Computer Vision ECE739
Feature Based Alignments and Image Stitching

By: Dónal Finnerty

Edited by: S. Shirani

Feature Based Alignment

•

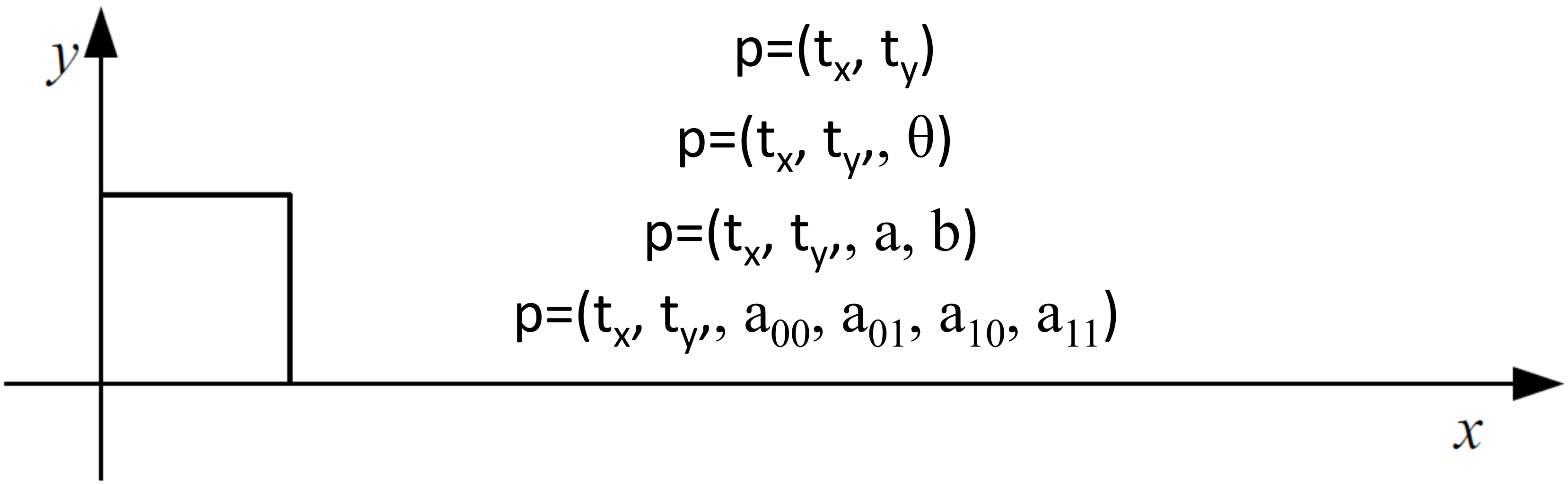
•

Chapter Contents

2D and 3D feature-based alignment

- – 2D alignment using least squares
- – Iterative algorithms
- – Robust least squares and RANSAC
- – 3D alignment
- Panography

2D Geometric Translations



p=(t_x, t_y)

p=(t_x, t_y,, θ)

p=(t_x, t_y,, a, b)

p=(t_x, t_y,, a_{00}, a_{01}, a_{10}, a_{11})

p=(h_{00}, h_{01}, h_{02}, h_{10}, h_{11}, h_{12} , h_{20}, h_{21}, h_{22})

[illegible]

Least Squares Method

$$E_{LS} = \sum_i \|r_i\|^2$$

r_i=f(x;p)-x’

Weighted Least Squares Method

$$E_{WLS} = \sum_i \frac{\|r_i{}^2\|}{\sigma^2}$$

r_i=f(x;p)-x’

