Small Satellite Research Laboratory

Franklin College of Arts and Sciences

UNIVERSITY OF GEORGIA

Selected Software Demonstrations from the Multi-view Onboard Computational Imager (MOCI) Satellite

Caleb Adams, Jackson Parker

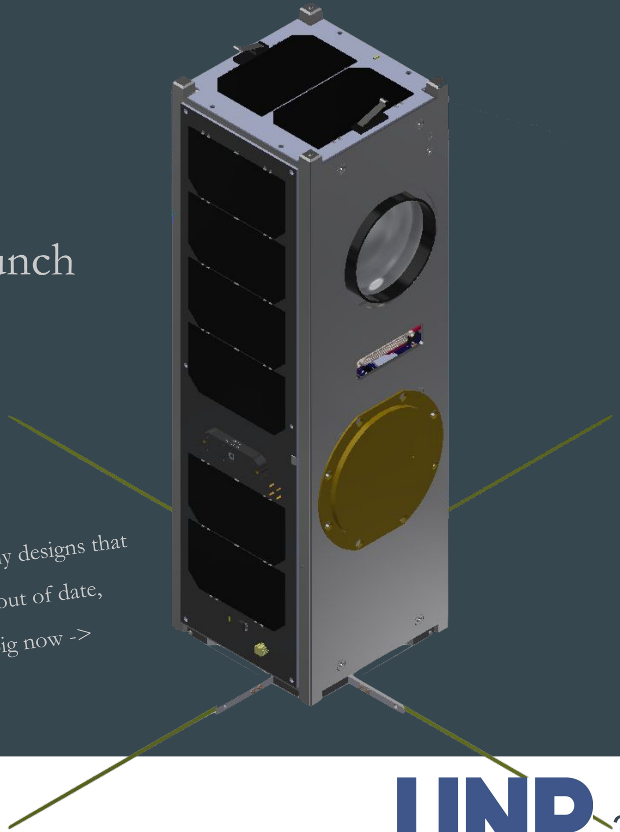
Presentation Expectations

- "Selected" demonstrations
- Overview of Concepts
- Hot swapping between machines to show the demos

MOCI: Multi-view Onboard Computational Imager

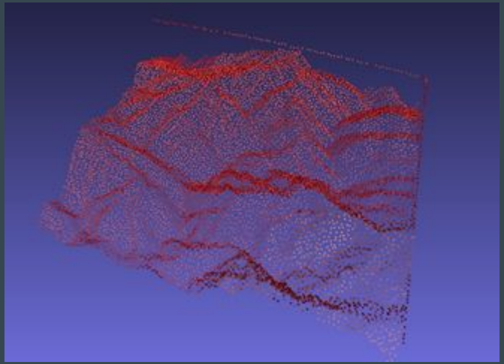
- Part of the 9th University Nanosatellite Program
- 9 winners since 1998 (MIT, GT, UC Boulder, … and us!)
- Competed against 9 schools for a chance to fully build and launch
  - UGA WON!
  - UGA was the first team without an aerospace department
  - First team to win on first try
- Launch in late 2020

We move quickly so many designs that make it to diagrams are out of date, we're actually twice as big now ->

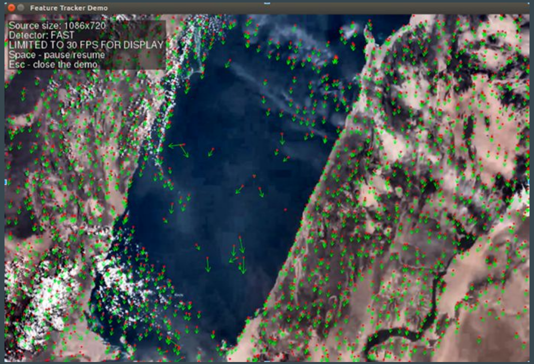


MOCI: Multi-view Onboard Computational Imager

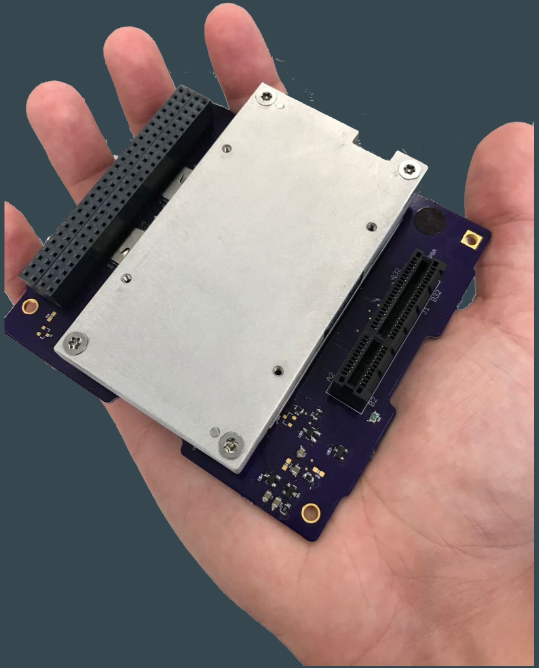
- Will produce 3D models of the Earth's surface from space
- Use computer vision to track objects on the surface
- Use neural networks to identify surface targets



Simulated computations of mountain ranges



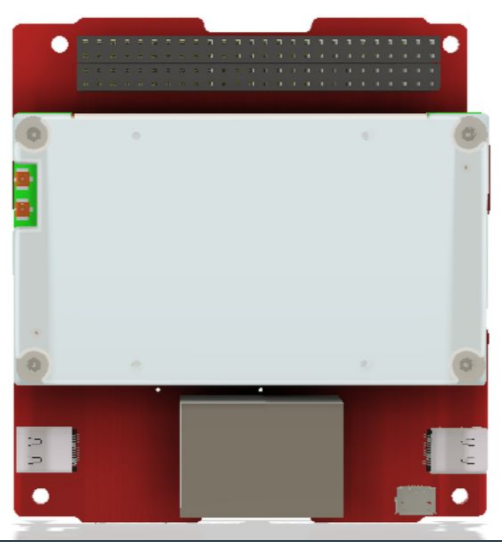
Simulation of the tracking of objects over the red sea from an ISS nadir video.

