

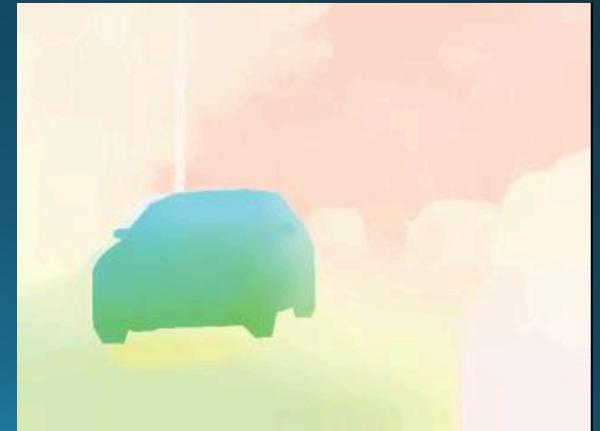
CSCI 4360/6360 Data Science II

Linear Dynamical Systems

Assignment 1 Lightning Review

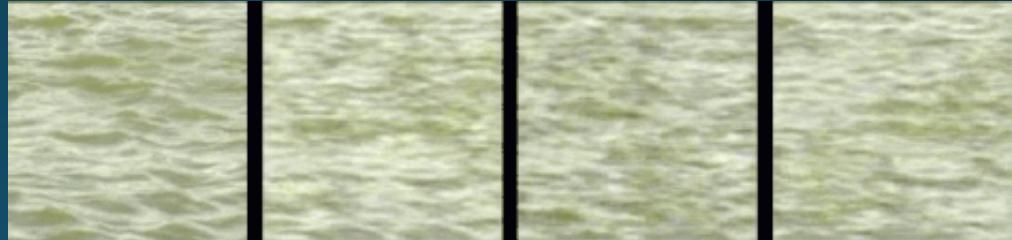
Last time...

- Motion analysis via optical flow
- Parametric vs energy-based formulations
- Importance of assumptions
- Modern formulations
 - Robustness to outliers (large optical flow)
 - Relatedness to markov random fields
 - Coarse-to-fine image pyramids



Today

- A specific type of motion: **dynamic textures**



Dynamic textures

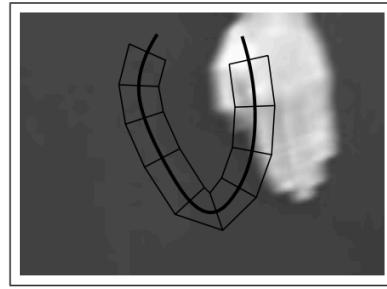
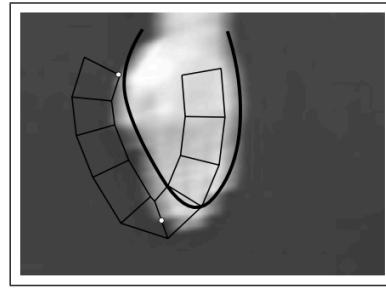
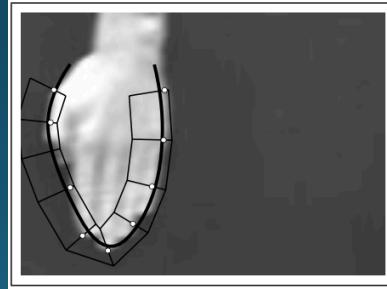
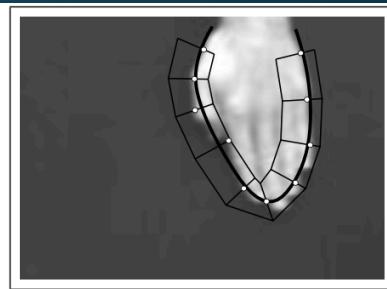
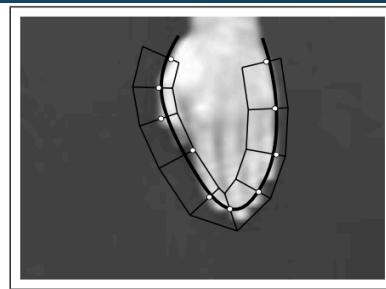
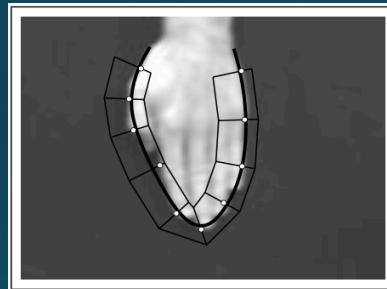
- “*Dynamic textured sequences are scenes with complex motion patterns due to interactions between multiple moving components.*”
- Examples
 - Blowing leaves
 - Flickering flames
 - Water rippling
- **Multiple moving components: problematic for optical flow**
- How to analyze dynamic textures?

