Kalman Filtering for arcjetCV

Kalman Filtering and RTS Smoothing for Arc-Jet Sample Edge Tracking

Ablation Workshop 2026
September 2026

Alexandre M. Quintart
Flying Squirrel

Magnus A. Haw
NASA Ames Research Center

Table of Contents

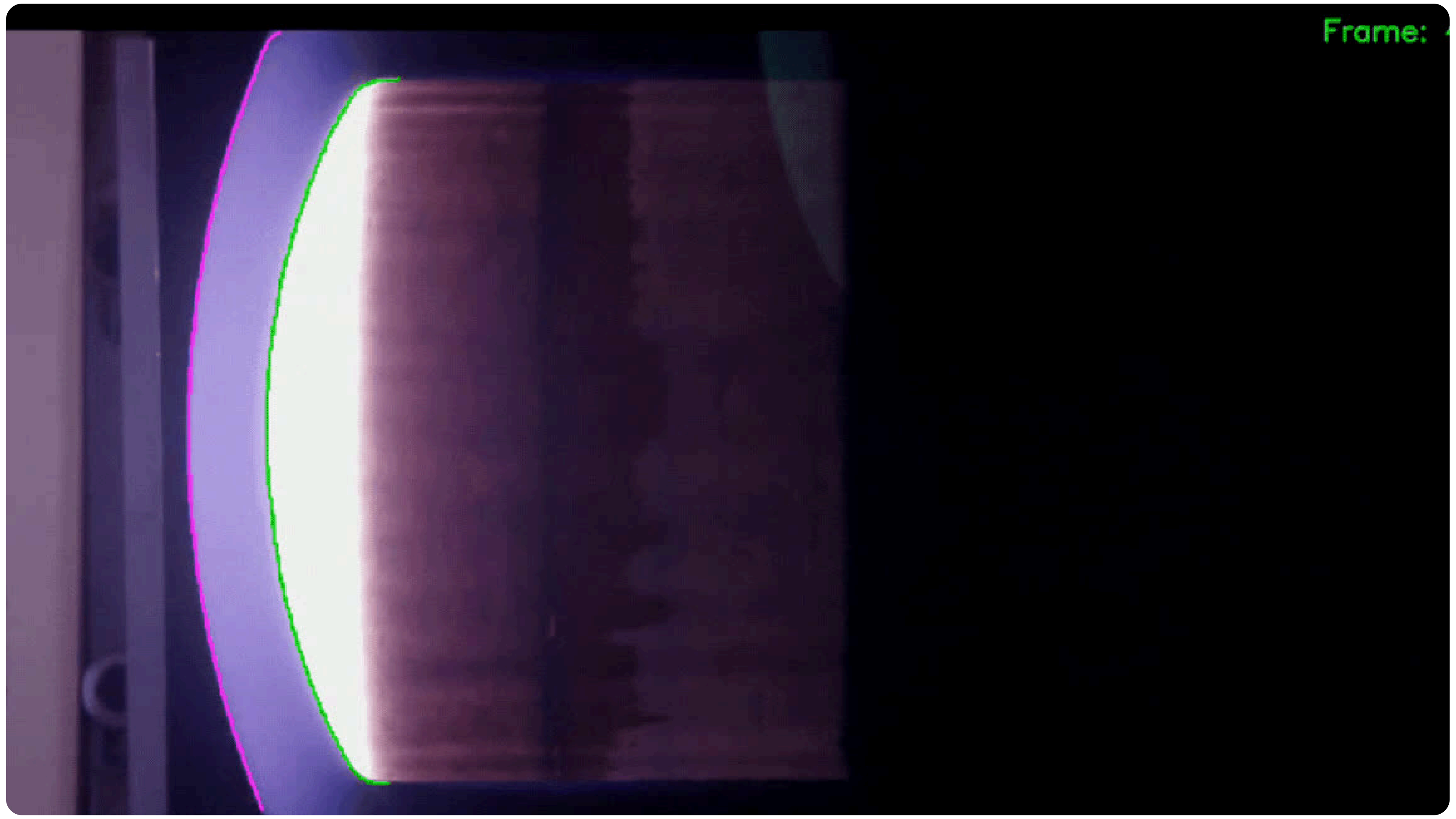
Introduction

Recession is a critical TPS material performance

BUT

- Recession is typically small (mm) & hard to measure in-situ
- Critical for material model validation

arcjetCV → automate in-situ recession measurements from video

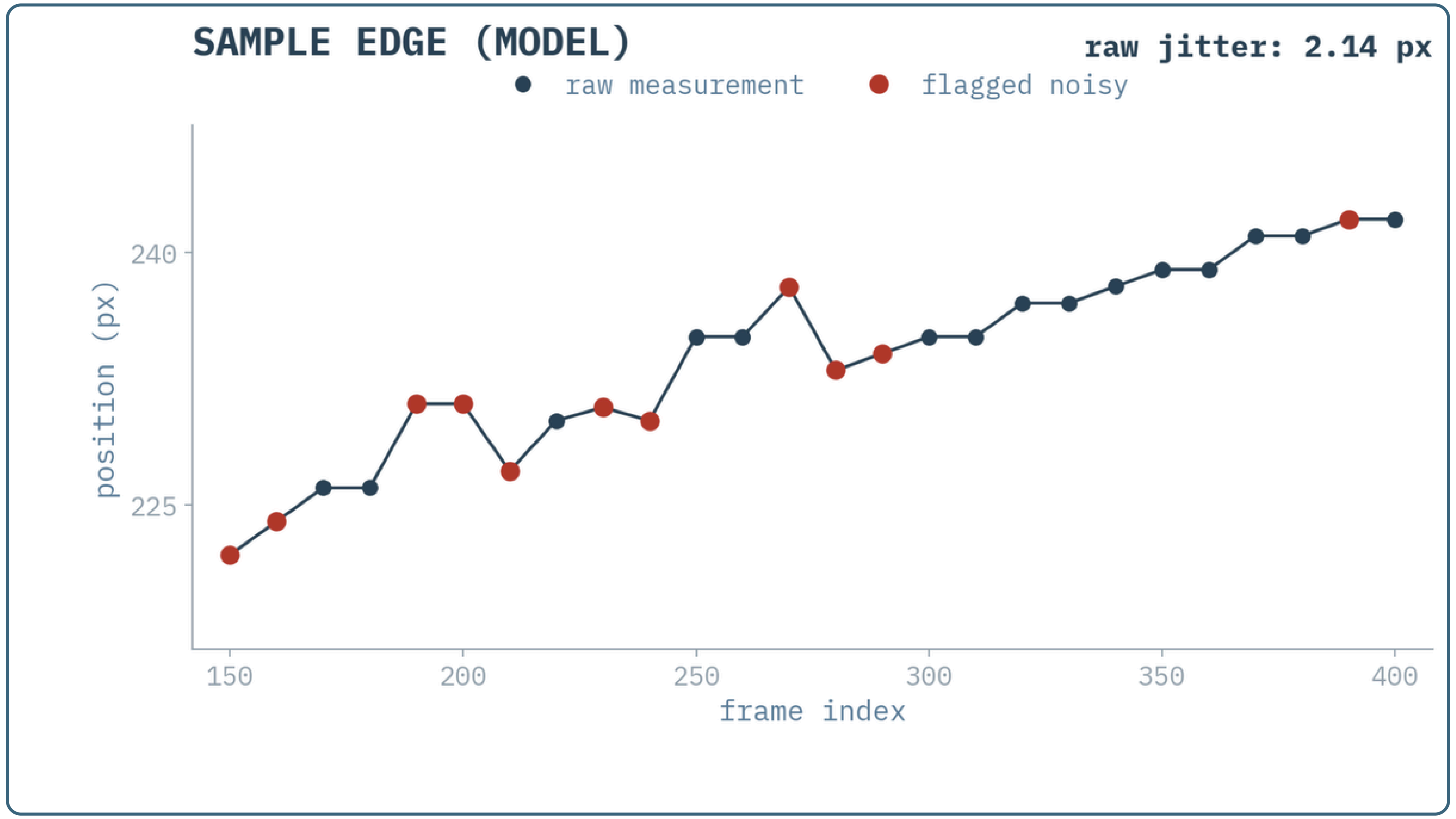


Recession is a Time-Series Problem, Not a Per-Frame One

arcjetCV segments the sample and shock **independently** frame by frame, with no notion of a smooth trajectory in time.