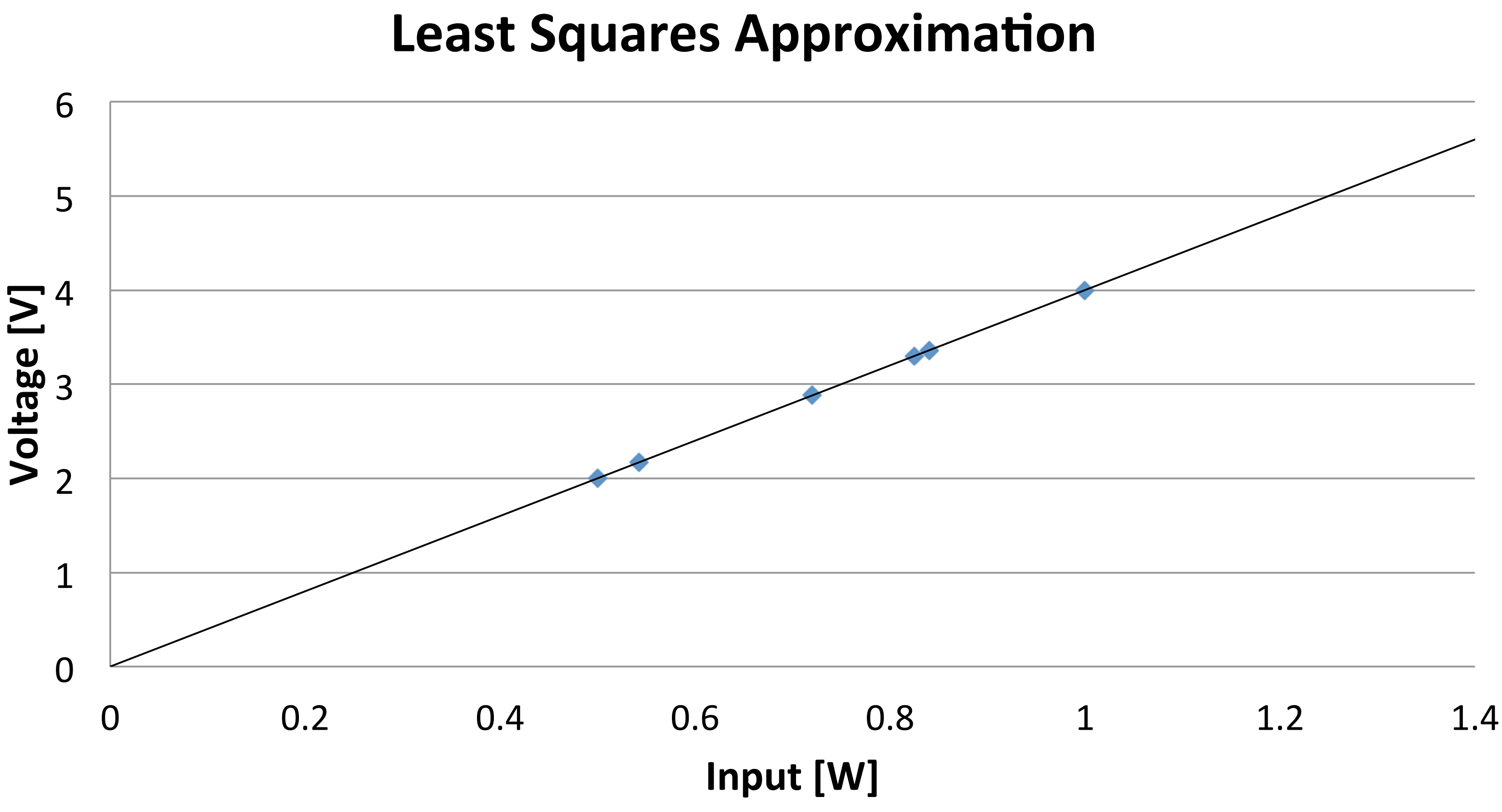
Linear Least Squares Method

- Many transformations have a linear relationship between the motion and the unknown parameters:
- $\Delta x = x - x' = J(x)p$
- $E_{LLS}(p) = \sum_i \| J(x_i)p - \Delta {x_i}' \|^2$
- $E_{LLS}(p) = p^T A p - 2p^T b + c$
- Minimum is found by solving : Ap=b
  - $A = \sum_i J^T(x_i) J(x_i)$
  - b= $\sum_i J^T(x_i) \Delta x_i$

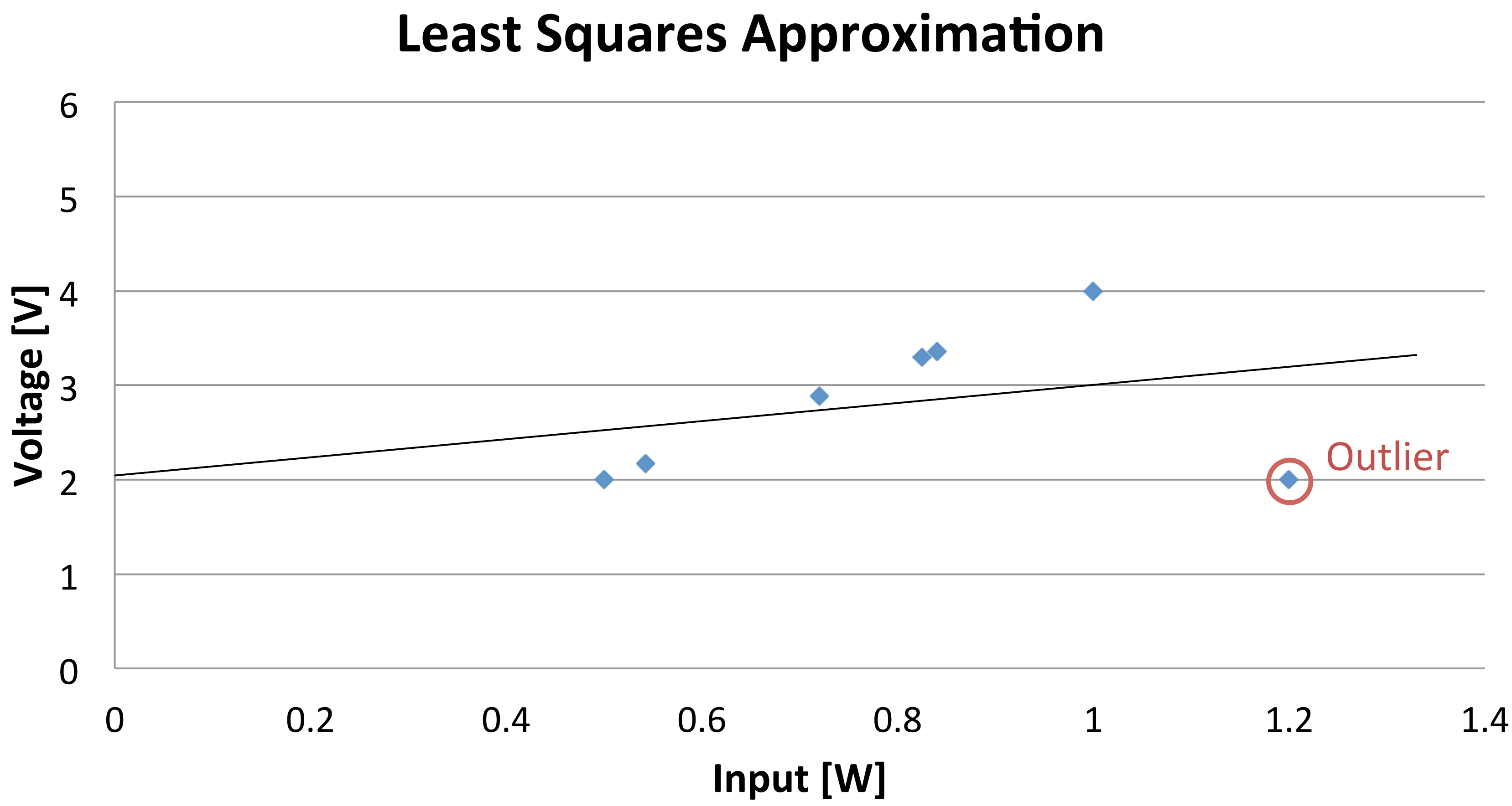
Non-Linear Least Squares Method

- With non-linear transformations we use iterative algorithms to solve for p
- We use the parameter Δp to update the parameters of the transformation.
- $p \leftarrow p + \Delta p$
- $E_{NLS}(\Delta p) = \sum_i \| f(x_i; p + \Delta p) - x_i' \|^2$
- $E_{NLS}(\Delta p) \approx \sum_i \| J(x_i; p)\Delta p - r_i \|^2$
- $E_{NLS}(\Delta p) = \Delta p^T A \Delta p - 2\Delta p^T b + c$
- Solve for Δp using $\big(A + \lambda diag(A)\big)\Delta p = b$