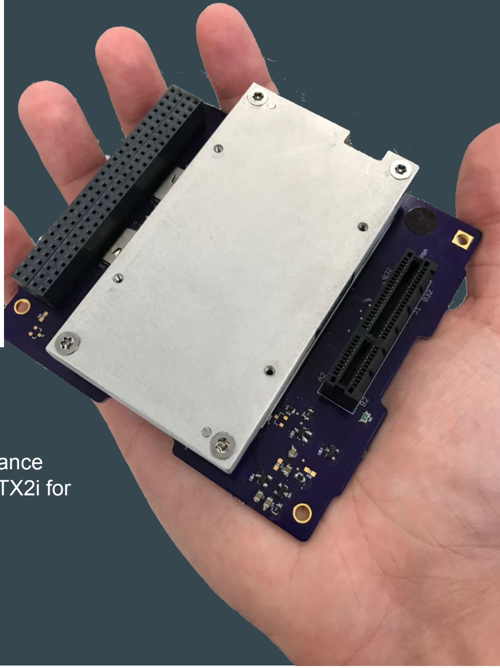


The UGA SSRL's miniaturized high performance computation unit developed around the Nvidia TX2i for the MOCI satellite

MOCI Hardware

- Required to be extremely computationally capable
- No existing space technology can satisfy our requirements. So we're building high-performance space systems.
- We're transitioning terrestrial high-performance embedded systems into space
  - From autonomous cars
  - From AI-optimized processors

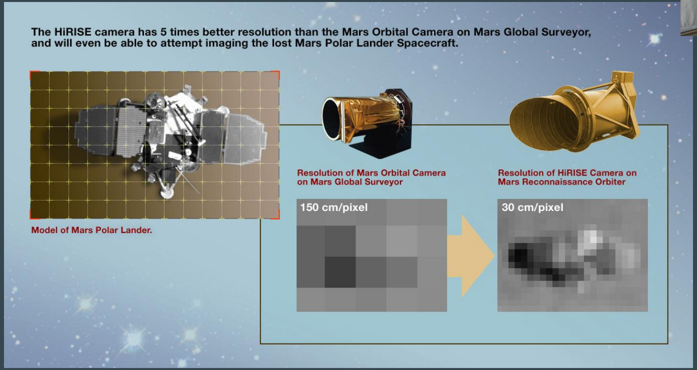




The UGA SSRL's miniaturized high performance computation unit developed around the Nvidia TX2i for the MOCI satellite

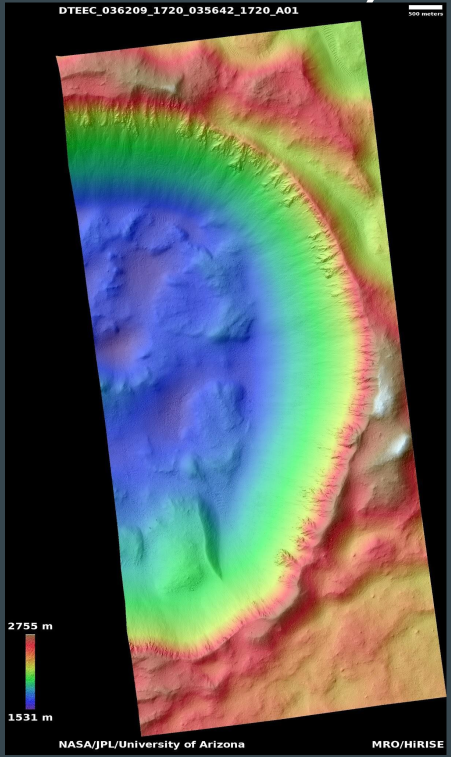
MOCI: The Fundamentals (How Others Do It)

- HiRISE on the MRO
- 3 foot diameter primary mirror
- Made at UC Boulder
- 30cm GSD
- MOCI has ~6m GSD



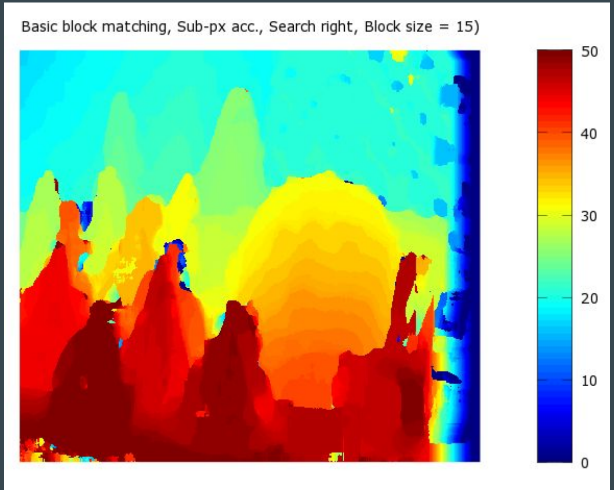
MOCI: The Fundamentals (How Others Do It)

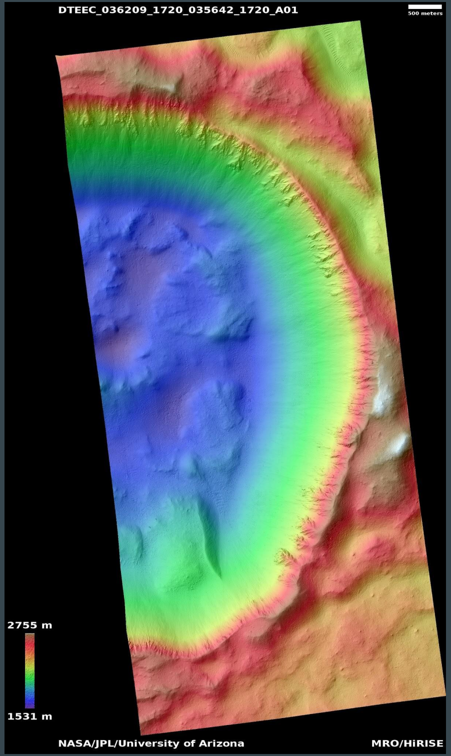
- MRO, at Mars, is the most similar large satellite to MOCI
- Multi-view geometry (parallax) used to deduce topography
- Doesn't seem so hard right?



