Highly efficient Machine learning for HoloLens

Andrew Fitzgibbon, Microsoft

@awfidius

Microsoft


Lowe's Hologram Experience
In-Store Pilot Program


Samsung
cu ft
Refrigerator
Whirlpool
cu ft
Refrigerator

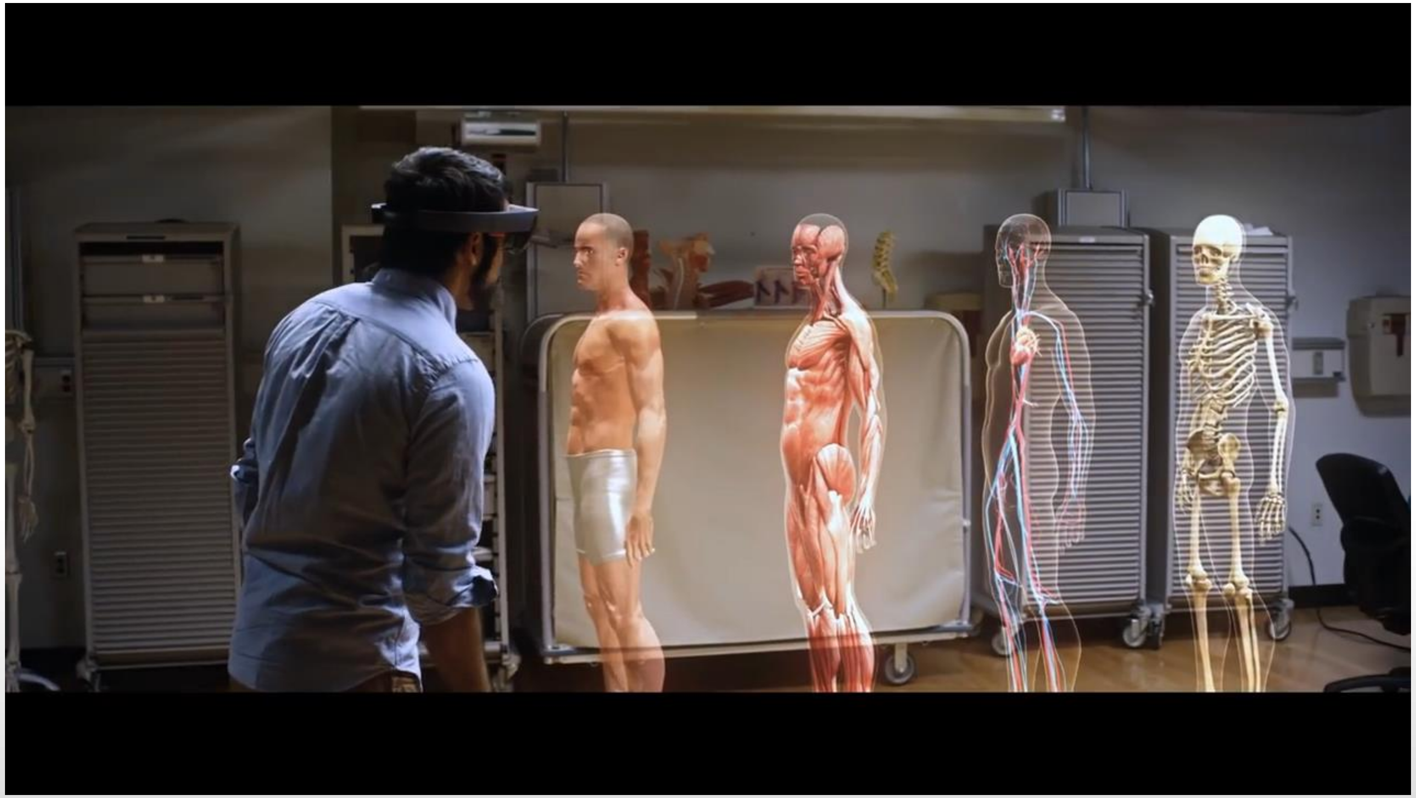
Applications of HoloLens

- Deskless workers
  - Merge real and digital world
- 3D designers / decision makers / learners
  - Create and communicate 3D concepts in 3D
- Everyone…

TASK WORKER


WELCOME JEFF
TAC 20 elevator
Video Tutorial
DANGER
Service Manuals
Location
San LaMonte Building
MAX Data

3D Learning / Medical



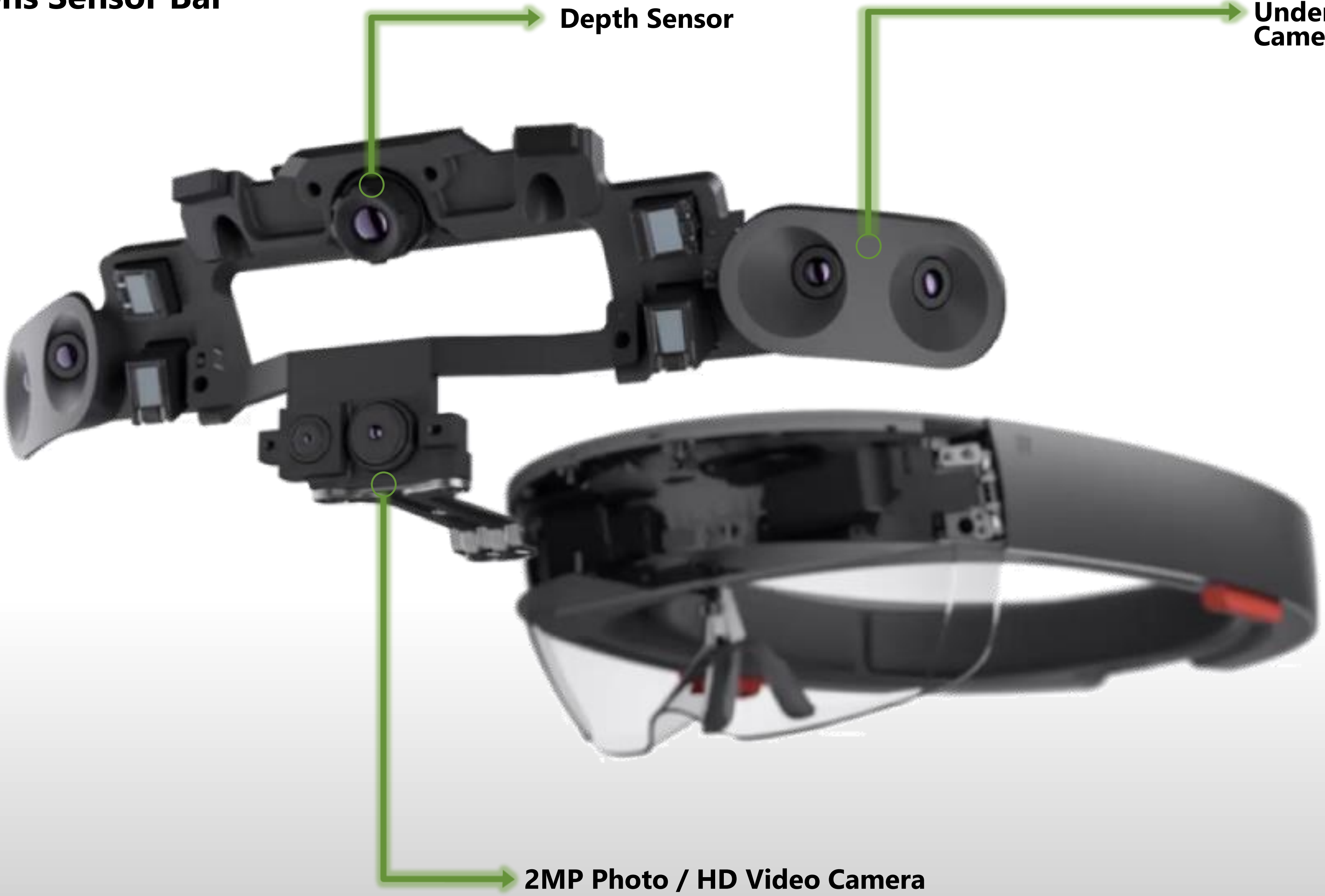
Features of HoloLens

- Fully self-contained computer
- Running Windows 10 Holographic
- Computer-vision based 3D localization
- Hand gesture recognition
- Onboard speech recognition
- Under power/thermal constraints

An Introduction to HoloLens

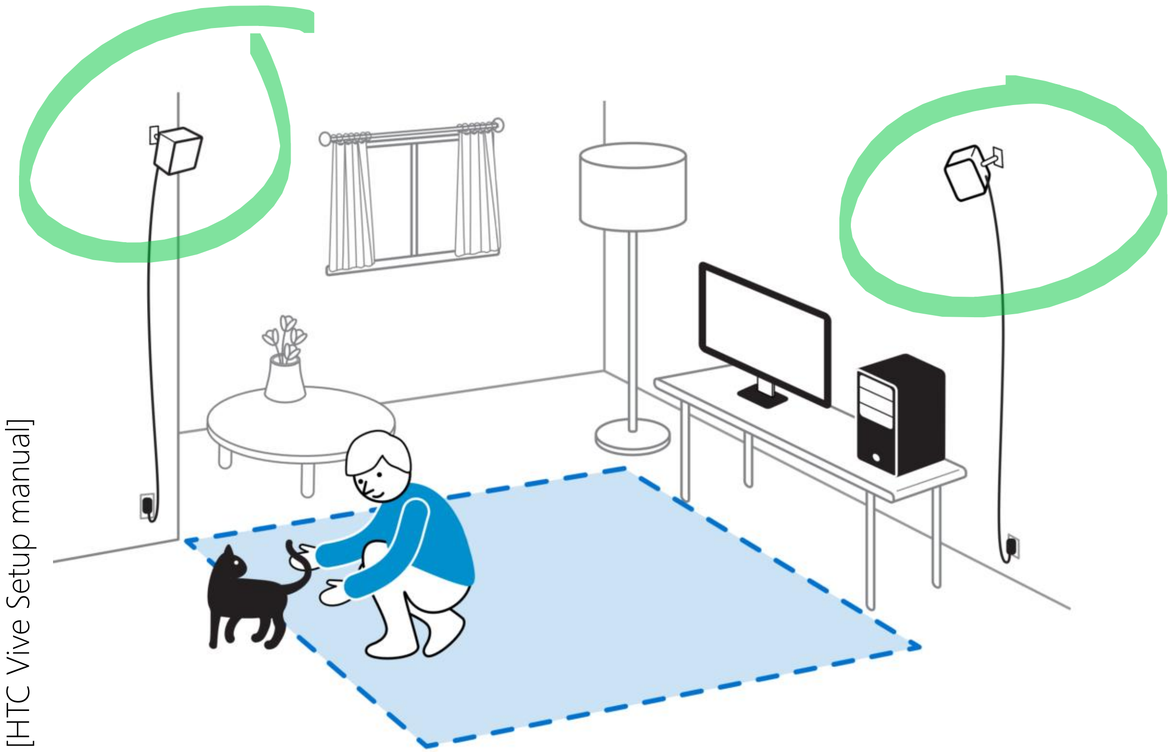
Hardware

HoloLens Sensor Bar



Head Tracking Technologies

[HTC Vive Setup manual]



