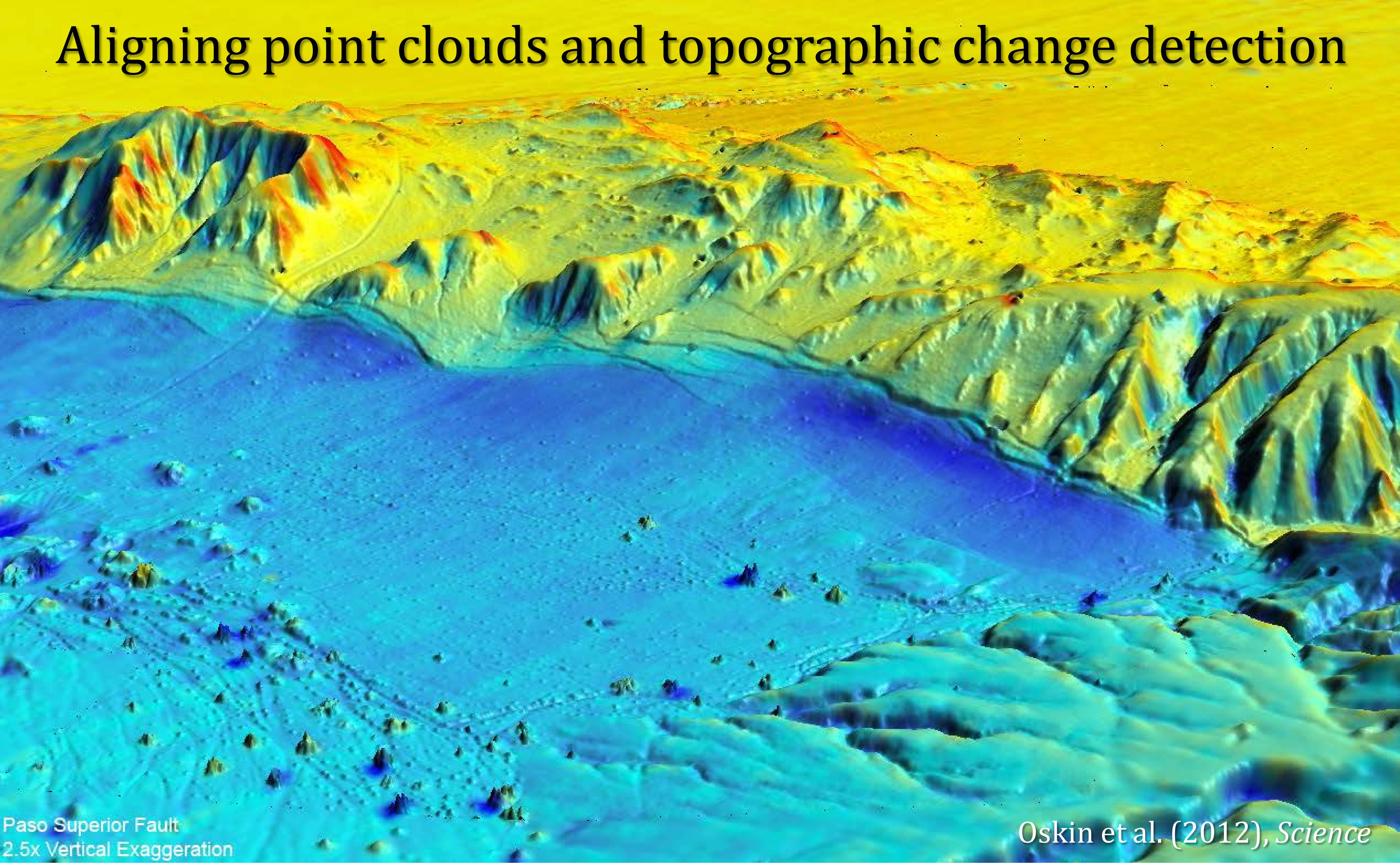
Aligning point clouds and topographic change detection



Edwin Nissen (Colorado School of Mines)

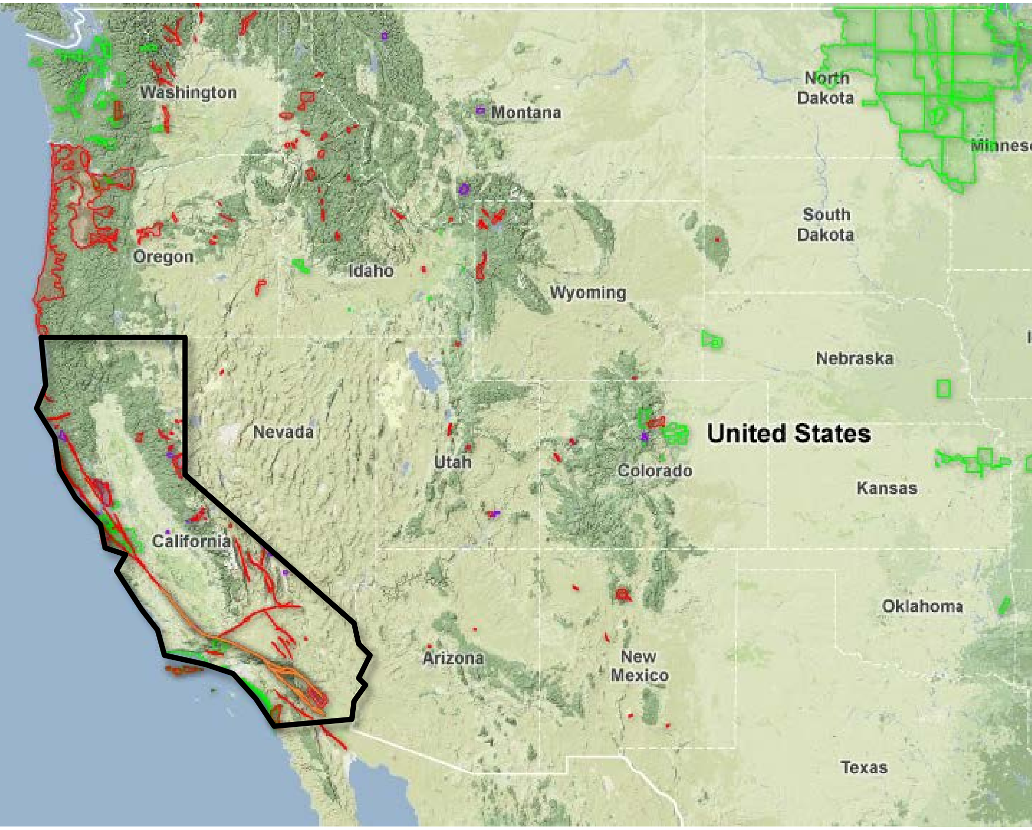
Thanks to: Ramon Arrowsmith, Srikanth Saripalli, Aravindhan Krishnan (ASU), Adrian Borsa (Scripps), Craig Glennie (Houston), Alejandro Hinojosa-Corona (CICESE), Tadashi Maruyama (AIST), Austin Elliott, Mike Oskin (UC Davis)

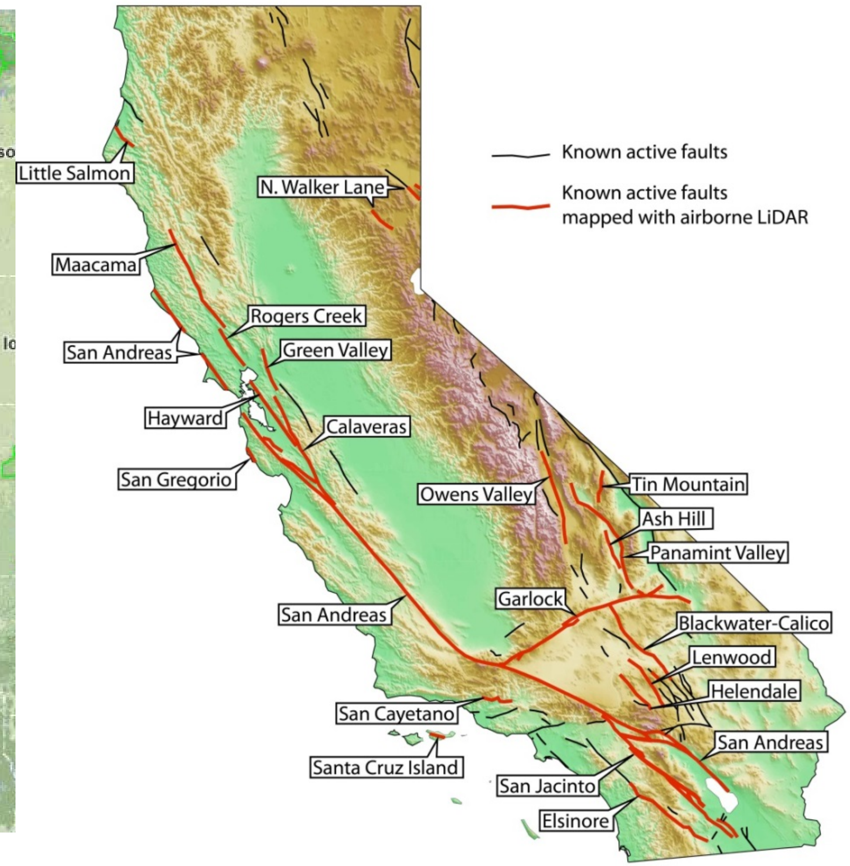
Aligning point clouds and topographic change detection

- Multi-temporal topography
- Earthquake examples:
    - scientific motivation
    - aligning (registering) topography data with ICP
    - 2008 Iwate earthquake (Japan)
    - 2011 Fukushima earthquake (Japan)
    - 2010 El Mayor Cucapah earthquake (Mexico)
- Other applications

Paso Superior Fault
2.5x Vertical Exaggeration

Aligning point clouds and topographic change detection

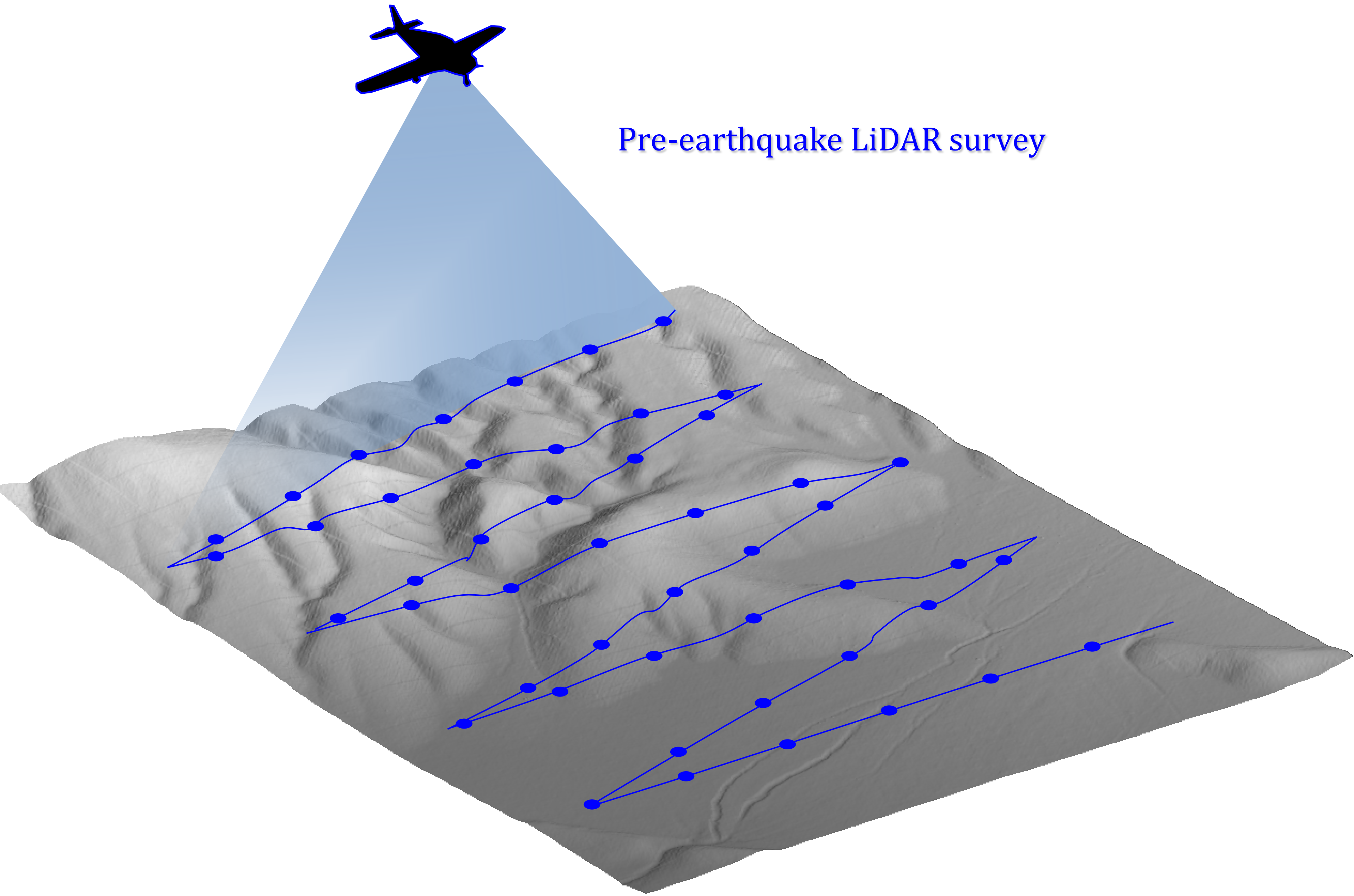




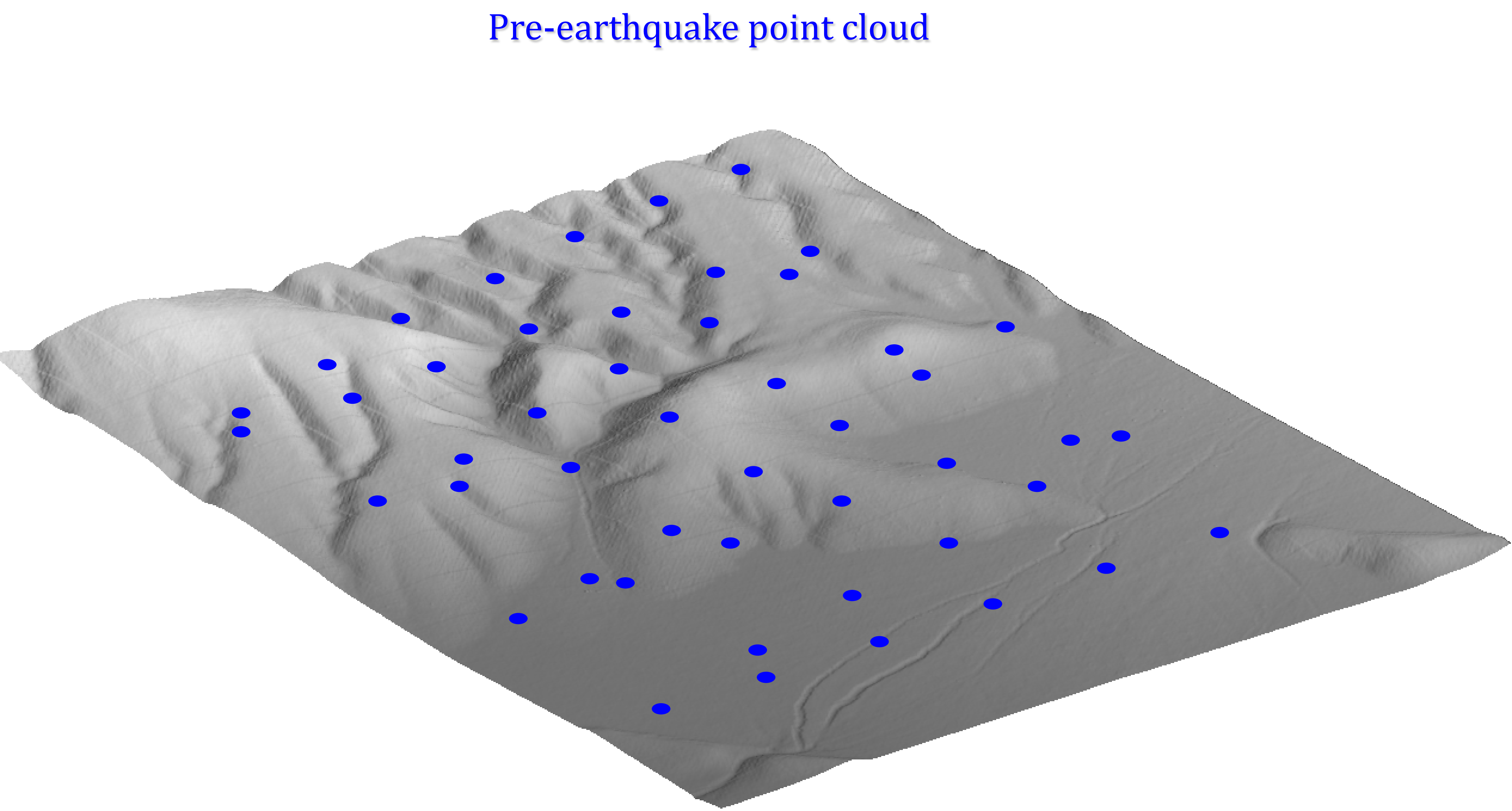
www.opentopography.org

- There is now a “baseline” of lidar topography on many active faults in the western US
- After an earthquake, repeat lidar data can be collected and differenced