Multiview Reconstruction

- Changing camera parameters (extrinsic and intrinsic)
  - Thermal, inertia, optical …
- Uncertain pose
- Stereo disparity results in poor fidelity models
- Sub-pixel resolution





Optical Design

- Made for orbital multiview reconstruction, need to be custom made each time! (no COTS)
- Bounding our unknowns
- HiRISE
- Our optical design also prioritizes high resolution
- Specs: 90/10 beam splitter, ~4K BW CMOS, ~1K RGB CMOS, 280 mm effective focal length

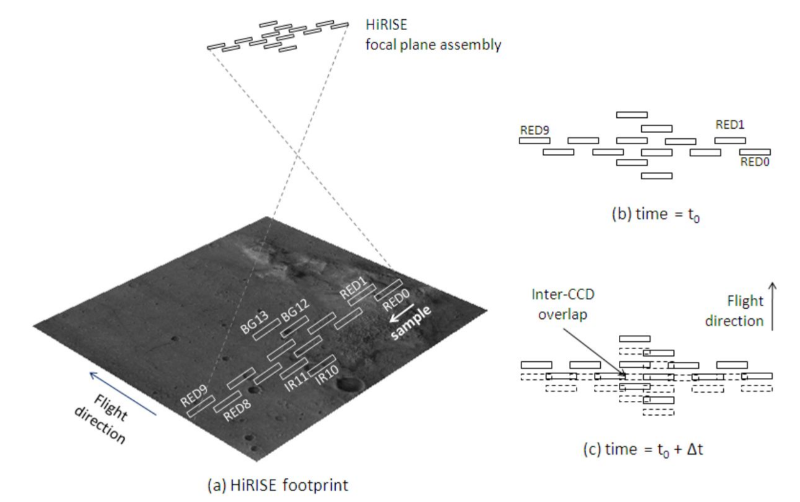


Figure 1. Illustration of jitter observation.


Current MOCI optical layout

Camera Matrices

- Assumption of pinhole
- Simple projection model

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

2D Image Coordinates

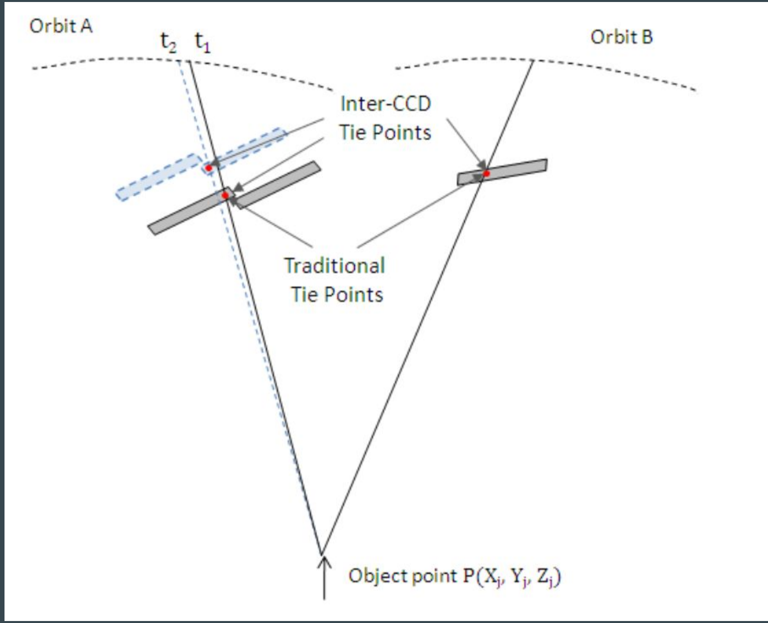
Intrinsic properties (Optical Centre, scaling)

Extrinsic properties (Camera Rotation and translation)

3D World Coordinates

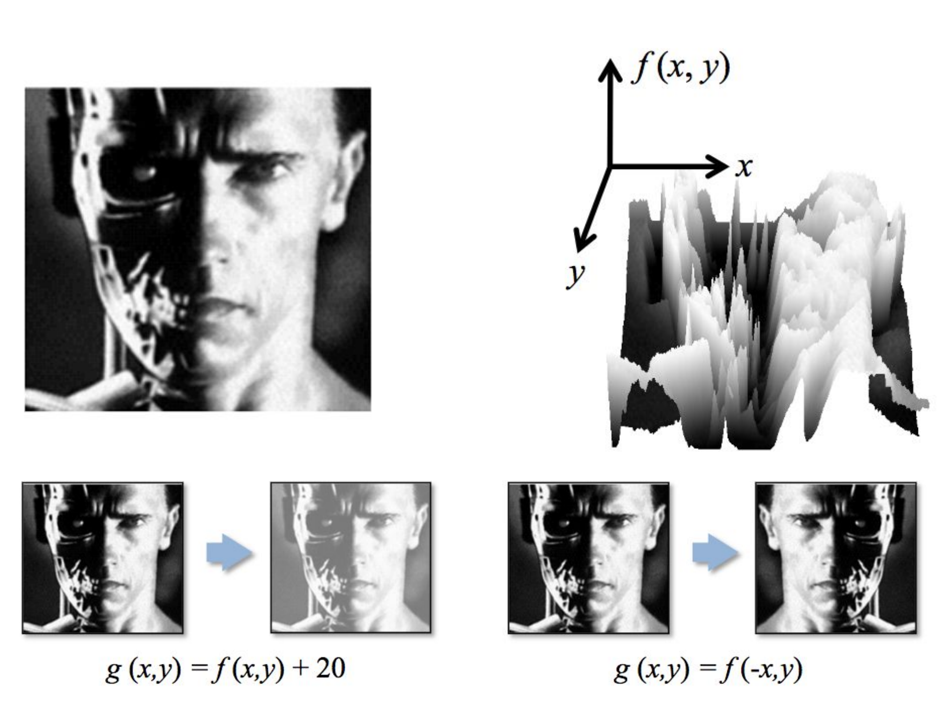
Concept: Features

- Stereo multiview produces poor matches
- Features can be used for better matching
- What makes a good feature?
- In remote sensing features are sometimes called “Tie Points”