“Outside-in” head tracking

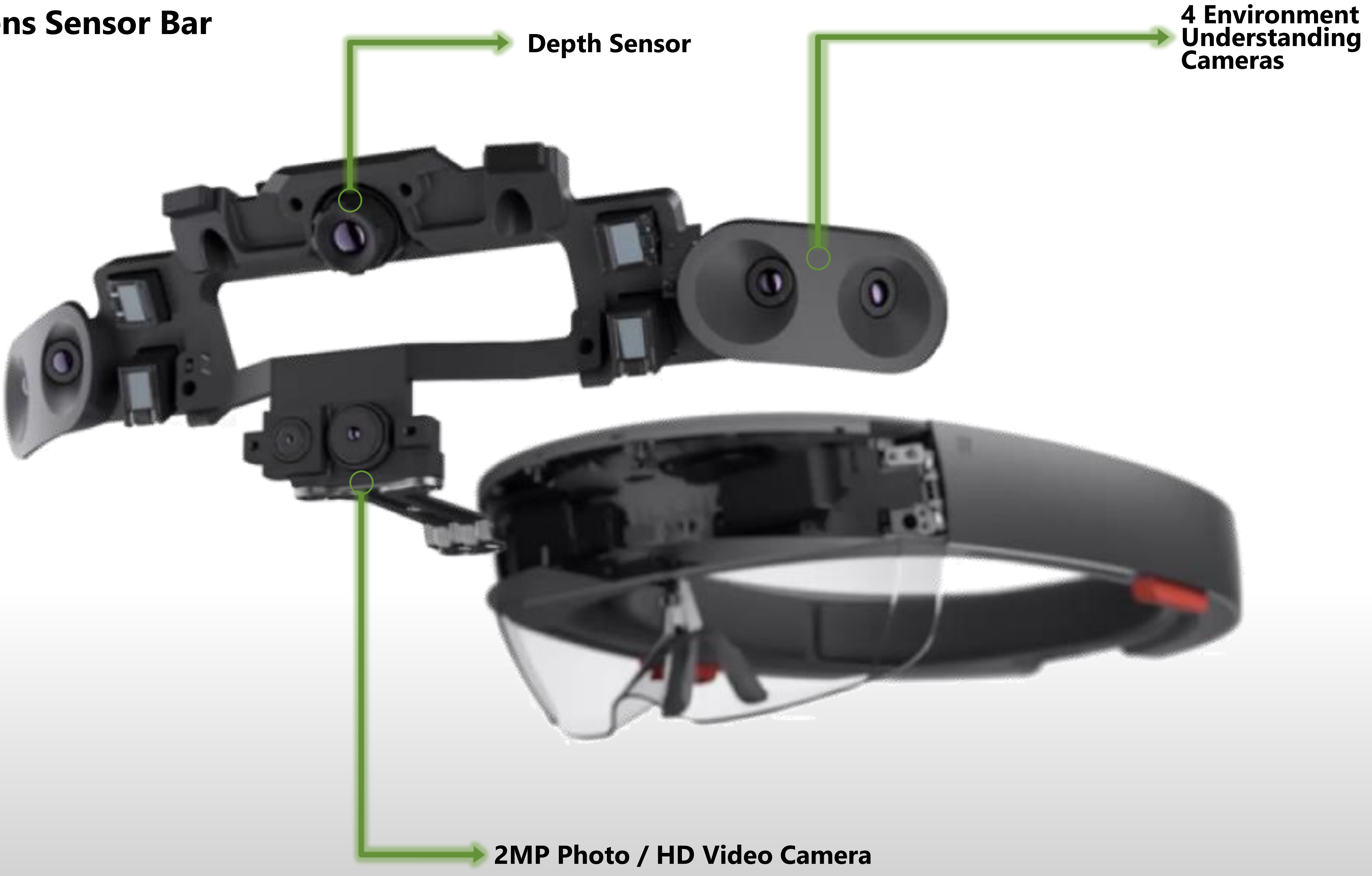
“Inside-out” head tracking

Microsoft

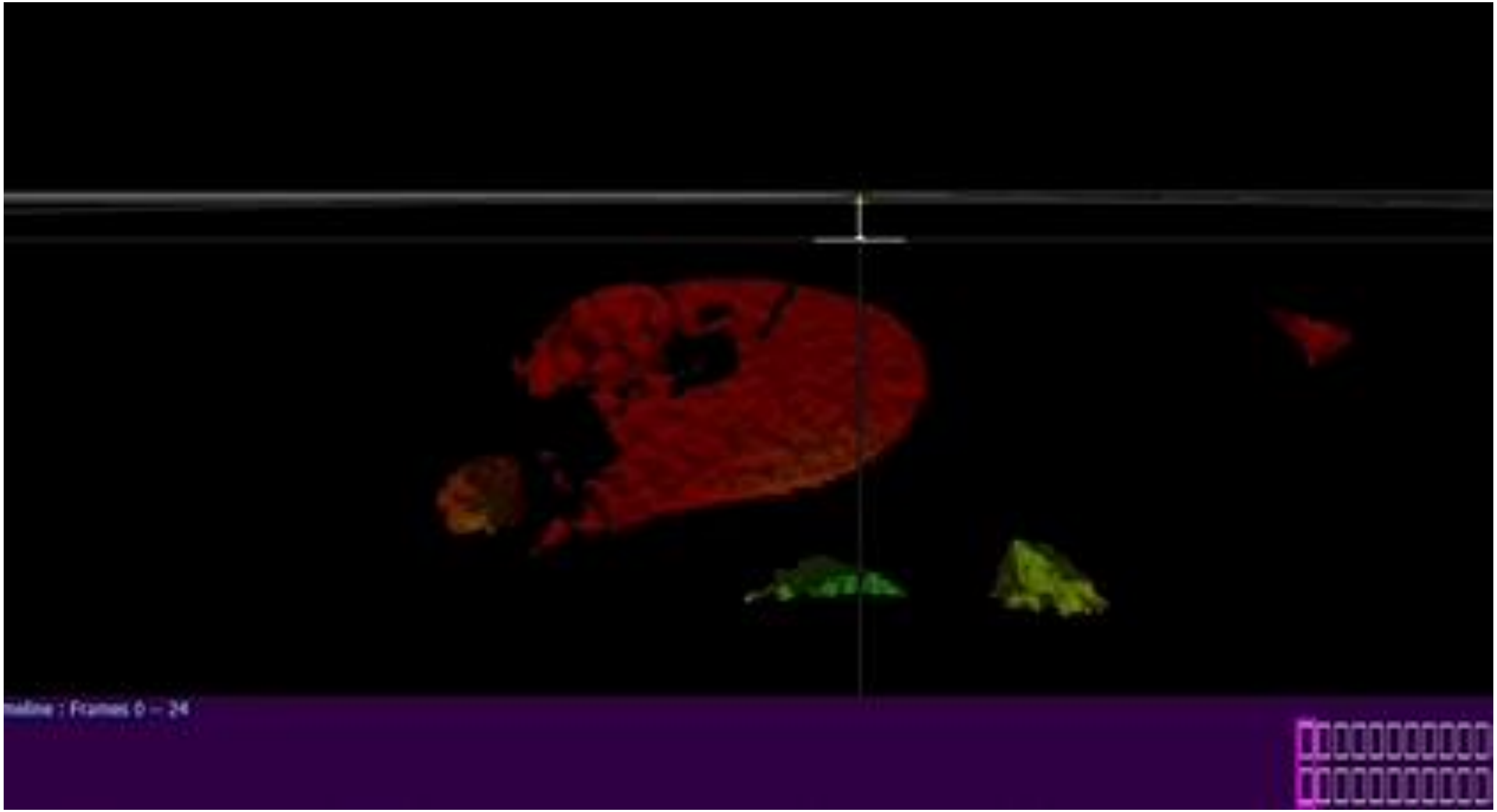
4 wide-angle tracking cameras

HEAD TRACKING CAMERAS: WORLD VIEW

HoloLens Sensor Bar



Input
3D Data



Gesture
Events

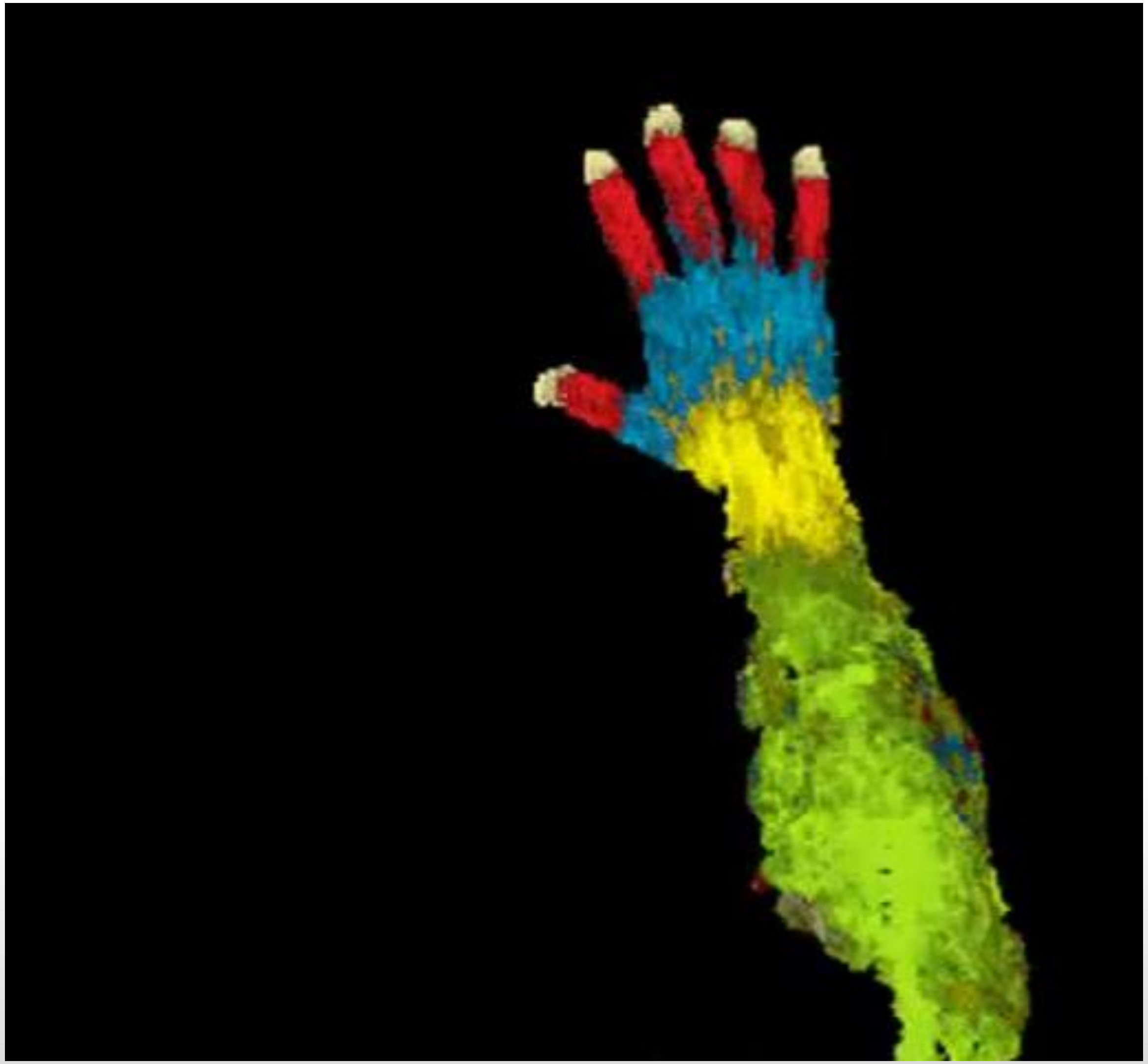
HAND GESTURE RECOGNITION: HOLOLENS V1

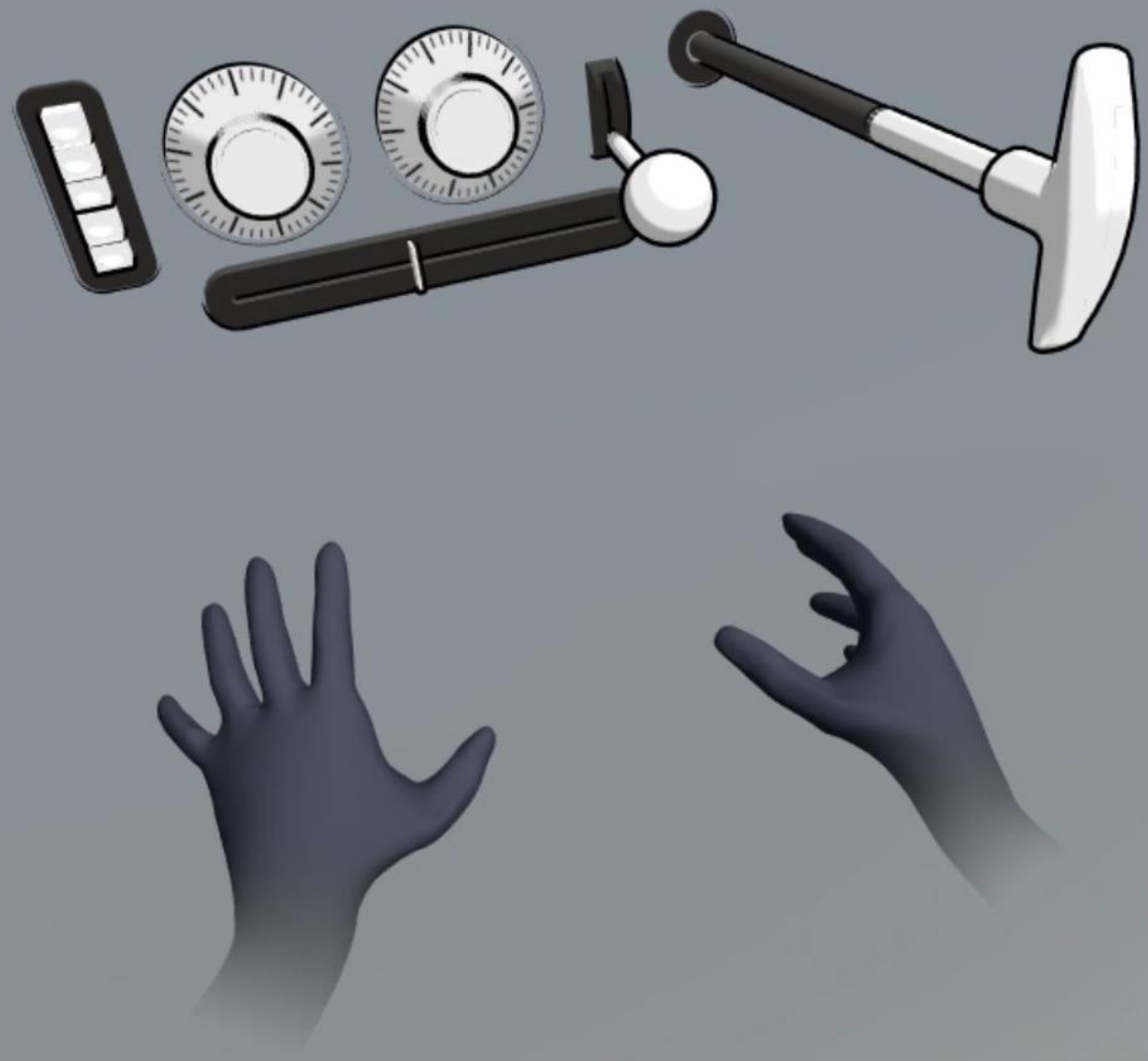
Hand gesture recognition

Machine learning

- Decision trees in V1
  - Based on Kinect Body Tracking
- Deep learning accelerator in V2

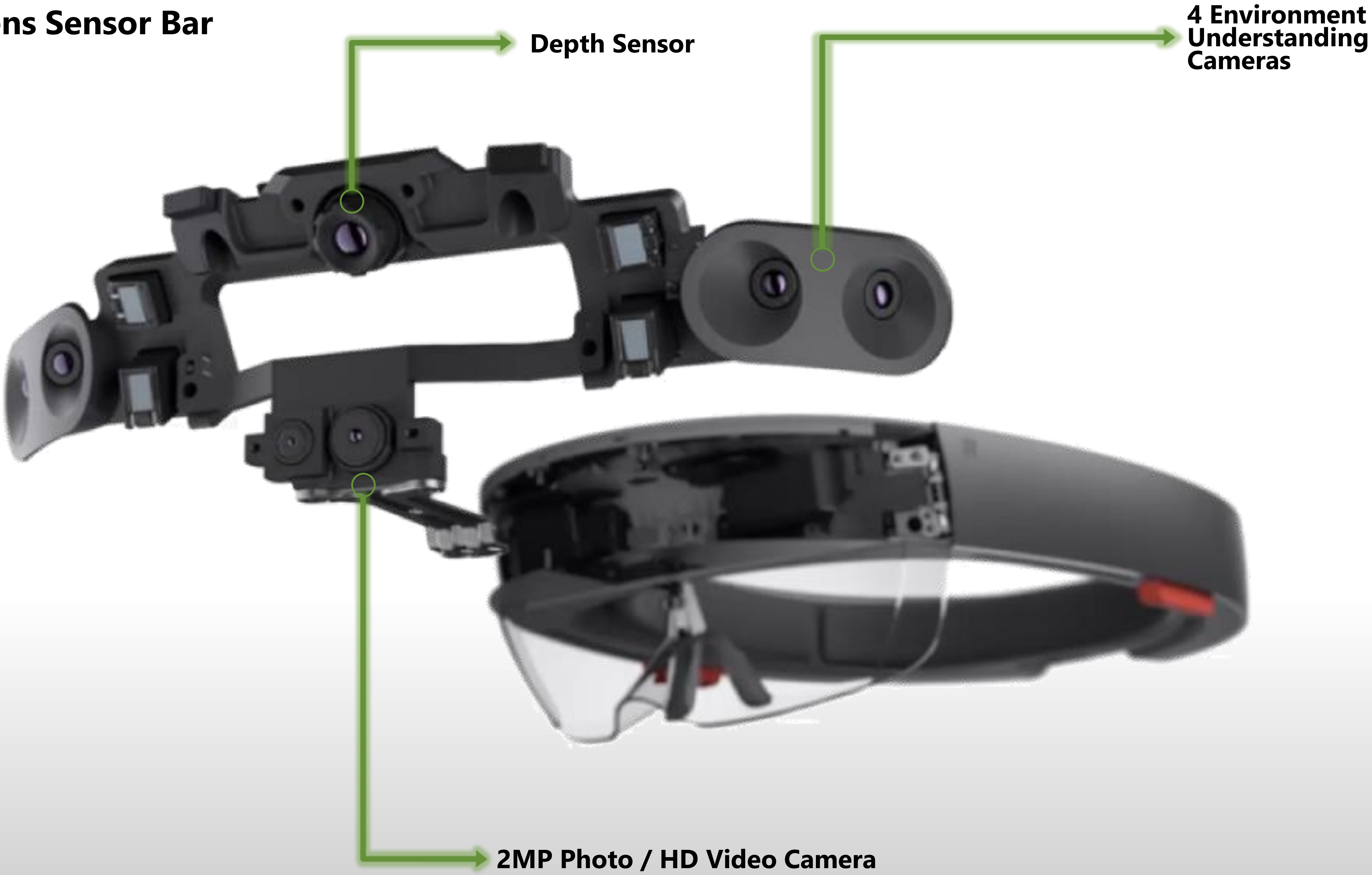
Gesture events and XYZ only





Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences
Taylor et al., *ACM Transactions on Graphics 35(4), pp. #143, 1–12, Proc. SIGGRAPH 2016*