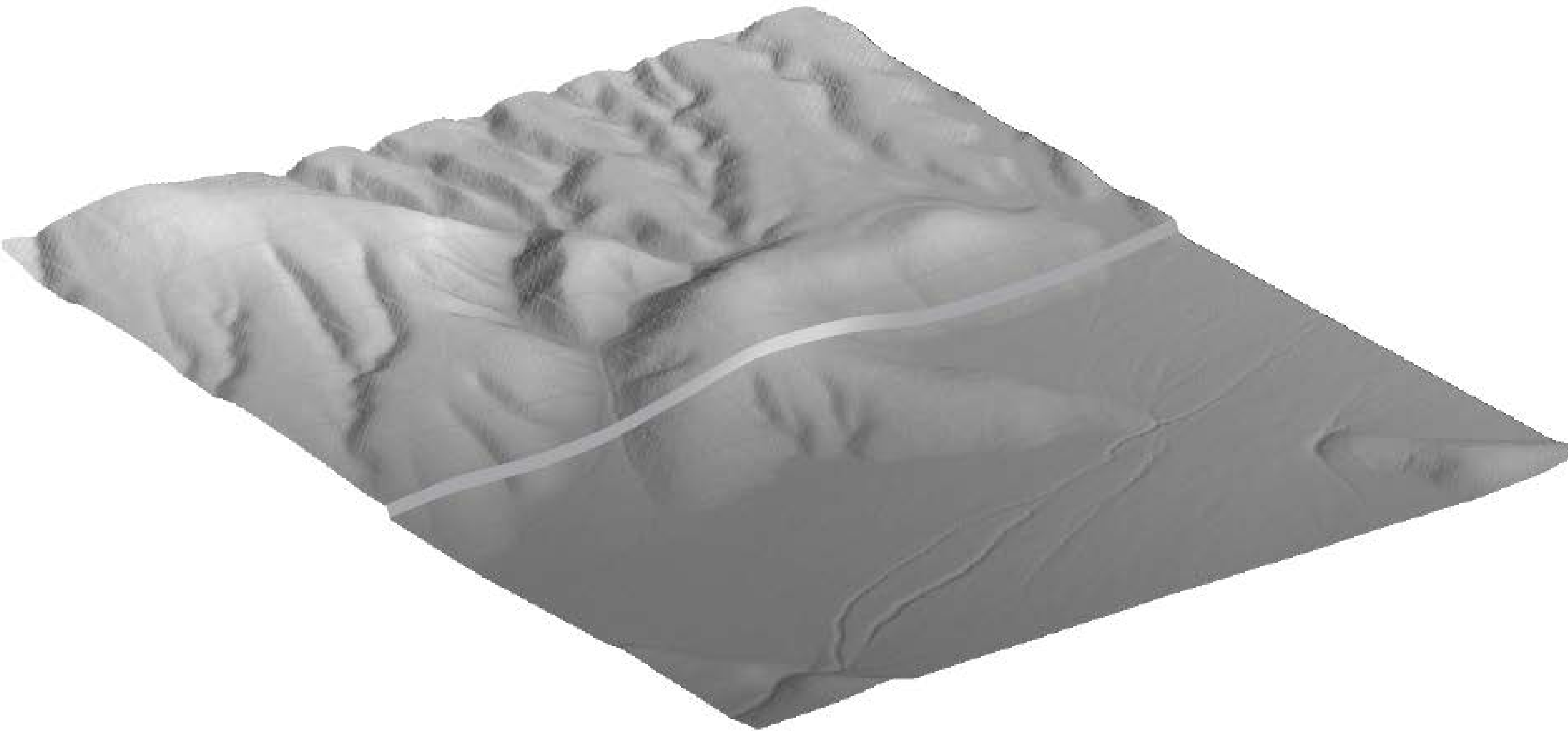
3-D earthquake deformation from repeat lidar



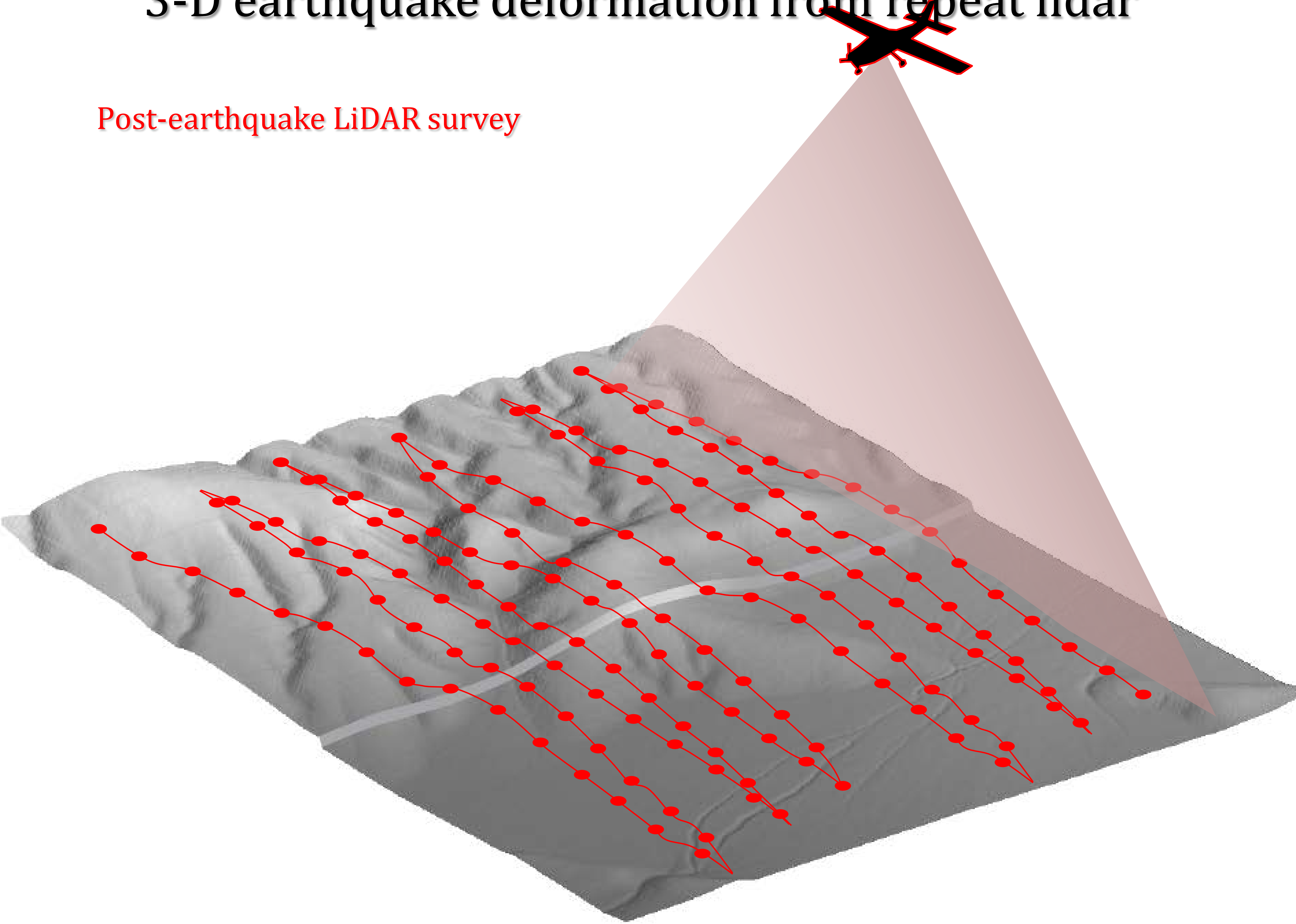
3-D earthquake deformation from repeat lidar



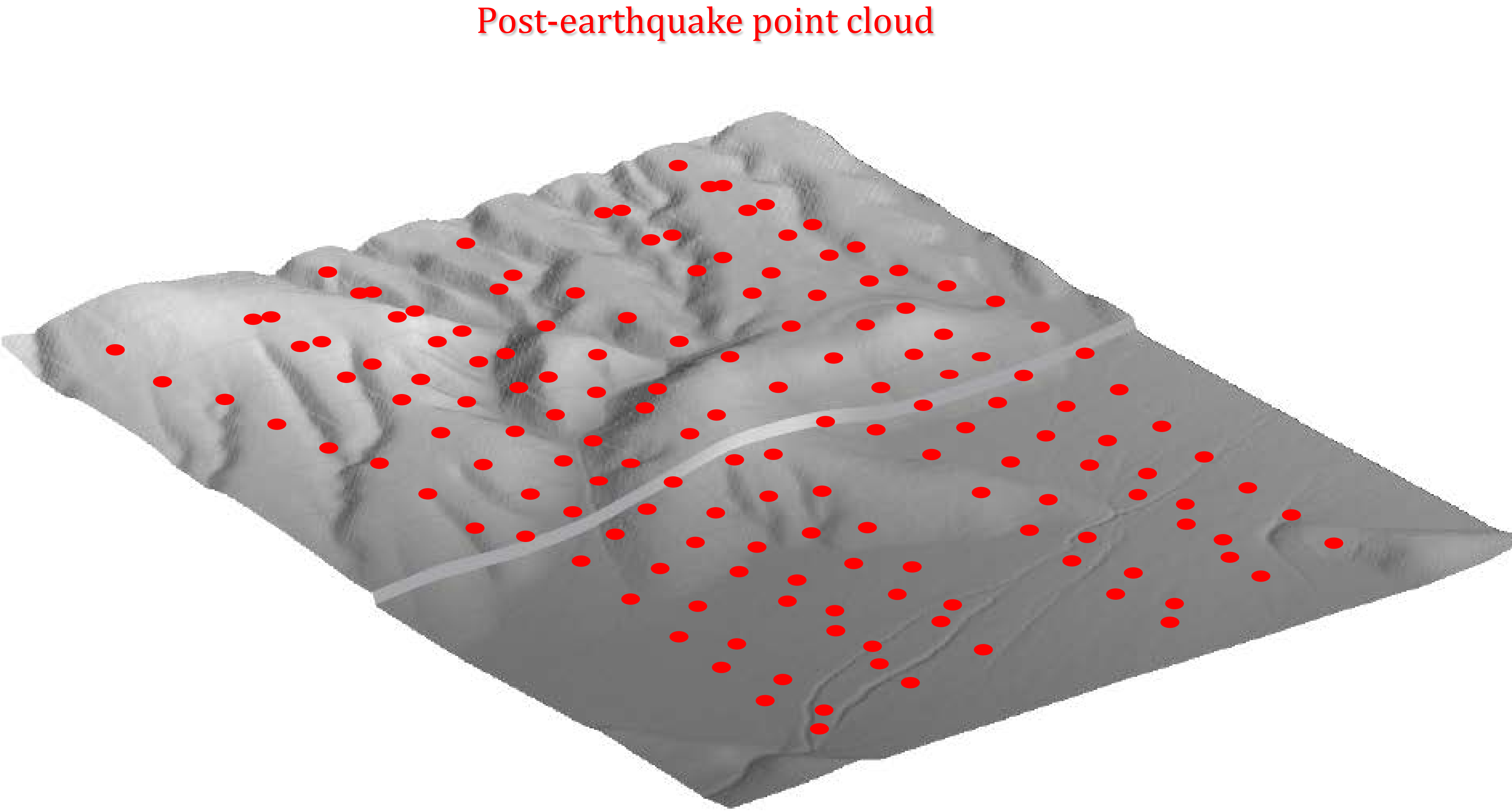
3-D earthquake deformation from repeat lidar



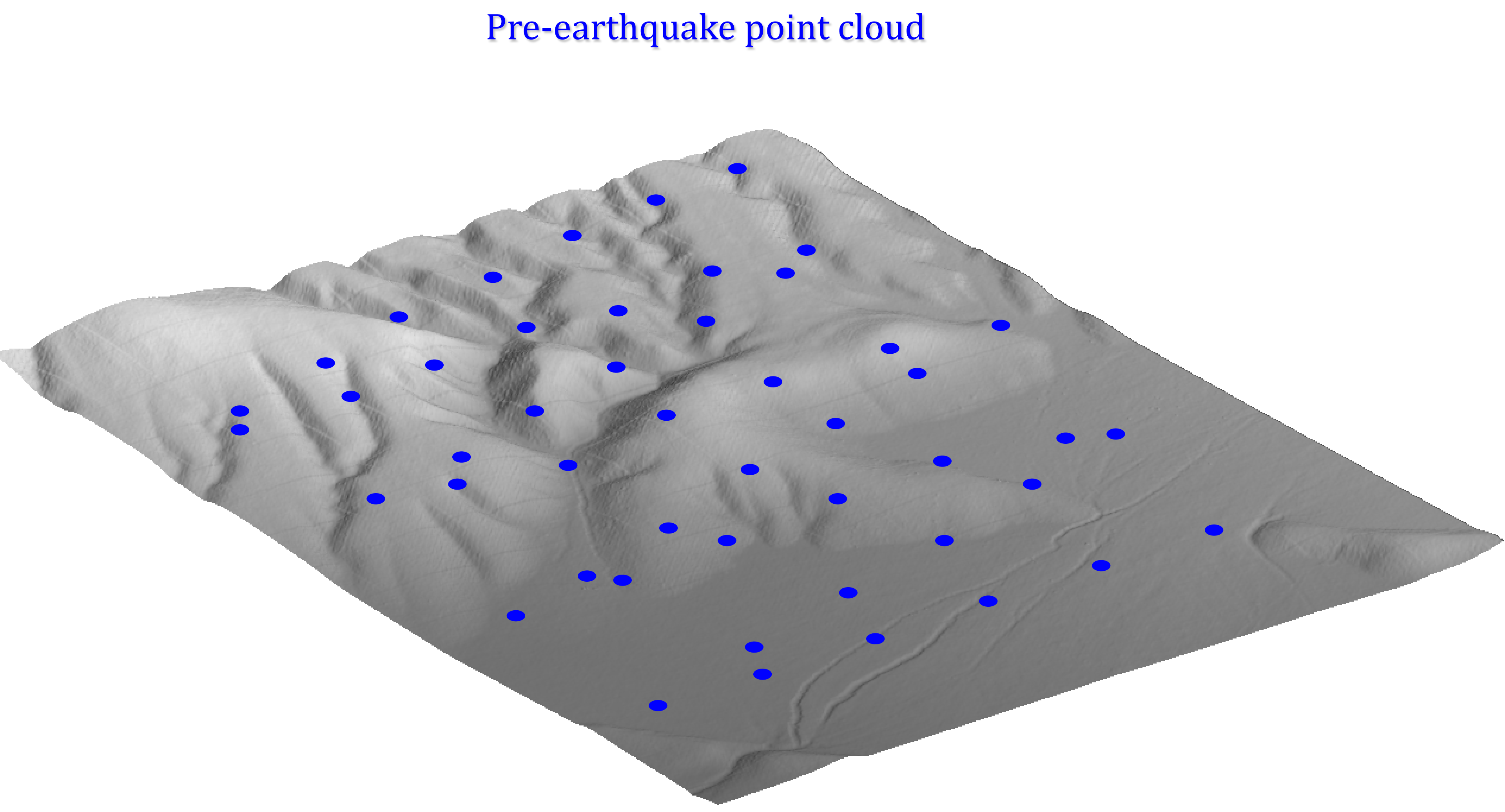
3-D earthquake deformation from repeat lidar