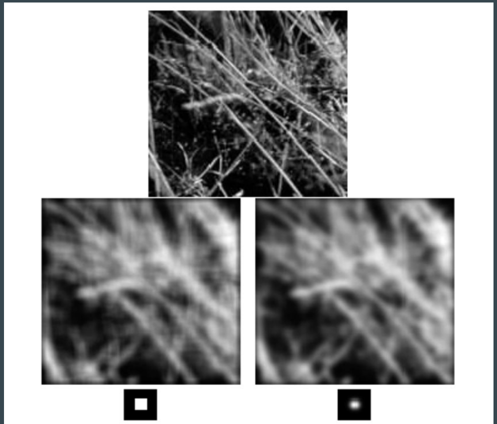
Step Back: Images

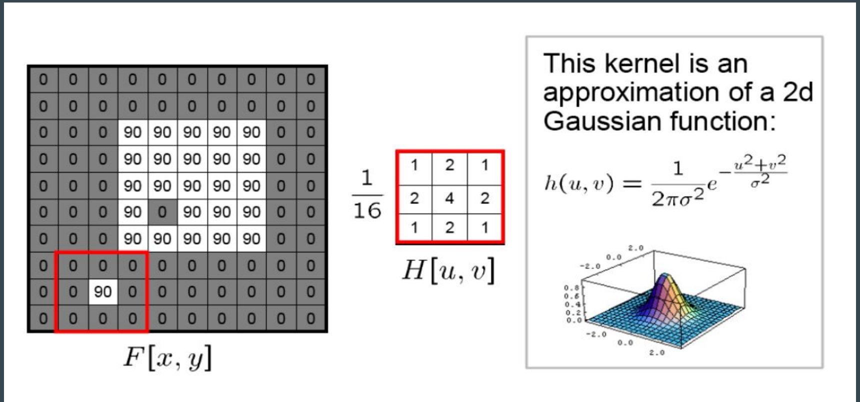
- Images as functions
- Operations can be performed on images
- Intensity from the real world projected onto our image plane



Reminder: Discrete Convolution

- “Filter” the image with a gaussian function
- Gaussian convolution is a better “blur” or “smoothing” of the image function
- Convolved with a “kernel”





Concept: Image Derivatives

- The “Sobel” kernel can be used to find image derivatives in the X and Y
- Gx or Gy is convolved with image
- Finds points of extreme change “extrema”
- The magnitude of the gradient of the image is stored

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

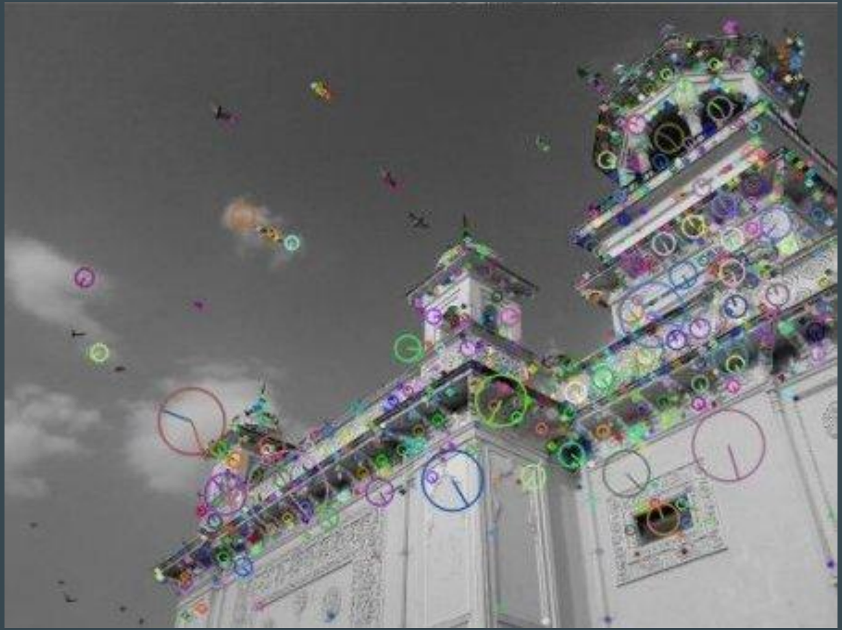
Gy

Concept: Scale Space

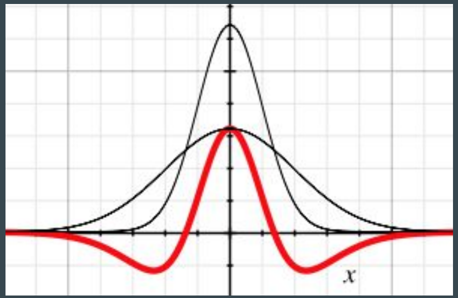
- A formal theory describing image structures at differing scales…
- Convolving images with gaussians of different sigma
- Similar effect to changing pixel sizes, resolution, GSD, focal length, etc …

The Scale Invariant Feature Transform (SIFT)

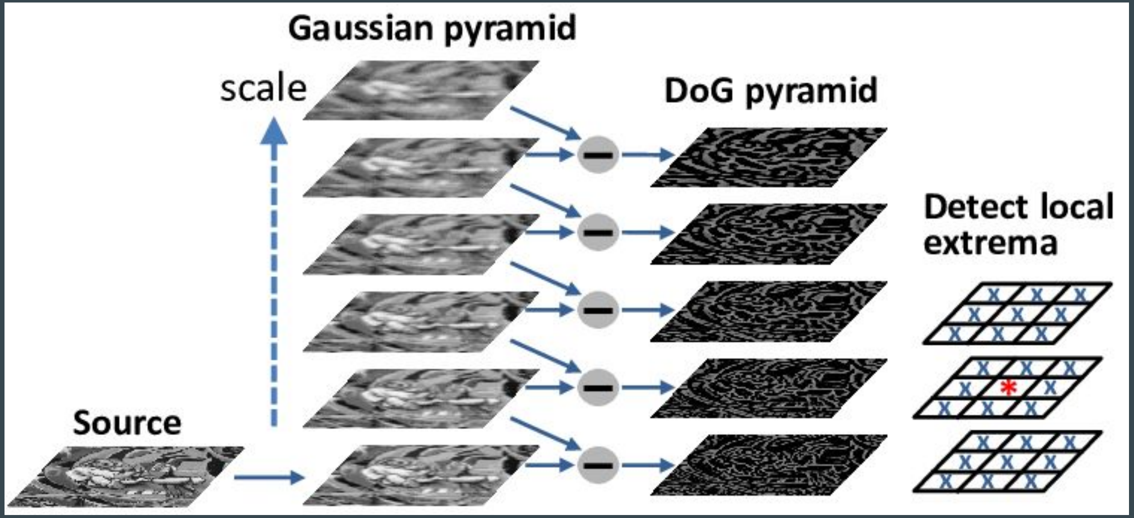
- When released created a mini revolution in Computer Vision
- Finds scale, rotation, and translation *invariant* features
- Generates features, which are the "Tie Points" we want



SIFT - how it works (briefly)