HoloLens Sensor Bar



Microsoft

HoloLens Optics and IMU

HoloLens MLB (Main Logic Board)

- Windows 10
- Custom-built
  Microsoft Holographic Processing Unit (HPU 1.0)
- 64GB Flash
- 2GB RAM (1GB CPU and 1GB HPU)
- x86 architecture

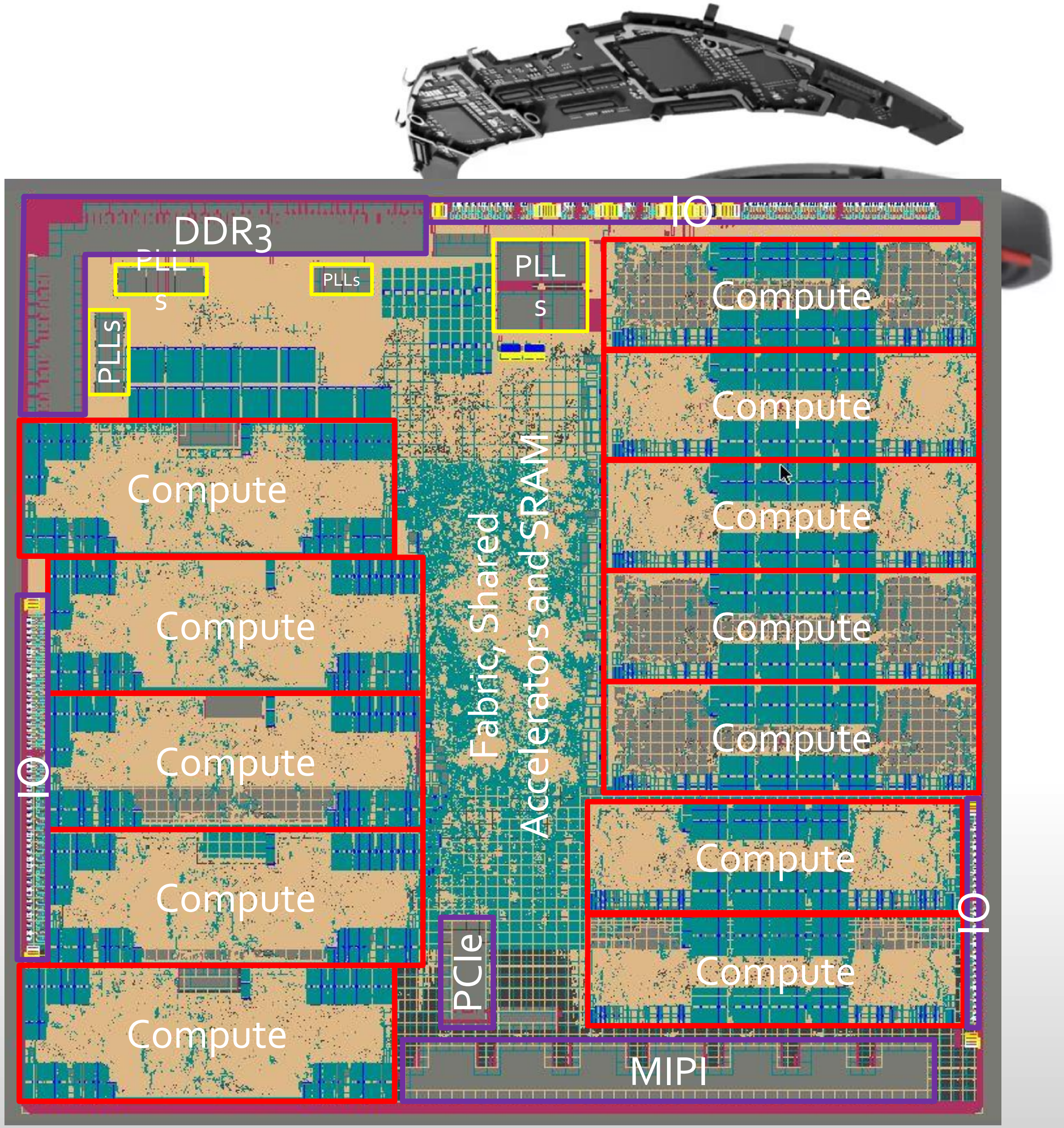
24 Processors, 500MHz each + DNN core

Programmed in C++, with SIMD intrinsics

Our research code was 10x more efficient than the best competitor

To move to HoloLens we needed another 100x.

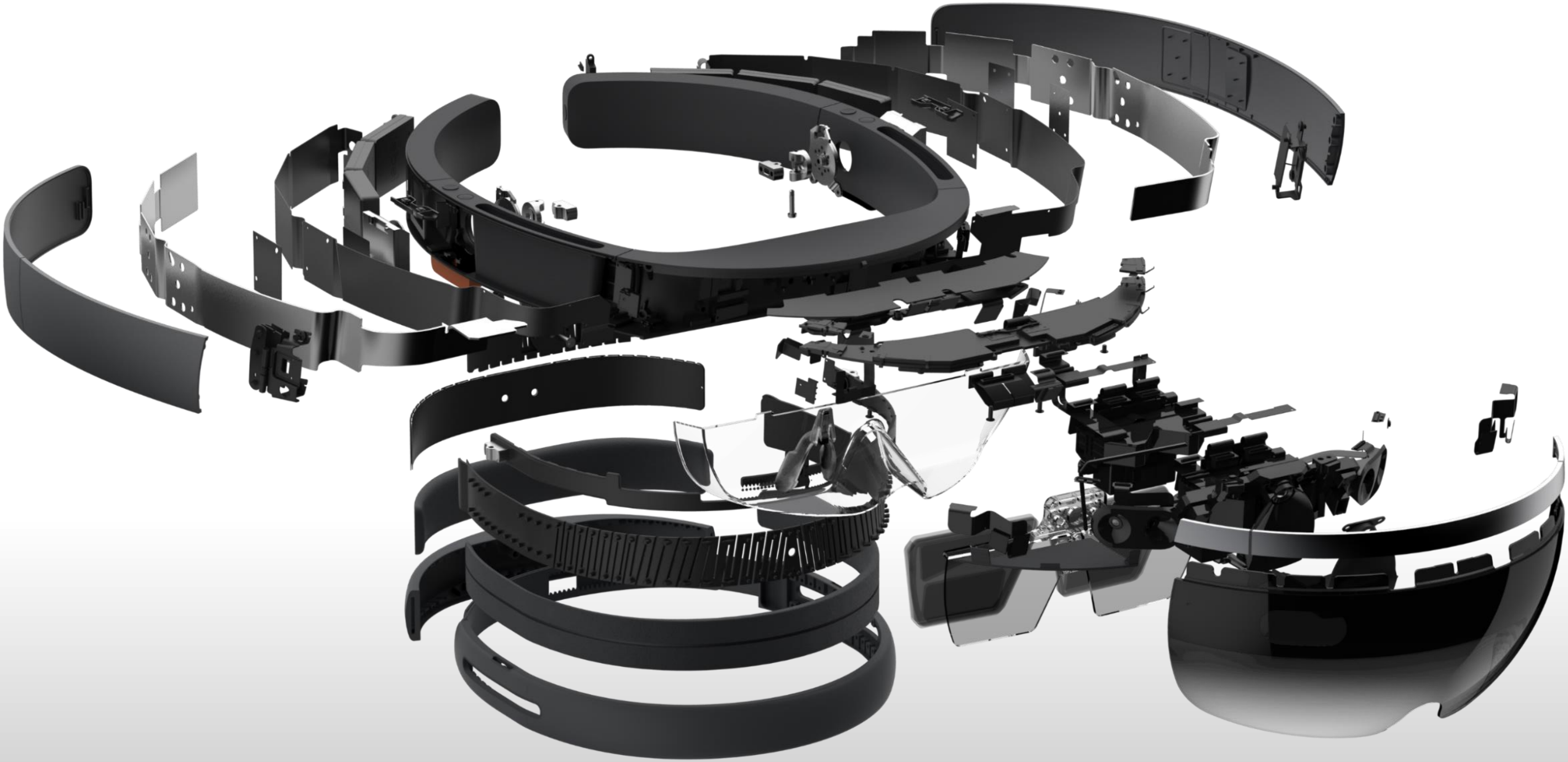
Even games programmers make mistakes today, and mistakes can mean 5x loss of efficiency.



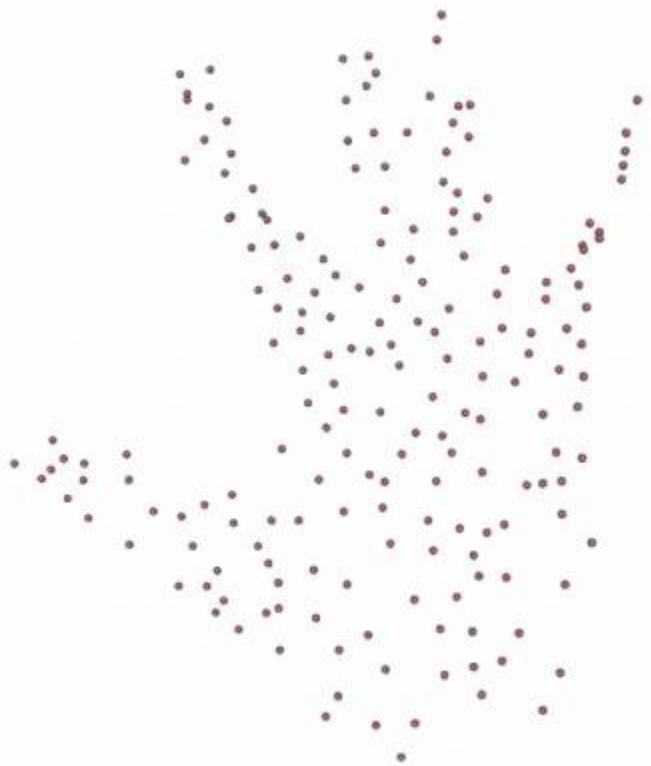
HPU (Holographic Processing Unit): Chip Plot

HoloLens Spatial Sound

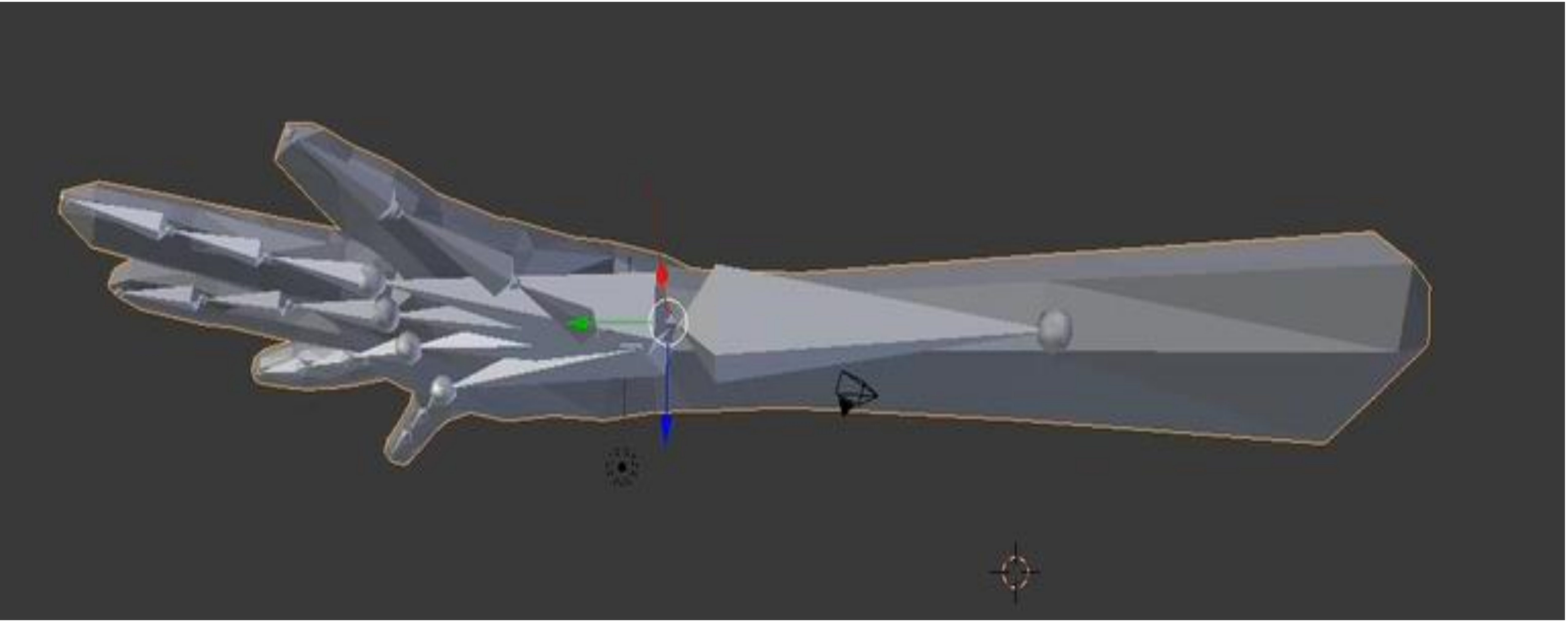
also 4 microphones for speech/beamforming