3-D earthquake deformation from repeat lidar

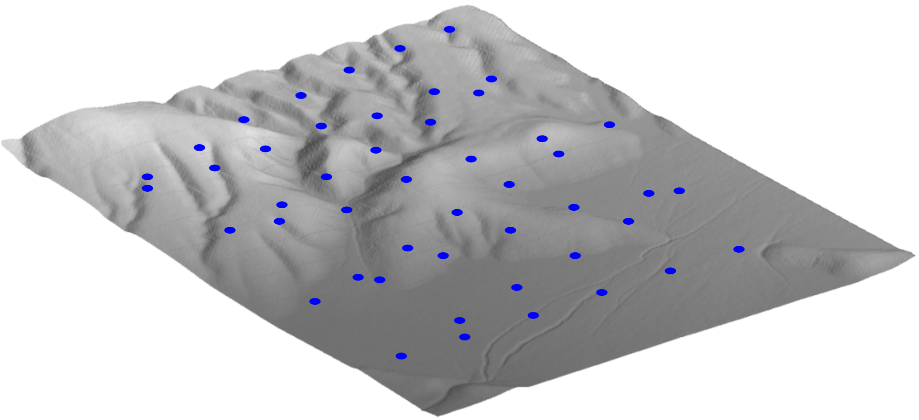


3-D earthquake deformation from repeat lidar

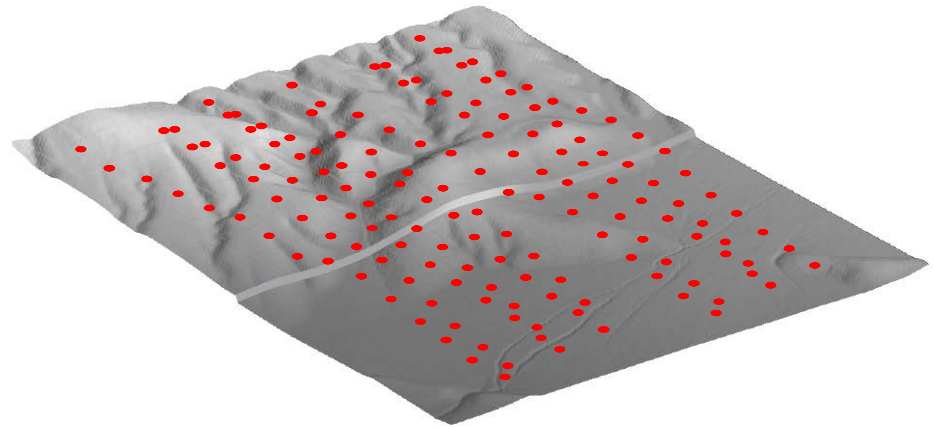


3-D earthquake deformation from repeat lidar

Pre-earthquake point cloud



Post-earthquake point cloud



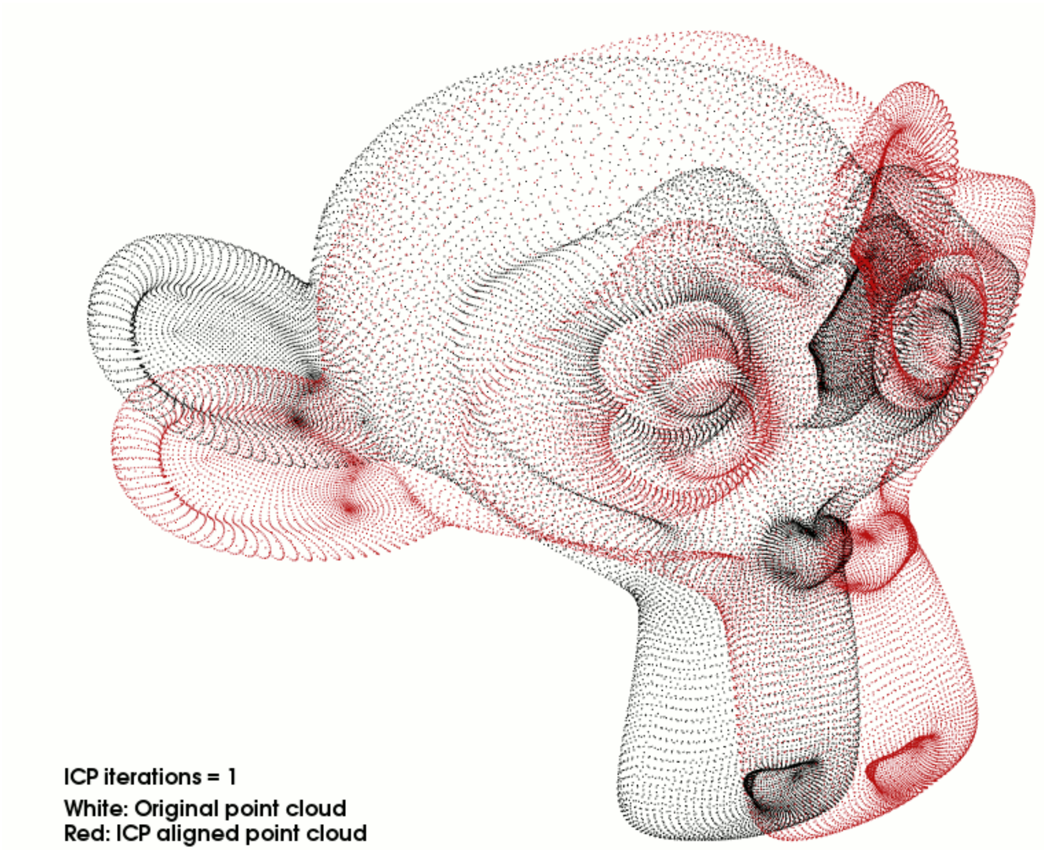
The Challenges of LiDAR differencing

- Data are irregularly spaced (we can rasterize them, but lose information doing so)
- There can be large mismatches in point density (legacy datasets vs modern surveys)
- … and mismatches in data quality and metrics (third party vs research-grade)
- Treatment of vegetation returns in forested areas

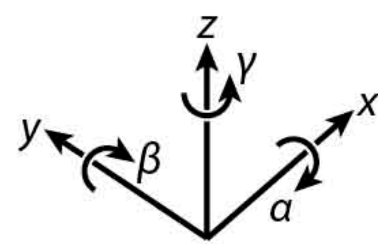
3-D earthquake deformation from repeat lidar

• The **iterative closest point** algorithm (ICP) is a method for registering (aligning) irregular point clouds, well known in computer vision and medical imaging

• ICP minimizes closest point pair distances using iterative **rigid-body transformations,** each one comprising a **translation** [t_x t_y t_z] and a **rotation** [α β γ]



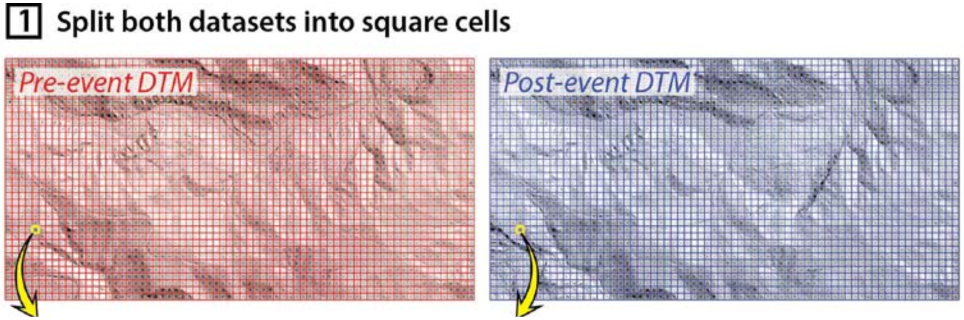
$$\Phi = \begin{pmatrix} 1 & -\gamma & \beta & t_x \\ \gamma & 1 & -\alpha & t_y \\ -\beta & \alpha & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



pointclouds.org/documentation/tutorials/interactive_icp.php

3-D earthquake deformation from repeat lidar

- The **iterative closest point** algorithm (ICP) is a method for registering (aligning) irregular point clouds, well known in computer vision and medical imaging
- ICP minimizes closest point pair distances using iterative **rigid-body transformations,** each one comprising a **translation** [t_x t_y t_z] and a **rotation** [α β γ]



- (1) the two LiDAR datasets are first split into square “cells”

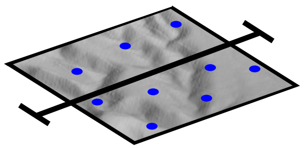
- (2) ICP is run on each equivalent pair of cells. The **translation** [t_x t_y t_z] corresponds to the cell displacement

- (3) this is repeated for the next pair of cells

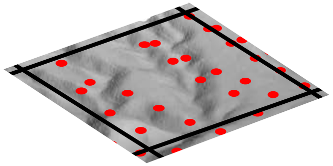
Nissen *et al.* (2012), *Geophys. Res. Lett.*

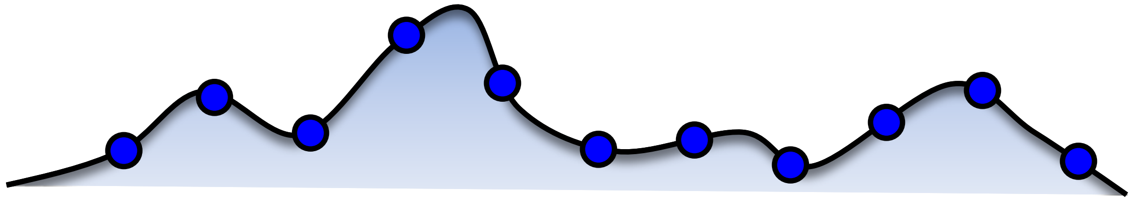
Iterative Closest Point algorithm (ICP)

Pre-earthquake cell

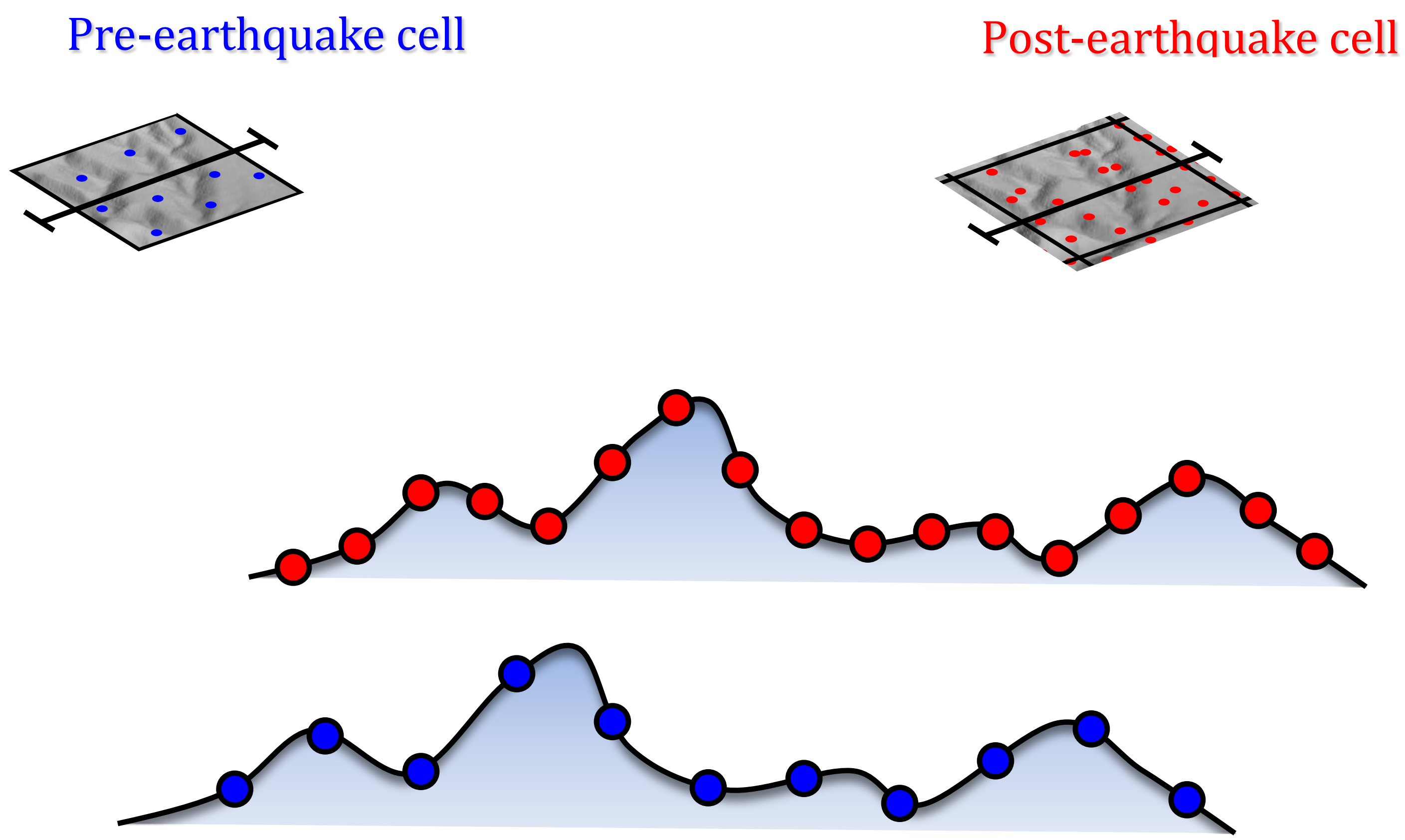


Post-earthquake cell



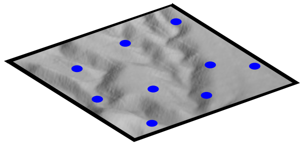


Iterative Closest Point algorithm (ICP)

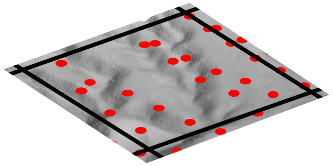


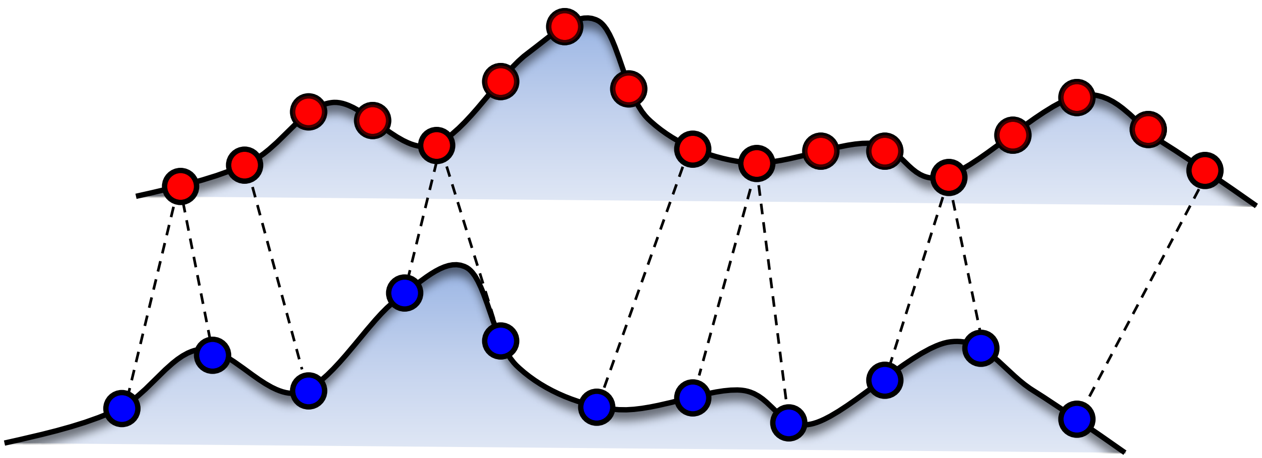
Iterative Closest Point algorithm (ICP)

Pre-earthquake cell



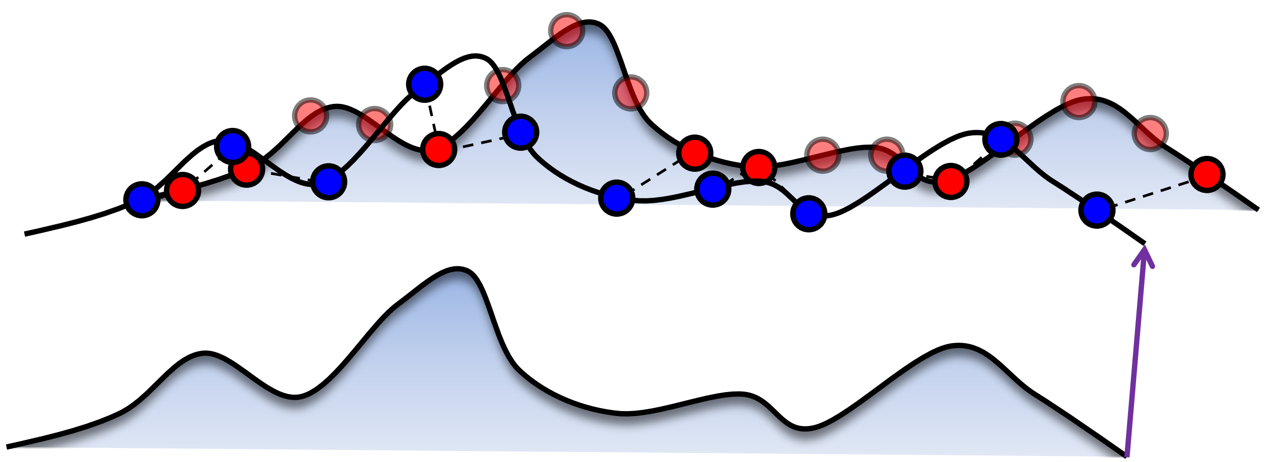
Post-earthquake cell





Find closest points

Iterative Closest Point algorithm (ICP)



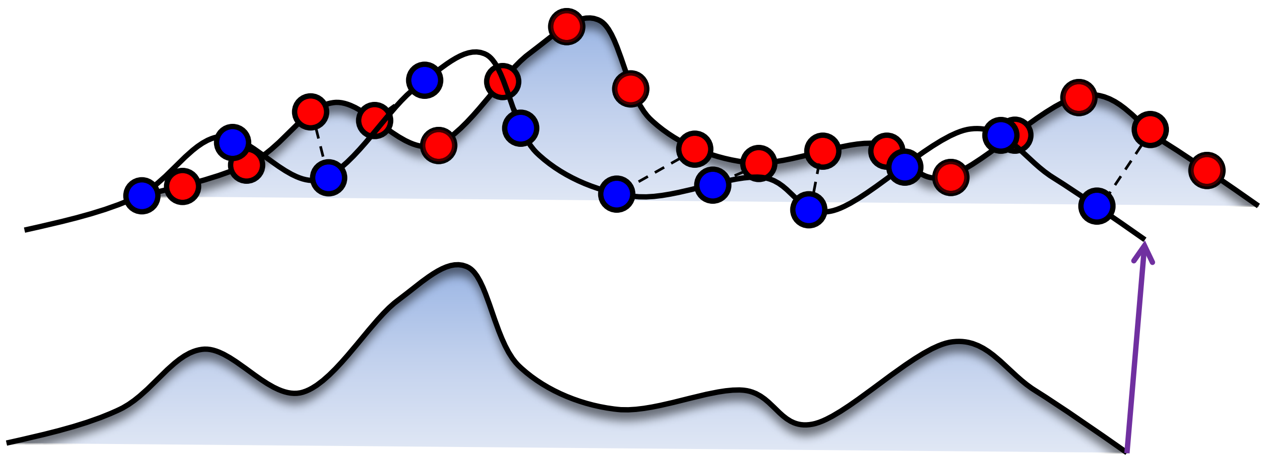
Iterate

Find closest points

Transform point cloud

$$\phi = \begin{pmatrix} 1 & -\gamma & \beta & t_x \\ \gamma & 1 & -\alpha & t_y \\ -\beta & \alpha & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