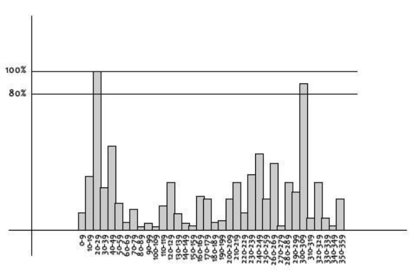


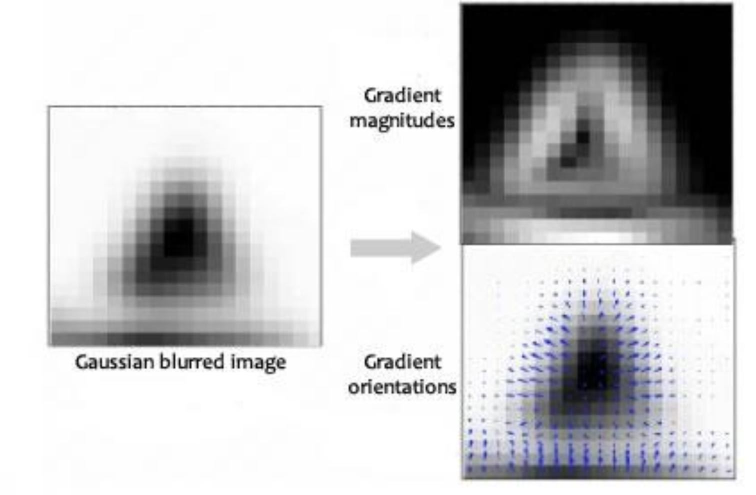
- Convolves Gaussian kernel with images while varying sigma
- Creates multiscale Gaussian pyramid
- Approximates a Laplacian of Gaussians (LoG) with a Difference of Gaussians (DoG)
- Detected extrema are stable features locations in scale space



SIFT - how it works (briefly)

- Around each detected extrema, calculate a Histogram of Oriented Gradients (HoG)
- Rotate the HoG to the greatest histogram bin
- HoG is used as a feature descriptor

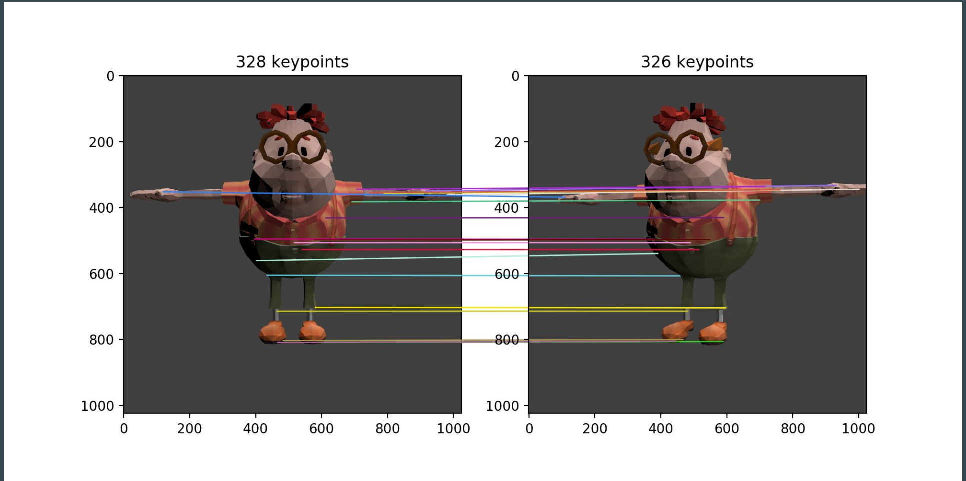




SIFT - how it works (briefly)

- Matching features is done by comparing the HoG of features
- Minimum Euclidean distance
- Min difference between the 128 feature vectors

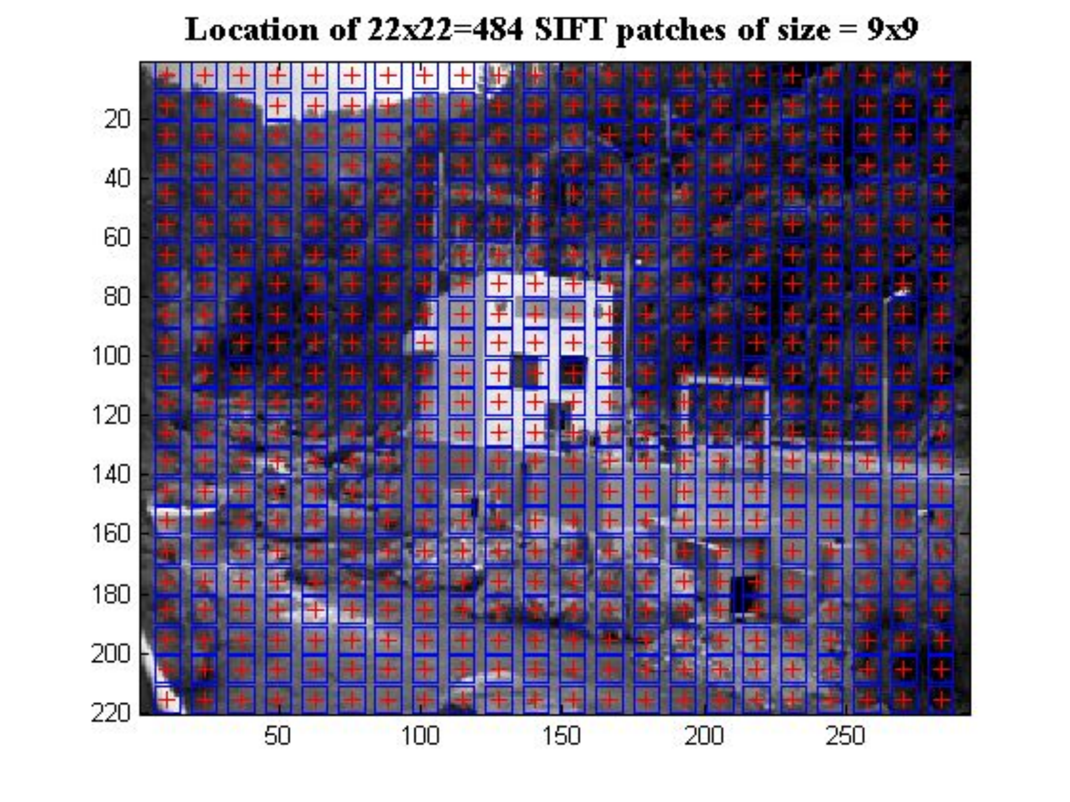
$$m = \sqrt{\sum_{0}^{127} (f_{I2} - f_{I1})^2}$$