EFFICIENT COMPUTER VISION & ML:
Learning + Model fitting



- Correspondences
- Data Points



Model driven by parameters $\theta \in \mathbb{R}^d$, e.g. $d = 28$

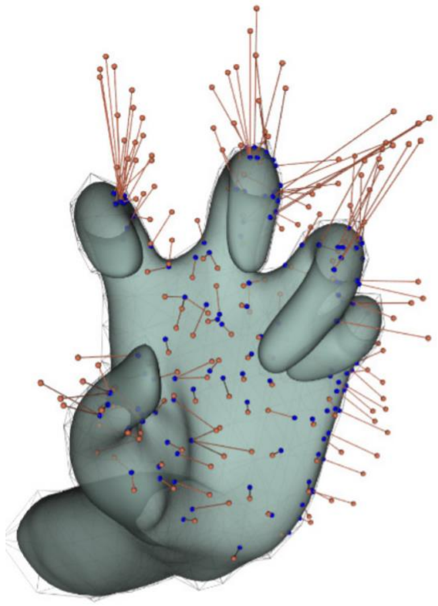
MODEL FITTING: FIRST MAKE A MODEL

Microsoft

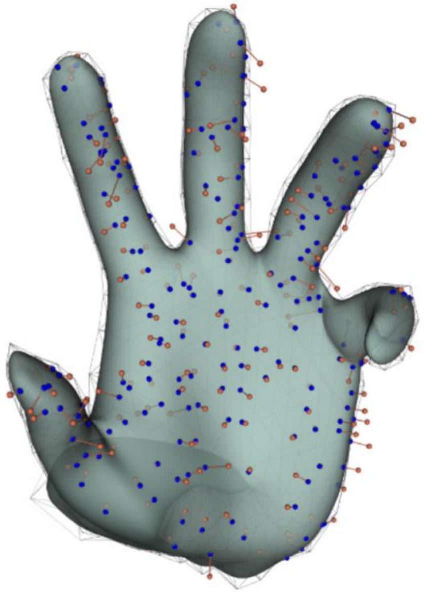
Model

Pose parameters $\boldsymbol{\theta}$

Energy Function



Bad pose $\boldsymbol{\theta}$



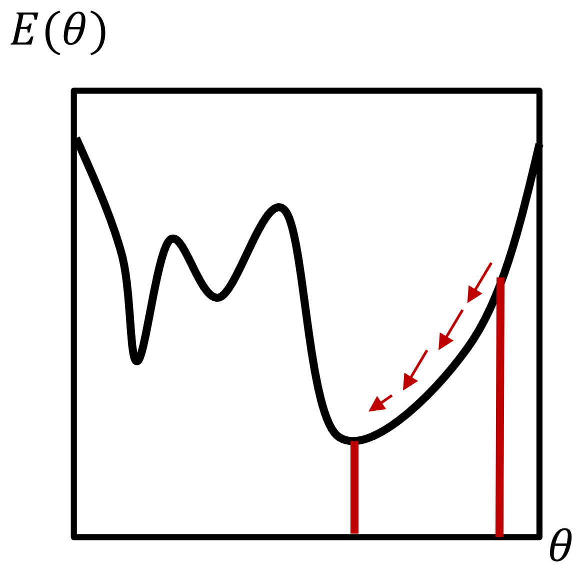
Good pose $\boldsymbol{\theta}$

Observed 3D data point

Closest point on model

Contribution to energy

Optimization



Given function

$$f(x): \mathbb{R}^d \mapsto \mathbb{R},$$

Devise strategies for finding x which minimizes f

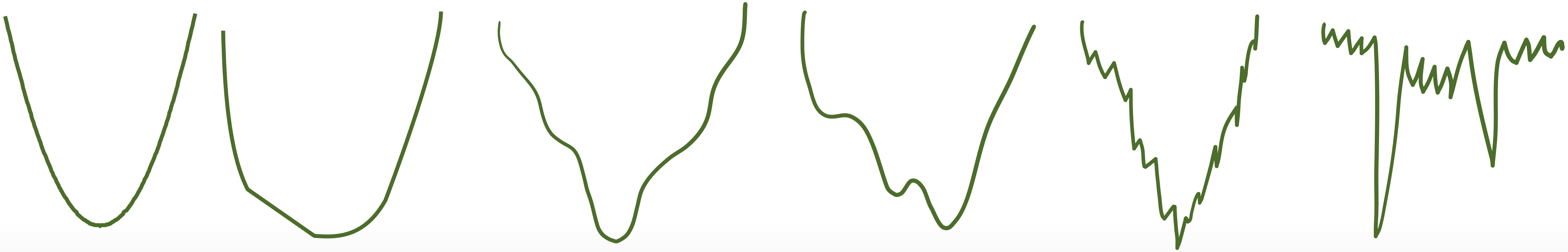
- Gradient descent++: Stochastic, Block, Minibatch
- Coordinate descent++: Block
- Newton++: Gauss, Quasi, Damped, Levenberg Marquardt, dogleg, Trust region, Doublestep LM, [L-]BFGS, Nonlin CG

- Not covered
  - Proximal methods: Nesterov, ADMM...

Given function

$$f(x): \mathbb{R}^d \mapsto \mathbb{R}$$

Devise strategies for finding x which minimizes f



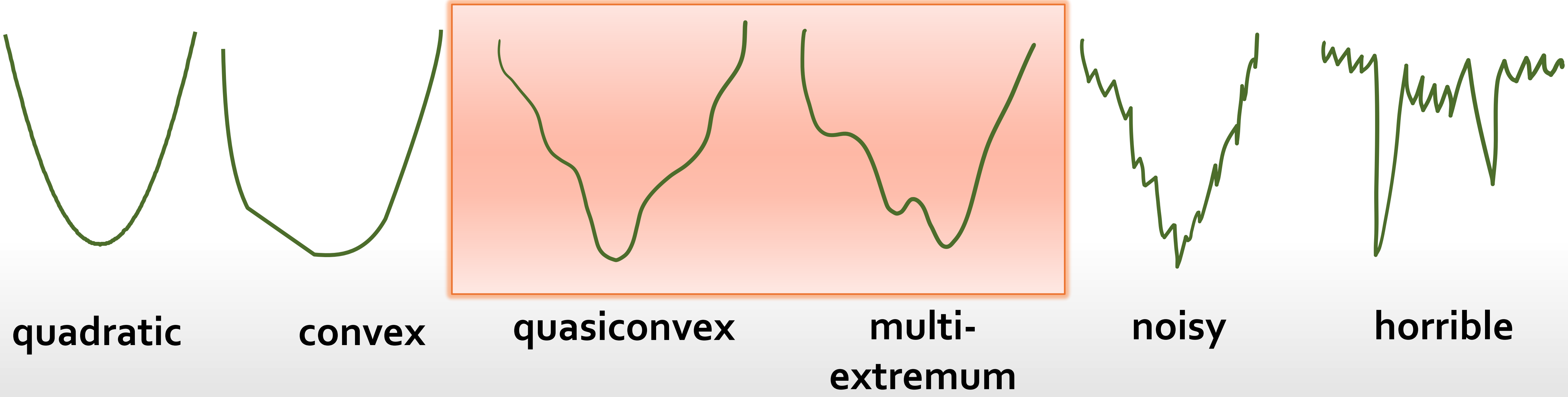
quadratic | **convex** | **quasiconvex** | **multi-extremum** | **noisy** | **horrible**

CLASSES OF FUNCTIONS

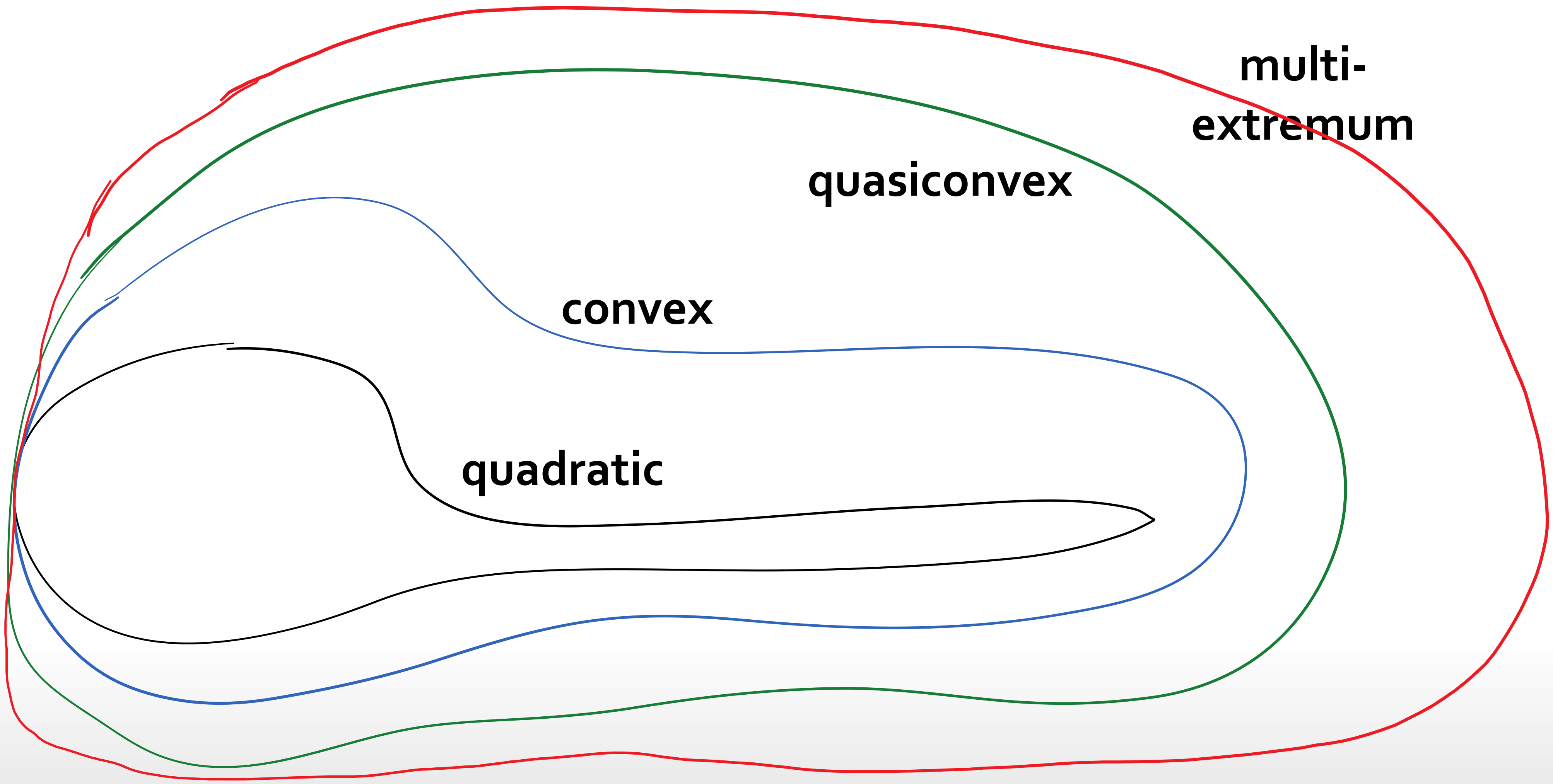
Given function

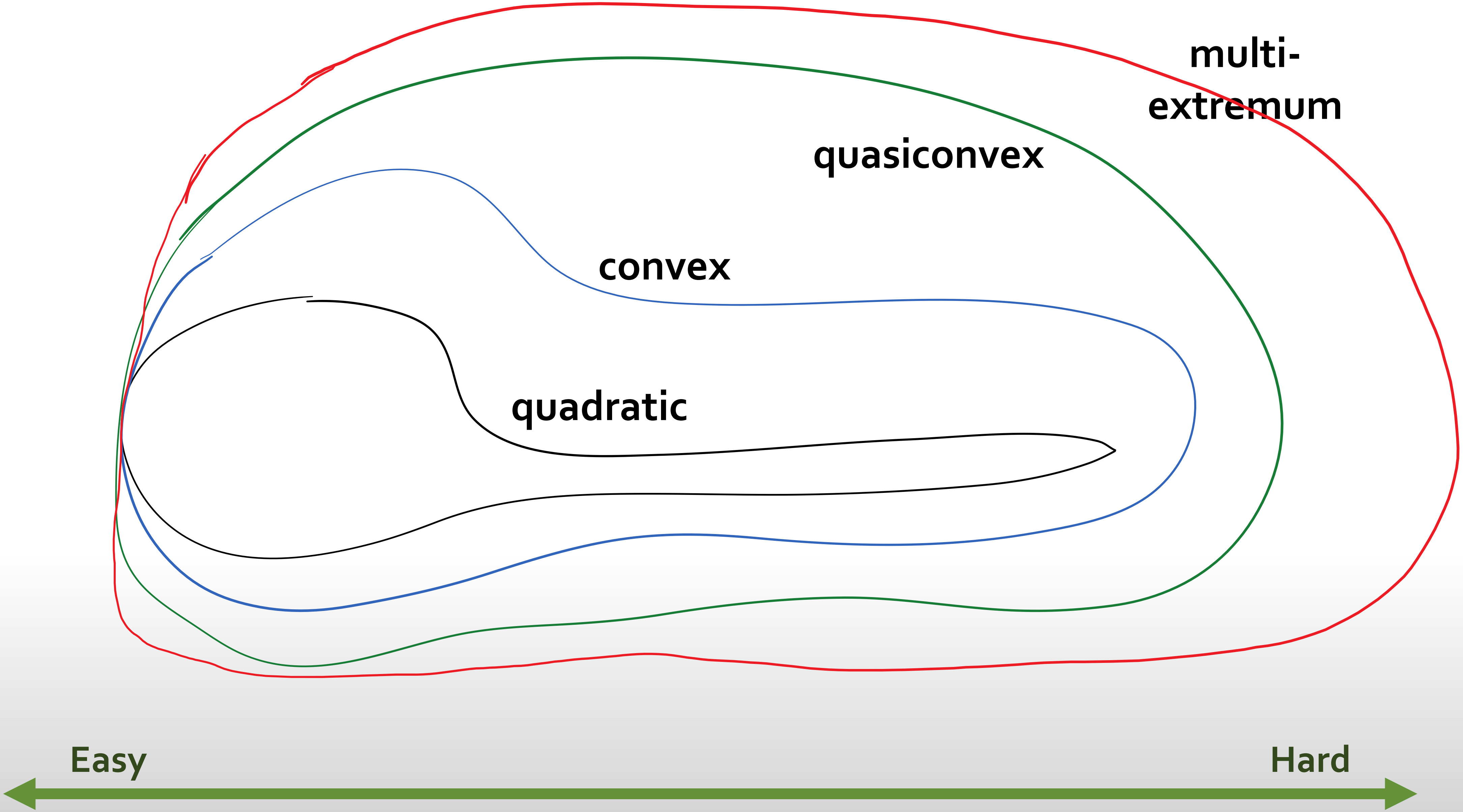
$$f(x): \mathbb{R}^d \mapsto \mathbb{R}$$