The Drawback of E_{LS}



The Drawback of E_{LS}



Robust Least Squares Method

- Use an “M-estimator” to reduce the negative influence of outliers.
- $E_{RLS}(\Delta p) = \sum_i \rho(\|r_i\|)$
- $\frac{\partial E_{RLS}(\Delta p)}{\partial p} = \sum_i \psi(\|r_i\|) \frac{\partial \|r_i\|}{\partial p} = 0$
- $w(\|r_i\|) = \frac{\psi(\|r_i\|)}{r_i} = \frac{\rho\prime(\|r_i\|)}{r_i}$
- $E_{IRLS}(\Delta p) = \sum_i w(\|r_i\|) \|r_i\|^2$
- w ← r ← p ← **w** ← **r** ← **p**

RANdom SAmple Concensus
RANSAC

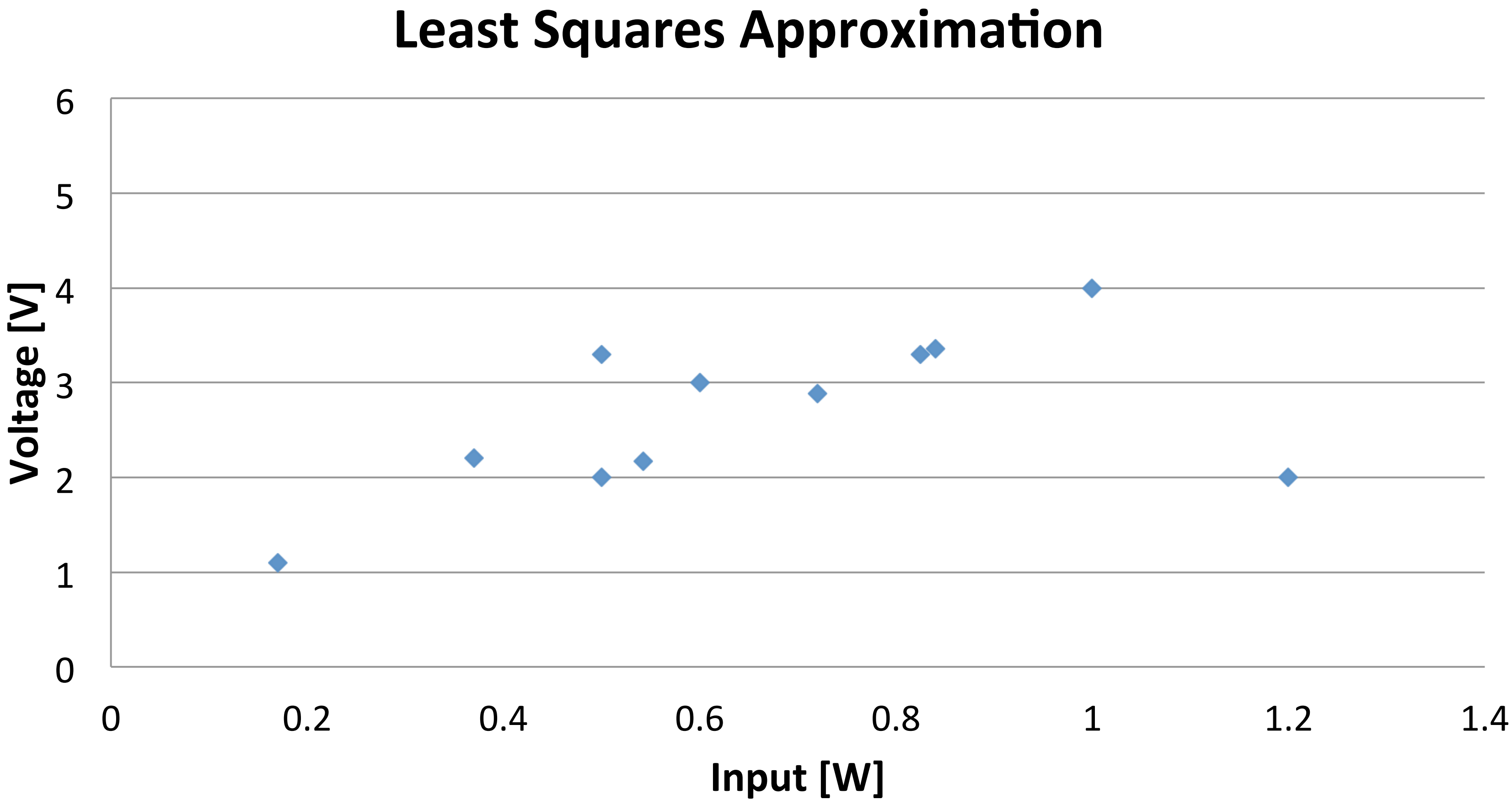
for i=1:1:S

1. Randomally select a subset of the data
2. Compute the transformation from this subset
3. Count the number of “inliers”
4. If the number of “inliers” is “sufficiently” large recalculate the transformation including “inliers”.