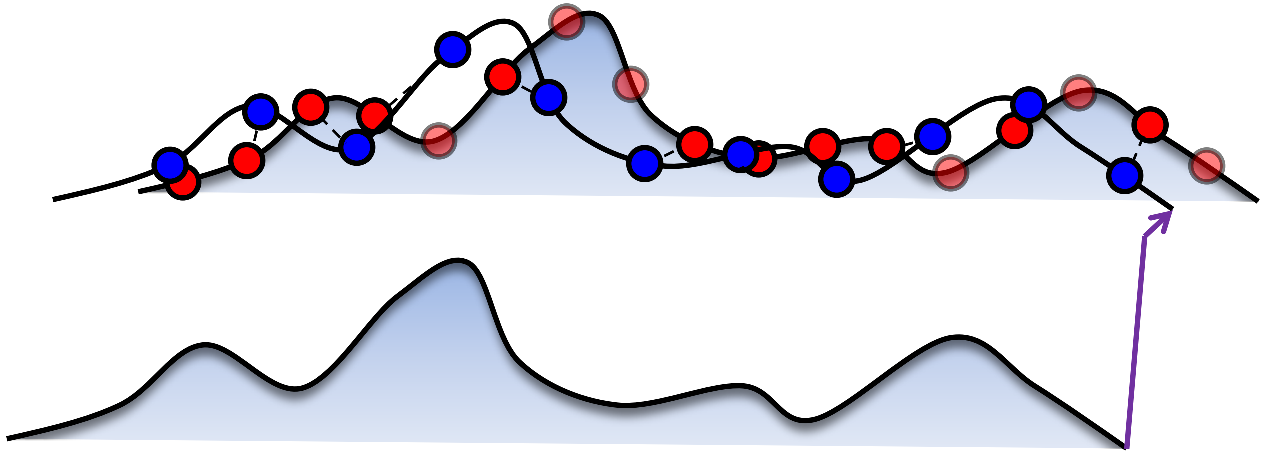
Iterative Closest Point algorithm (ICP)



Find closest points

Transform point cloud

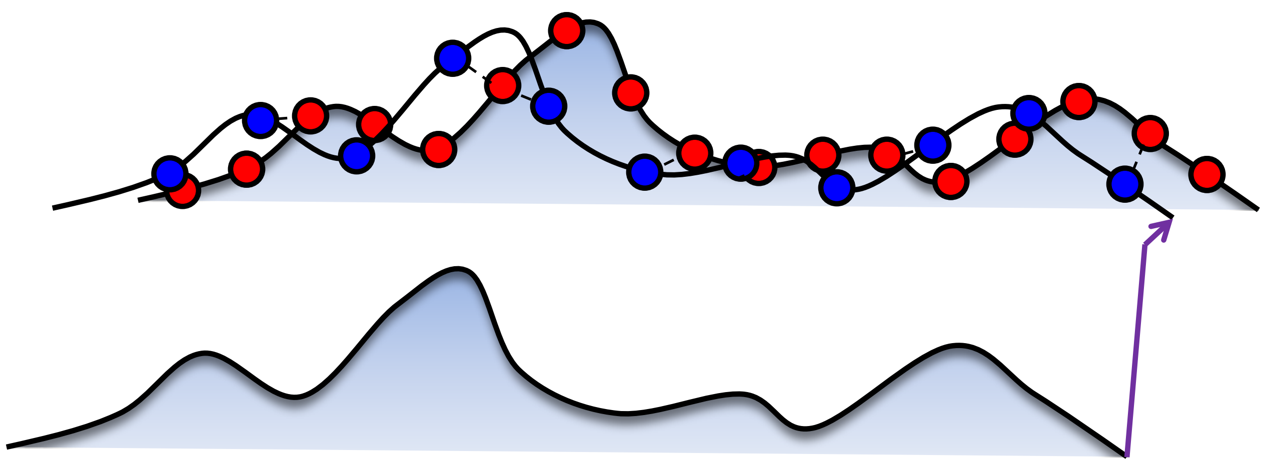
Iterative Closest Point algorithm (ICP)



Find closest points

Transform point cloud

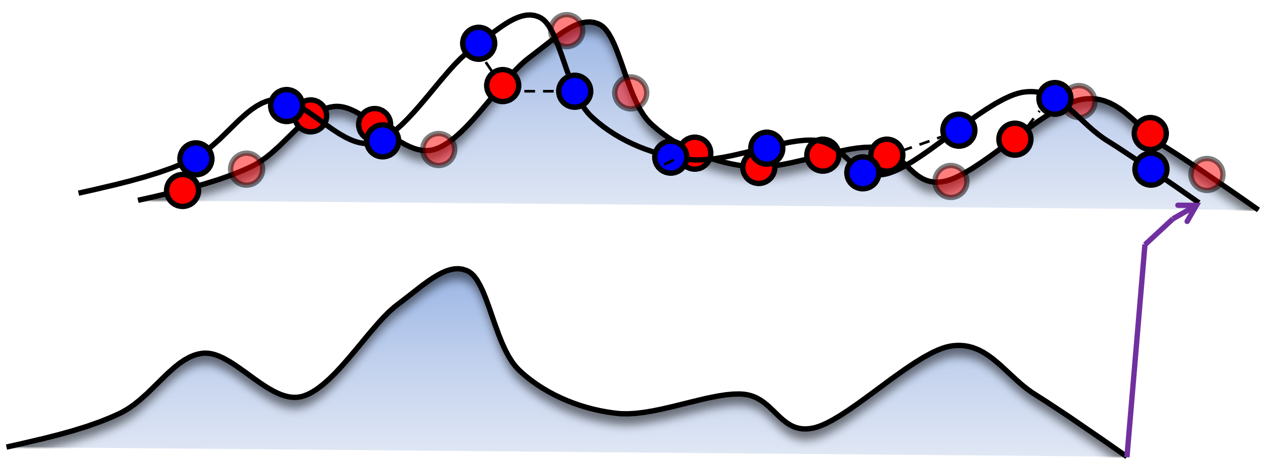
Iterative Closest Point algorithm (ICP)



Find closest points

Transform point cloud

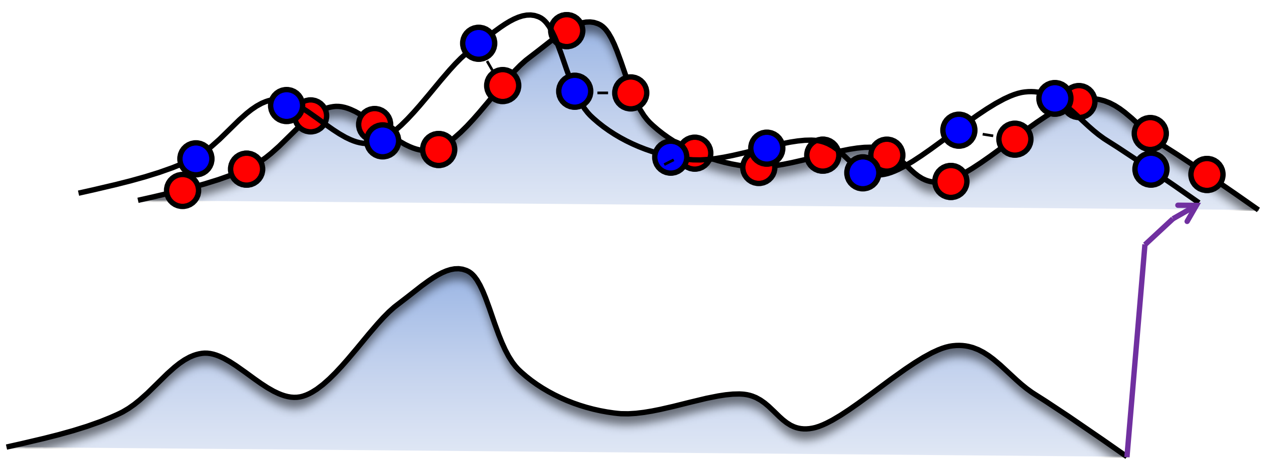
Iterative Closest Point algorithm (ICP)



Find closest points

Transform point cloud

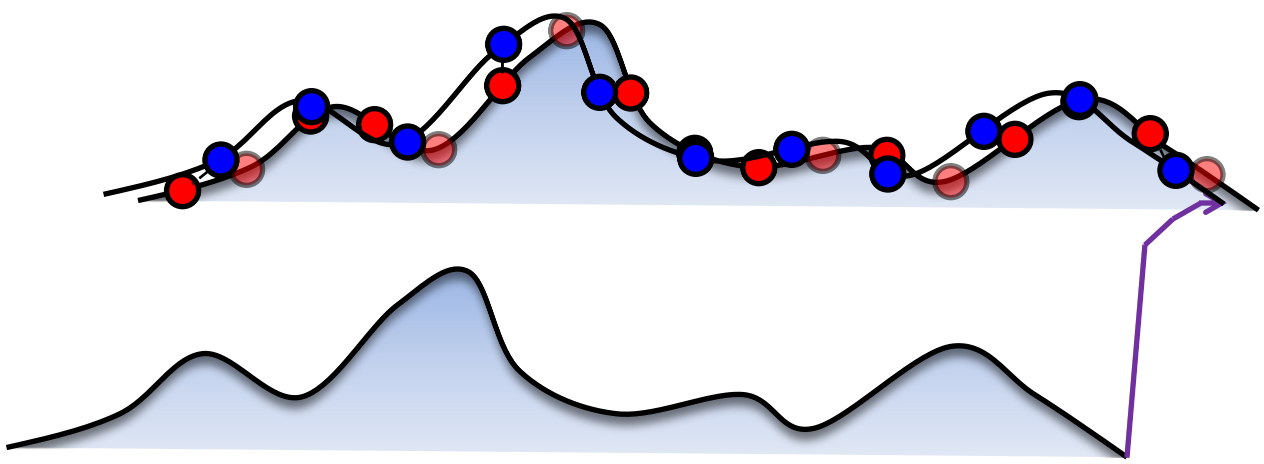
Iterative Closest Point algorithm (ICP)



Find closest points

Transform point cloud

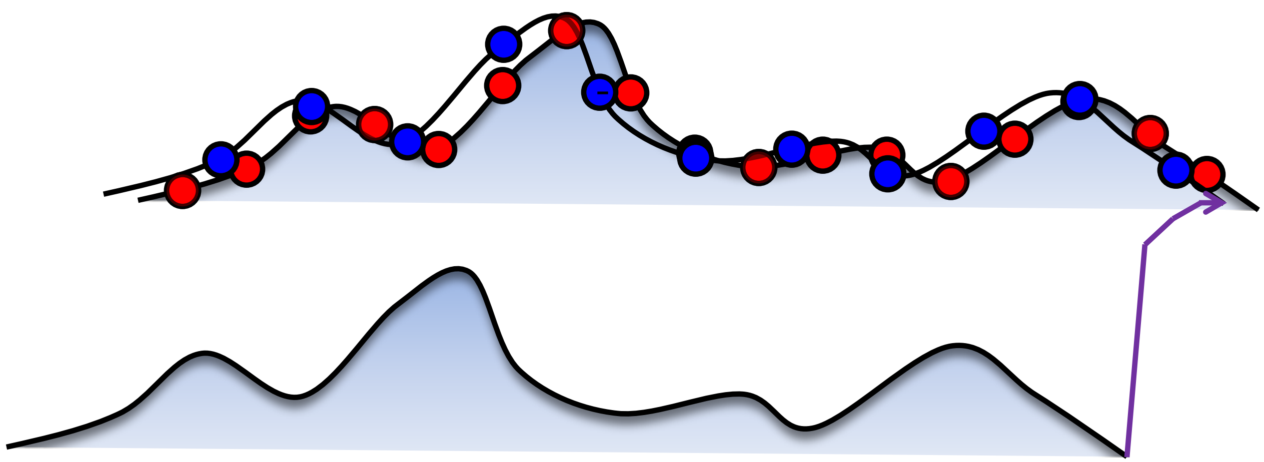
Iterative Closest Point algorithm (ICP)



Find closest points

Transform point cloud

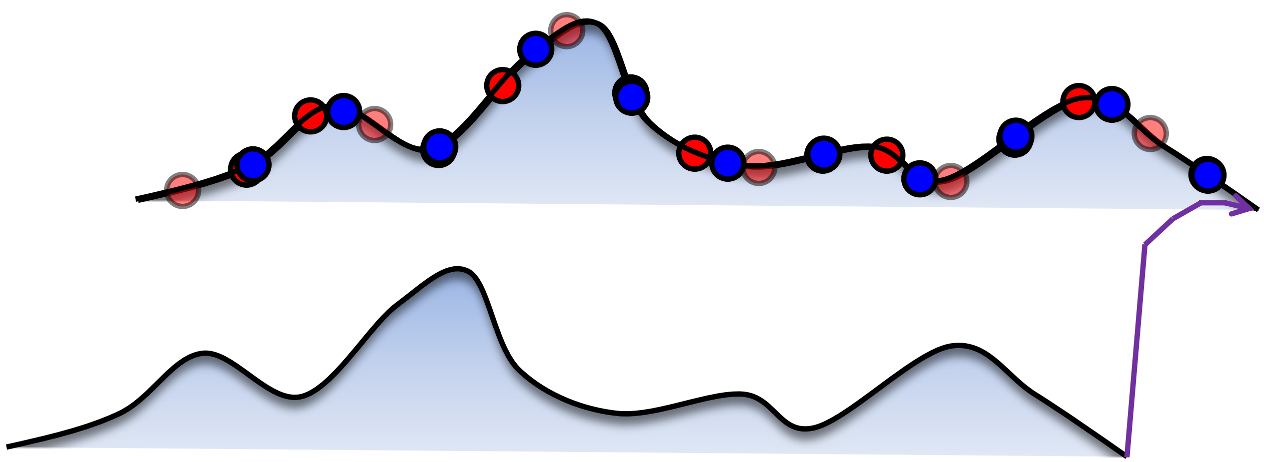
Iterative Closest Point algorithm (ICP)



Find closest points

Transform point cloud

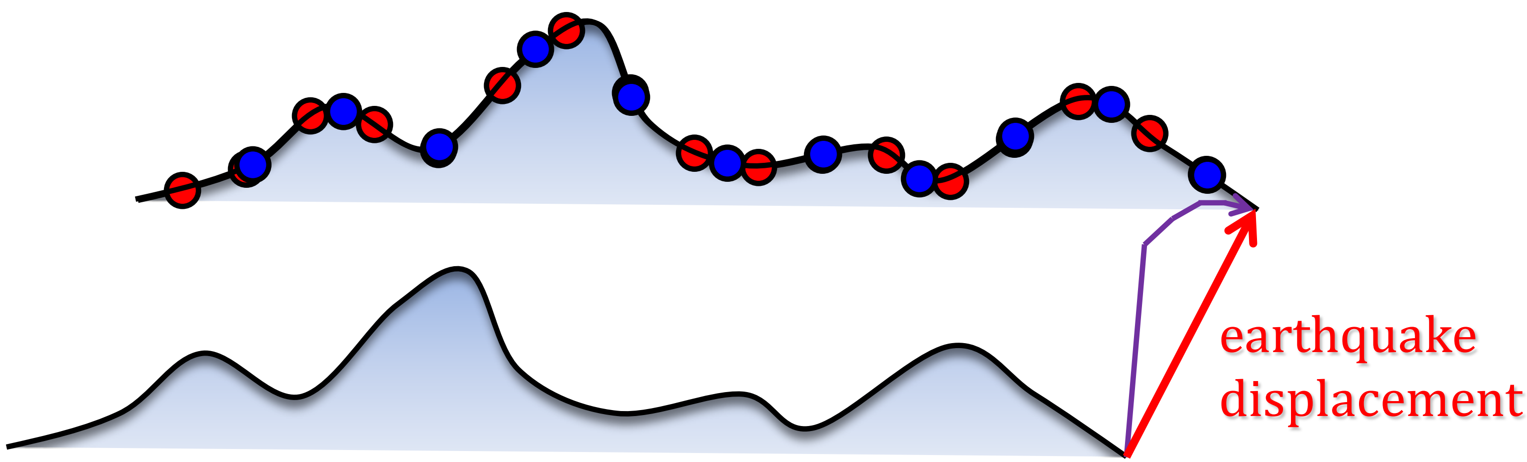
Iterative Closest Point algorithm (ICP)



Find closest points

Transform point cloud

Iterative Closest Point algorithm (ICP)