Devise strategies for finding x which minimizes f



CLASSES OF FUNCTIONS

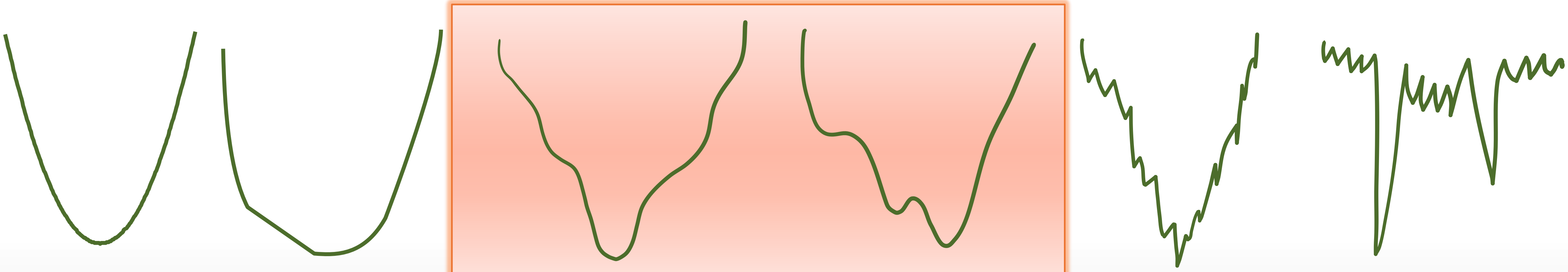

multi-
extremum
quasiconvex
convex
quadratic


multi-
extremum
quasiconvex
convex
quadratic
Easy
Hard

Given function

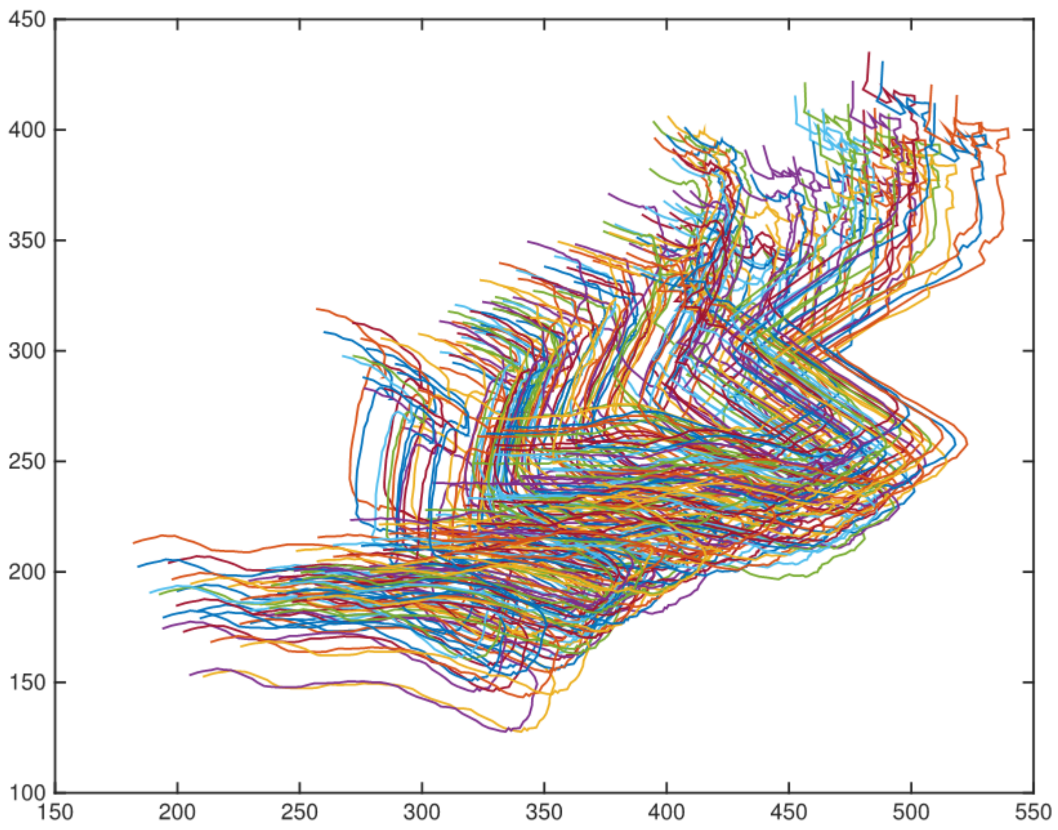
$$f(x): \mathbb{R}^d \mapsto \mathbb{R}$$

Devise strategies for finding x which minimizes f

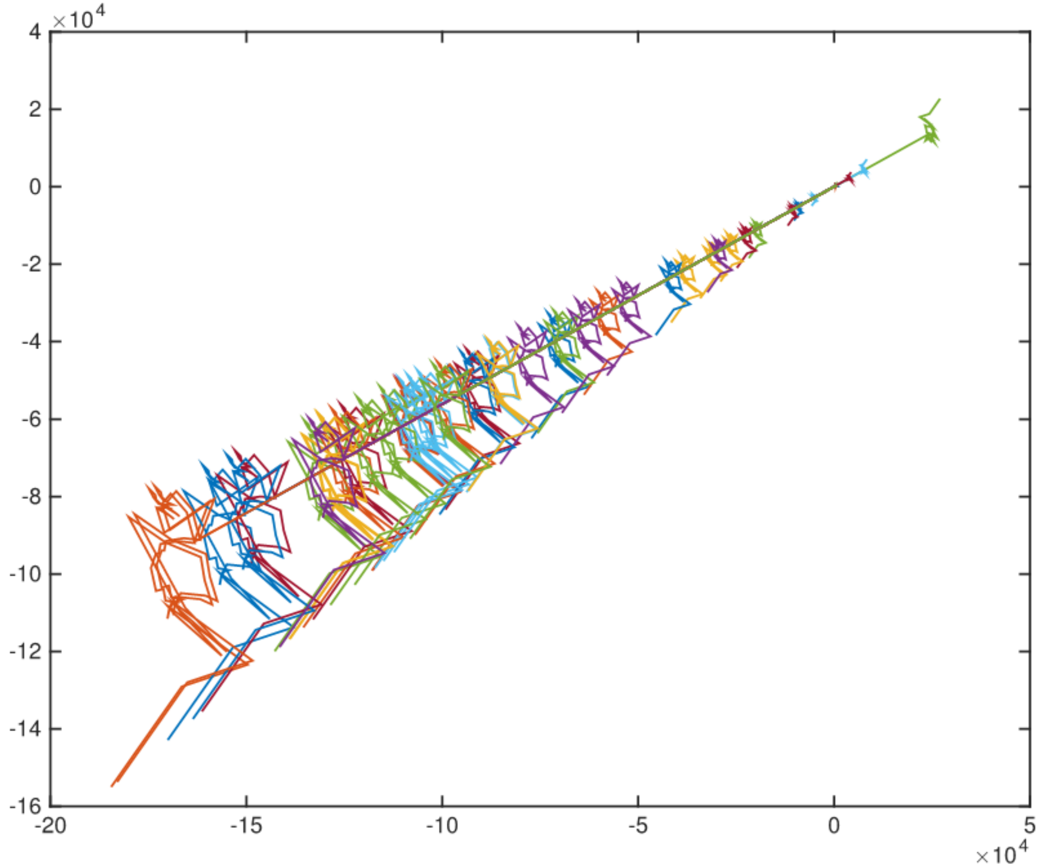


quadratic | convex | quasiconvex | multi-extremum | noisy | horrible

CLASSES OF FUNCTIONS



(a) Best known minimum (0.3228)



(b) Second best solution (0.3230)

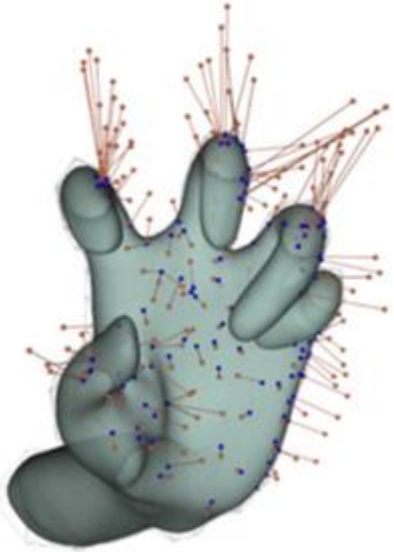
It's worth it to get to the optimum

Microsoft

Given function

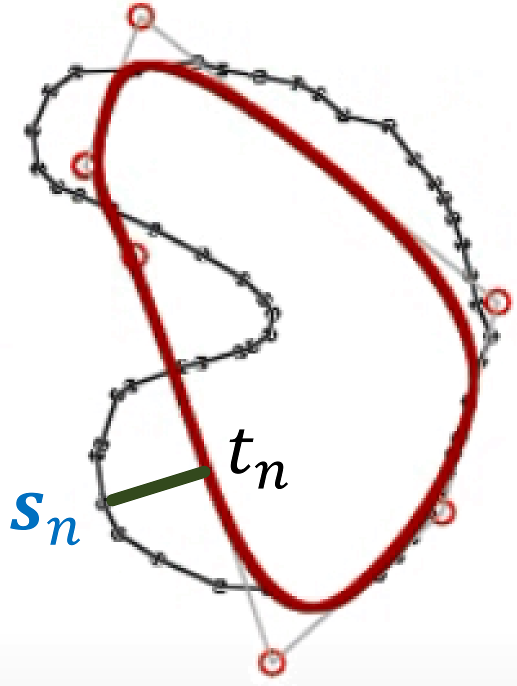
$$f(x): \mathbb{R}^d \mapsto \mathbb{R}$$

$f(x) = \sum_{n=1}^{N} f_n(x)$	Stochastic gradient descent
$f(x) = \sum_{n=1}^{N} f_n(x)^2$	[Damped] Gauss-Newton Levenberg-Marquardt
$f(x) = \sum_{n=1}^{N} \min_{t_n} f_n(x, t_n)$	Block coordinate descent VarPro?

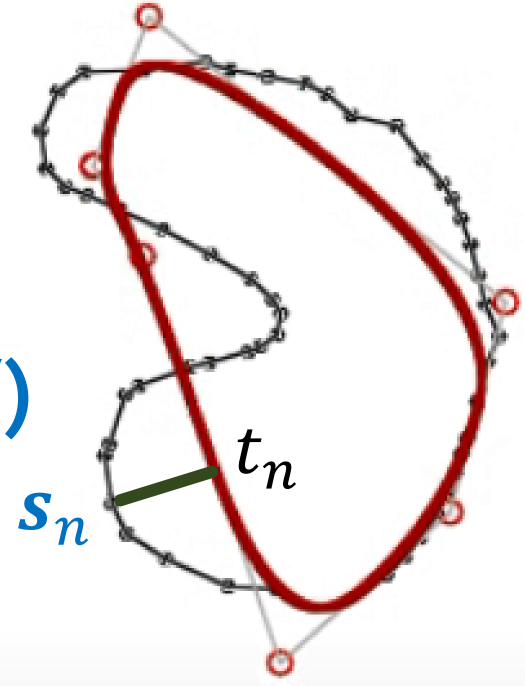


$$\min_x \sum_{n=1}^{N} \min_{t_n} f_n(x, t_n)$$

SLAM, model fitting, recommenders,...



$$\min_x \sum_{n=1}^{N} \min_{t_n} f_n(x, t_n)$$



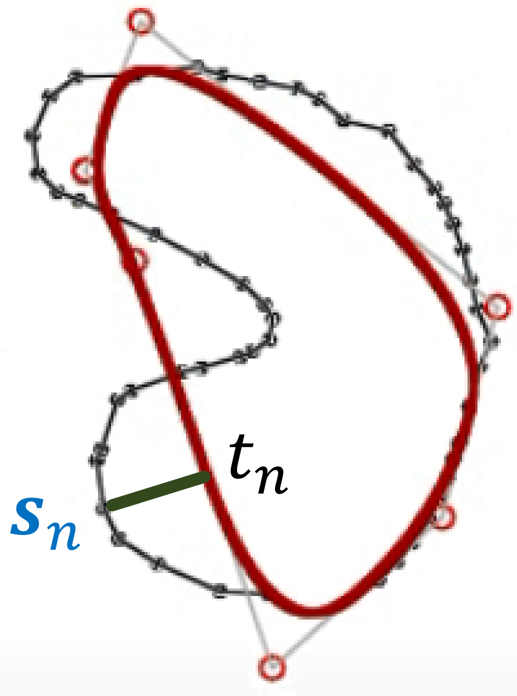
Solution 1: Block coordinate descent ("ICP")

while (something):

$$\forall n: \; t_n = \operatorname*{argmin}_t f_n(x, t)$$

$$x := \operatorname*{argmin}_x \textstyle\sum_{n=1}^{N} f_n(x, t_n)$$

$$\min_x \sum_{n=1}^{N} \min_{t_n} f_n(x, t_n)$$



Solution 2: Joint optimization ("lifting")