end

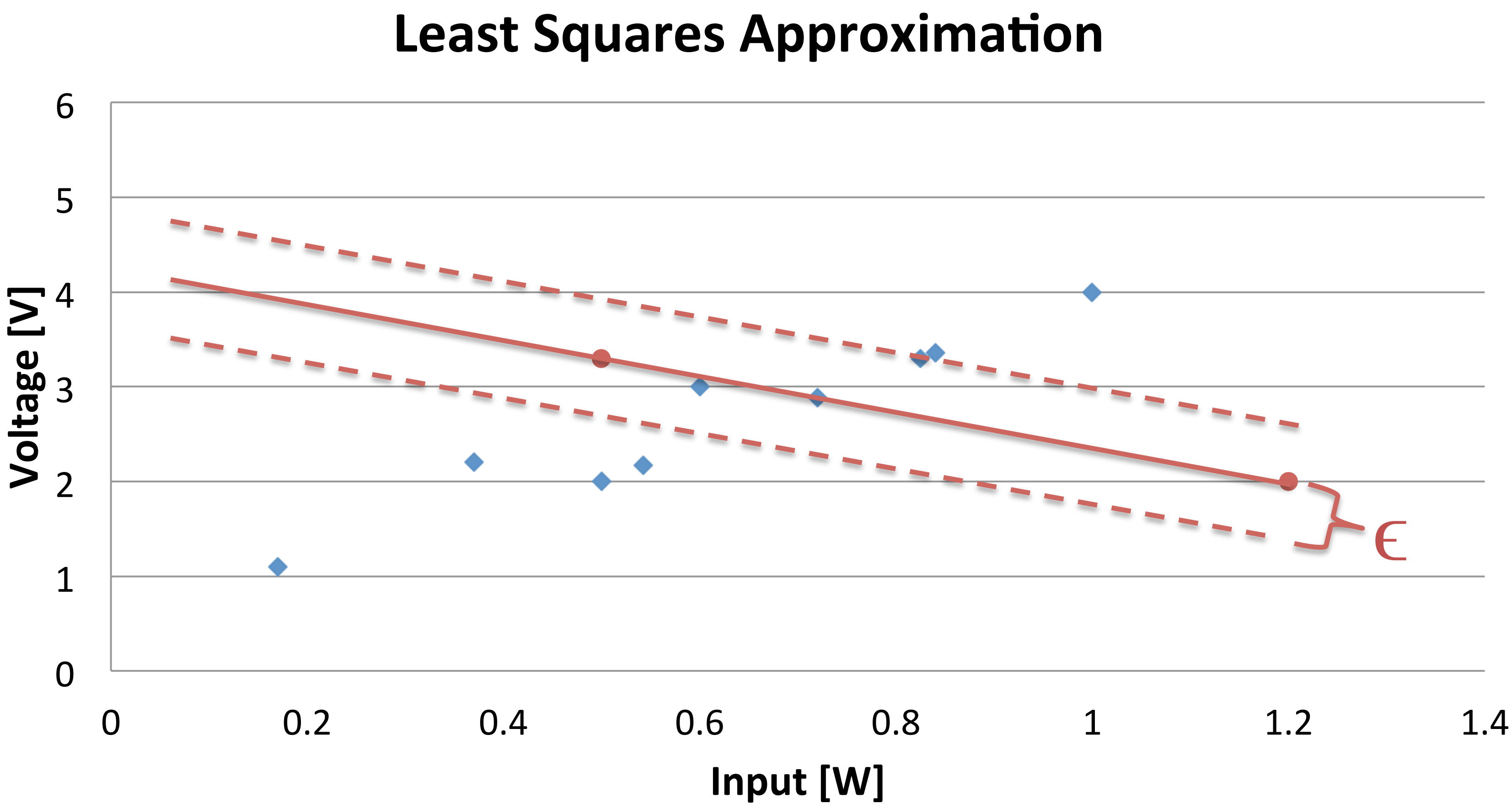
RANdom SAmple Concensus

RANSAC



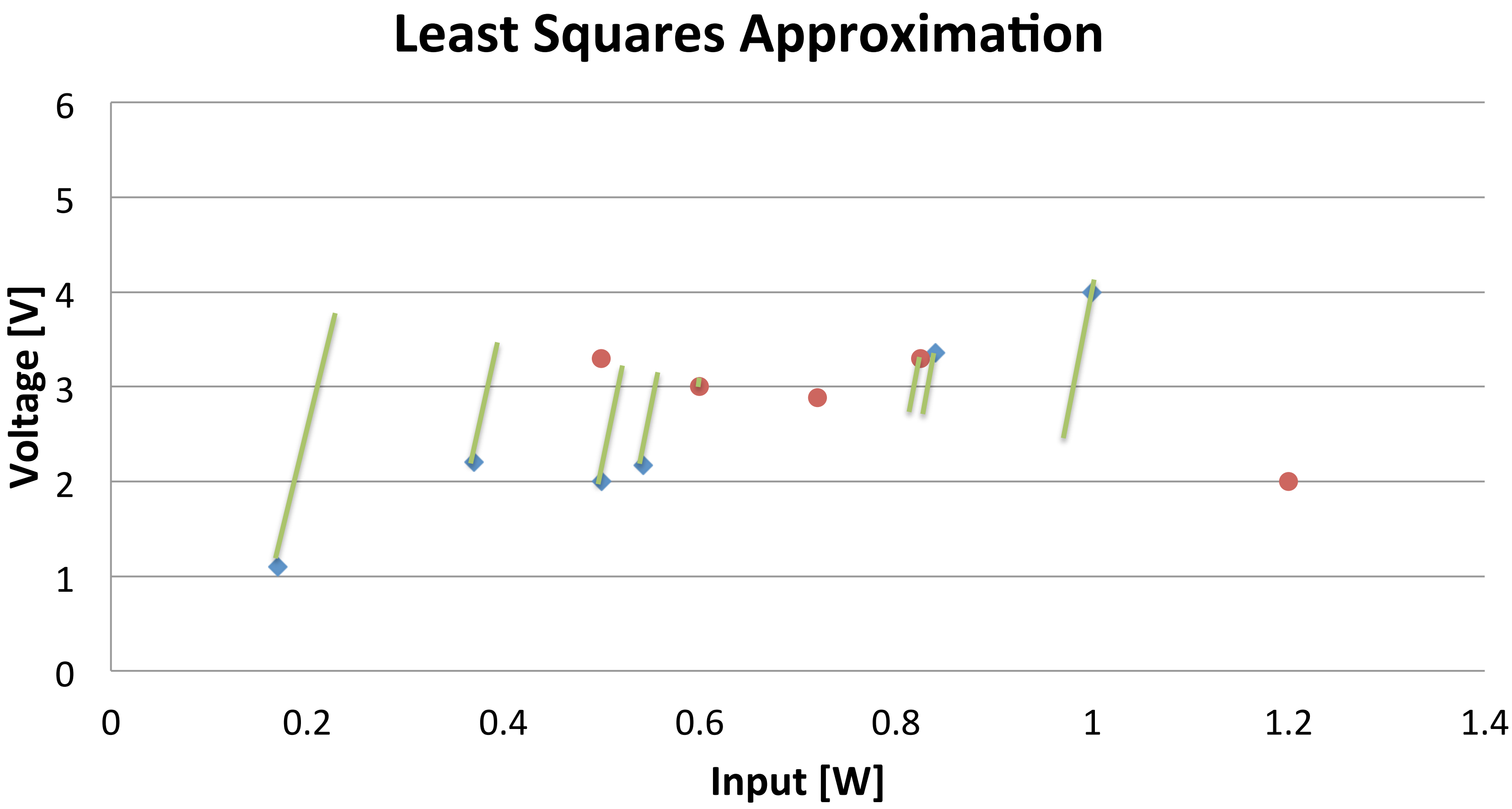
RANdom SAmple Concensus

RANSAC