Find closest points

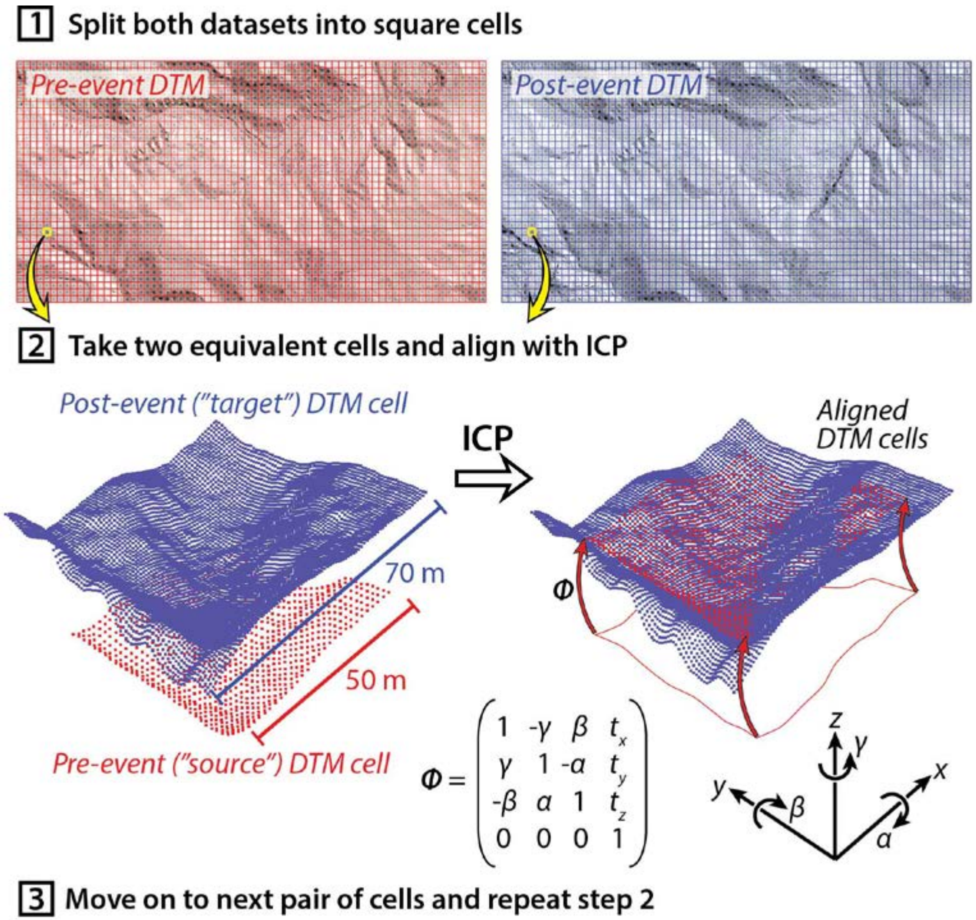
Transform point cloud

3-D earthquake deformation from repeat LiDAR point clouds

Caveats

• ICP will not work if there are large changes to the shape of the cell, e.g. through landsliding

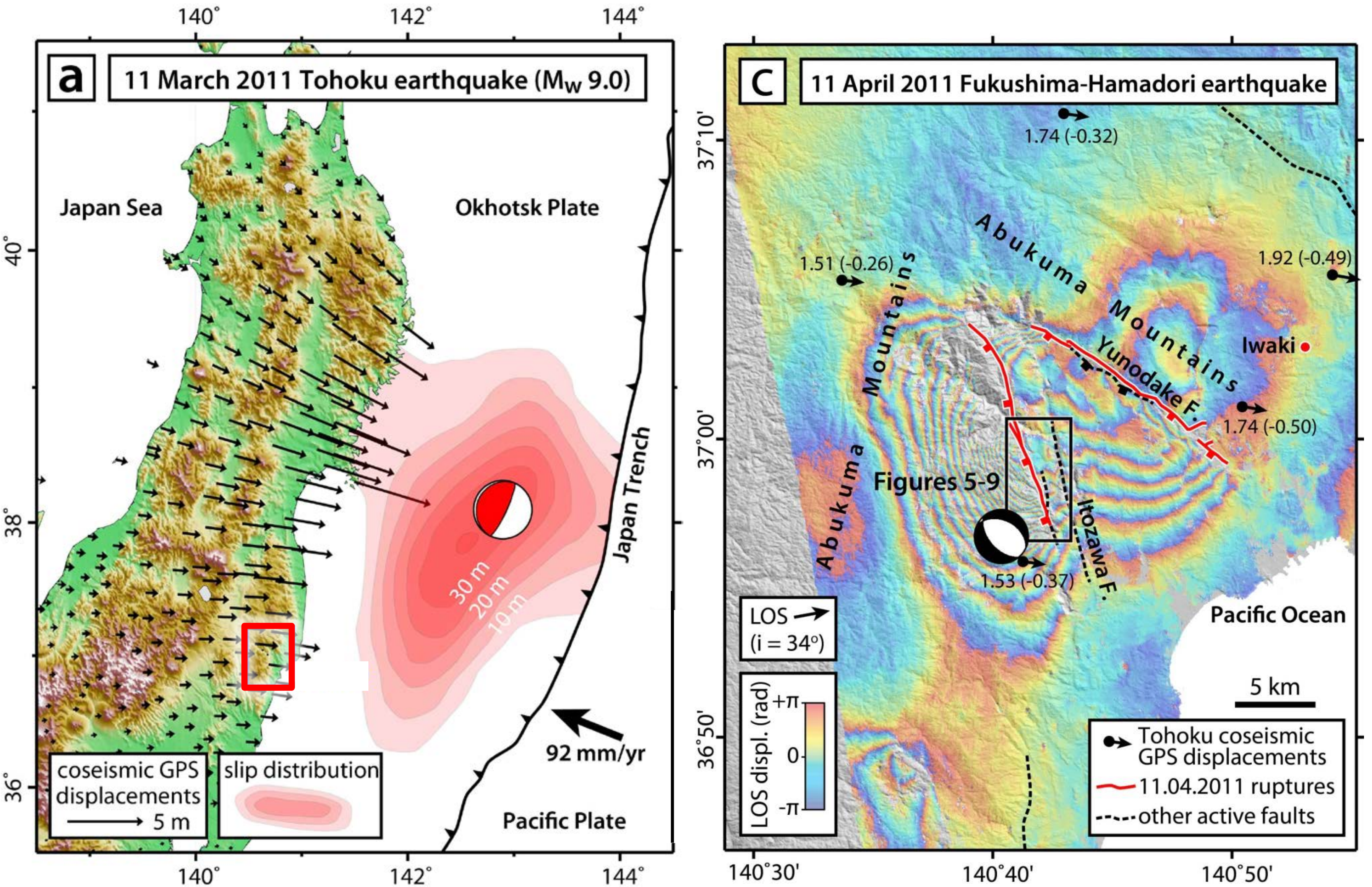
• ICP will generate spurious results in areas that are very planar



$$\Phi = \begin{pmatrix} 1 & -\gamma & \beta & t_x \\ \gamma & 1 & -\alpha & t_y \\ -\beta & \alpha & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

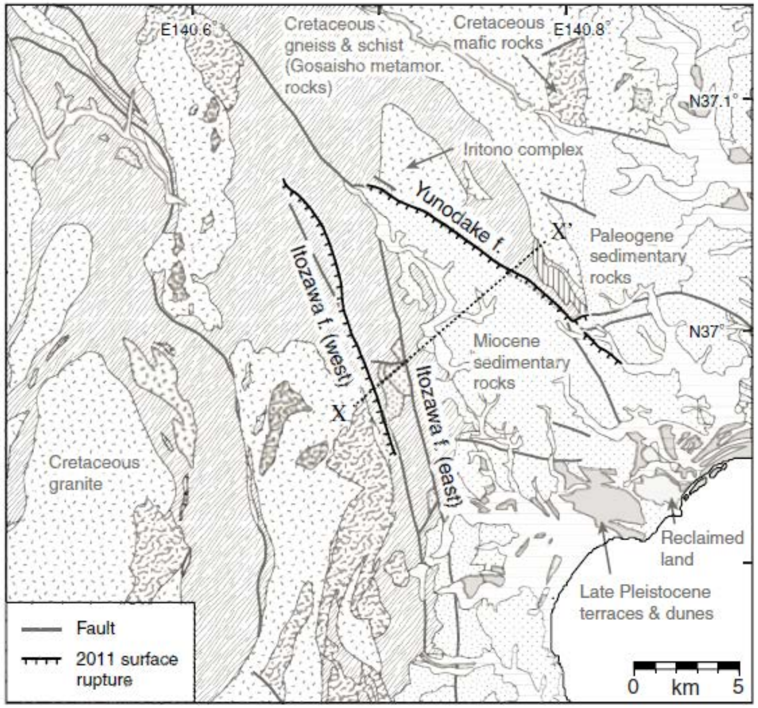
Nissen *et al.* (2012), *Geophys. Res. Lett.*

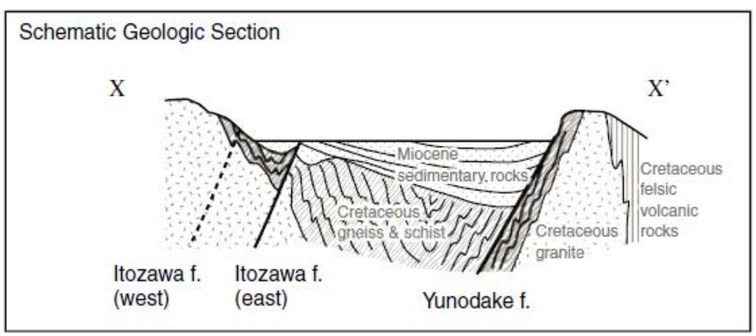
11 April 2011 Fukushima-Hamadori earthquake



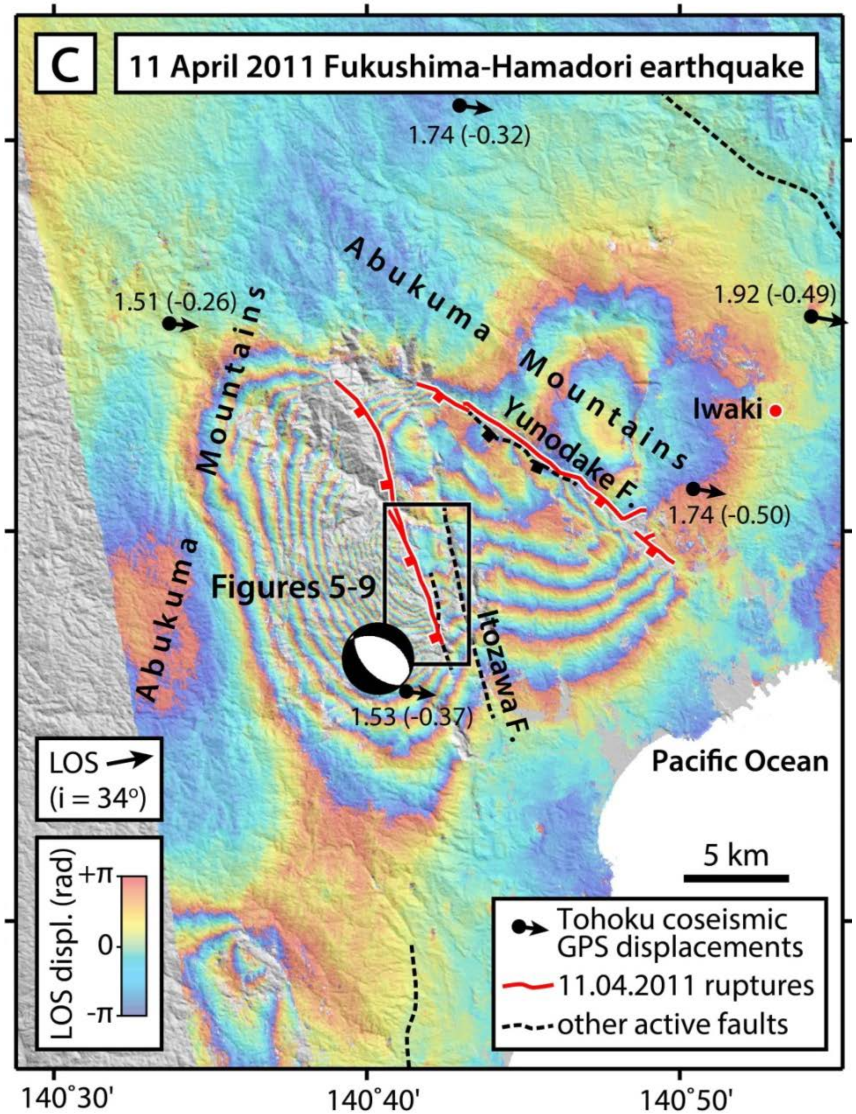
Nissen *et al.* (2014), *Earth Planet. Sci. Lett.*

