Smooth Animations to Visualize Gaussian Uncertainty

Charles R. Hogg III
Google, Inc.

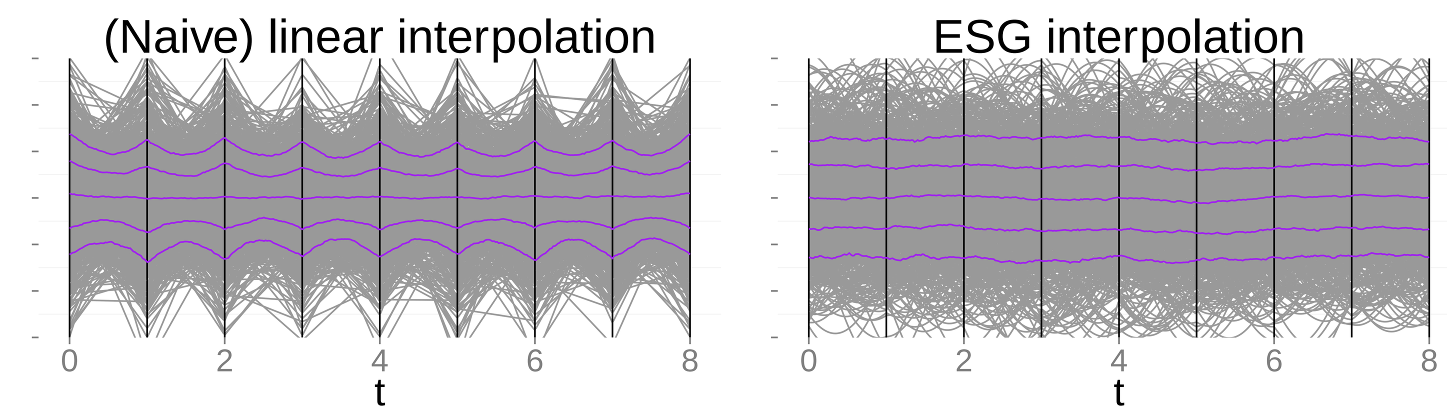
Google

Introduction

- **Goal:** Visualize uncertainty in *curves and surfaces*
  - Specifically: using **Gaussian processes** (see refresher at bottom)
- **Approach:** animations
  - Each frame shows one draw from posterior
  - Consecutive frames show similar curves (i.e., *continuous* animations)
  - *Reducible*: find **single "Gaussian oscillator"**; use copies as needed
- **New Results:**
  - **Smooth, keyframe-free** animations
  - **New framework** for all future work in Gaussian animations

Existing approach: interpolate between I.I.D. Gaussian draws

Linear interpolation: **variance too small** between keyframes



- Ehlschlaeger, Shortridge, Goodchild (**ESG**) solved in 1997 (see right figure)
- Problem with *both* approaches: keyframes are 'special'
  - Motion changes discontinuously
  - Even at $\Delta t = 1$, correlation can be surprisingly high (up to 0.5)

New approach: eliminate keyframes entirely

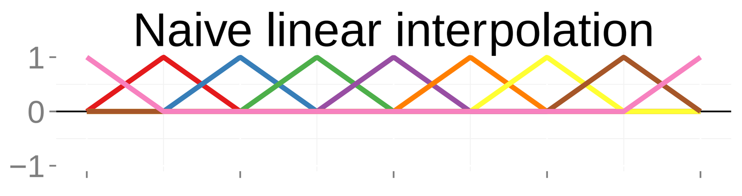
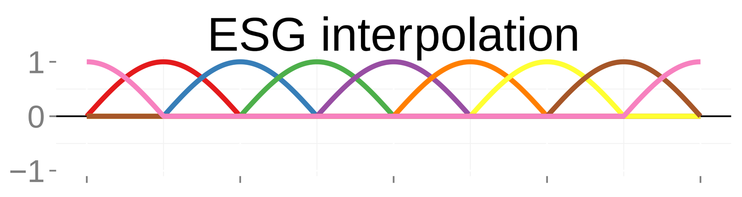
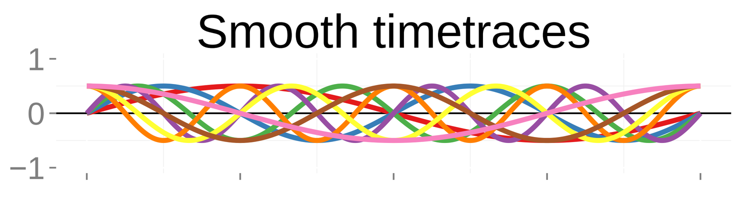
- Still use I.I.D. normals, $\{\epsilon_i\}$, but *de-localize* rather than interpolate