This causes:

- Frame-to-frame **jitter**, even when the physical recession is smooth
- Dropped frames from glare or plume occlusion break the sequence
- Noisy recession-rate and shock-standoff estimates propagate downstream



2.14 px
raw jitter · sample edge

0.76 px
raw jitter · shock

≈ 10%
Total recession · sample edge

≈ 4%
Total recession · shock edge

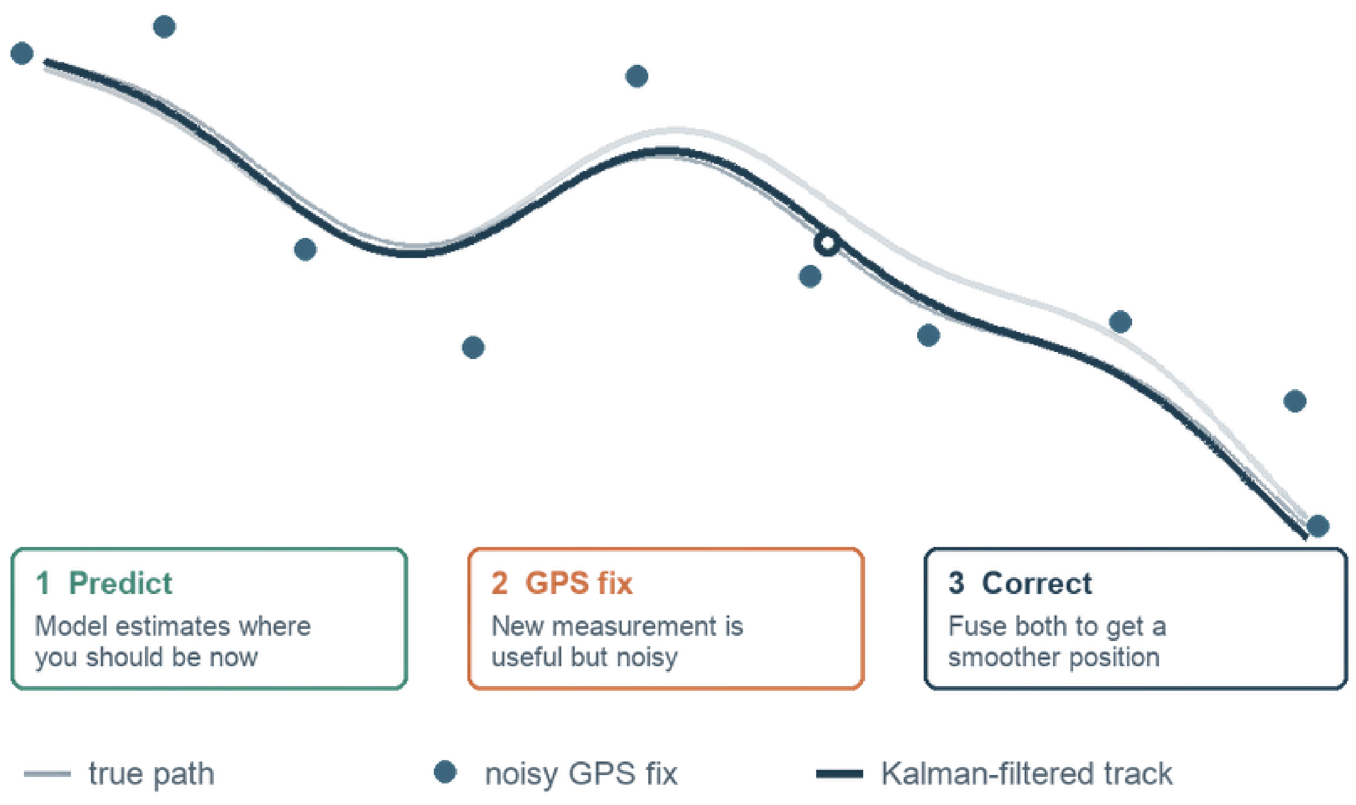
> **Edge tracking is a time-series problem, not a per-frame classification problem.**

Reducing Pixel Noise: The Same Idea as Phone GPS

Your phone does not rely on GPS alone. A GPS measurement is useful, but noisy: buildings, atmosphere, signal and reflections can make the position jump by several meters.

- **Predict:** estimate your current position from the previous position, speed, and direction
- **Correct:** compare that prediction with the new GPS measurement
- **Result:** a smoother path than GPS alone, with reasonable motion between updates