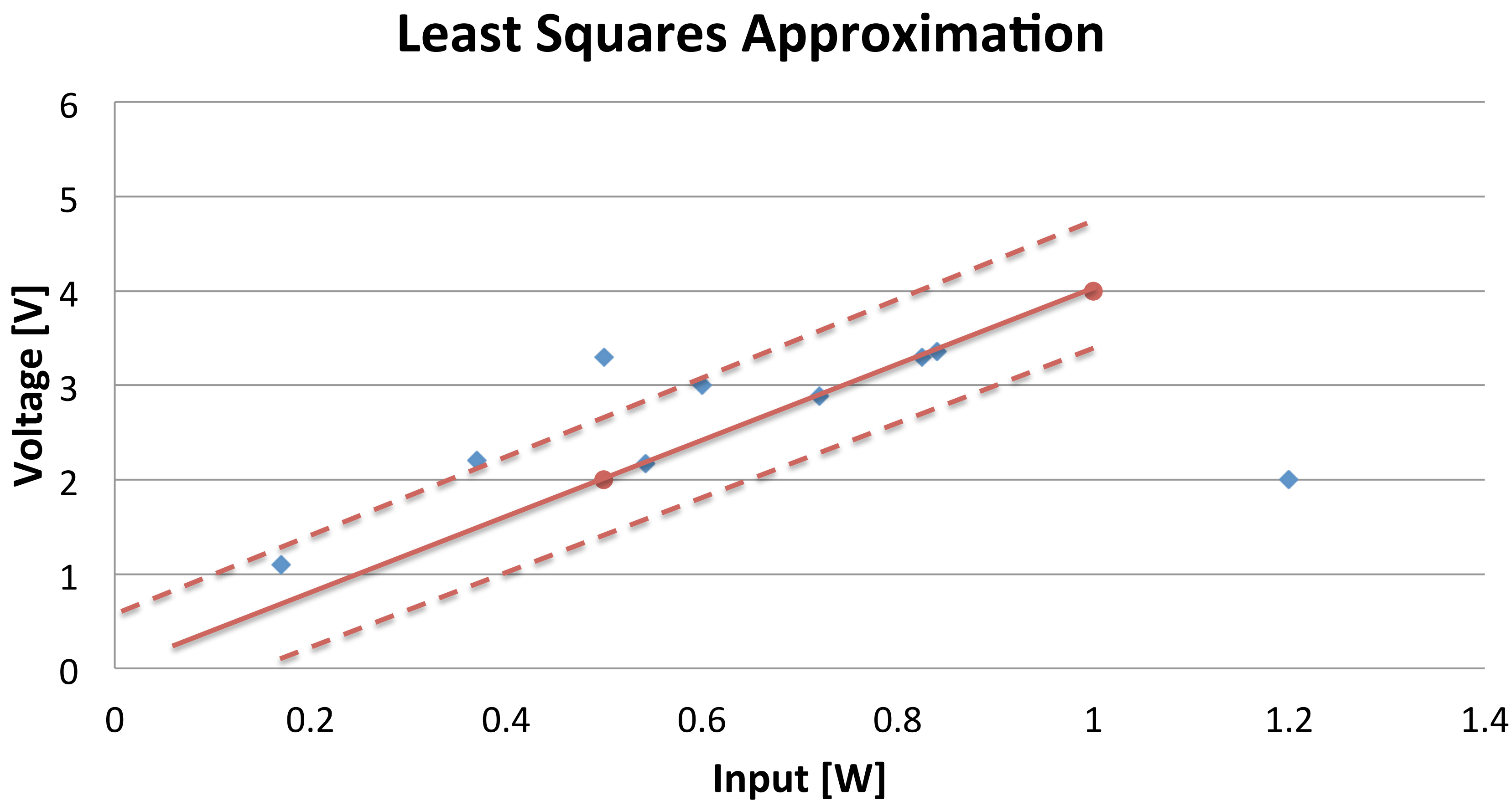
RANdom SAmple Concensus

RANSAC



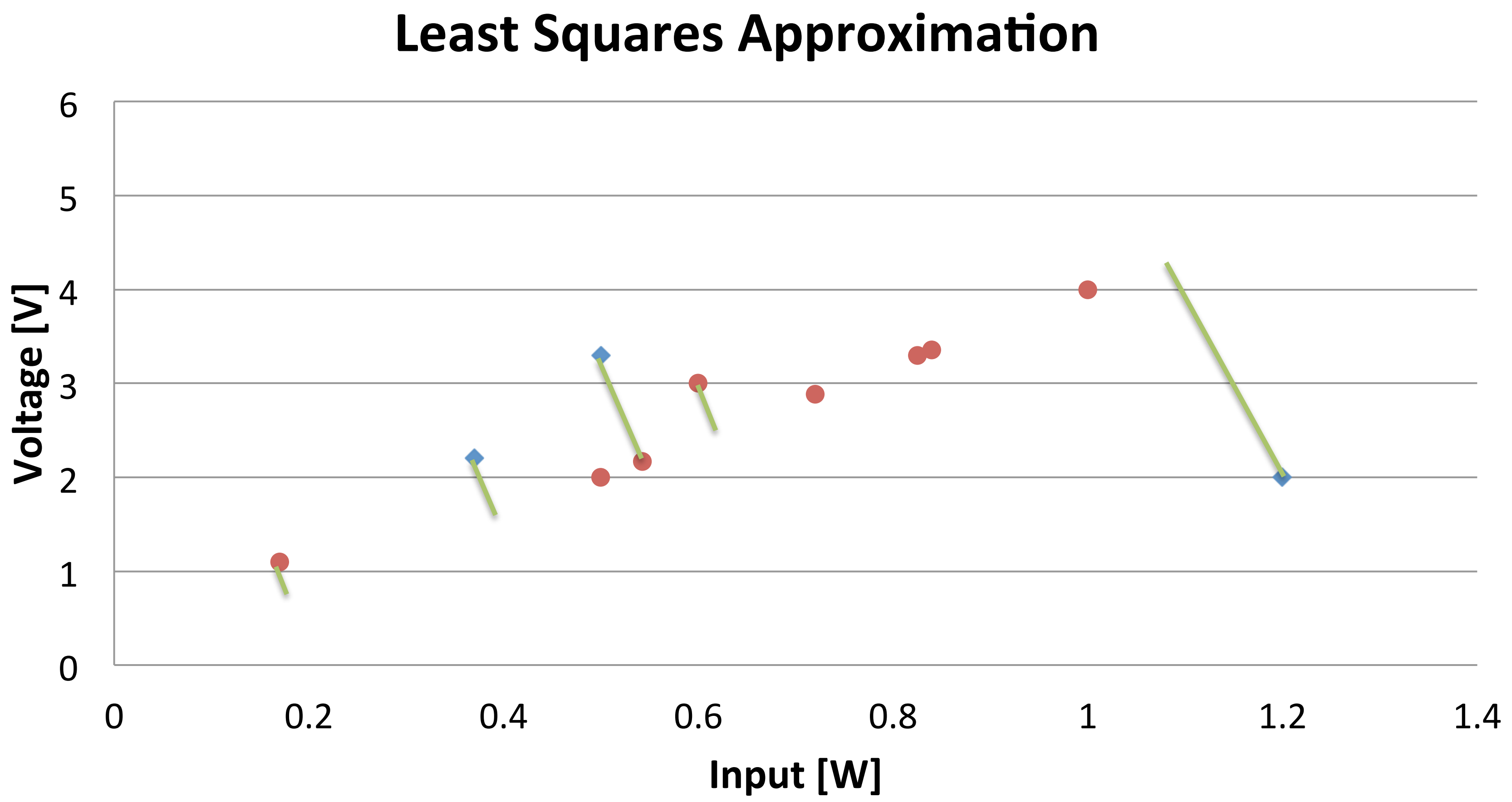
RANdom SAmple Concensus

RANSAC



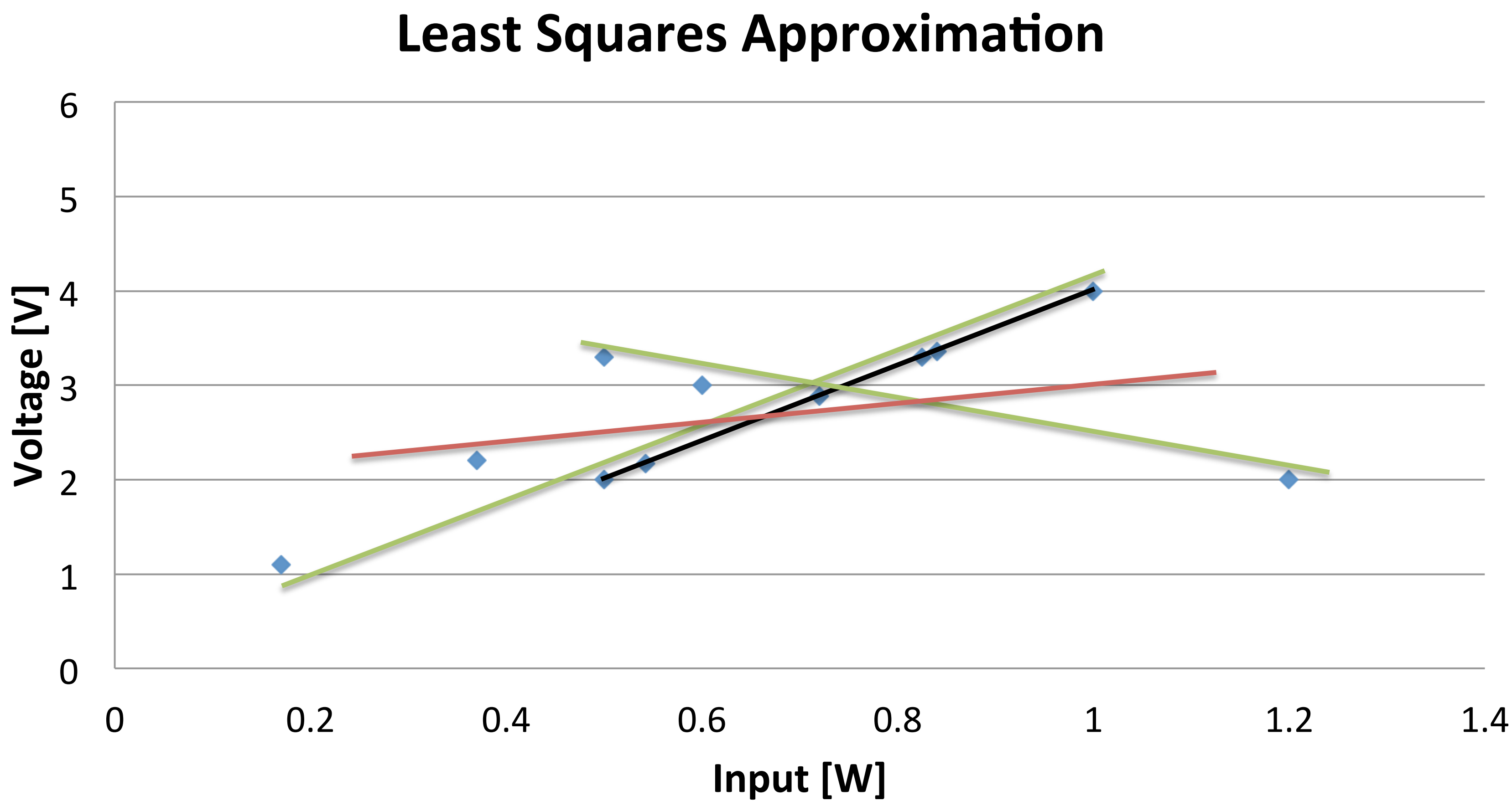