Demo - Standard SIFT

Dense SIFT

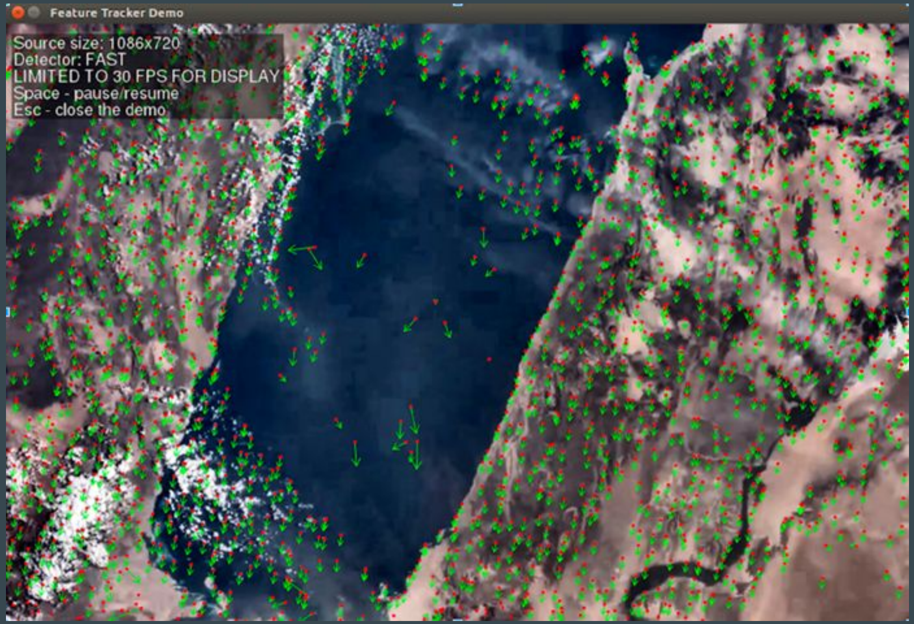
- Essentially the computation of SIFT descriptors across all pixels
- Creative approaches often needed due to memory constraints
- Subpixel precision must be interpolated within the matching stage



Demo - Dense SIFT

Optical Flow

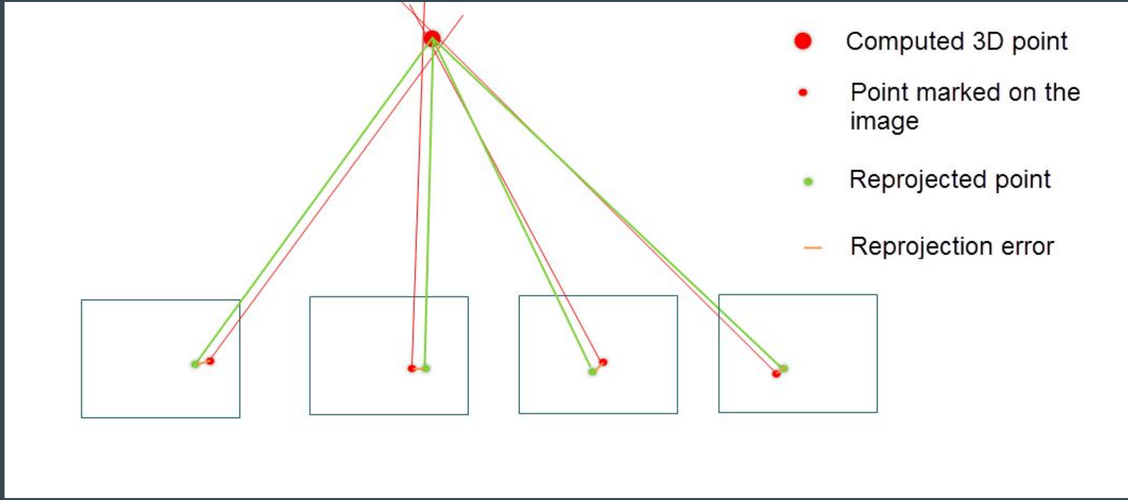
- Each Frame:
  - Feature locations determined
  - Feature descriptors calculated
- Track Objects
- A single step in Structure from Motion



demo - Optical Flow

Triangulation

- Skew lines are practically guaranteed. An estimated point is generated at closest approach with least squares.
- Reprojection Error: The distance between the estimated point projection and the feature location
- Errors in projection and feature detection must be accounted for



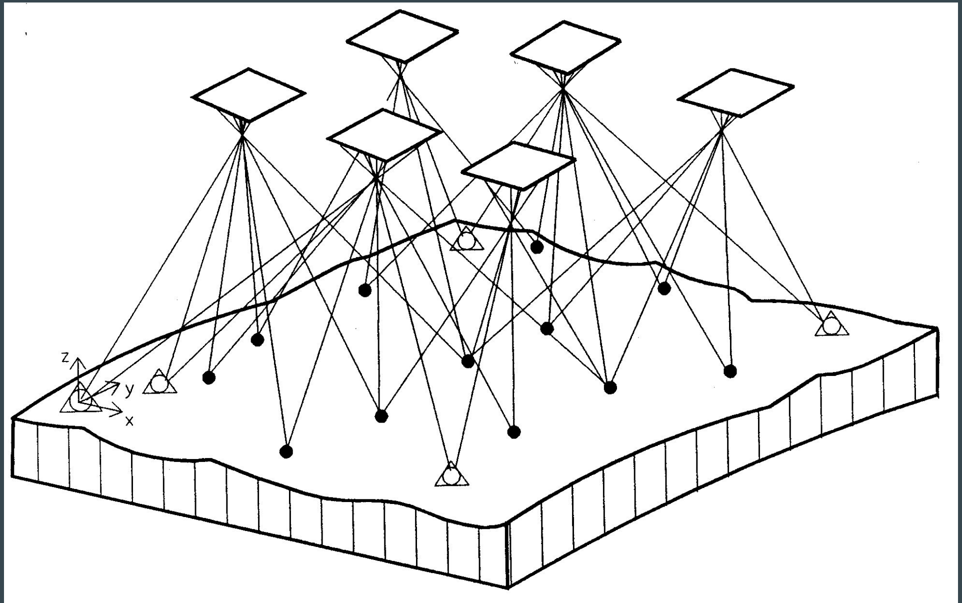
$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} f_x & s_k & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

demo - Triangulation

- 2 view discussion
  - 2-view Least Squares
  - 2-view skew line
- 2-view Least Squares demo
- n-view discussion