Noisy fixes are corrected by a motion model

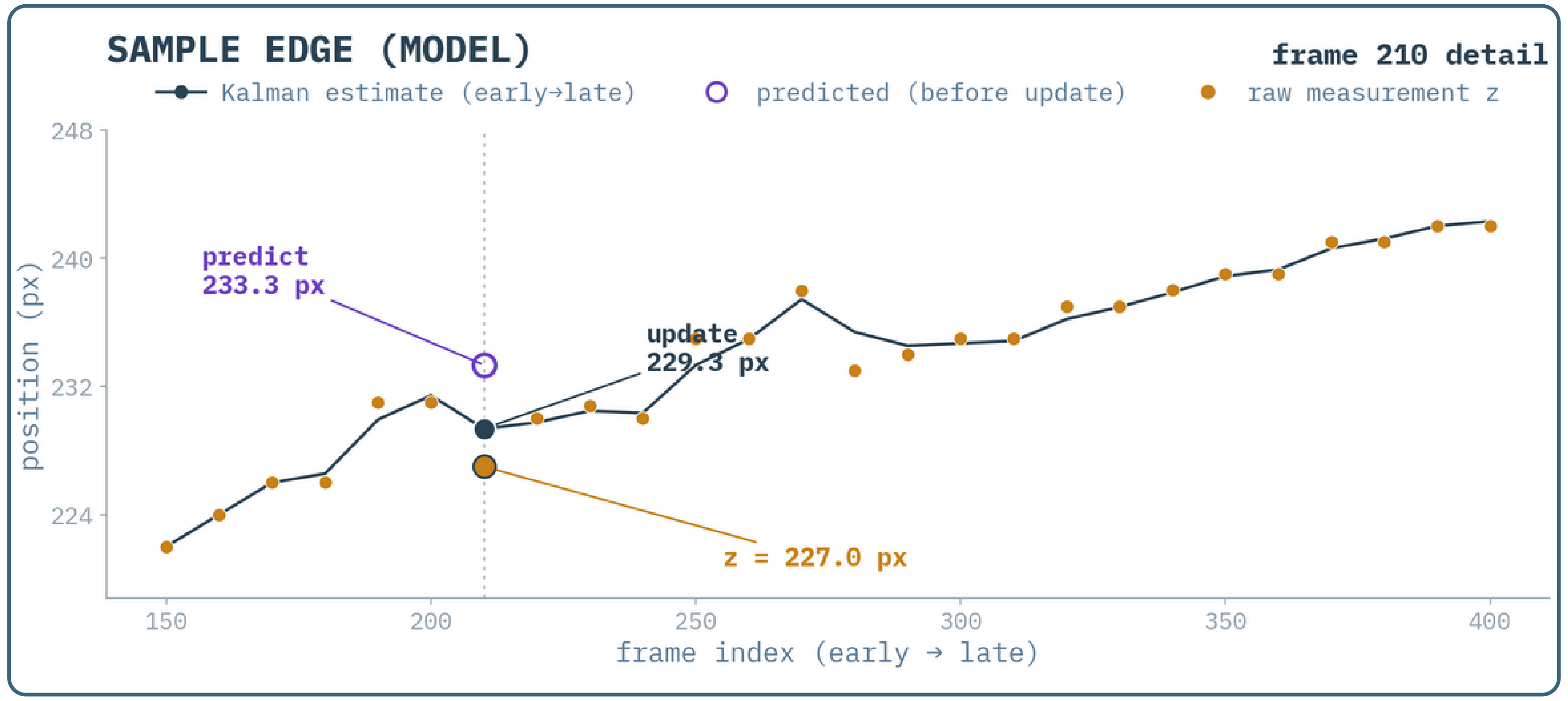


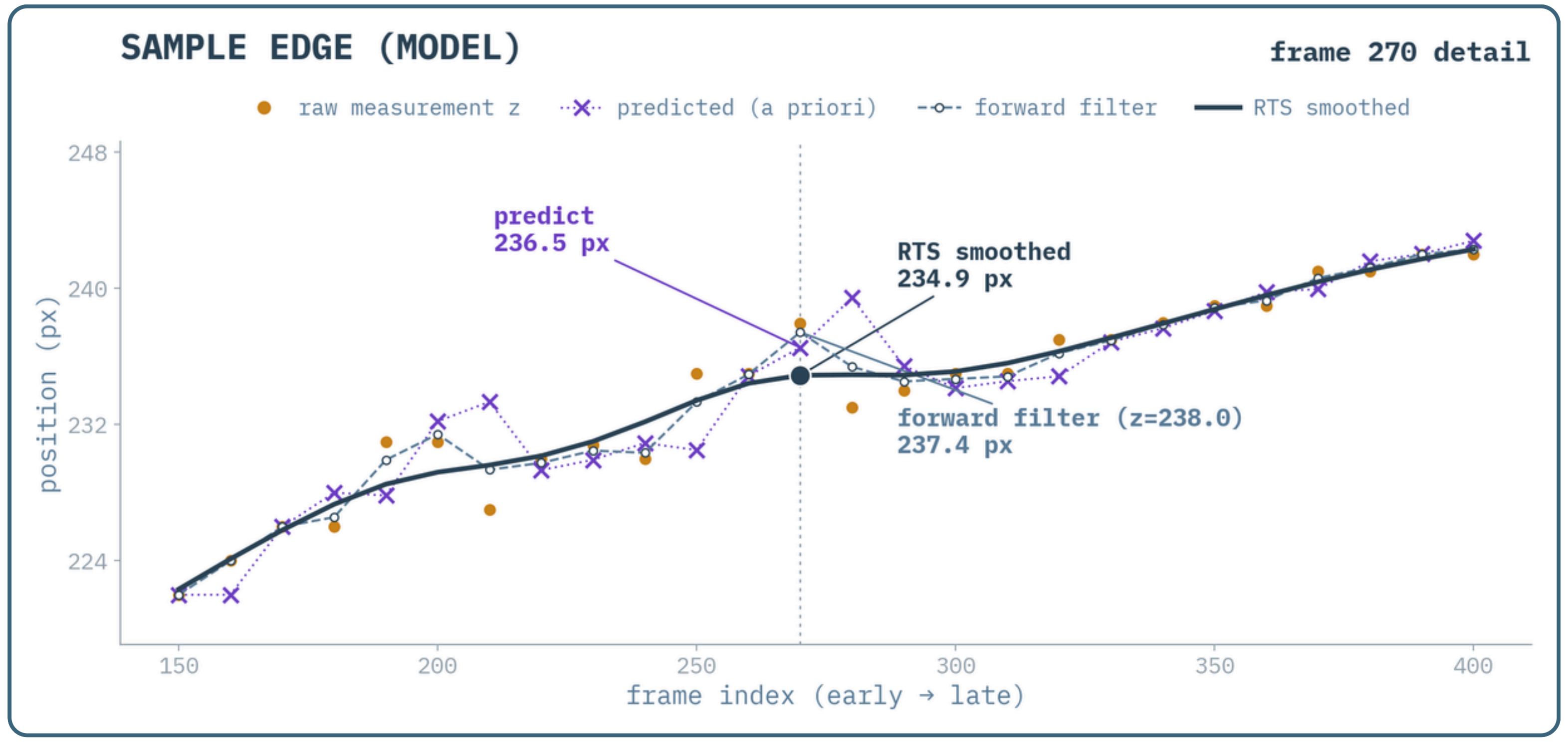

```
Same recipe as arcjetCV: state = [position, velocity].
Predict with the motion model, then correct with the noisy measurement.
```

Kalman Filter + RTS Smoother

Building a Kalman filter means specifying three things: a system model, a measurement model, and a noise estimation. RTS = Rauch–Tung–Striebel, the standard backward smoother pass.

- **System model:** state = [position, velocity], constant-velocity dynamics predict each frame's position from the last
- **Measurement model:** the noisy per-frame segmentation is a direct observation of position
- **Noise estimation:** how much to trust the model vs. the data, then fuse the two, forward in time, backward with RTS





Why Not SAM3 or a Temporal CNN?

Kalman filtering **improves** measurement quality because:

- Explicit model of recession dynamics, **not a black box**
- Quantified position uncertainty
- Quantified recession rate and its uncertainty
- Uses **past** and **future frames** to improve every estimation
- Propagates uncertainty into later recession measurements.
- Computationally cheap
- A clean route to multi-camera 3D state estimation

SAM3 = Segment Anything Model version 3
CNN = Convolutional Neural Network

Point-wise vs Shape-Parameter

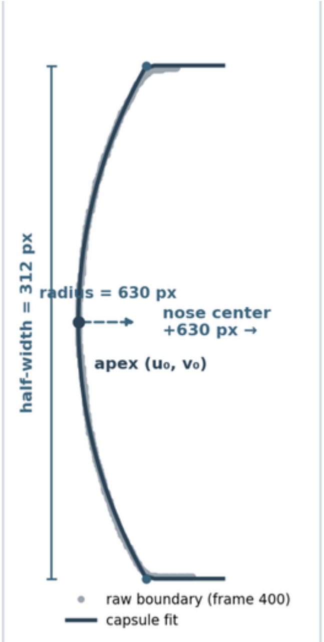
Point-wise

- Resample each segmented edge to 61 correspondence points (arc-length parameterization)
- Track all 2 × 61 = 122 coordinate channels independently, in parallel
- No geometric assumption: **works for any shape** (hemisphere, ISO-Q, shock front)

Shape-parameter

- Reduce each segmented edge to a few geometry-dependent parameters
- Capsule samples: circular-arc nose blending into flat sides (nose center, apex, radius, half-width) → 4 channels
- Shock boundary: radial profile with a offsets, geometry-agnostic