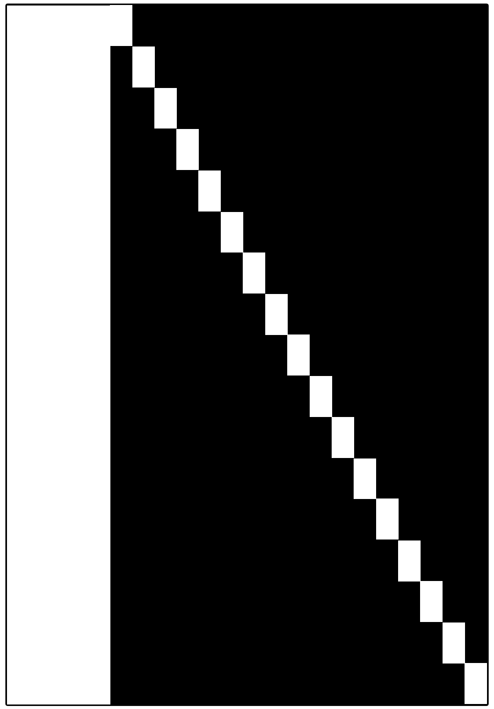
$$\min_{x,t_1,\ldots,t_N} \sum_{n=1}^{N} f_n(x, t_n)$$

A d-dimensional problem becomes $N + d$

Much much faster in practice, if problem structure used well

$$\min_{x} \sum_{n=1}^{N} \min_{t_n} f_n(x, t_n)$$

x $\quad t_1 \ldots t_N$



Jacobian $\left[\frac{\partial f_n}{\partial x} \,\middle|\, \frac{\partial f_n}{\partial t_{1..n}}\right]$

Solution 2: Joint optimization ("lifting")

$$\min_{x, t_1, \ldots, t_N} \sum_{n=1}^{N} f_n(x, t_n)$$

A d-dimensional problem becomes $N + d$

Much much faster in practice, **if** problem structure used well

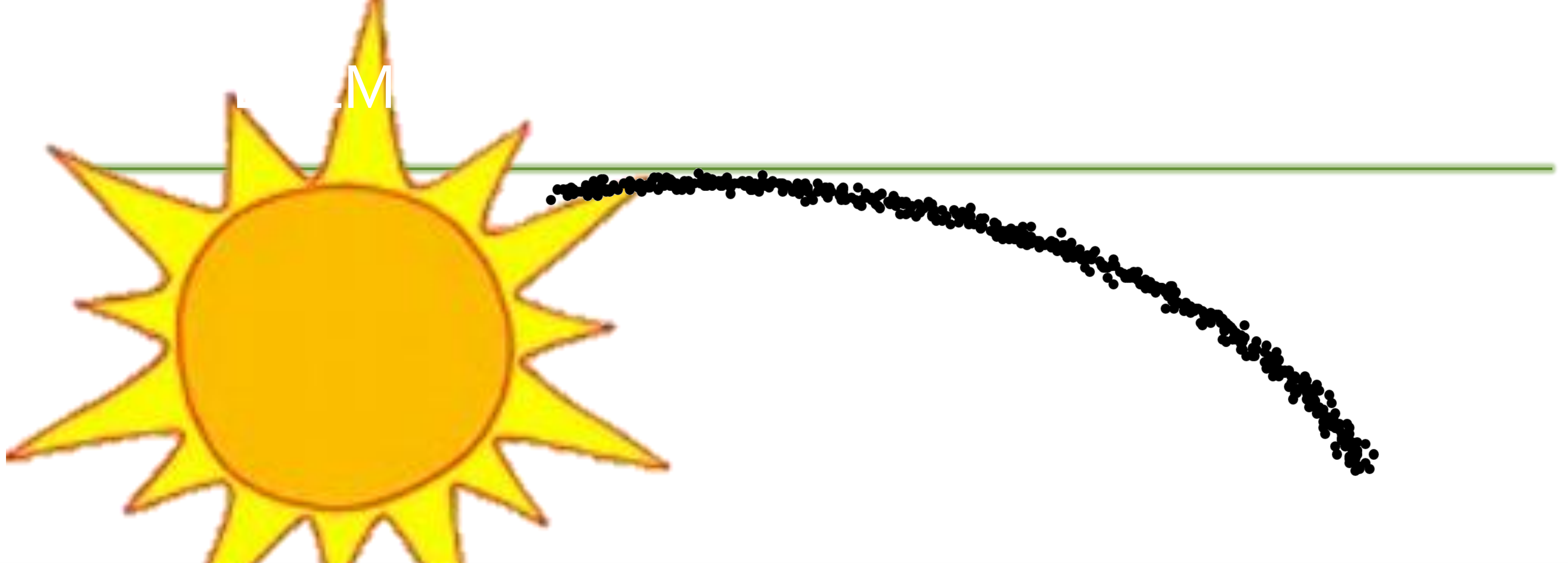
AN EXEMPLARY PROBLEM

"Based on a true story", not necessarily historically accurate

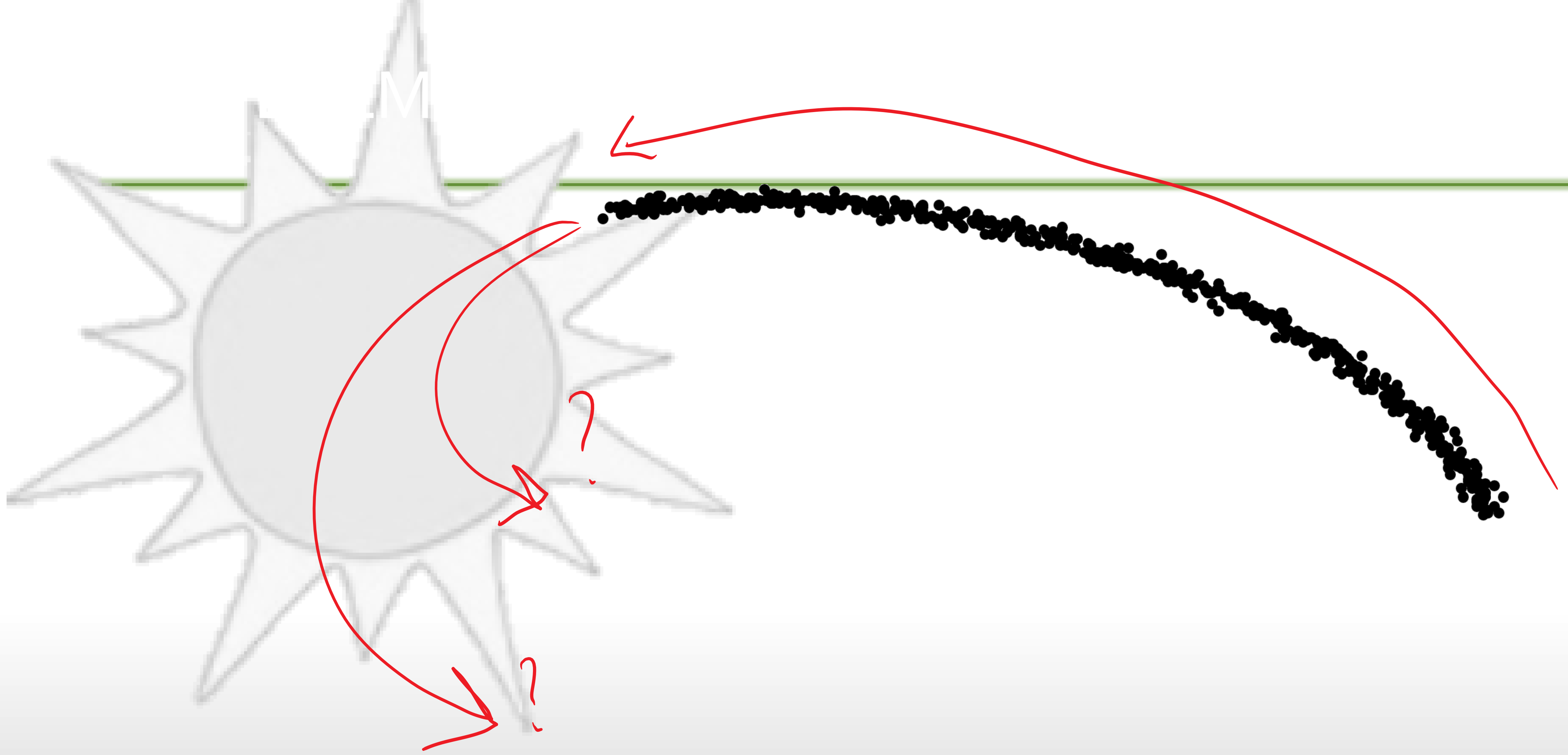
Note well: this problem is a good proxy for much more realistic problems:

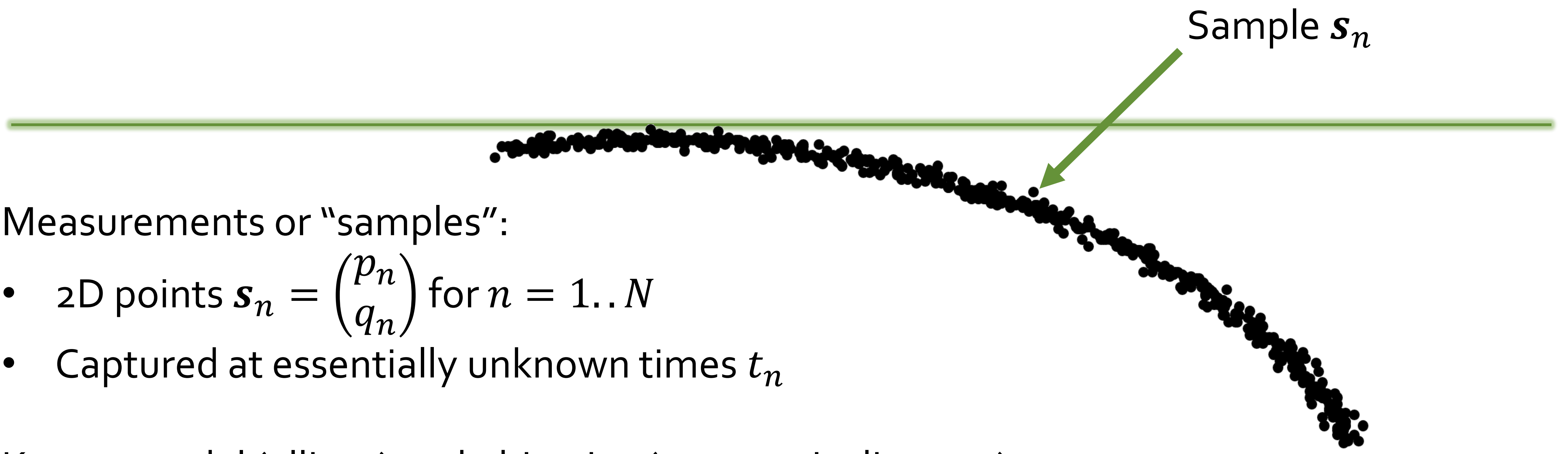
1. Stereo camera calibration
2. Multiple-camera bundle adjustment
3. Surface fitting, e.g. subdivision surfaces to range data, realtime hand tracking
4. Matrix completion
5. Image denoising.

[From Neil Lawrence]



The year: 1801
The hot topic: A "guest planet", named Ceres
The big question: Where will it reappear?





Measurements or "samples":

- 2D points $\boldsymbol{s}_n = \begin{pmatrix} p_n \\ q_n \end{pmatrix}$ for $n = 1..N$
- Captured at essentially unknown times t_n

Known model (ellipse) and objective (geometric distance):

$$f(x) = \sum_{n=1}^{N} \min_t f_n(x, t)$$

$$f_n(x, t) = \left\| \begin{pmatrix} x_1 \cos t + x_2 \sin t + x_3 - p_n \\ x_4 \cos t + x_5 \sin t + x_6 - q_n \end{pmatrix} \right\|^2$$

Measurements or “samples”:

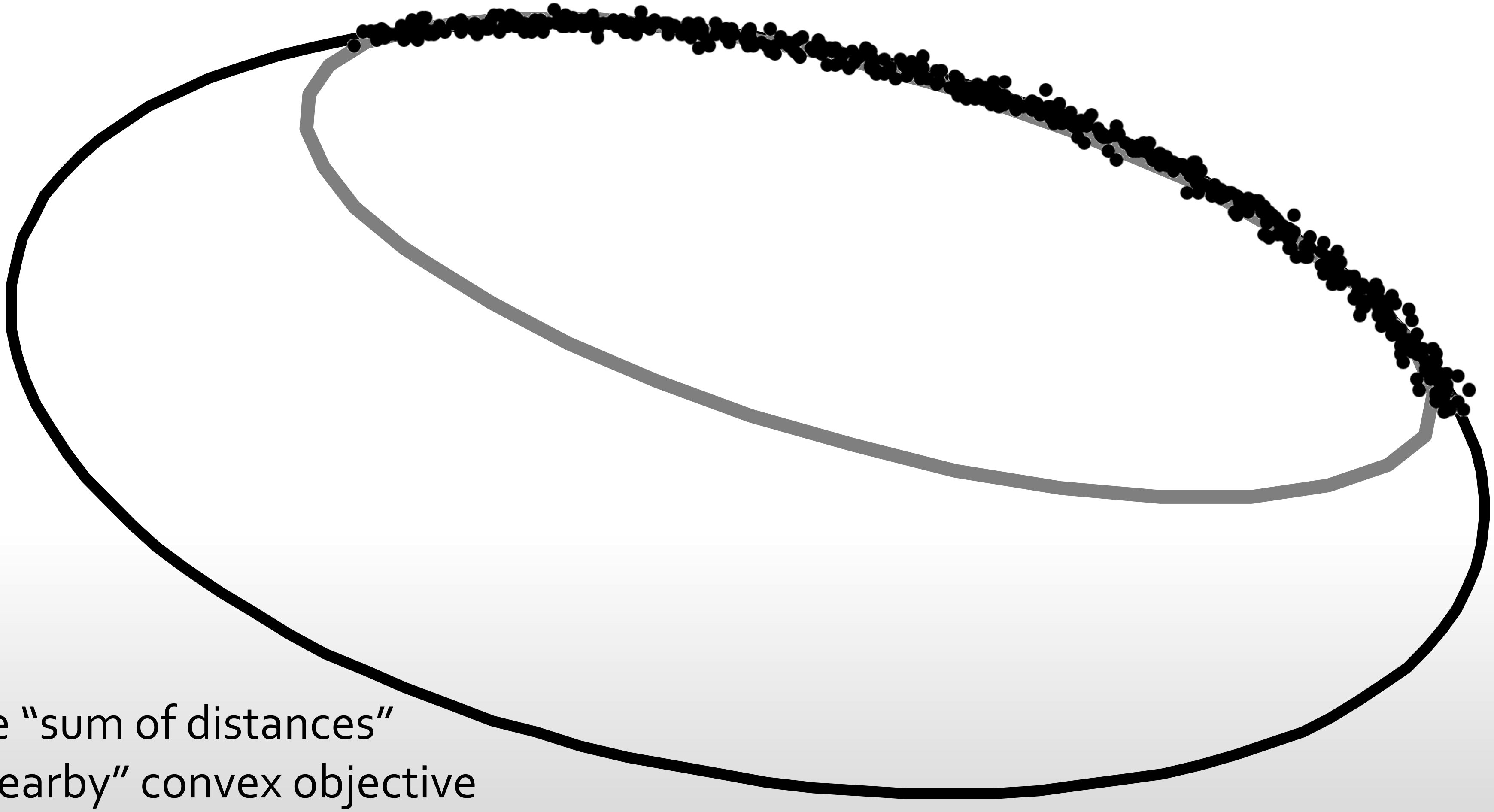
- 2D points $s_n = \begin{pmatrix} p_n \\ q_n \end{pmatrix}$ for $n = 1..N$
- Captured at essentially unknown times t_n