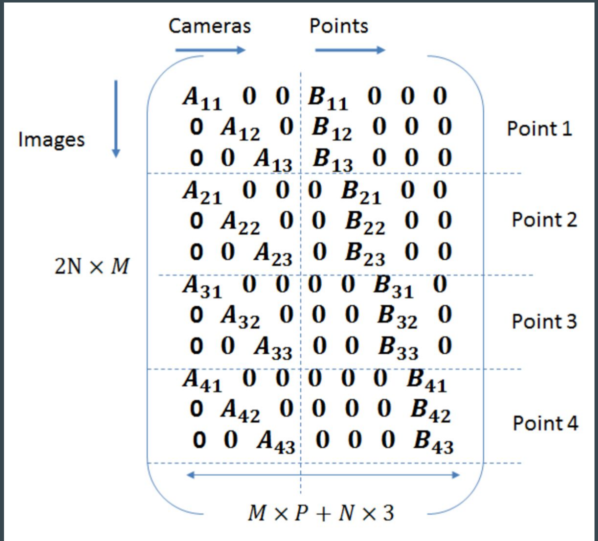
Bundle Adjustment

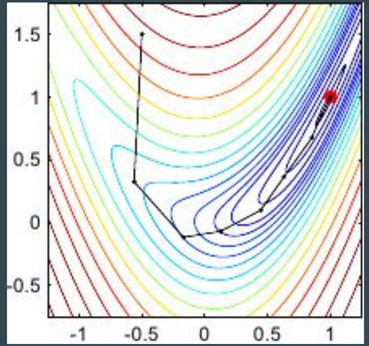
- We seek to minimize the reprojection error
- Consider the relationship between all variables
- Generate massive matrix to optimize
- Final step to generate 3D model cloud (sort of)
- Very difficult to compute



Solving Bundle Adjustment

- Large matrix optimization problem
- Gauss-Newton methods are typically used. The cost function is the reprojection error.
- Levenberg–Marquardt algorithm is current best general approach
- Jacobians and Hessians over a massive matrix… this is why we need a GPU in space!





MOCI Simulations

- We test our algorithms with simulations
- Realistic 3D models
- GPUs are needed
- Decent results so far!

`./main.py config.json`

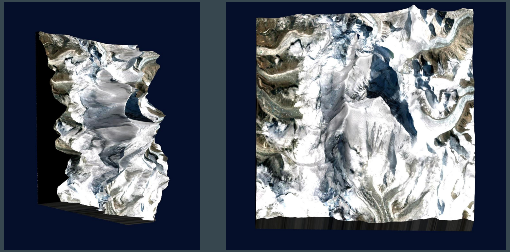
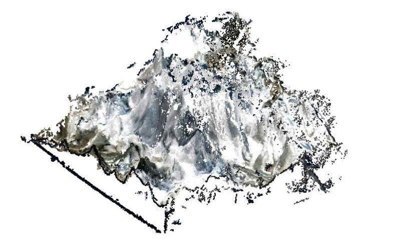
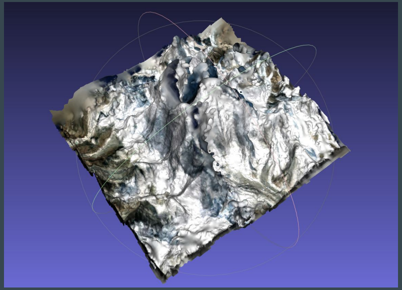


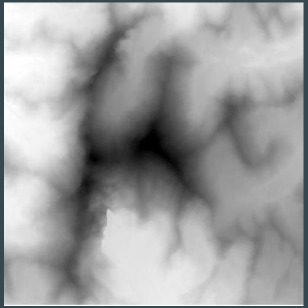
image acquisition



point cloud generation



surface reconstruction



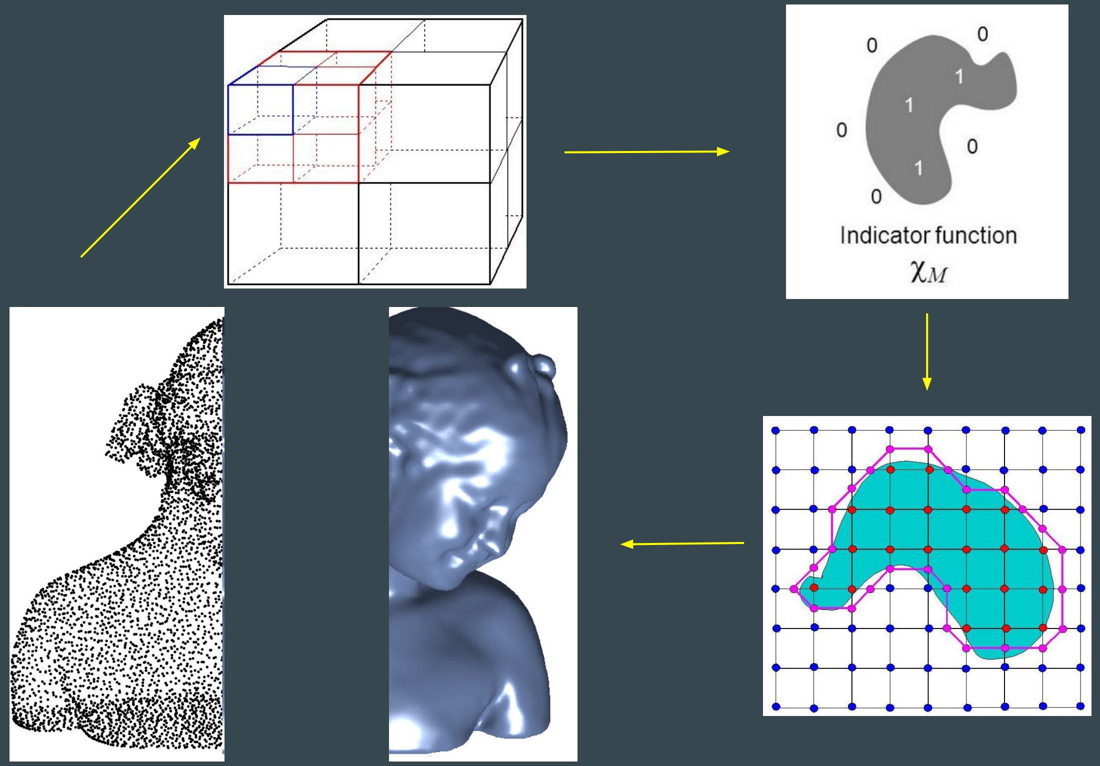
GeoTiff Generation (Rasterization)

demo - Blender Simulations

Small Satellite Research Laboratory
Franklin College of Arts and Sciences
UNIVERSITY OF GEORGIA