61 Points

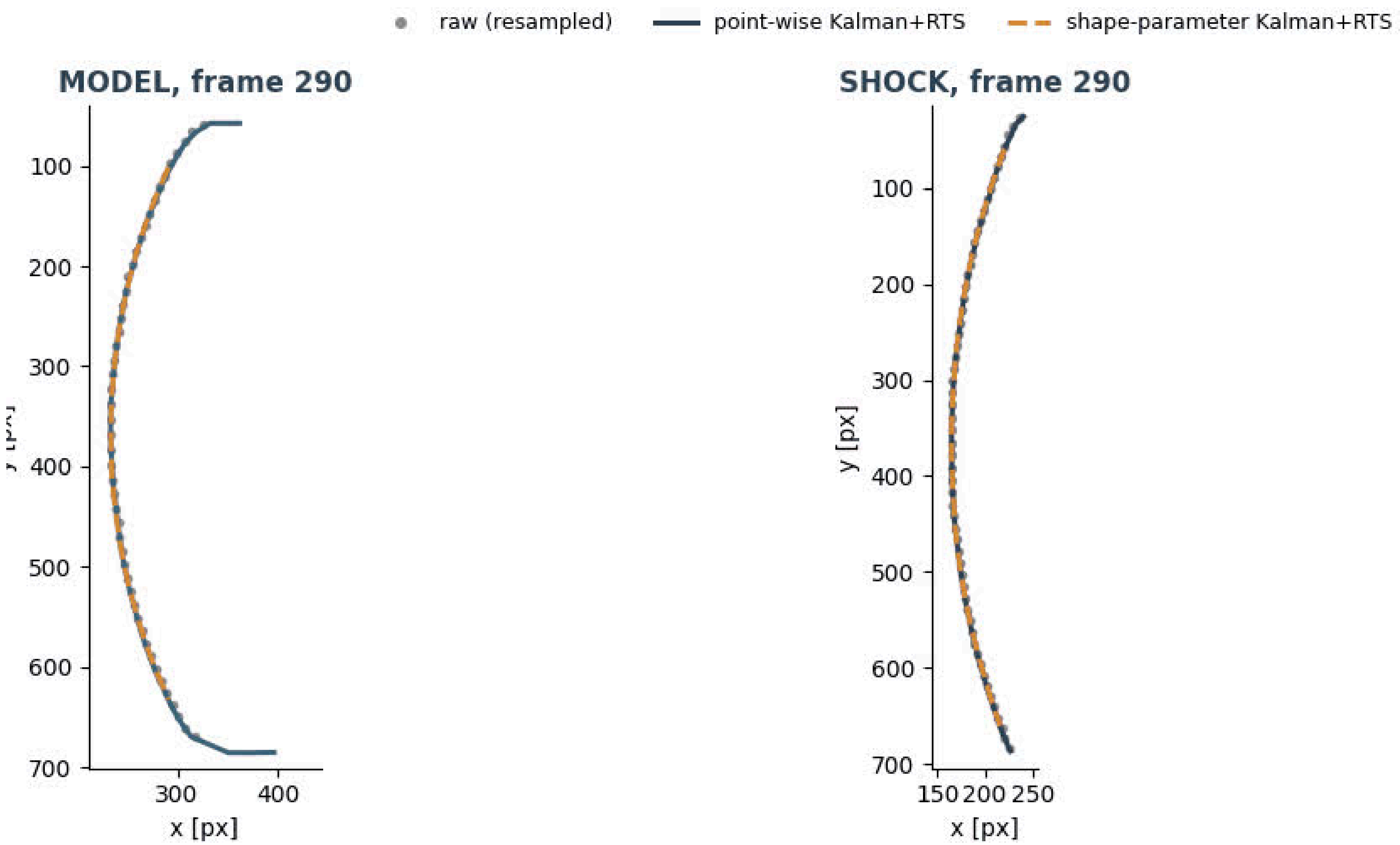


Capsule fit - 4 channels

> **Other approaches are possible and should be explored.**

Testing on a Real Arc-Jet Video

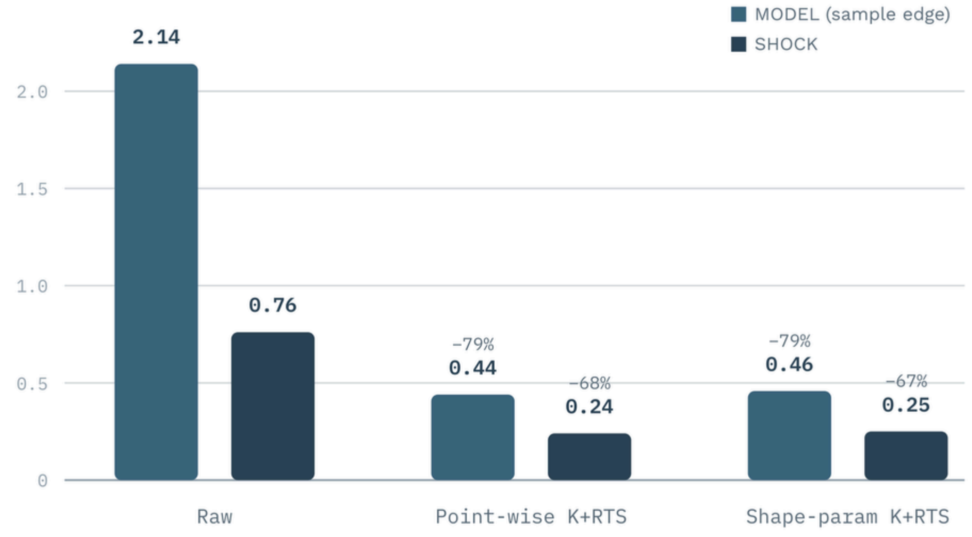
- Input: arcjetCV per-frame segmentation output, frames 150 to 400 of a real test
- Sample edge tracked with:
  - the capsule (arc-nose) shape-parameter geometry;
  - point-wise (61 correspondence points) for comparison
- Metric: frame-to-frame jitter of the stagnation-point tracking, defined as the standard deviation of the frame-to-frame position difference.



Real tracking output on the validation video: source frame with overlaid boundaries (left) and the reconstructed MODEL and SHOCK boundaries from both filtering approaches (right), animated across frames 150 to 400.

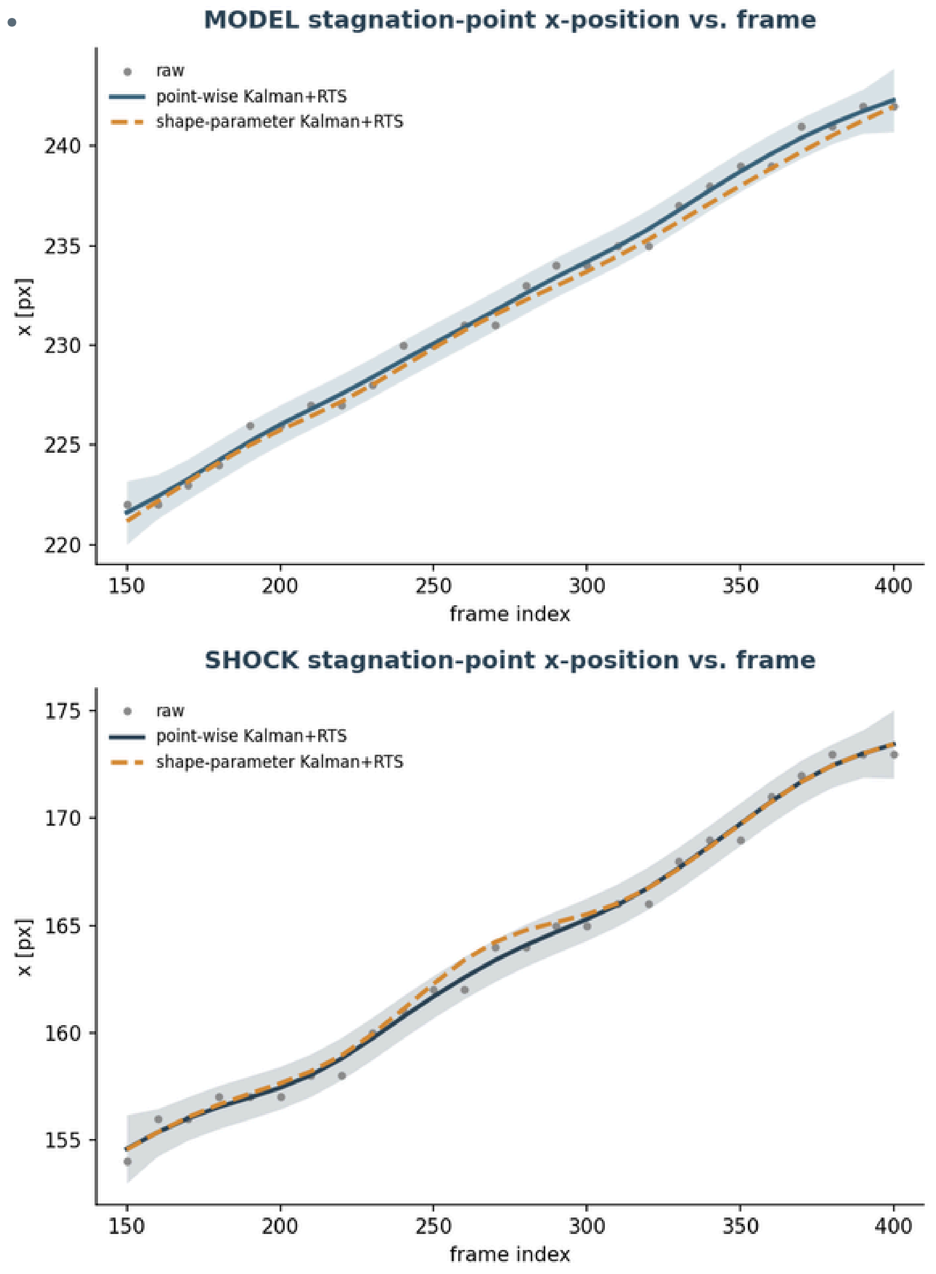
Jitter Reduction

Frame-to-frame jitter at the stagnation point is measured as the standard deviation of position changes between consecutive frames. Kalman + RTS reduces jitter on both the sample edge and shock boundary, using the same video data and no extra hardware.