Known model (ellipse) and objective (geometric distance):

$$f(x) = \sum_{n=1}^{N} \min_t f_n(x,t)$$

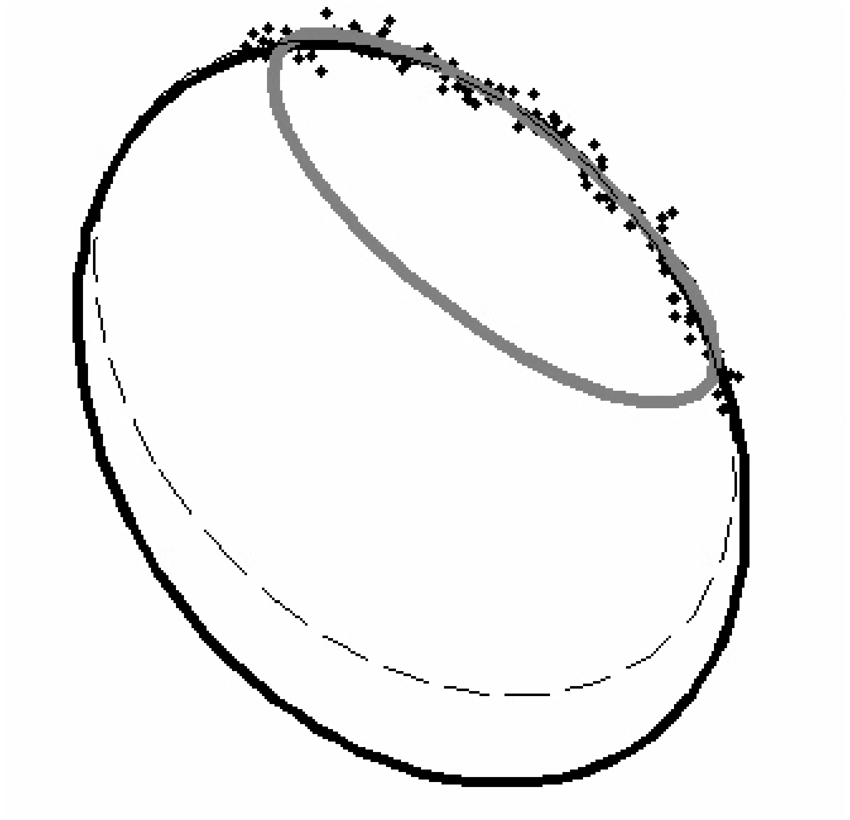
$$f_n(x,t) = \left\| \begin{pmatrix} x_1 \cos t + x_2 \sin t + x_3 - p_n \\ x_4 \cos t + x_5 \sin t + x_6 - q_n \end{pmatrix} \right\|^2$$

“Direct least squares fitting of ellipses”
[Fitzgibbon et al, 1999]

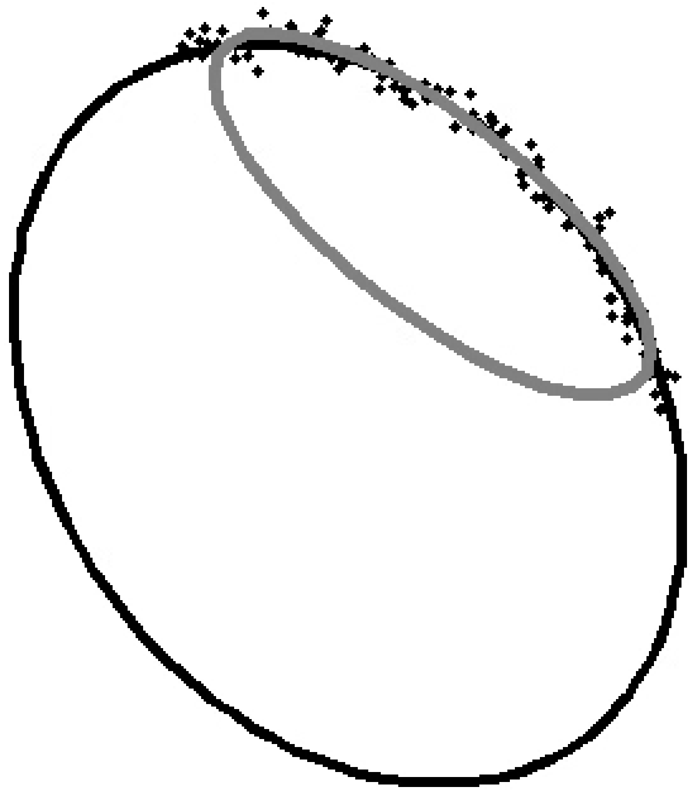


Does not minimize “sum of distances” objective, but a “nearby” convex objective

RUNNING AN OFF-THE-SHELF FITTER DOES NOT.



