What to turn in
  - Describe your algorithm detail enough so other can follow it to rewrite it
  - Describe why you made the major design choices that you did
  - Describe the method to evaluate the performance of your algorithm
  - Present the performance of your algorithm
  - Describe the strengths and weaknesses
  - Describe extra credit item you did

Extra Credit

- Here is a list of suggestions for extending the program for extra credit
  – Implement a nonlinear homography RANSAC and make a comparison between the linear homography RANSAC vs nonlinear homography RANSAC
  – Implement a homography RANSAC for planer surface detections
  – You are encouraged to come up with your own extensions as well!