Method	MODEL (sample edge)	SHOCK
Raw	2.14 px	0.76 px
Point-wise K+RTS	0.44 px (−79%)	0.24 px (−68%)
Shape-param K+RTS	0.46 px (−79%)	0.25 px (−67%)

Stagnation-point jitter (px, std of frame-to-frame difference)



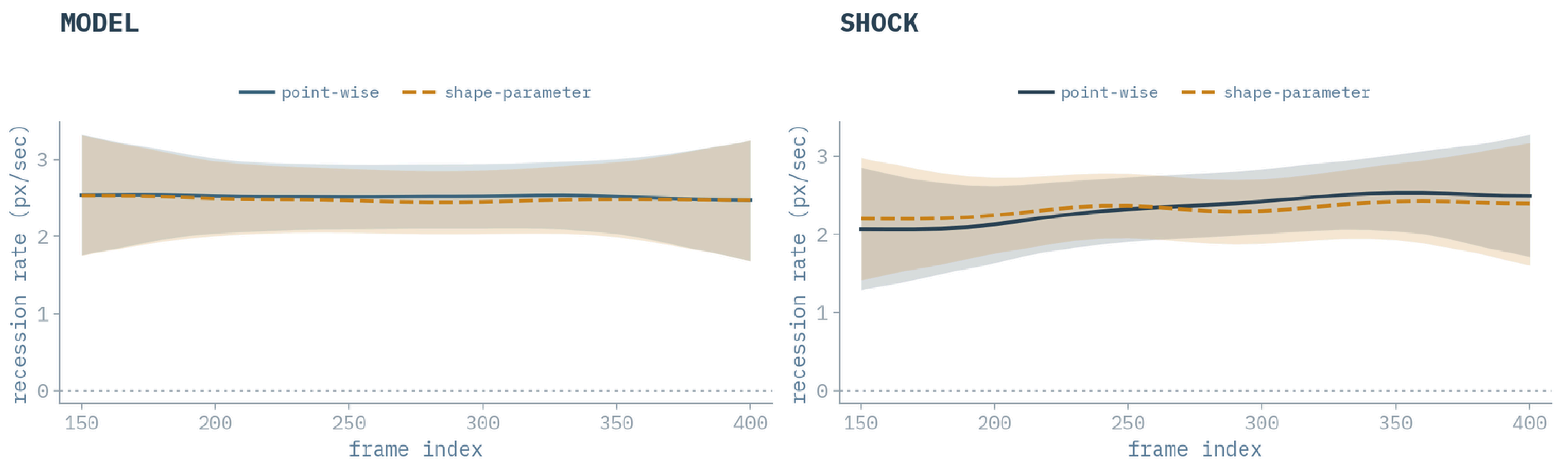
Real data behind the jitter numbers: raw stagnation-point position is noisy frame to frame; both Kalman + RTS variants track a smooth trend.

Recession Rate, With Uncertainty

Recession Rate Comes Directly From the Kalman filter
The Kalman filter is not only position. It is:
state = [position, velocity]
That velocity term gives the recession rate directly from the RTS-smoothed estimation.

Why this matters

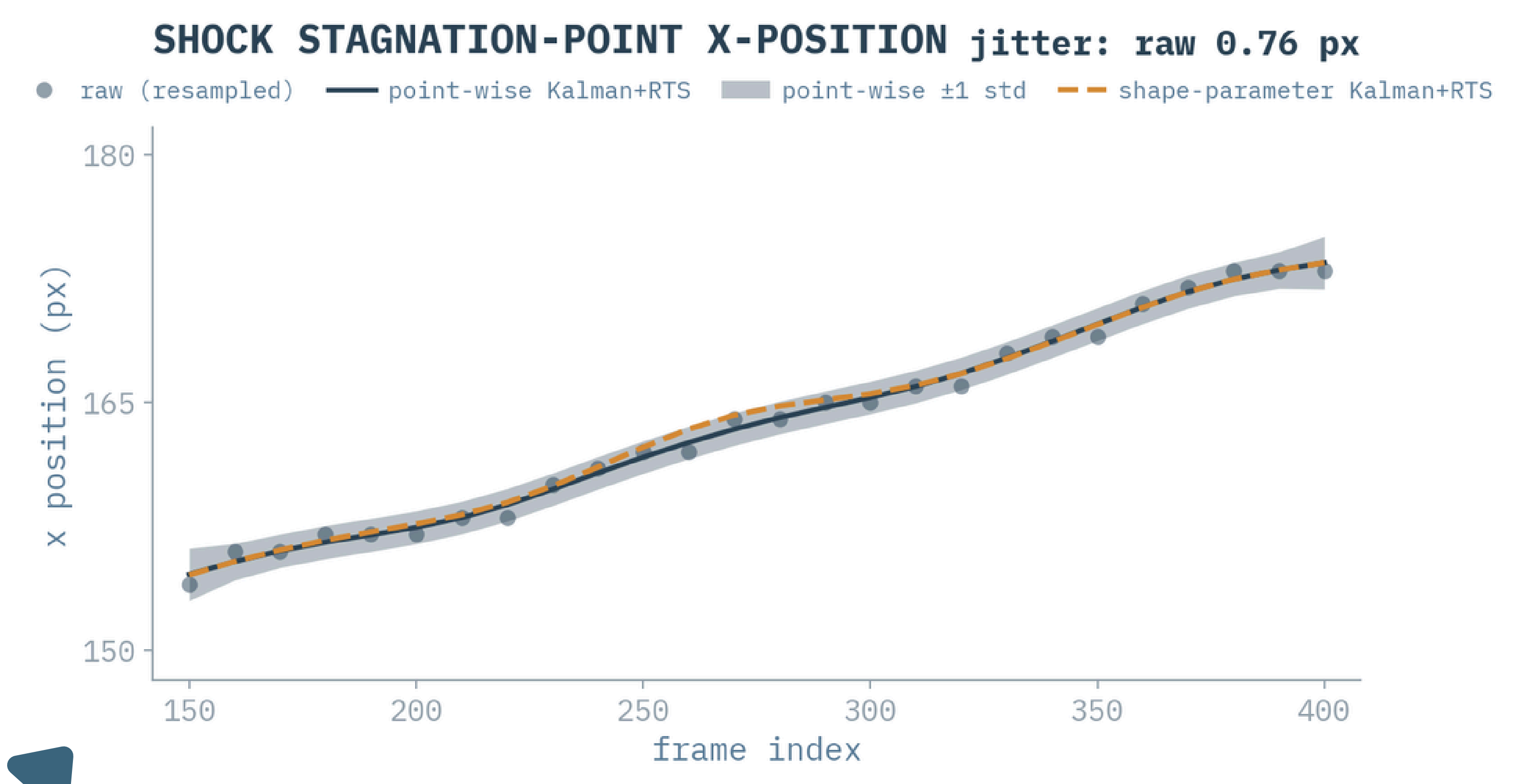
- It is not a manual frame-to-frame slope.
- It comes with its own ±1σ (standard-deviation) uncertainty band from the Kalman/RTS covariance.
- The uncertainty is propagated automatically by the filter, not added afterward.