$$f(t) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} \left[\epsilon_{2i-1} \sin\left(\frac{\pi i t}{N}\right) + \epsilon_{2i} \cos\left(\frac{\pi i t}{N}\right) \right]$$

- Correct statistical properties: $\langle f(t) \rangle = 0; \quad \langle f(t)^2 \rangle = 1 \quad \forall t$

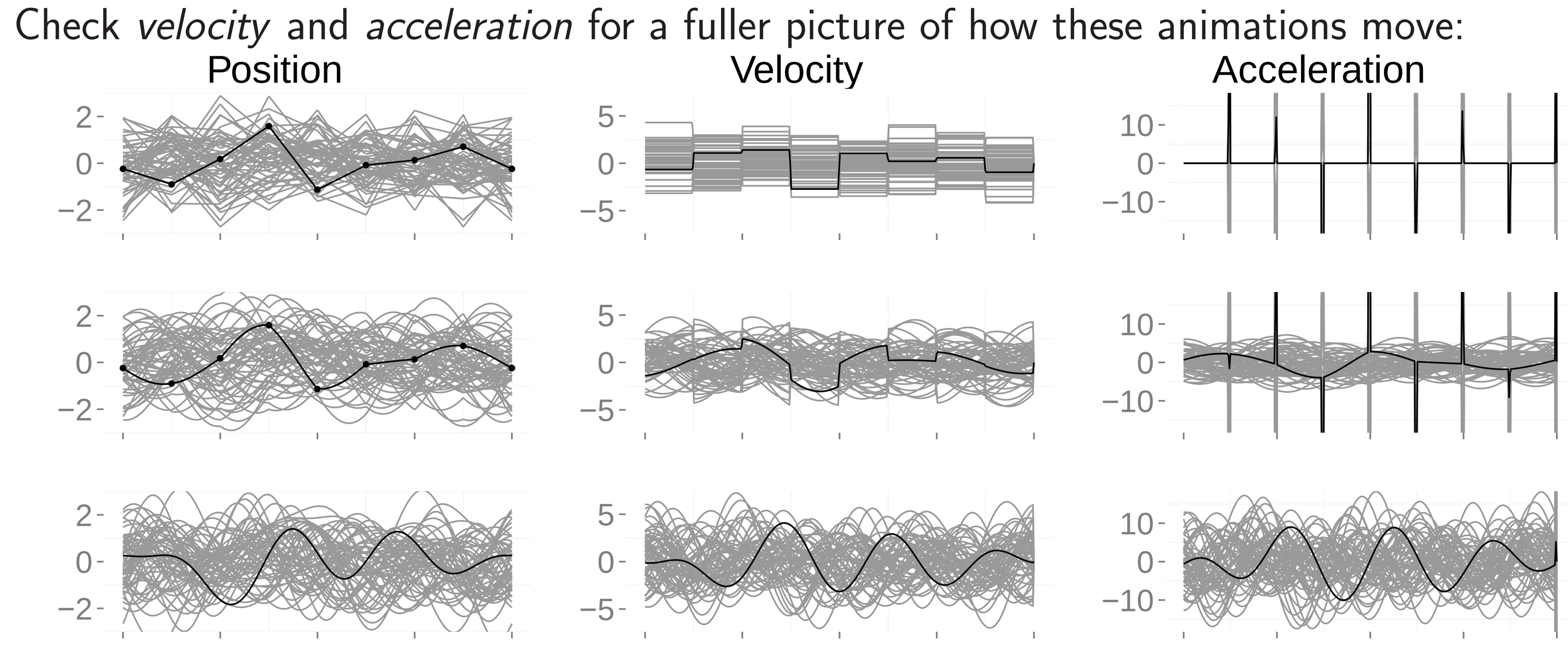
Basis function view

$$f(t) = \sum_{i=1}^{N} \epsilon_i b_i(t)$$

Animation Method	Statistically Correct	Stationary	Smooth
Naive linear interpolation 	✗	✗	✗
ESG interpolation 	✓	✗	✗
Smooth timetraces 	✓	✓	✓

Physical motion: basic kinematics

Check *velocity* and *acceleration* for a fuller picture of how these animations move:



Motion is *not different* at the keyframes **because they do not exist.**

The true nature of $f(t)$

- Observations about $f(t)$:
  - Infinite set of Gaussian random variables
  - Indexed by continuous variable, t
  - Well-defined covariance between every pair of points:

$$\langle f(t) f(t+\tau) \rangle = \frac{1}{N} \sum_{i=1}^{N} \left[\cos\left(\frac{\pi i \tau}{N}\right) \right]$$

- **Implication:** $f(t)$ is *itself a Gaussian process* (in the *time* domain)
- **Benefit:** Gaussian animations *revealed to belong to a well-studied framework*
  - Future animations can leverage existing Gaussian Process work (e.g., try new covariance functions)

R implementation

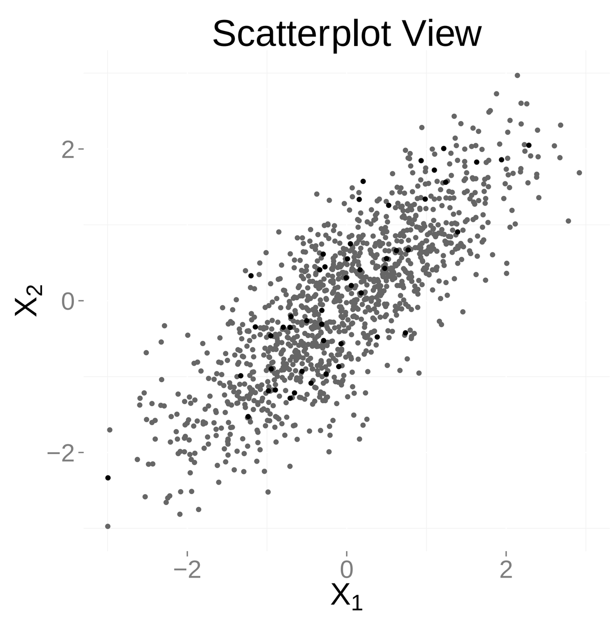
```
# Matrix to turn 2N random values into Gaussian oscillator.
GaussianOscillatorMatrix <- function(N, t) {
  cbind(outer(t, 1:N, function(x, y) cos(pi * x * y / N)),
        outer(t, N:1, function(x, y) sin(pi * x * y / N)))
}
```