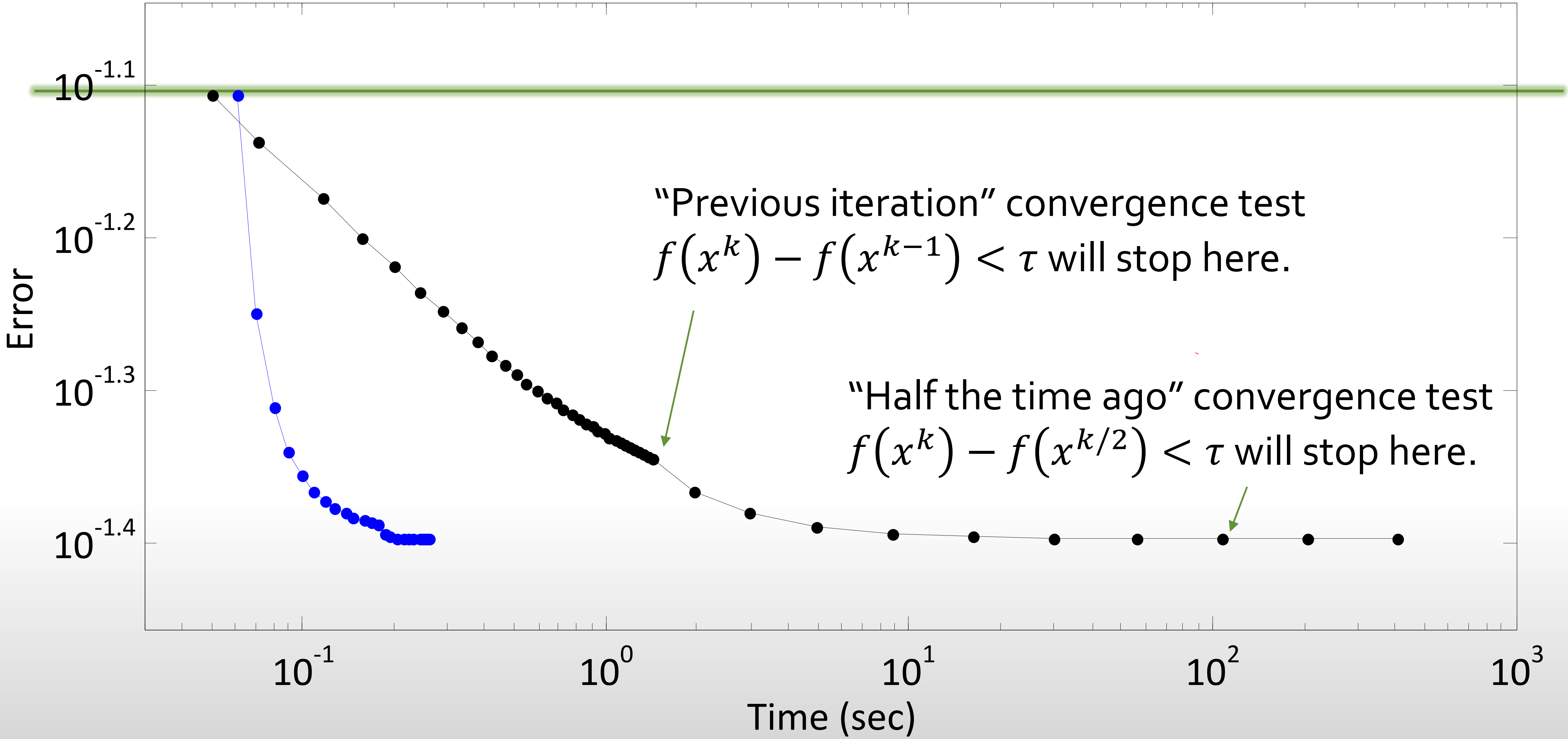
A slow method

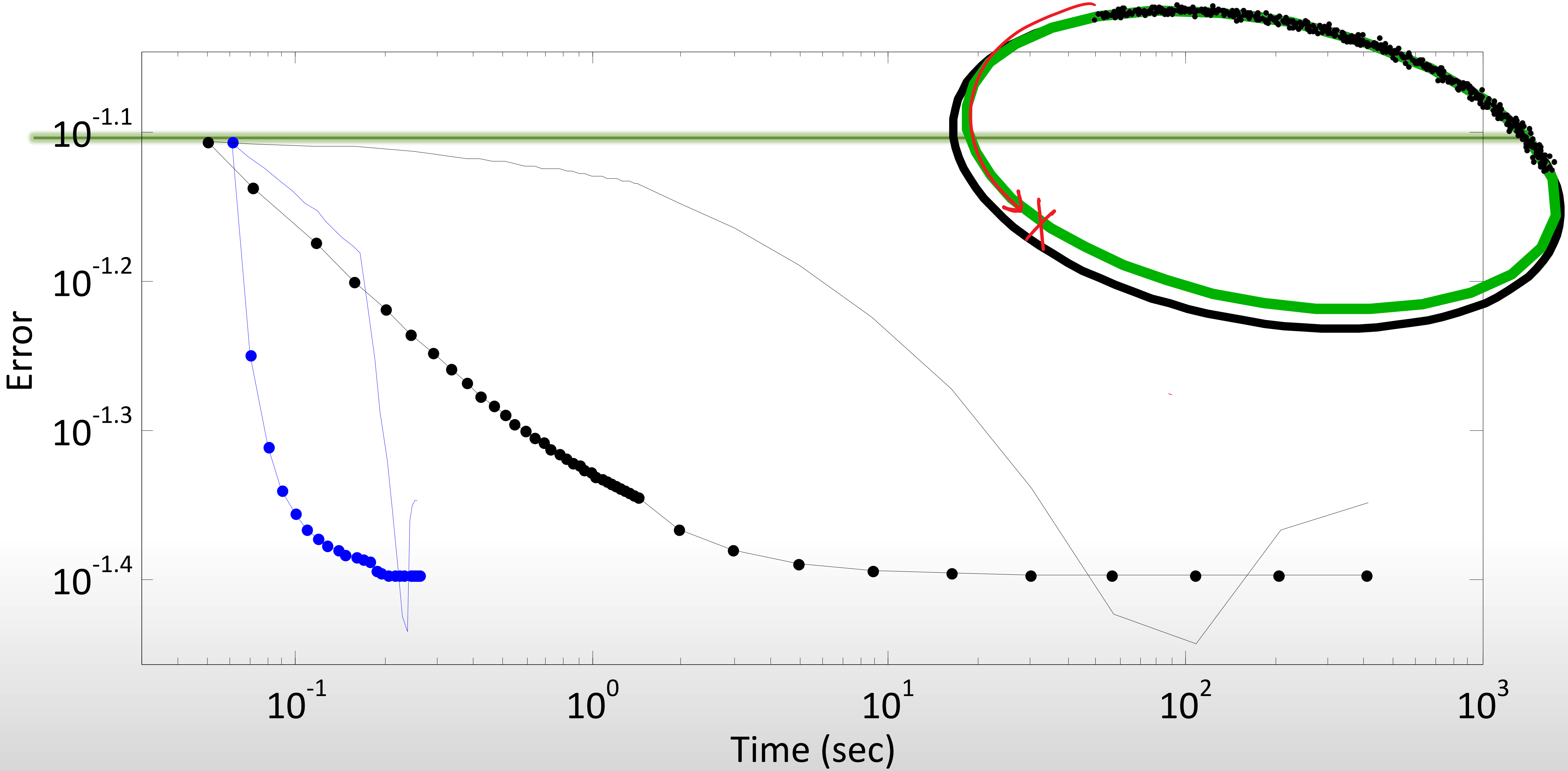


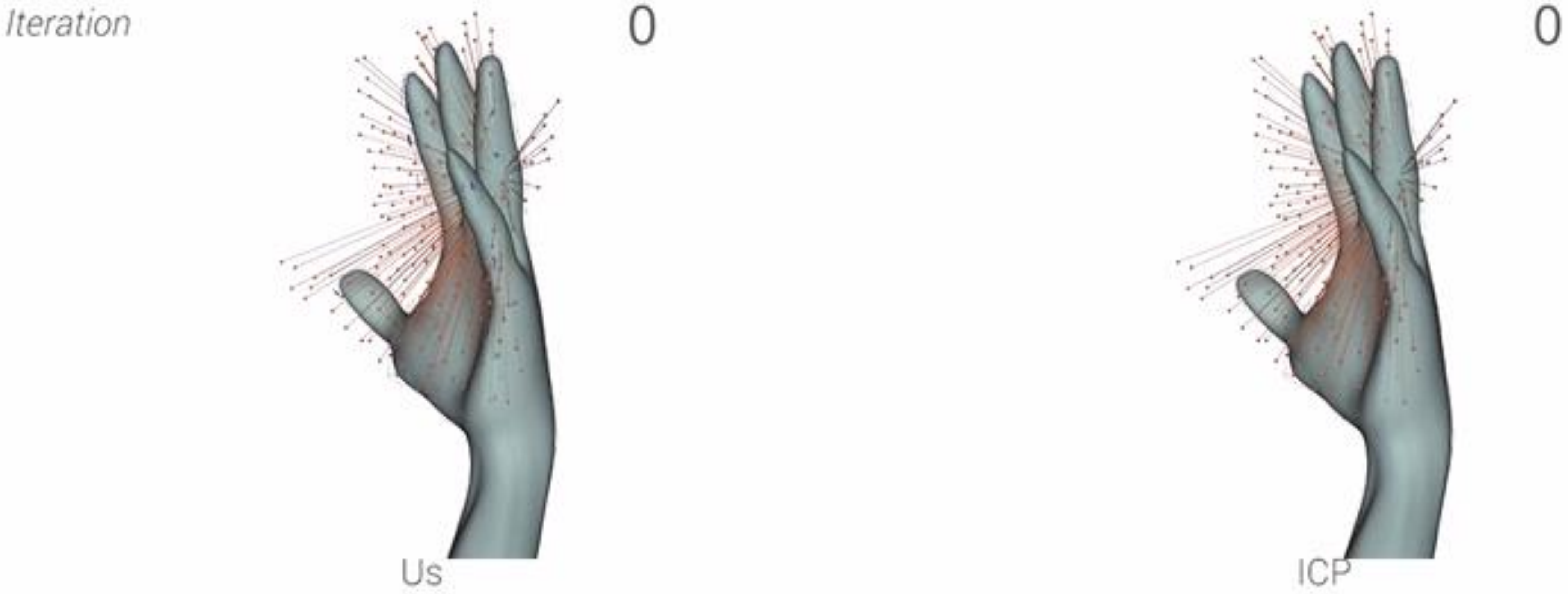
A fast method, slowed down 10x

SPEED RESULTS: SNEAK PREVIEW

Microsoft


"Previous iteration" convergence test
$f(x^k) - f(x^{k-1}) < \tau$ will stop here.
"Half the time ago" convergence test
$f(x^k) - f(x^{k/2}) < \tau$ will stop here.
Error
$10^{-1.1}$
$10^{-1.2}$
$10^{-1.3}$
$10^{-1.4}$
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
Time (sec)


Error
10^-1.1
10^-1.2
10^-1.3
10^-1.4
10^-1
10^0
10^1
10^2
10^3
Time (sec)



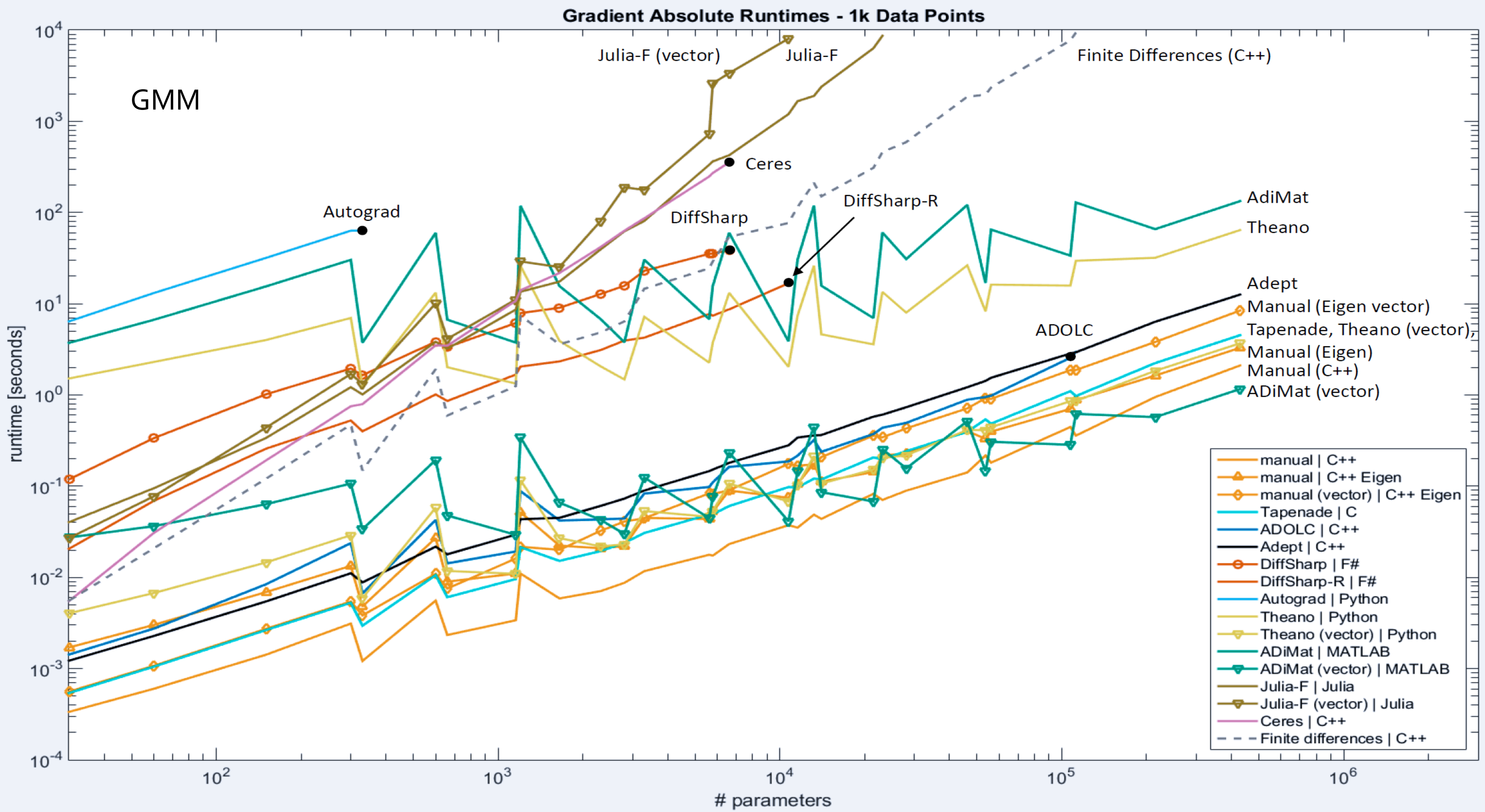
And in the real world

Microsoft

BETTER THAN STATE-OF-THE-ART, 10X FASTER

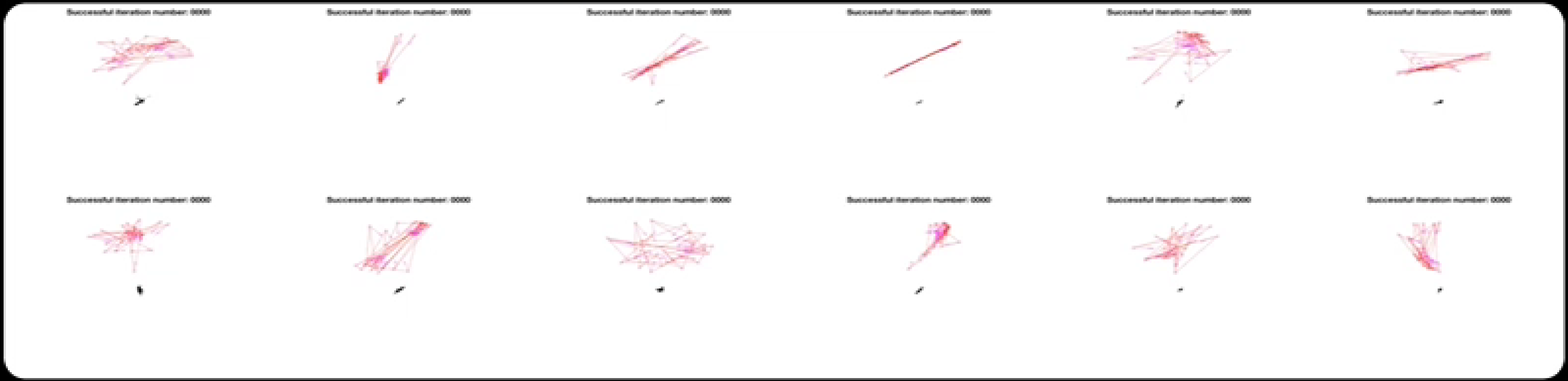
Microsoft

Bonus material

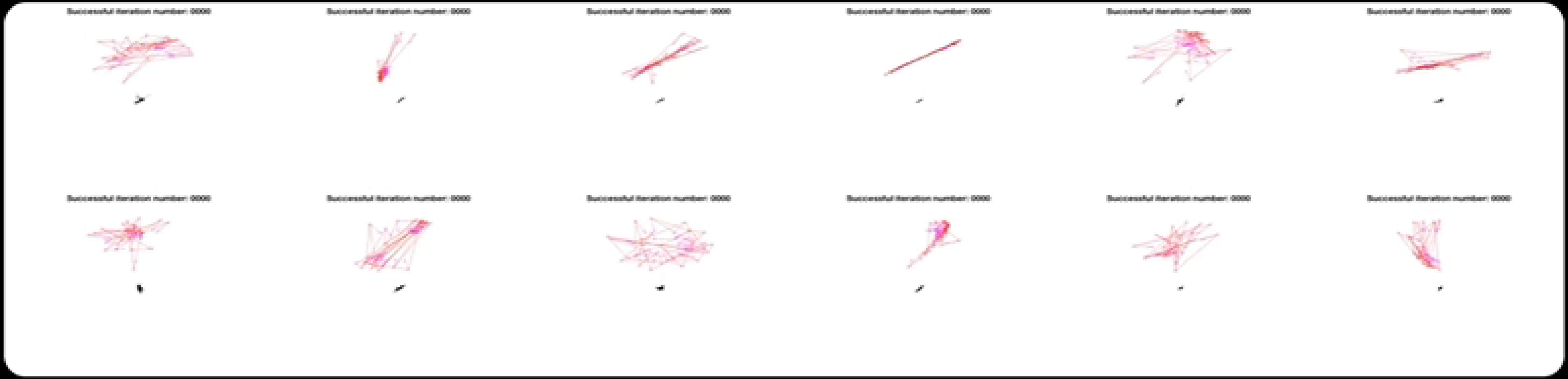


Srajer et al., AutoDiff '16: Benchmark

VarPro

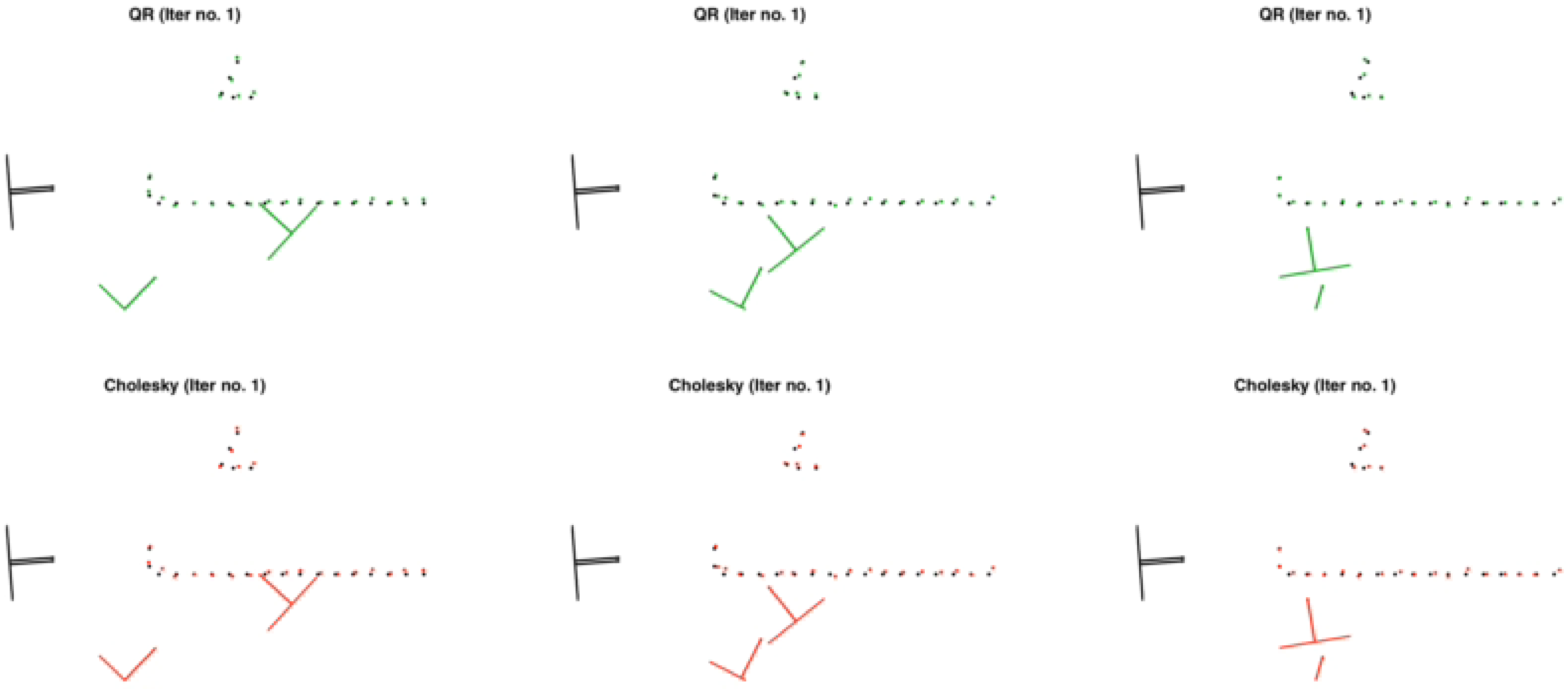


Joint+EPI



HONG ET AL, CVPR'17: BUNDLE ADJUSTMENT

Microsoft



SVOBODA ET AL. 2018: LM IN SINGLE PRECISION

- Use discriminative machine learning (e.g. DNNs) to get near an optimum (e.g. down to 10 pixels)

- Use model fitting to get quality solutions (e.g. down to 0.1 pixels).

CONCLUSION

Microsoft

Clichés I want you to stop using

- “Non convex optimization is slow”
- “There’s no point in getting doing better than 1% off the optimum”
- “There’s no point in optimizing my code”
- “Bundle adjustment needs a good initialization”