Built to Deal with Bad Frames

Robustness

- Missing or dropped frames are handled: the filter **keeps predicting through** gaps instead of breaking the sequence.
- Two noise settings control how much the filter trusts the **motion** model versus the **raw segmentation**.
- The same filter settings are used for all tracked points, so the method stays simple and consistent.
- Filtering reduces both frame-to-frame uncertainty and the observed nonlinearity in the raw segmentation

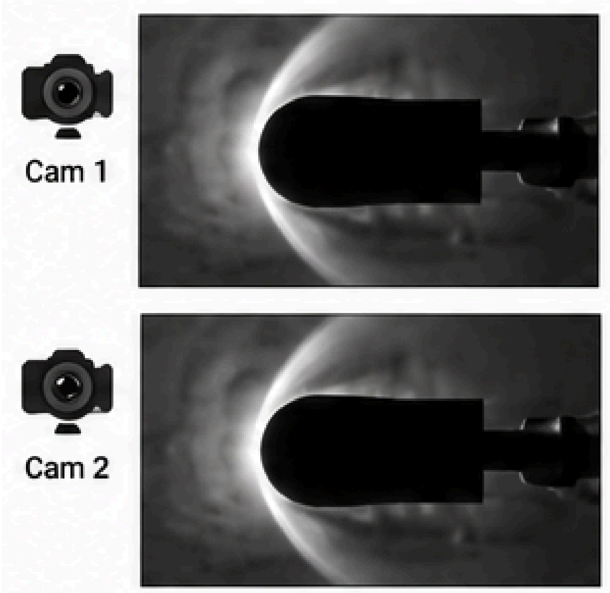


> **Is that nonlinearity real sample-response physics, or sting vibration ?**

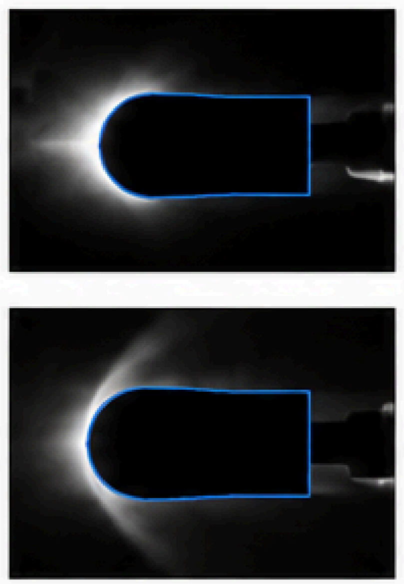
Kalman Filtering: Next Steps

1 · MULTI-VIEW VIDEO

Hemisphere-cylinder article



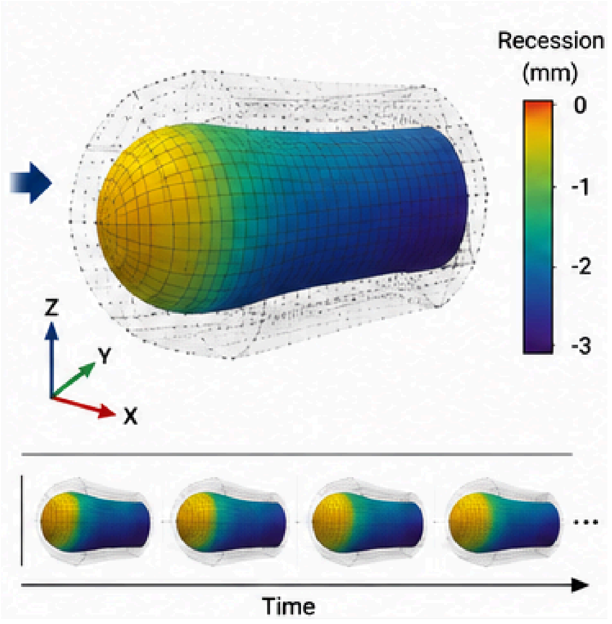
2 · EDGE / SEGMENTATION



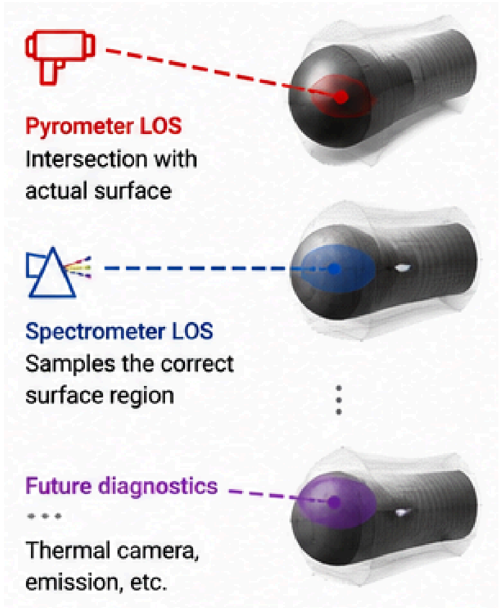
3 · KALMAN SURFACE ESTIMATOR

- Fuses observations across cameras & time
- Enforces temporal consistency
- Quantifies uncertainty in geometry

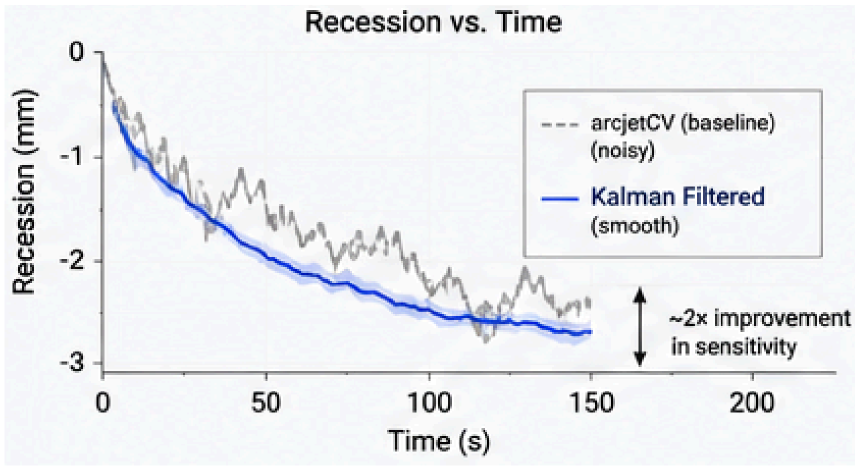
4 · EVOLVING 3D SURFACE (UQ)



5 · REGISTERED DIAGNOSTICS

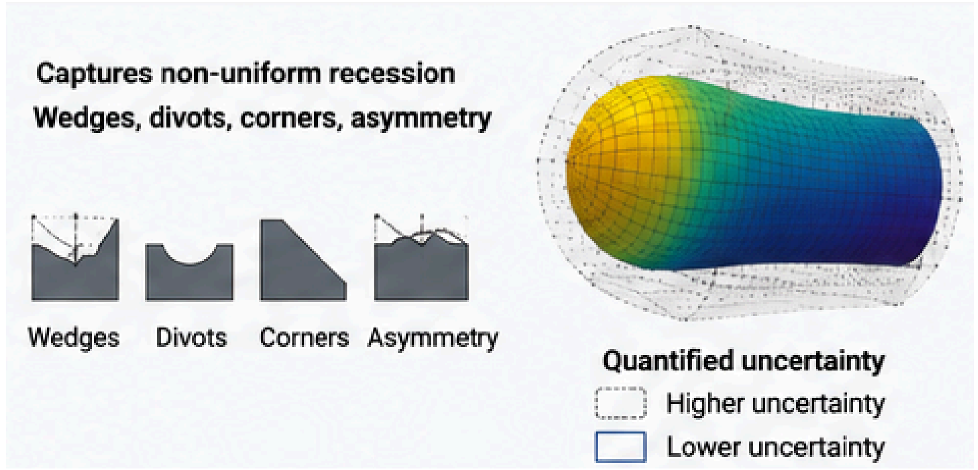


1. ~4× Recession Sensitivity



Reduces noise and reveals subtle recession. Detect smaller changes with confidence.

2. 3D Surface Mapping with UQ



Reveals 3D features that single-view tracking cannot see.

3. Common, Time-Evolving Geometry for Other Diagnostics

- ✓ **Spatially registered measurements**
- ✓ **Accurate LOS / surface intersection**
- ✓ **Enables quantitative analysis & comparison**

More accurate data. Lower uncertainty. Deeper insight into material response

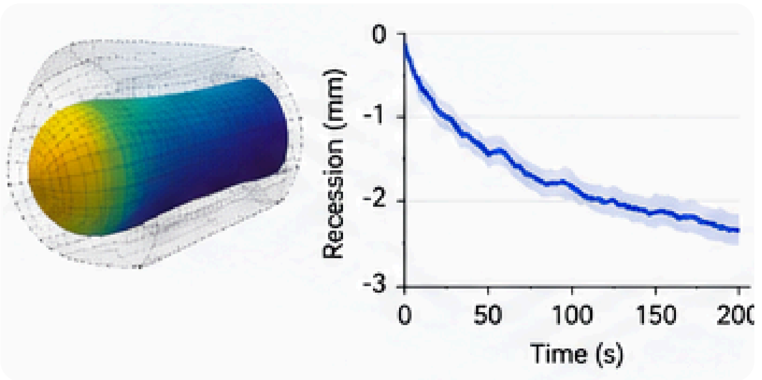
Kalman filtering has potential to transform multi-view images into an evolving, uncertainty-aware 3D surface.