RANdom SAmple Concensus

RANSAC



RANdom SAmple Concensus

RANSAC



Problems with RANSAC

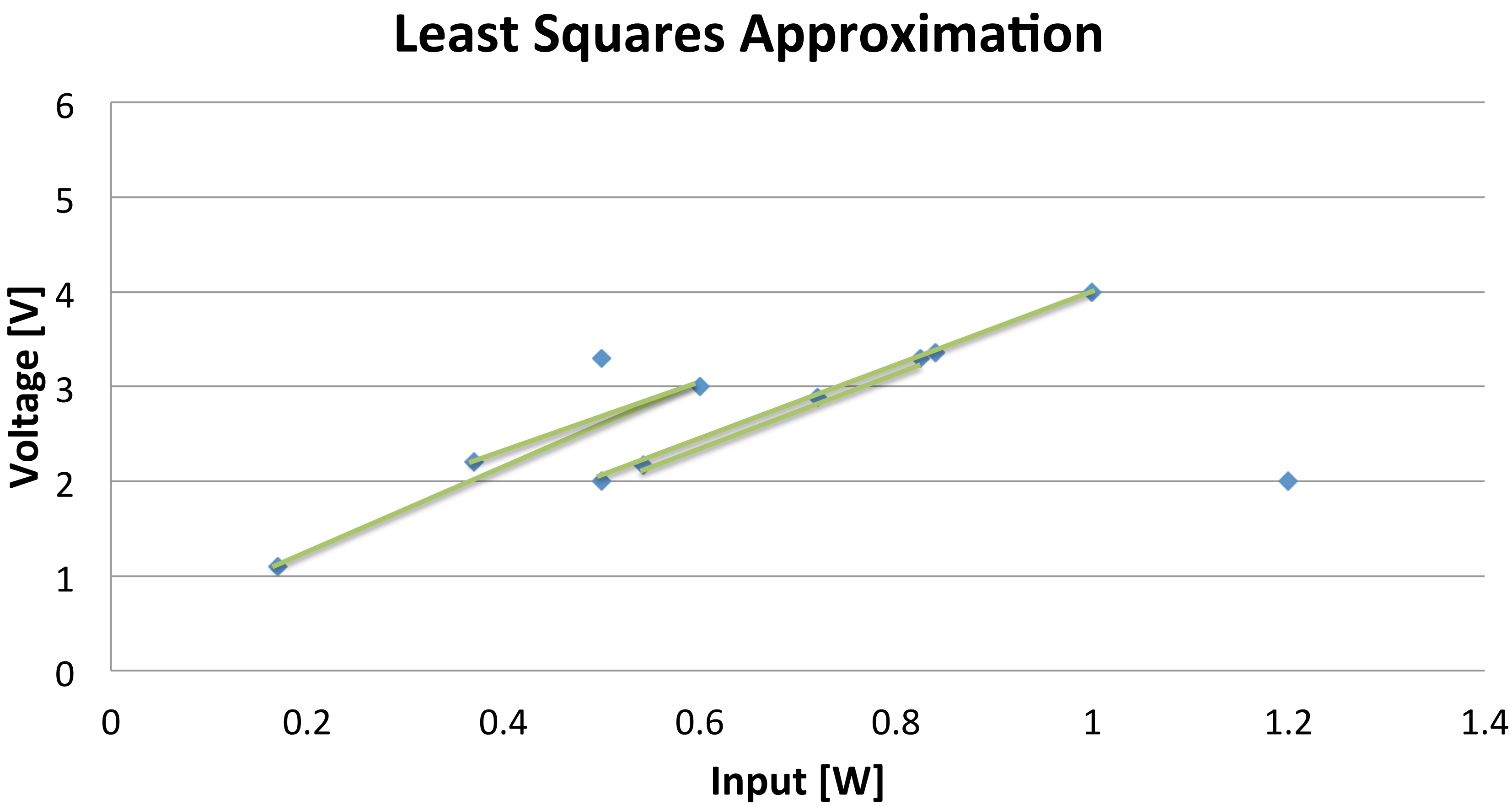
- k: the amount of samples initially taken for the data subset
- p: the probability that a randomly chosen sample is an "inlier" to its own transform.
- S the number of times to iterate RANSAC for a 99% probability of success