Dynamical Models

- Goal: an effective procedure for tracking changes over sequences of images, while maintaining a certain coherence of motions

Dynamical Models

- Hand tracking
- Top row: slow movements
- Bottom row: fast movements
- **Fixed curves or priors cannot exploit coherence of motion**



Linear Dynamical Models

- Two main components (using notation from Hyndman 2006):

Appearance
Model

$$y_t = Cx_t + u_t$$

State Model

$$x_t = Ax_{t-1} + Wv_t$$

Autoregressive Models

- This is the definition of a 1st-order autoregressive (AR) process!

$$x_t = Ax_{t-1} + Wv_t$$

- Each observation (x_t) is a function of previous observations, plus some noise
- **Markov model!**

Autoregressive Models

- AR models can have higher orders than 1
- Each observation is dependent on the previous d observations

$$x_t = A_1 x_{t-1} + A_2 x_{t-2} + \dots + A_d x_{t-d} + W v_t$$

Appearance Model

- y_t : image of height h and width w at time t , usually flattened into $1 \times hw$ vector
- x_t : state space vector at time t , $1 \times q$ (where $q \ll\ll hw$)
- u_t : white Gaussian noise
- C : output matrix, maps between spaces, $hw \times q$

$$y_t = Cx_t + u_t$$

Output matrix

Image in a sequence

Low-dimensional “state”

Noise inherent to the system

Appearance Model

$$y_t = Cx_t + u_t$$

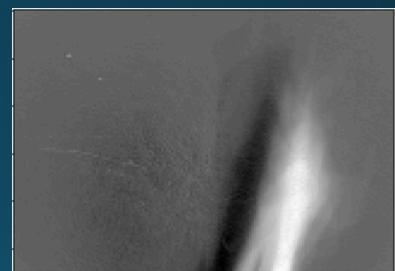


Each of these is 1 column of C .

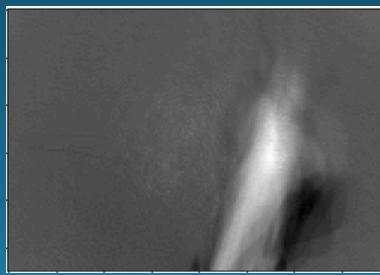
There are q of them
(first 4 shown here).



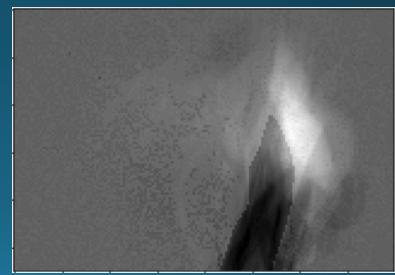
PC 1



PC 2



PC 3



PC 4

Appearance Model

- How do we learn the appearance model?
- Choose state-space dimension size q
- Noise term is i.i.d Gaussian

$$y_t = Cx_t + u_t$$

\hat{U} -hat is a matrix of the first q columns of U

$$Y = [\vec{y}_1, \vec{y}_2, \dots, \vec{y}_f]^T$$

$$Y = U\Sigma V^T$$

$$C = \hat{U}$$

$$X = \hat{\Sigma} \hat{V}^T$$

\hat{V} -hat is a matrix of the first q columns of V , and sigma-hat is a diagonal matrix of the first q singular values

State Model

- x_t and x_{t-1} : state space vectors at times t and $t-1$, each $1 \times q$ vector
- A : transition matrix, $q \times q$ matrix
- W : driving noise, $q \times q$ matrix
- v_t : white Gaussian noise

$$x_t = Ax_{t-1} + Wv_t$$

State transition matrix

Low-dimensional state at time t

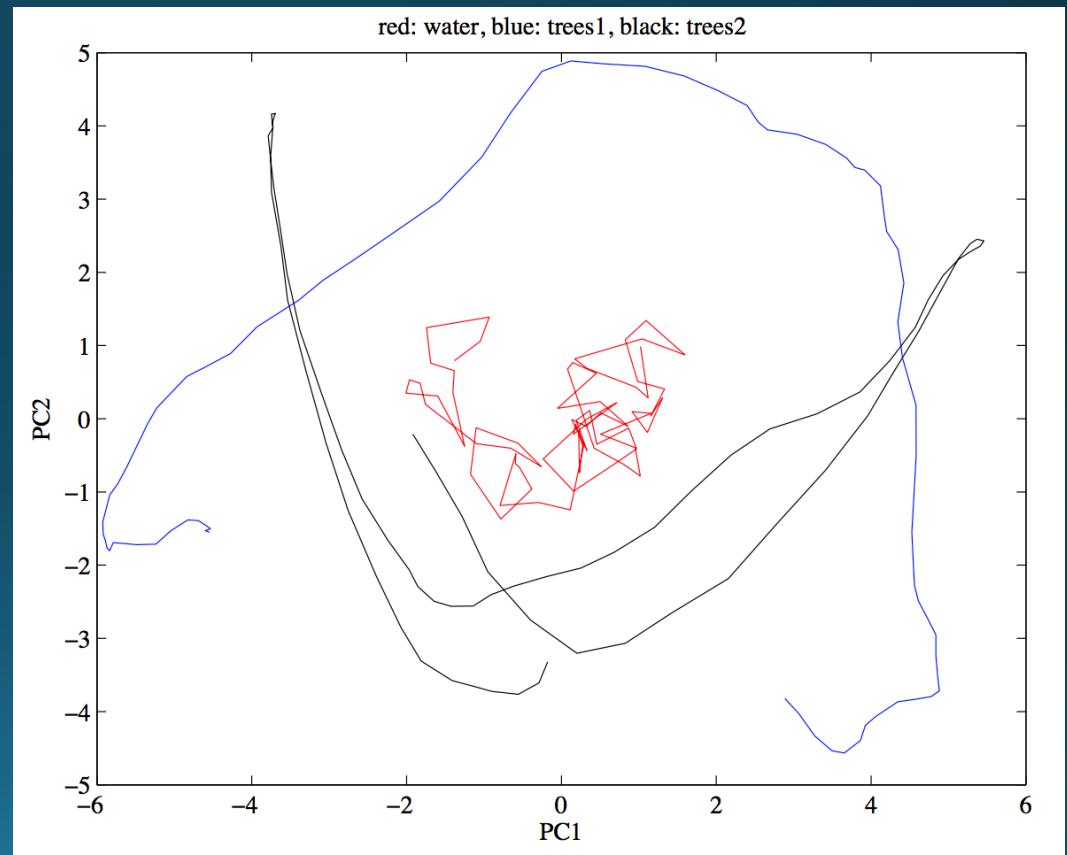
Low-dimensional state at $t - 1$

Driving noise

State Model

$$x_t = Ax_{t-1} + Wv_t$$

- Three textures
- $q = 2$



State Model

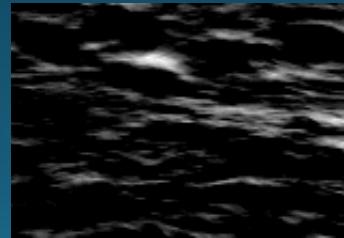
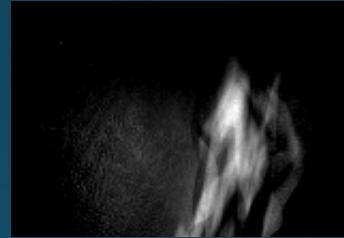
- How do we learn the state model?

$$x_t = Ax_{t-1} + Wv_t$$

- See Assignment 3!

LDS as Generative Models

- Once we've learned the parameters, we can *generate new instances*



- Major strength of LDS!

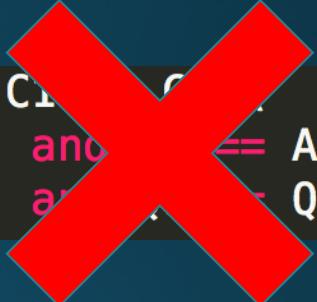
Problems with LDS

- PCA = Linear + Gaussian
- What if the *state space* isn't linear, or data aren't Gaussian?
- Nonlinear appearance models
 - Wavelets
 - IsoMap
 - LLE
 - Kernel PCA
 - Laplacian Eigenmaps
- These introduce their own problems!

Problems with LDS

- Comparing LDS models
- Given a sequence Y :
- New sequence Y' :
- **How do we compare these systems?**
- Despite linear formulation, θ are NOT Euclidean
- Valid distance metrics include spectral methods and distribution comparators

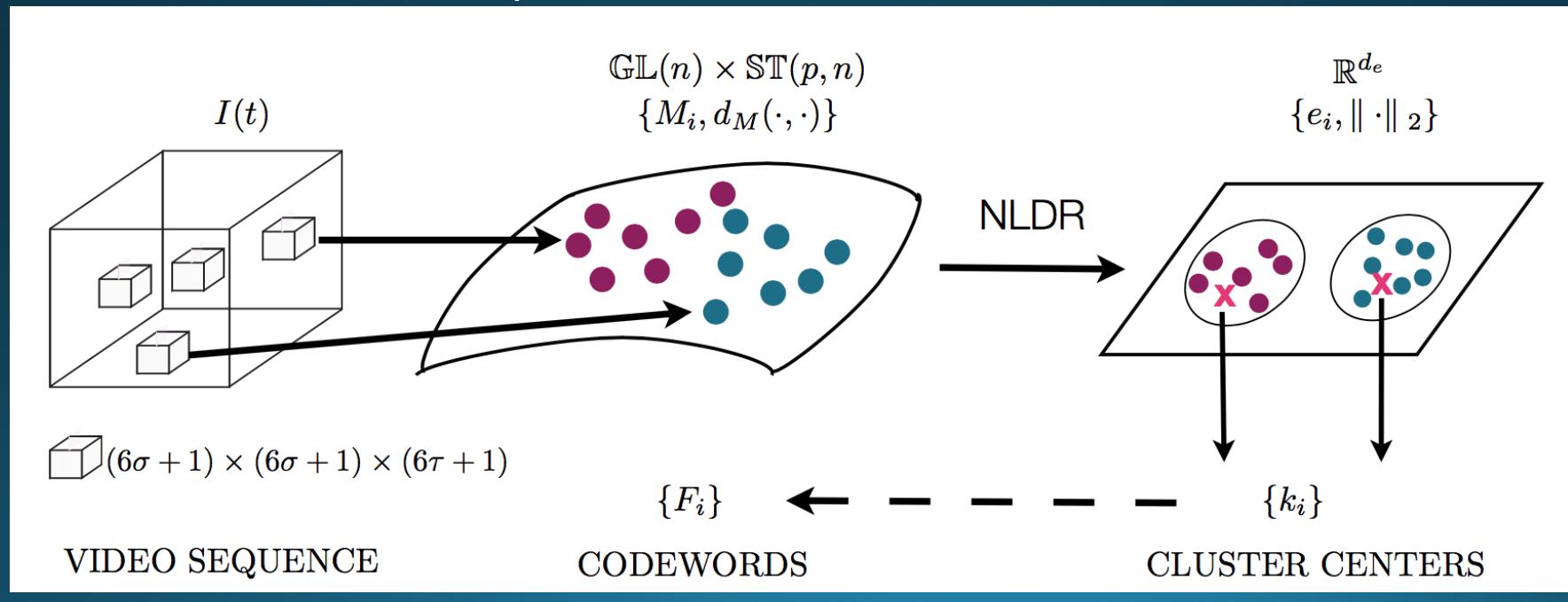
$$\begin{aligned}\theta &= (C, A, Q) \\ \theta' &= (C', A', Q')\end{aligned}$$



```
if C1 == A1 and C2 == A2 \n    Q1 = Q2:\n        print("Match")\n    else:\n        print("No Match")
```

Comparing LDS

- Select multiple, non-overlapping patches from each video
- Build LDS for each patch



Assignment 3

- Implement LDS!
- Now on AutoLab: **due September 28**

References

- Hyndman *et al*, "Higher-order Autoregressive Models for Dynamic Textures", BMVC 2007
<http://www.cs.toronto.edu/~fleet/research/Papers/DynamicTextures.pdf>
- Shumway and Stoffer, *Time Series Analysis and Its Applications* (3rd ed.), Chapters 3 and 6, <http://www-db.ucsd.edu/static/TimeSeries.pdf>
- Blake and Isard, *Active Contours*, Chapter 9-11,
http://www.vavlab.ee.boun.edu.tr/courses/574/material/books/blake_Active_Contours.pdf
- Doretto *et al*, "Dynamic Textures", IJCV 2003
<https://escholarship.org/uc/item/2m5of2fb>
- Woolfe *et al*, "Shift Invariant Dynamic Texture Recognition", ECCV 2006
https://link.springer.com/chapter/10.1007%2F11744047_42?LI=true
- Ravichandran *et al*, "View-invariant dynamic texture recognition using a bag of dynamical systems", CVPR 2009
<http://ieeexplore.ieee.org/abstract/document/5206847/>