Kalman Filtering Beyond arcjetCV

Once we know the **surface**, we can register **temperature** and other **diagnostics** to it and fuse recession, surface temperature, and internal temperature with the material model.

1 · Surface Recession vs. Time

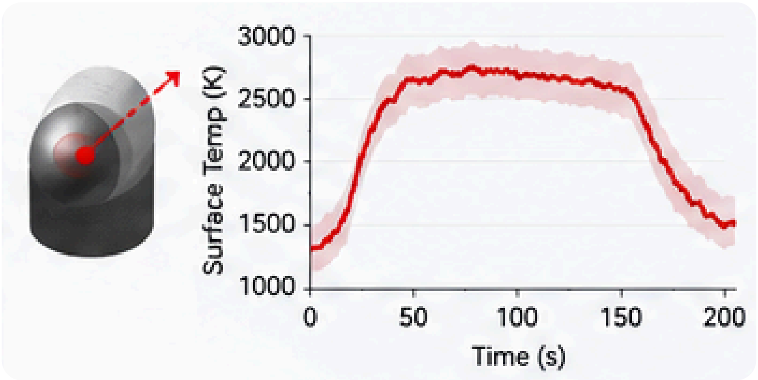
from multi-view arcjetCV



Geometry registered, time-evolving, with quantified uncertainty

2 · Surface Temperature vs. Time

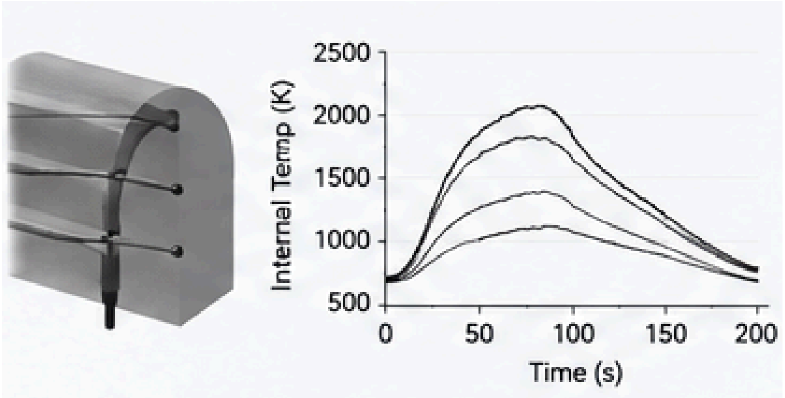
from pyrometer (registered)



Temperature on the actual surface, with uncertainty

3 · Internal Temperature vs. Time

Hemisphere-cylinder article



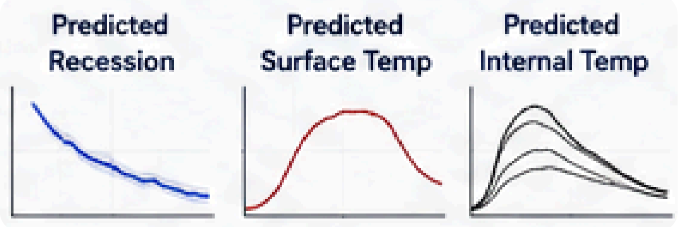
Temperature history within the material, with uncertainty

MATERIAL-RESPONSE MODEL + STATE ESTIMATION

Model Inputs

- Material properties (k, ρ, C_p
- Ablation / recession model
- Boundary conditions (heat flux, pressure, chemistry, radiation)
- Geometry (from surface state)

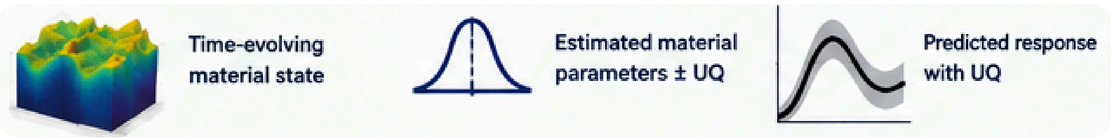
Predicted Recession



Data Assimilation (update)

- Compare predictions to all three observables
- Update states and uncertain parameters
- Propagate uncertainty

OUTPUT: INFERRED MATERIAL STATE & PARAMETERS WITH UNCERTAINTY



Enhanced material model validation & extrapolation
Quantitatively compare predictions and measurements across conditions.

WHAT THIS ENABLES

Self-consistent parameter estimation
Jointly infers uncertain material properties and boundary conditions.

Quantified uncertainty
Posterior uncertainty on parameters and predicted material response.

Predictive capability
Predicts unmeasured quantities and future behavior with uncertainty.

Stronger extrapolation
Constraints improve confidence for new conditions and designs.

Decision support
Better data for material development, validation, and risk reduction.

Conclusion

Recession is critical and hard to measure
arcjetCV turns noisy video segmentation into usable recession and shock-standoff time series.
Kalman filtering adds the missing time model: smooth the signal, explicit uncertainty, and with zero new hardware.

UQ
In-Situ recession UQ reduction

4x
jitter reduction · sample edge

0
new hardware

- Kalman + RTS reduces frame-to-frame jitter
- Works directly on arcjetCV's existing per-frame segmentation output
- Produces cleaner recession-rate and shock-standoff histories for material-model validation
- Supports two modes: point-wise tracking for general shapes, shape-parameter tracking when geometry is known
- Gives uncertainty on each measurement, so downstream models can use error bars instead of a single clean curve
- The same predict/update recipe extends naturally to multi-view 3D models, and other state estimation

arcjetCV Is Open Source

arcjetCV is free and open source today. The Kalman filter and RTS smoother extension shown in this talk is on its way into the main package, coming soon.

> **github.com/magnus-haw/arcjetCV**

Thank You for Your Attention

Team
Alexandre M. QuintartFlying
Squirrelalex@flying-squirrel.space
Magnus A. HawNASA Ames Research
Centermagnus.haw@nasa.gov

Funding
Mars Sample Return Project
Entry Modeling & Instrumentation Project
TSM Branch IRAD

Code
github.com/magnus-haw/arcjetCV

Thank you for your attention!

Team
Alexandre M. Quintart — Flying Squirrel — alex@flying-squirrel.space
Magnus A. Haw — NASA Ames Research Center — magnus.haw@nasa.gov

Funding
Mars Sample Return Project
Entry Systems Modeling Project
TSM Branch IRAD

Code
github.com/magnus-haw/arcjetCV

Conclusion

Kalman + RTS turns noisy frame-by-frame segmentation into a physically consistent time series.

It reduces jitter without changing hardware or the CNN, then carries uncertainty into recession rate, shock standoff, and downstream material-response models.

The same recipe can scale from 2D tracking to multi-view 3D geometry and registered thermal diagnostics.