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

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

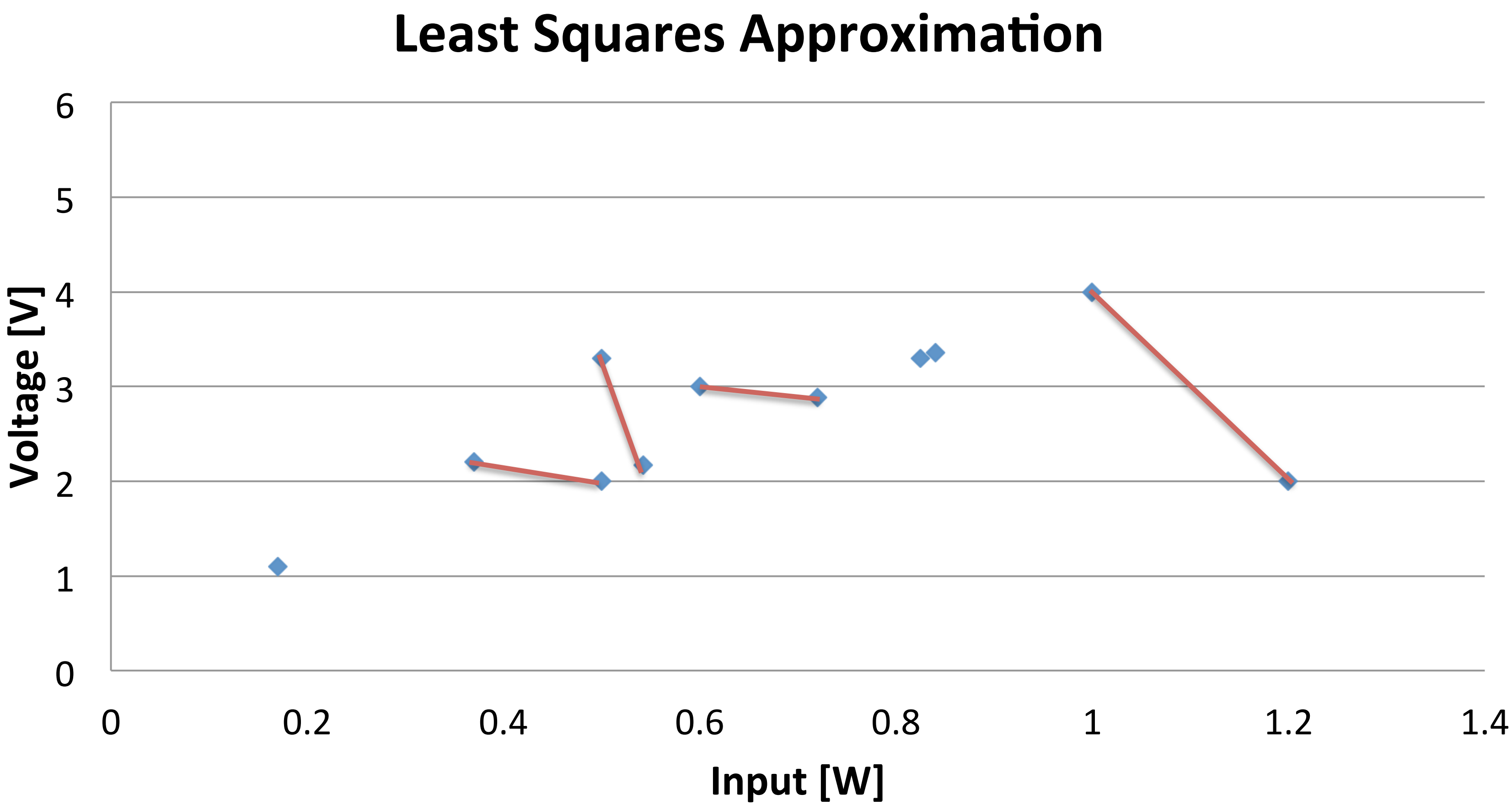
PROgressive SAmple Concensus

PROSAC



PROgressive SAmple Concensus

PROSAC



PROgressive SAmple Concensus
PROSAC

- The initial subset of data is chosen in a “semi-random” process.

Samples	RANSAC	PROSAC
Average	106,534	9
Time	10.76 [s]	0.06 [s]
Min	97,702	5
Max	146,069	29

Panography

- Three images translated together and averaged

[Szeliski]

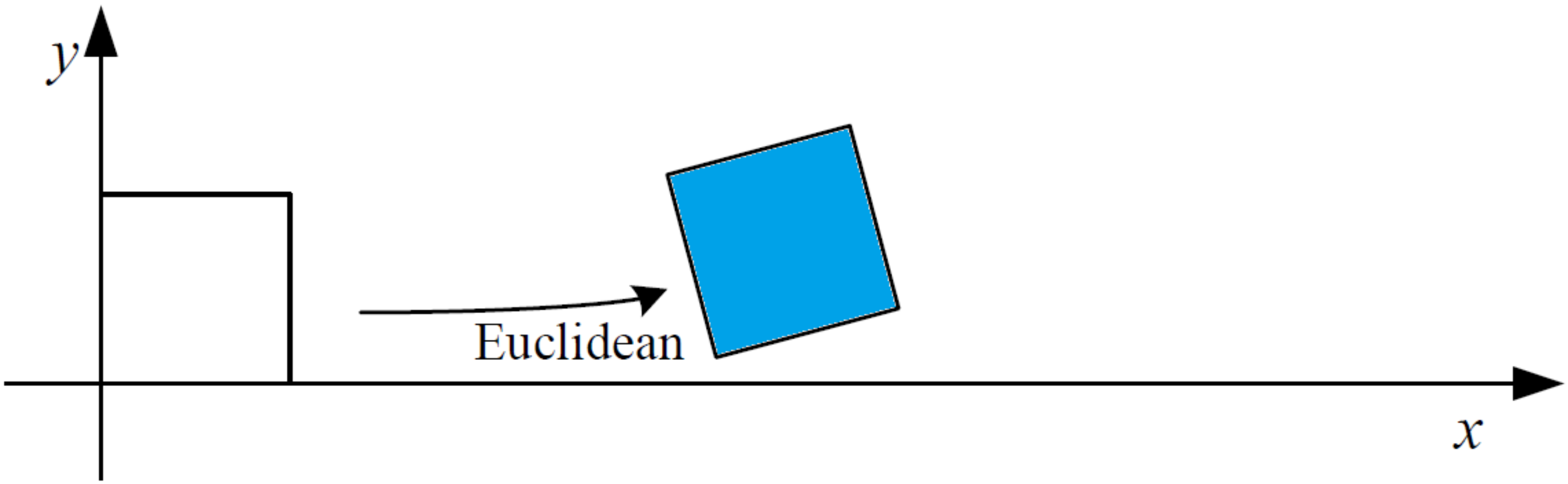
Panography

Francisco Diez

3D Alignment

- Aligning 3D points instead of 2D features.
- "The biggest difference between 2D and 3D coordinate transformations is that the parameterization of the 3D rotation matrix is not as straightforward."

Rigid Euclidian Motion in 3D



$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

p=(t_x, t_y,, θ)

Rigid Euclidian Motion in 3D

- Translation component can be estimated from the difference in centroids.
  - Orthogonal Procrustes algorithm

    Perform a SVD on the corrolation matrix

    $$C = \sum_i \hat{x}' \hat{x}^T = \boldsymbol{U \Sigma V^T}$$

    The Rotation matrix $\boldsymbol{R = UV^T}$
  - Absolute Orientation Algorithm

    Estimates the Unit quarturnion associated with **R**

    Convert C into a 4x4 matrix and find the eigenvector associated with the largest eigenvalue