11 April 2011 Fukushima-Hamadori earthquake

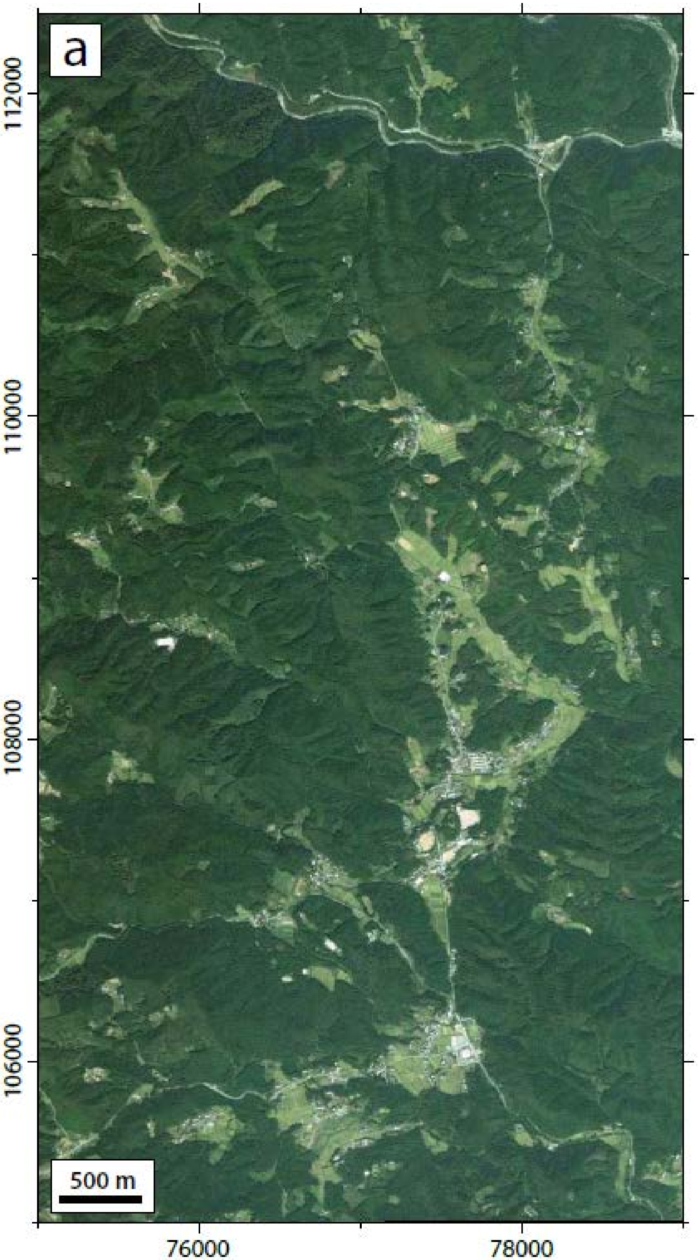


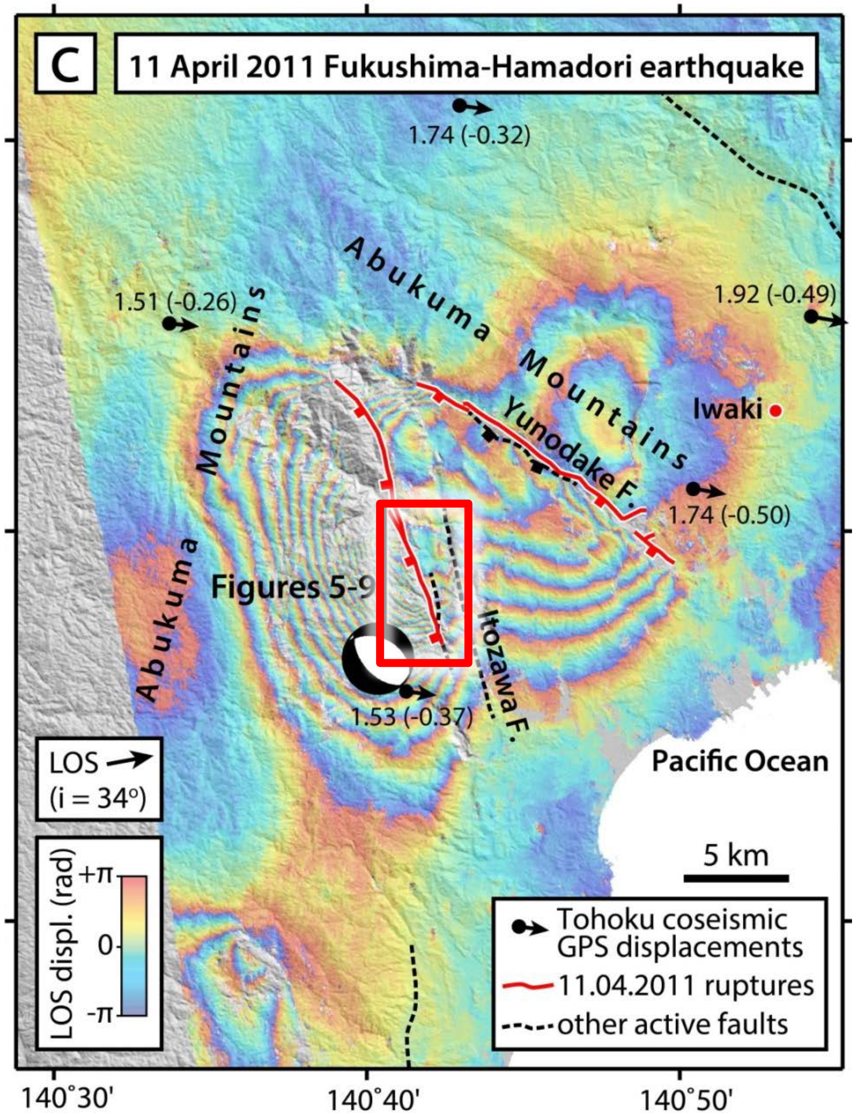


Toda & Tsutsumi (2013), *BSSA*

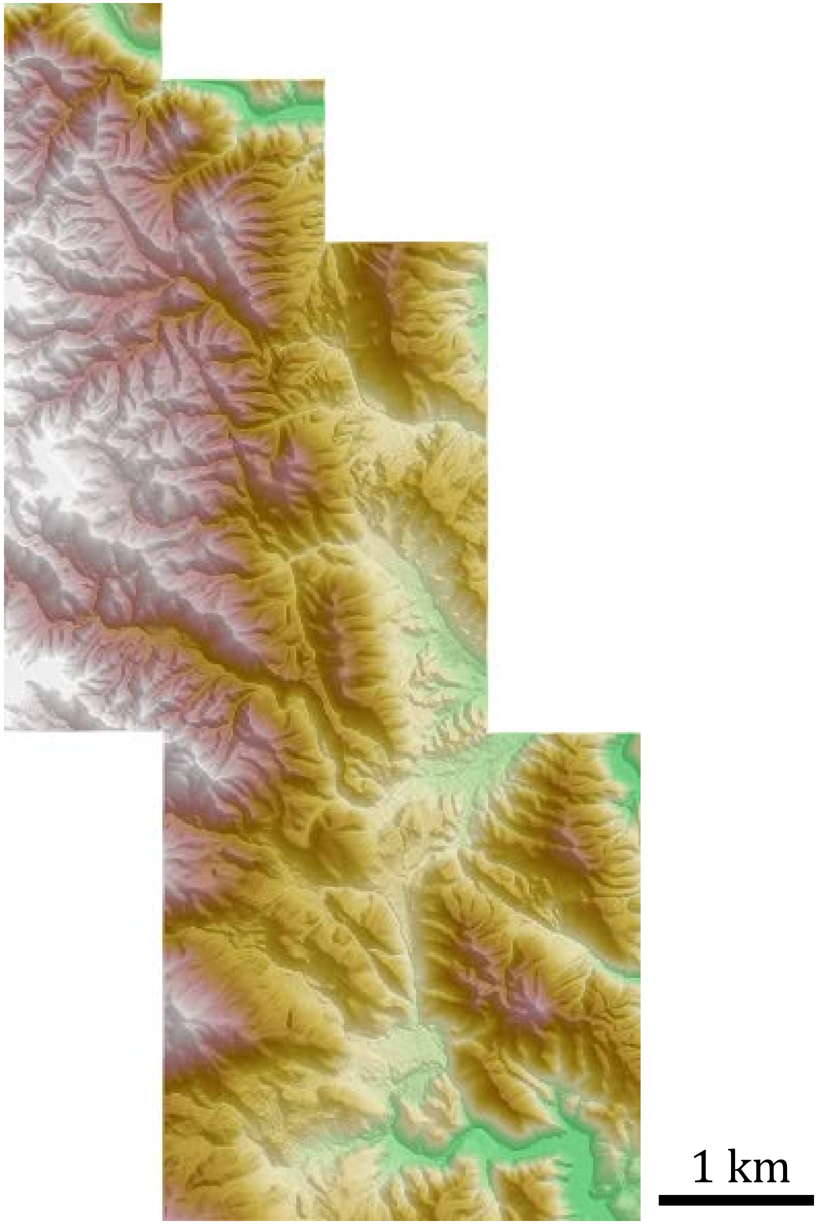


11 April 2011 Fukushima-Hamadori earthquake

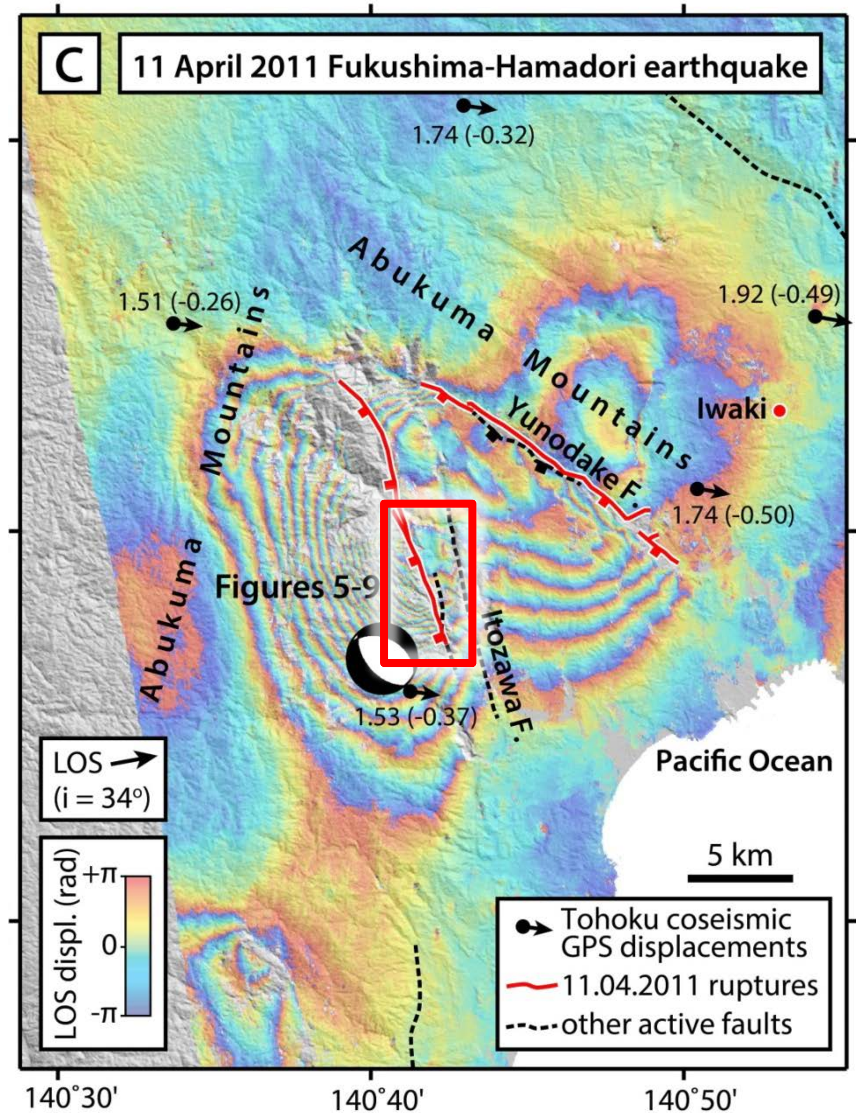




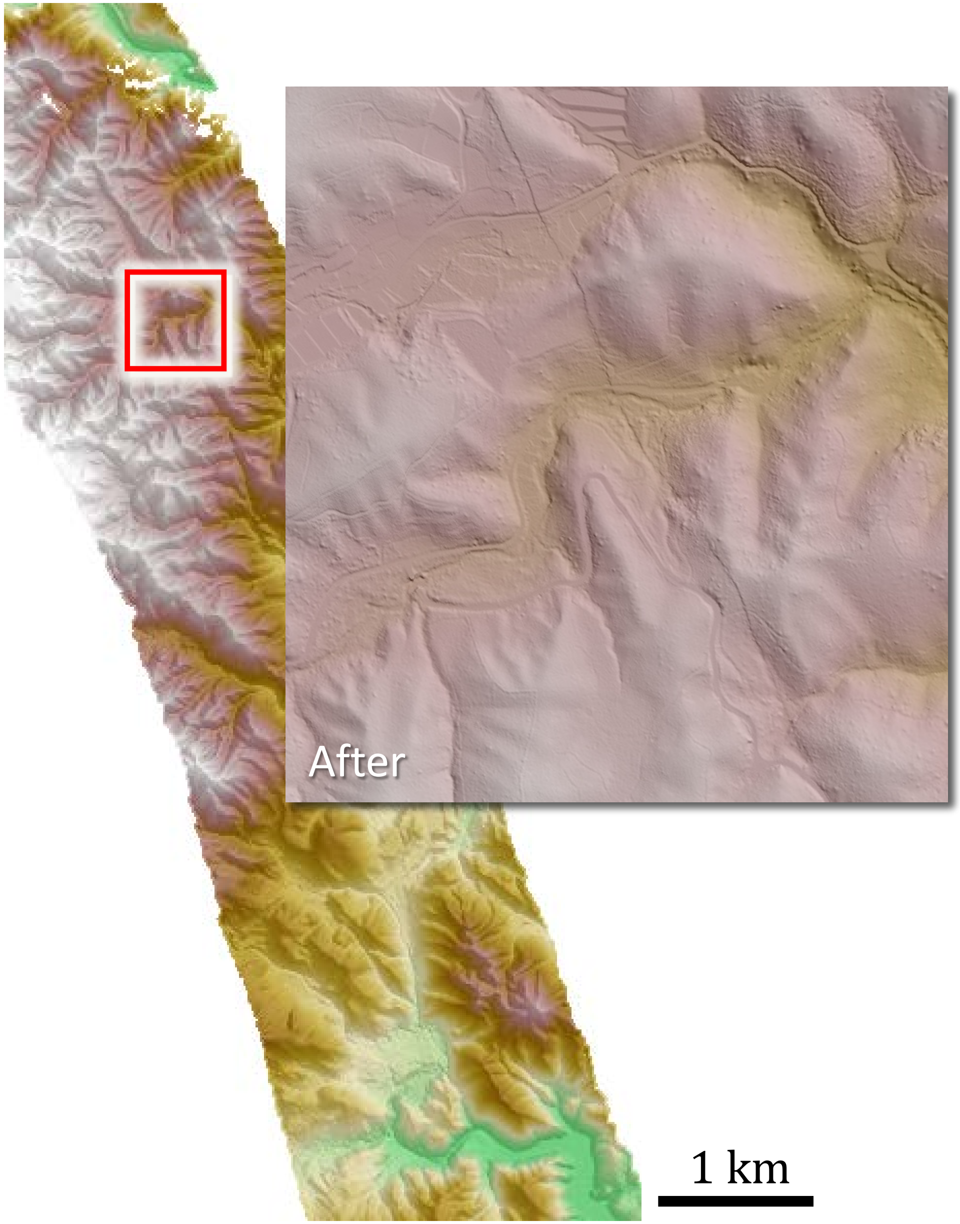
11 April 2011 Fukushima-Hamadori earthquake

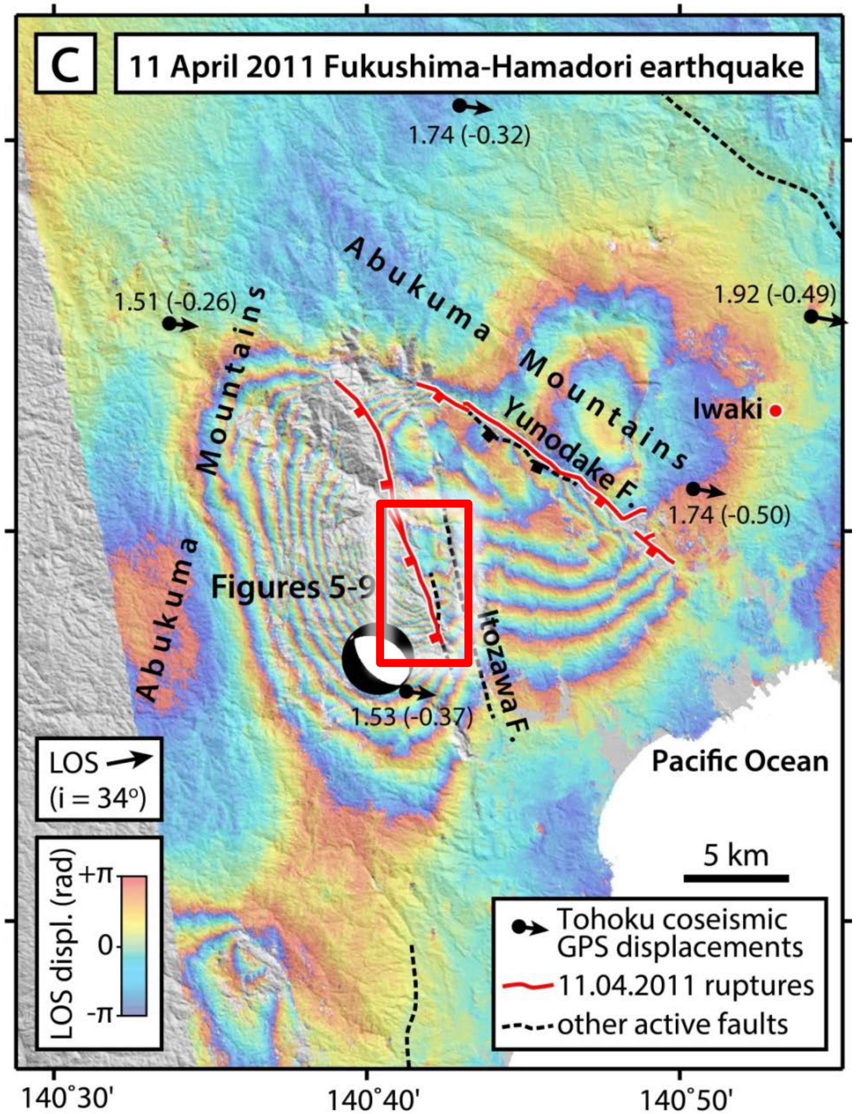


2006 pre-event 2 m DEM KOKUSAI KOGYO GROUP



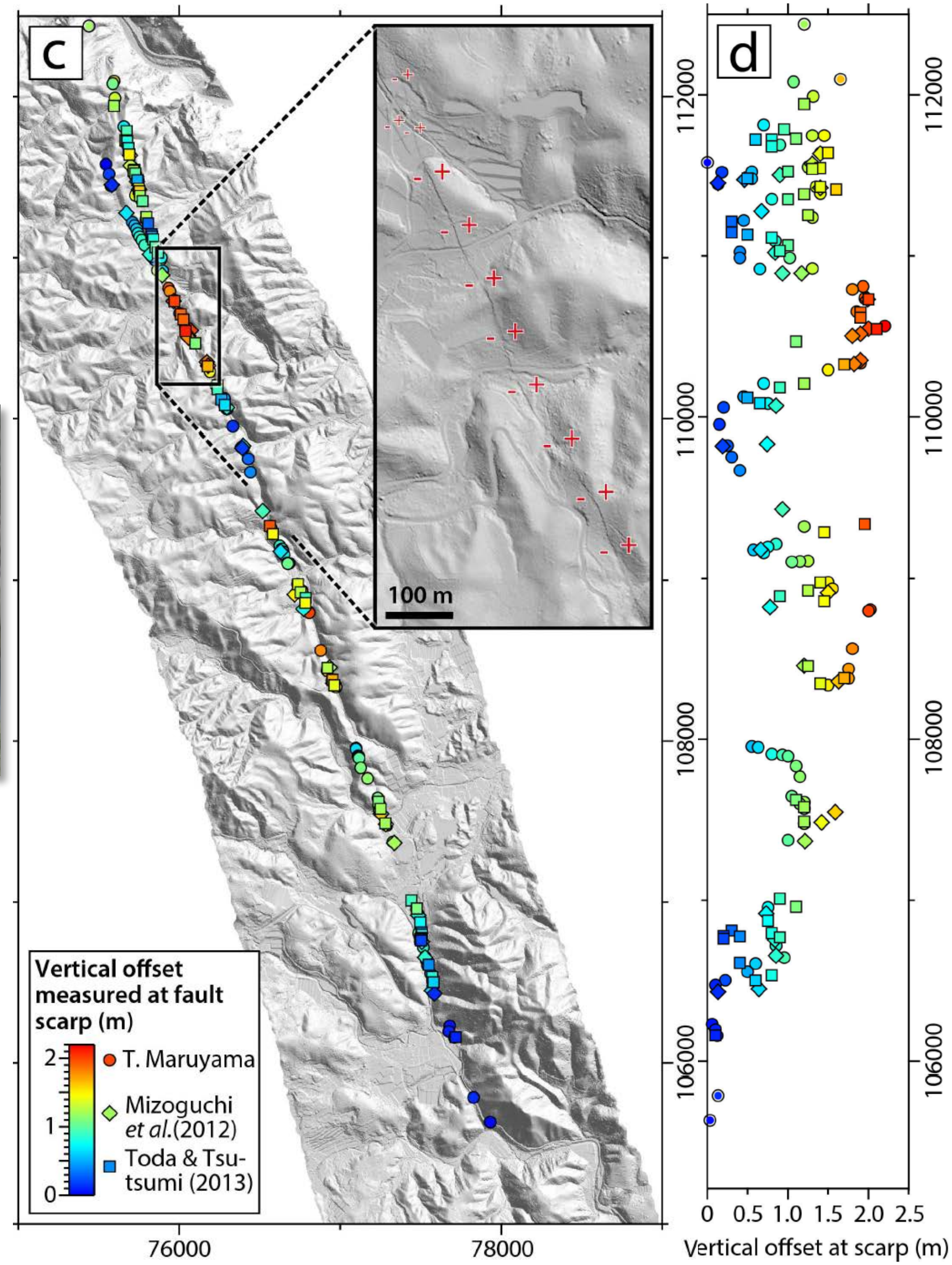
11 April 2011 Fukushima-Hamadori earthquake





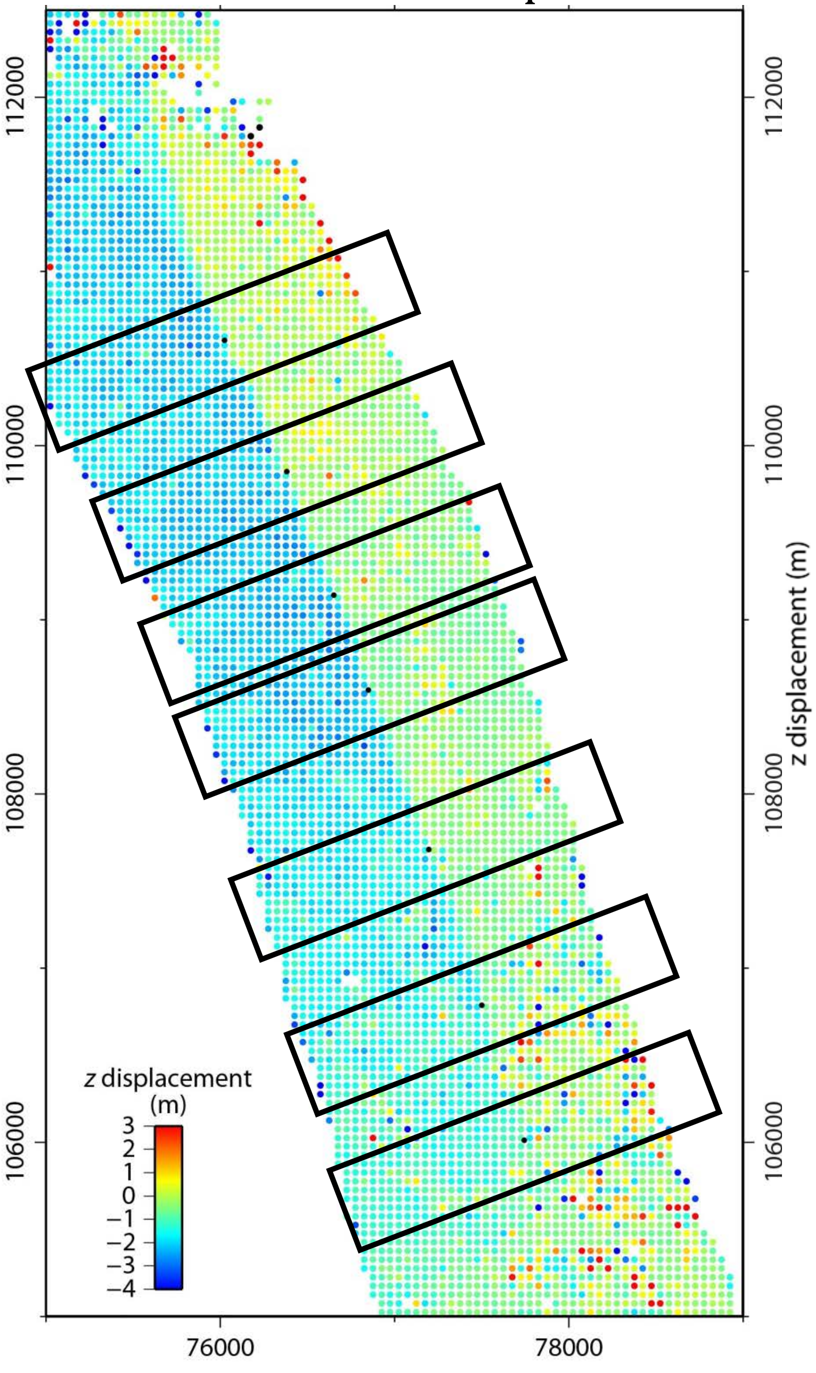
2011 post-event 1 m DEM AERO ASAHI CORPORATION

11 April 2011 Fukushima-Hamadori earthquake

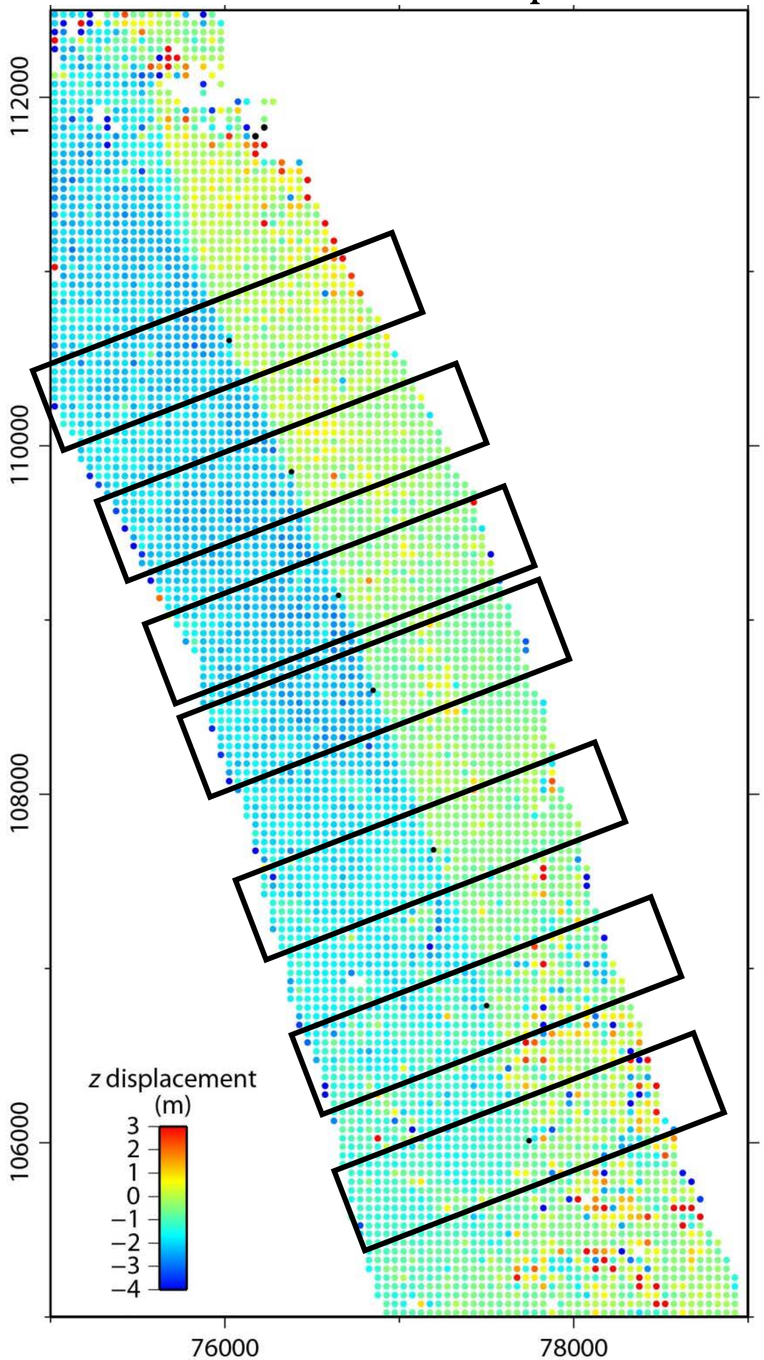


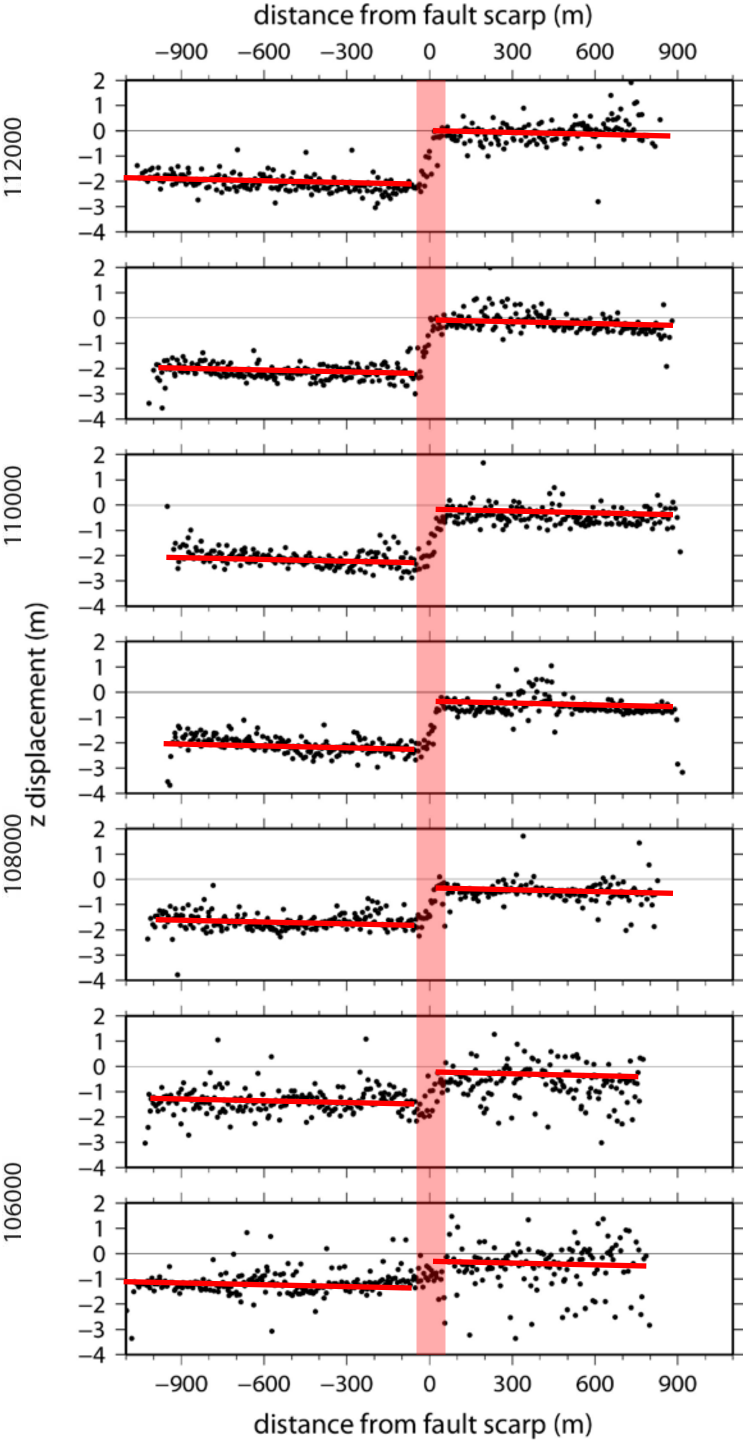
Photos from Toda & Tsutsumi (2013), *BSSA*

2005-2011 vertical displacements

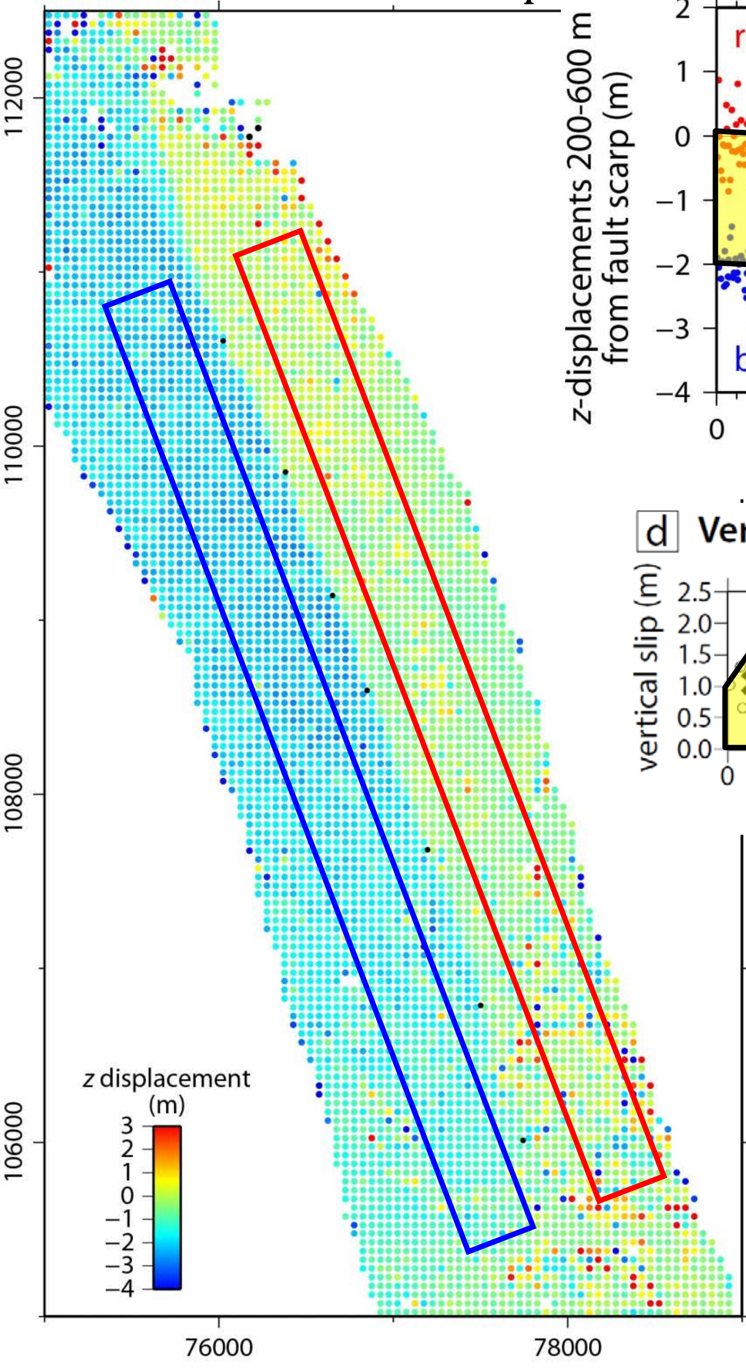


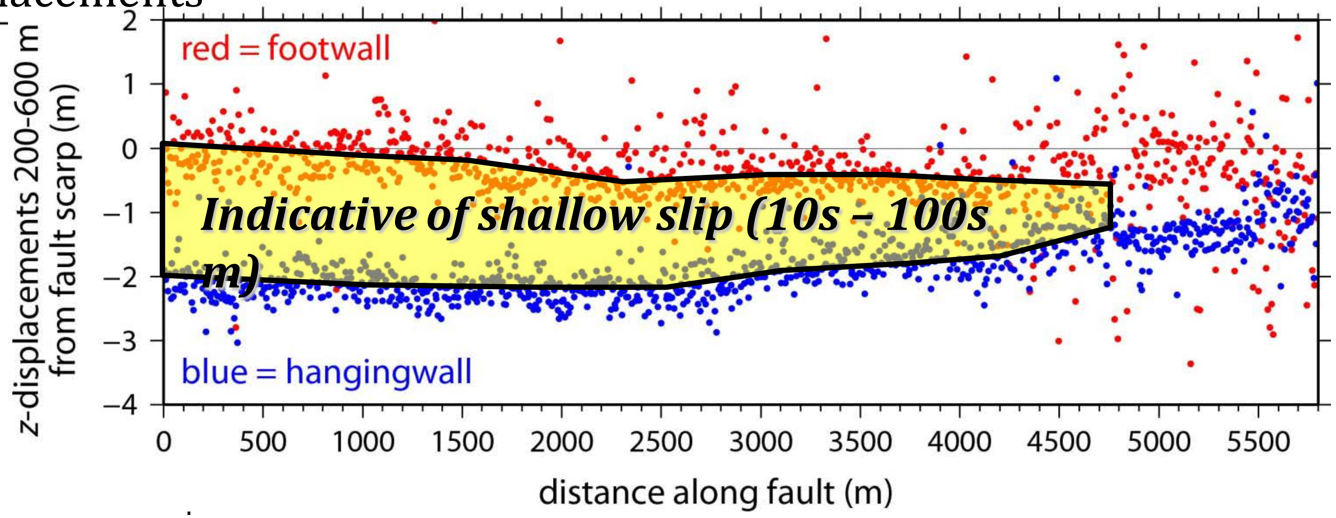
2005-2011 vertical displacements

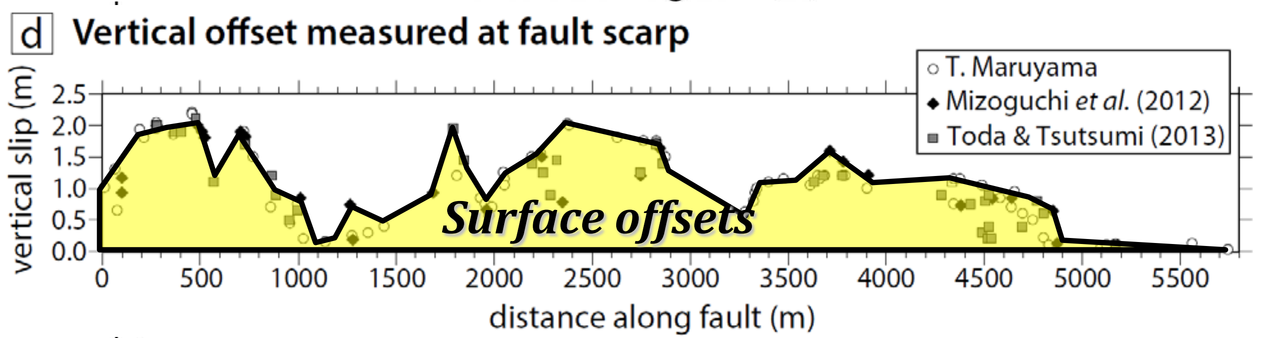




2005-2011 vertical displacements

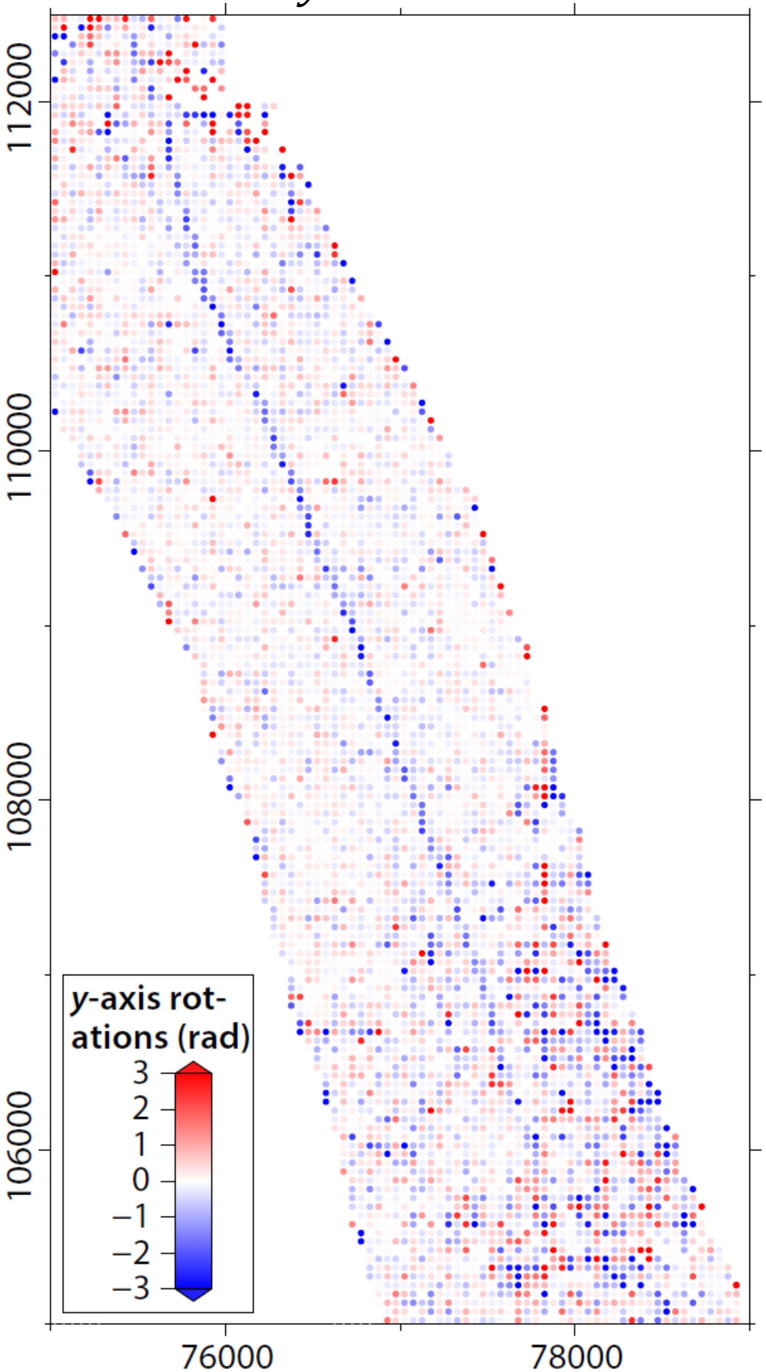


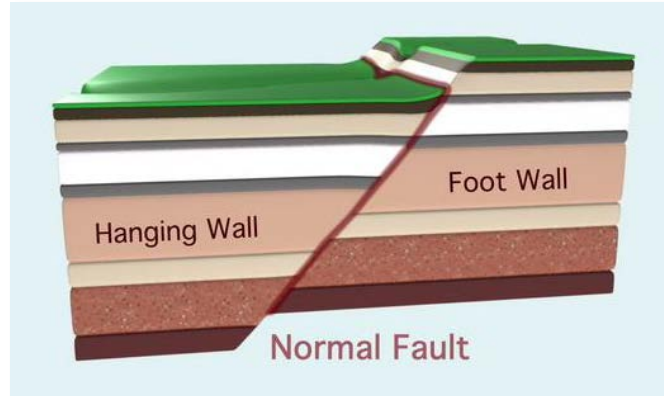


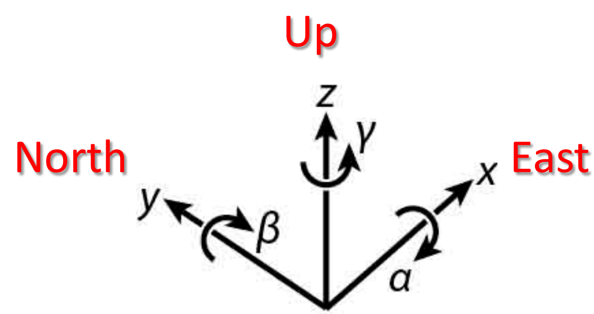


- Slip at depths of a few hundred meters appears to vary smoothly
- In many places, only a small proportion of the slip makes it to the surface

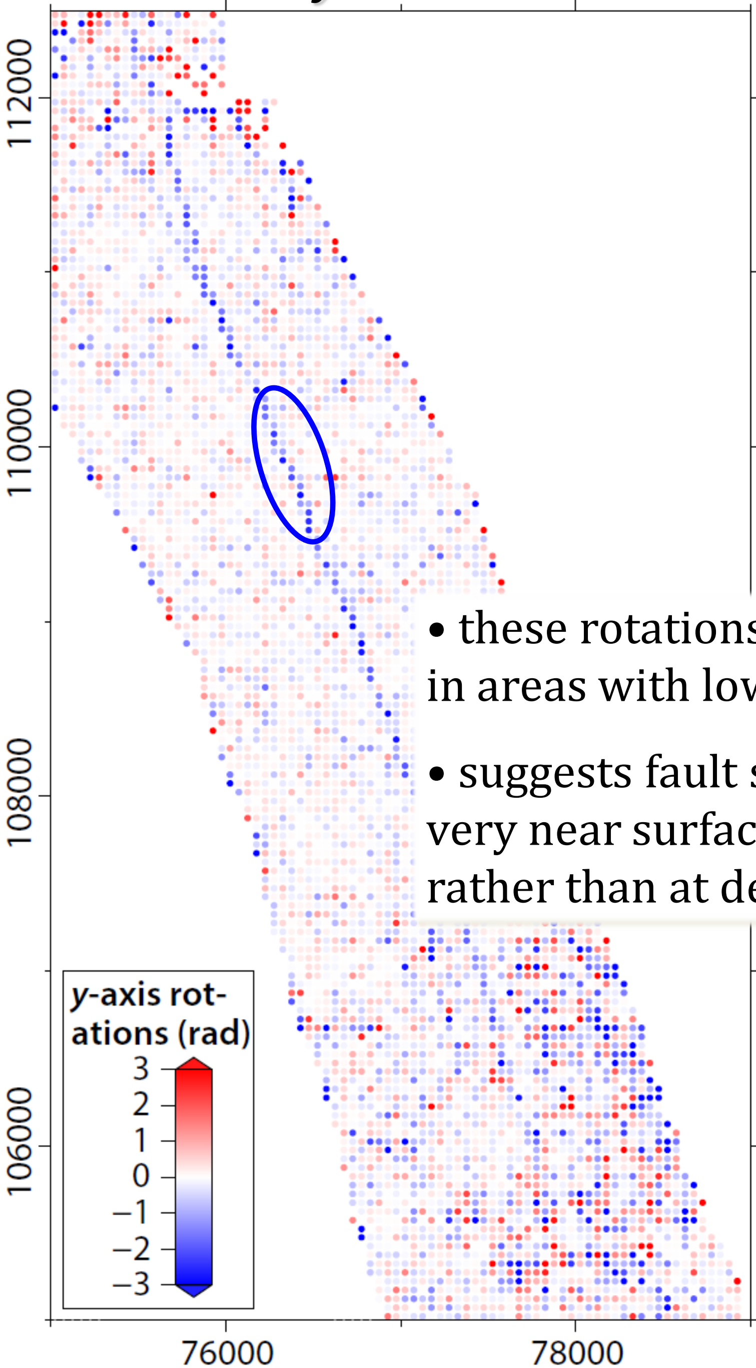
2005-2011 *y*-axis rotations

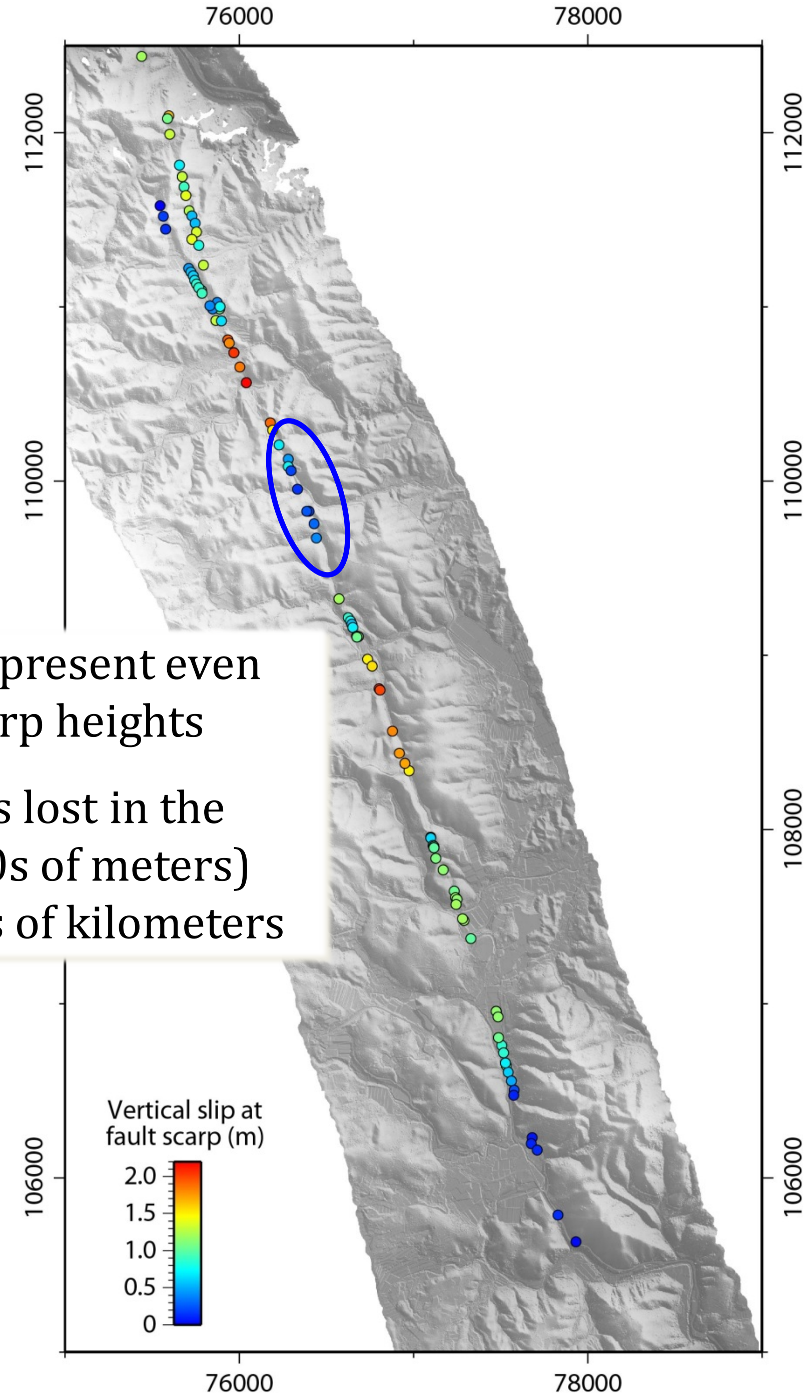






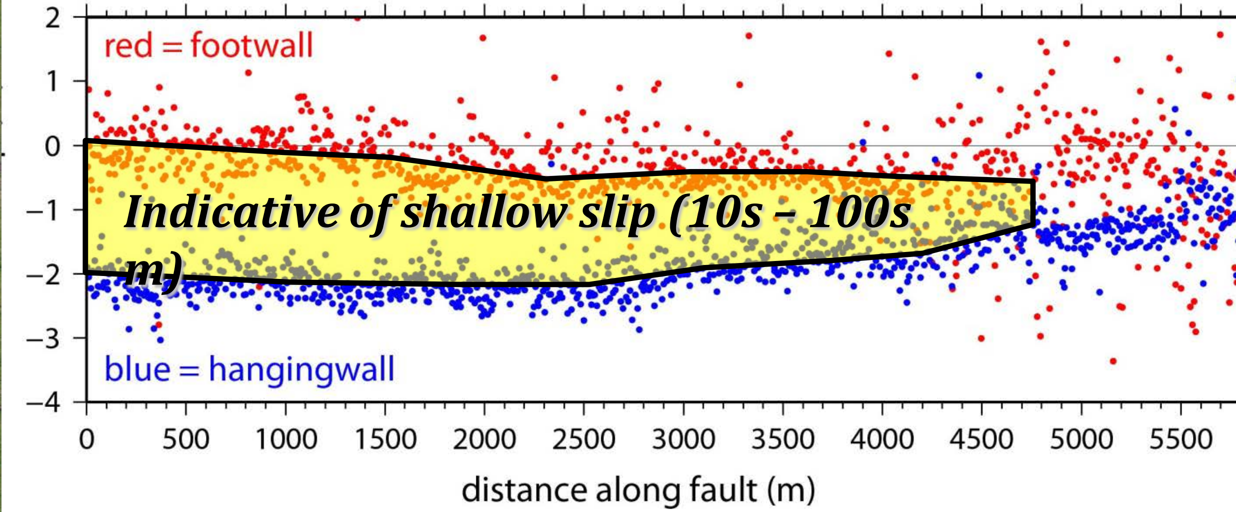
2005-2011 *y*-axis rotations

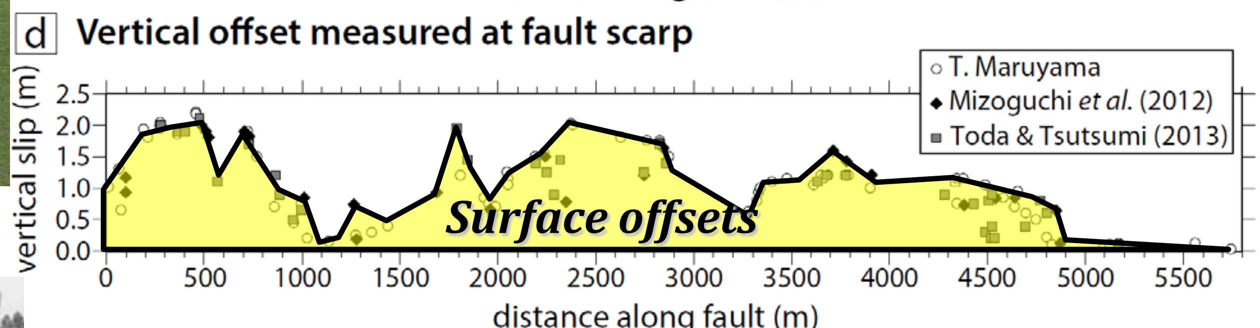




- these rotations are present even in areas with low scarp heights
- suggests fault slip is lost in the very near surface (10s of meters) rather than at depths of kilometers

Darfield rupture (Quigley *et al.* 2010)





Izmit rupture (Rockwell *et al.* 2002)

- Slip at depths of a few hundred meters appears to vary smoothly
- In many places, only a small proportion of the slip makes it to the surface
- Reflects off-fault deformation in the shallow subsurface?