Kalman + RTS, in Full

1 · SYSTEM MODEL position + velocity

state $x = [\text{ position, velocity }]^T$

model $x^- = F\,x$ constant-velocity model

2 · MEASUREMENT MODEL what the segmentation gives us

measurement $z = H\,x$ H = [1, 0]: observe position only

3 · NOISE ESTIMATE how much to trust each

process Q trust the model more → larger Q

measurement R trust the data more → smaller R

forward filter, each frame t

predict $x^-,\quad P^- = F\,P\,F^T + Q$

gain $K = \dfrac{P^-}{P^- + R}$ Kalman gain

update $x = x^- + K\,(z - x^-)$ fuse with measurement z

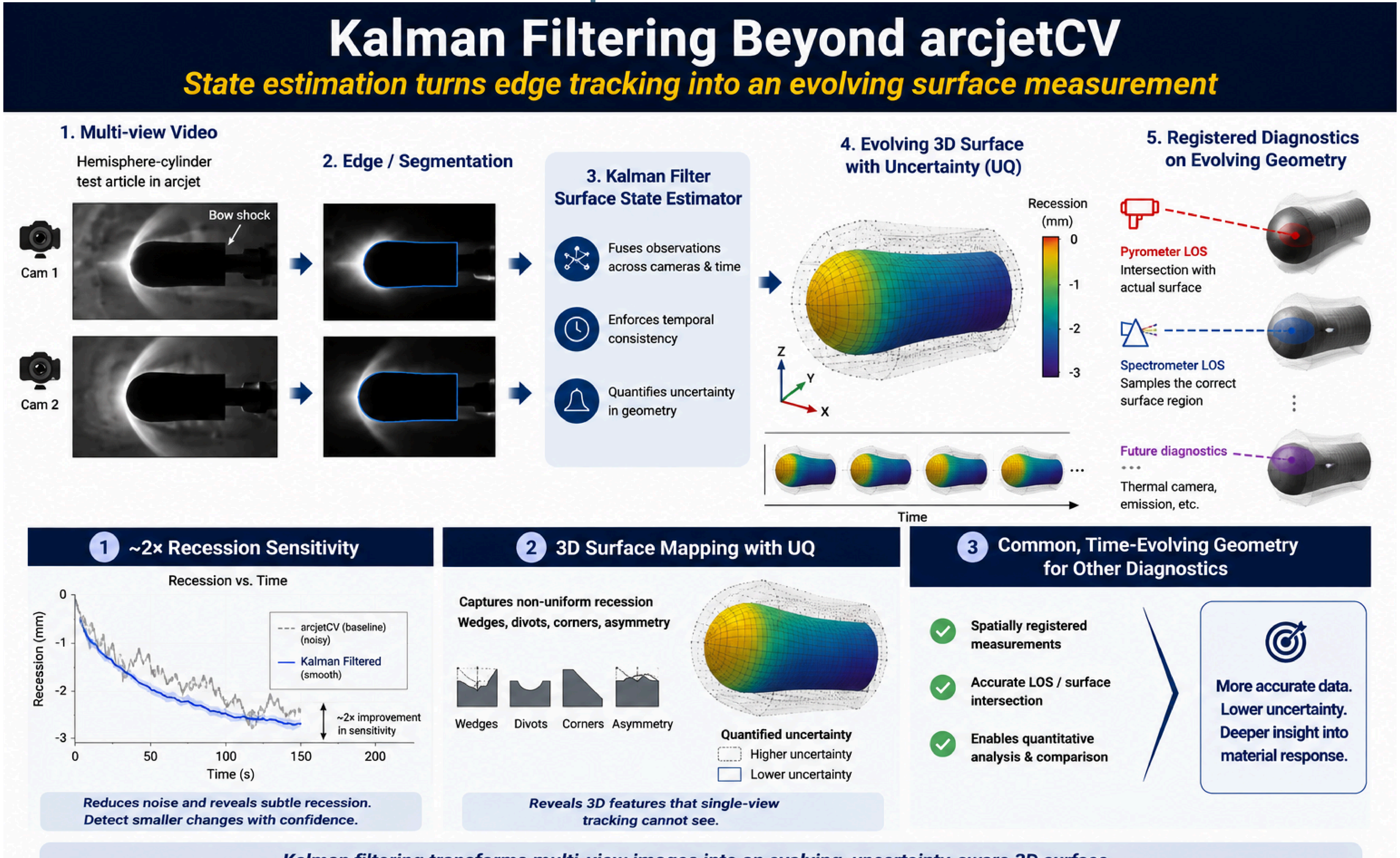
backward RTS pass, t = T−1 → 0

smooth gain $C_t = \dfrac{P(t)\,F^T}{P^-(t+1)}$

smooth $x_{smooth}(t) = x(t) + C_t\,[\,x_{smooth}(t+1) - x^-(t+1)\,]$

Q = process noise, how much to trust the model. R = measurement noise, how much to trust the data.

Robustness & Next Steps



Kalman + RTS, in Full

1 · SYSTEM MODEL position + velocity

state $x = [\ \text{position},\ \text{velocity}\ ]^T$

model $x^- = F\,x$ constant-velocity model

2 · MEASUREMENT MODEL what the segmentation gives us

measurement $z = H\,x$ H = [1, 0]: observe position only

3 · NOISE ESTIMATE how much to trust each

process Q trust the model more → larger Q

measurement R trust the data more → smaller R

Too fix

forward filter, each frame t

predict $x^-,\ P^- = F\,P\,F^T + Q$

gain $K = \dfrac{P^-}{P^- + R}$ Kalman gain

update $x = x^- + K\,(z - x^-)$ fuse with measurement z

backward RTS pass, t = T−1 → 0

smooth gain $C_t = \dfrac{P(t)\,F^T}{P^-(t+1)}$

smooth $x_{smooth}(t) = x(t) + C_t\,[\,x_{smooth}(t+1) - x^-(t+1)\,]$

Q = process noise, how much to trust the model. R = measurement noise, how much to trust the data.

Backup · How the Jitter Value Is Computed

Jitter is the spread of frame-to-frame jumps, not the maximum jump.

Definition

For each frame, take the stagnation-point position:

```
x_t,   t = 1, ..., T
```

Compute the frame-to-frame jump:

```
Delta x_t = x_t − x_{t−1}
```

Then compute the population standard deviation:

```
jitter = sqrt( (1/N) sum_t (Delta x_t − mean_Delta)^2 )
N = T − 1  (25 frame−to−frame jumps here)
```

Raw jitter values

Sample edge	**2.14 px**
Shock	**0.76 px**

Same calculation for both boundaries.

Interpretation

Higher jitter means more irregular frame-to-frame motion.
A smooth physical recession gives nearly constant jumps, so the standard deviation stays small.

Backup · Where the 4.9× Jitter Reduction Comes From

Same metric, same channel: raw segmentation vs. Kalman + RTS output

Meaning

The ratio answers: how many times smaller did the frame-to-frame jitter become after filtering?

Raw jitter

2.14 px

Unfiltered sample-edge position
std(diff(raw_x))

→

Filtered jitter

0.44 px

Point-wise Kalman + RTS
std(diff(smoothed_x))

→

Reduction ratio

4.9×

Raw jitter ÷ filtered jitter
2.14 ÷ 0.44 = 4.86 ≈ 4.9×

Ratio form

2.14 px / 0.44 px = 4.86 ≈ 4.9×

The filtered jitter is about 4.9 times smaller than the raw jitter.

Percent form

1 − 0.44 / 2.14 = 0.794 ≈ 79%

The same result expressed as percent reduction instead of a multiple.

Takeaway: 4.9× smaller jitter and 79% reduction are two ways of saying the same thing.

Backup · How the ±1σ Error Band Is Computed

The shaded band comes from the Kalman/RTS covariance, not from a post-hoc fit.

What the band means

At each frame, the line is the smoothed position estimate.

```
band = position ± σ
```

The shaded area is a ±1 standard-deviation uncertainty interval.

Where σ comes from

```
state: x = [position, velocity]
```

The filter keeps a 2×2 covariance matrix P for each tracked point.

```
σ = sqrt(P[0,0])
```

P[0,0] is the position variance after RTS smoothing.

Not this

- × Not simulated
- × Not bootstrapped
- × Not hand-drawn

It is filter bookkeeping.

How covariance P changes

1 Forward Kalman filter

```
Predict:  P_minus = F P F^T + Q   (uncertainty grows)
Update:   P = (I − K H) P_minus   (uncertainty shrinks)
```

If a segmentation frame is missing, the update is skipped; P grows through the gap.

2 Backward RTS smoother

Uses the whole sequence, including future frames, to tighten P where later data adds information.

What controls band width

```
process_var = 1.0 px²
```

Model uncertainty between frames. Larger means less smoothing.

```
measurement_var = 4.0 px²
```

Raw segmentation uncertainty. Larger means more weight on the model.

Practical reading

Narrow band: dense, consistent measurements. Wide band: noisy or dropped frames.

Takeaway: the ±1σ band reflects the filter's confidence in each position estimate.