Surface Reconstruction - Overview

- Octree Generation
- Normal Determination
- Poisson Surface Reconstruction
  - Indicator Function Determination
  - Marching Cubes

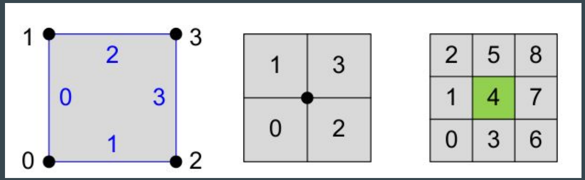


The Octree

- Built on GPU - can be used on CPU or GPU
- Nodes
  - Pointer to contained points
  - Children, Neighbor and Parent pointers
  - Vertices, Edge and Face pointers
- Non redundant Vertice, Edge and Face Arrays

MAJOR STEPS

1. **Build nodes around points**
2. **Fill in missing siblings and generate parents**
3. **Neighborhood determination**
4. **Vertex, Edge and Face generation**



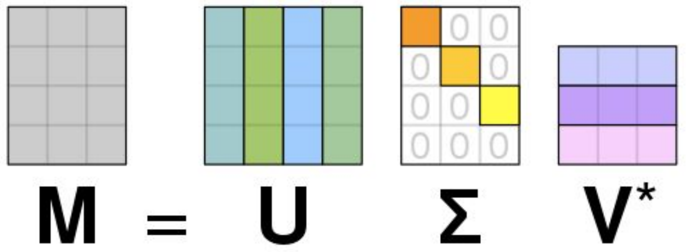
demo - The Octree

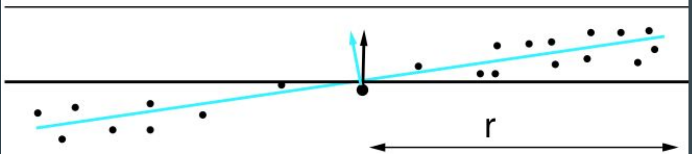
Point Normal Calculations

$$Q_{i,k} = \{q_1, q_2, \ldots, q_k\}$$

$$m = \frac{1}{k} \cdot \sum_{q \in Q} q$$

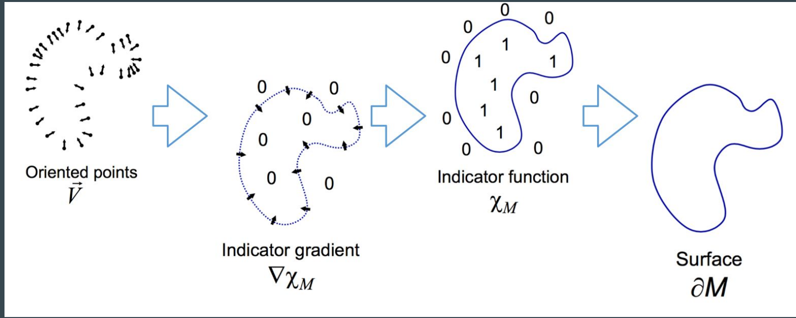
- Find K nearest Points
- Calculate centroid
- Build matrix $M_{i,k} = \{q \in Q_{i,k} \mid q - m\}$
- Singular Value Decomposition (SVD) on covariance matrix $M^T M$
- Normal = Last eigenvector (row in V^T)
  - orthogonal to neighbor plane
- Eliminate directional ambiguity
  - if direction is ambiguous relative to camera positions and normals of neighbors $n *= \{-1,-1,-1\}$





demo - Point Normal Caclulations

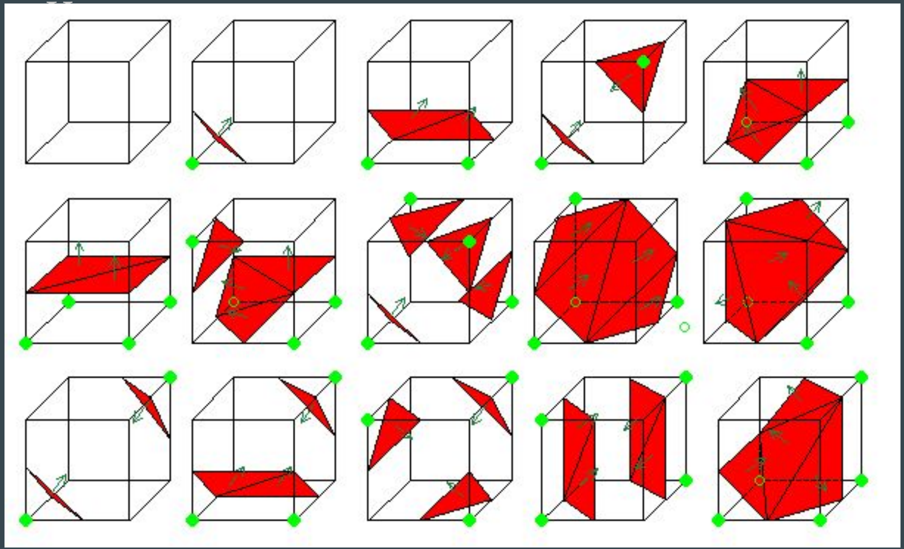
Poisson Surface Reconstruction



- Build a linear system $\mathbf{Lx} = \mathbf{b}$
  - $\mathbf{L}$ = Laplacian matrix
  - $\mathbf{b}$ = divergence vector
  - $\mathbf{x}$ = implicit function
- Multigrid conjugate gradient solver to solve for $\mathbf{x}$
- Compute the isovalue as an average of the implicit function values at sample points
- Distribute point implicit function values to octree vertices
- Indicator function = if edge's vertex implicit values are opposite in sign edge crosses surface
- Extraction of isosurface using marching cubes

Marching Cubes

- Look up table for all possible cube configurations
- If edge crosses surface its midpoint is used in triangulation
- Needs good indicator function to function properly
    - currently using oversimplified vertex implicit calculation using the closest point's normal



demo-
Marching Cubes

What's next?

- n-view matching
- bundle adjustment
- poisson reconstruction

Questions?

smallsat.uga.edu
University of Georgia