Conclusions

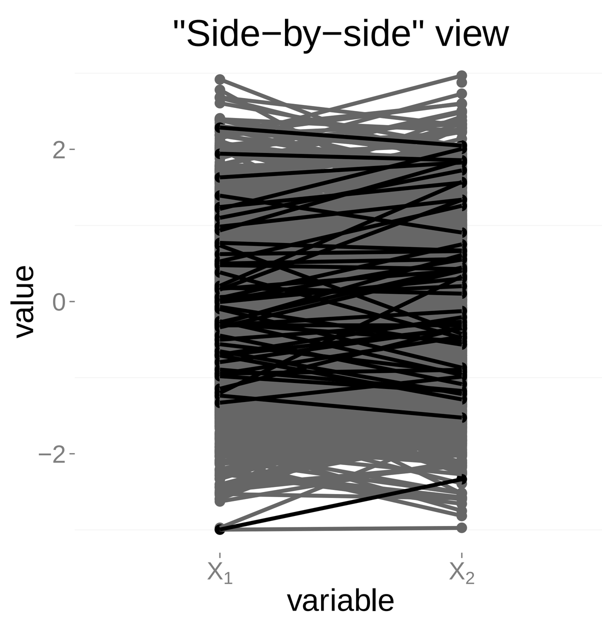
- First *statistically correct* Gaussian animations with **smooth and natural motion**
- Moving beyond interpolation: **keyframes entirely eliminated**
- **Time-domain Gaussian Processes** enable *animated visualization* of **Space-domain Gaussian Processes**
  - To eliminate keyframes, use *stationary* covariance function

Gaussian Processes refresher: Probabilities for Functions

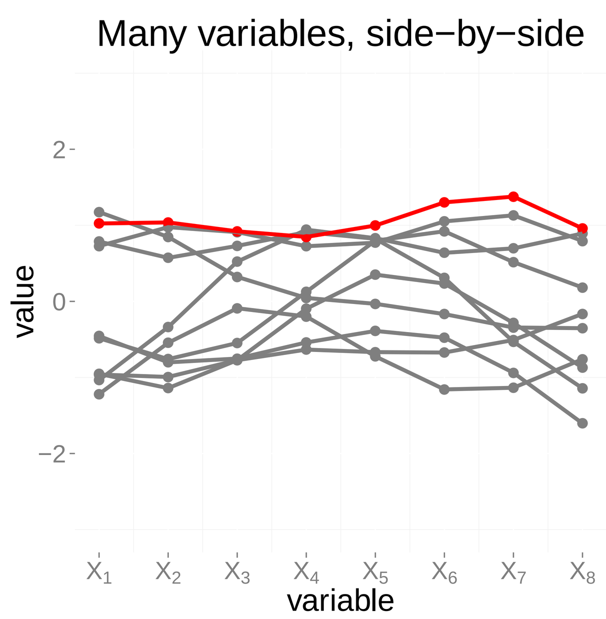
- Random curves and surfaces: *infinitely many* random variables!
- **Gaussian Processes:** work with *any* **finite** *subset*
  - Assume *joint Gaussian distribution*
- Simple example: Start with 2 variables, work up from there...



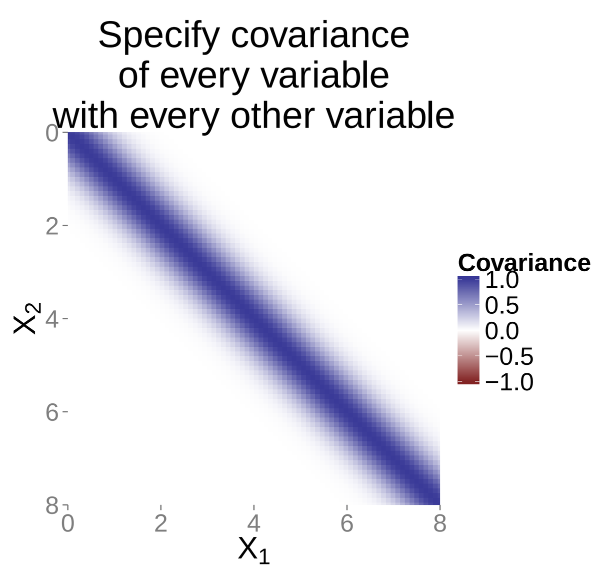
- Highly correlated → **close to diagonal**
- Works well for two variables



- Highly correlated → **horizontal lines**
- Works well for more variables...



- Variables indexed by *position*



Get *continuous* random function from *I.I.D.* normal draws: