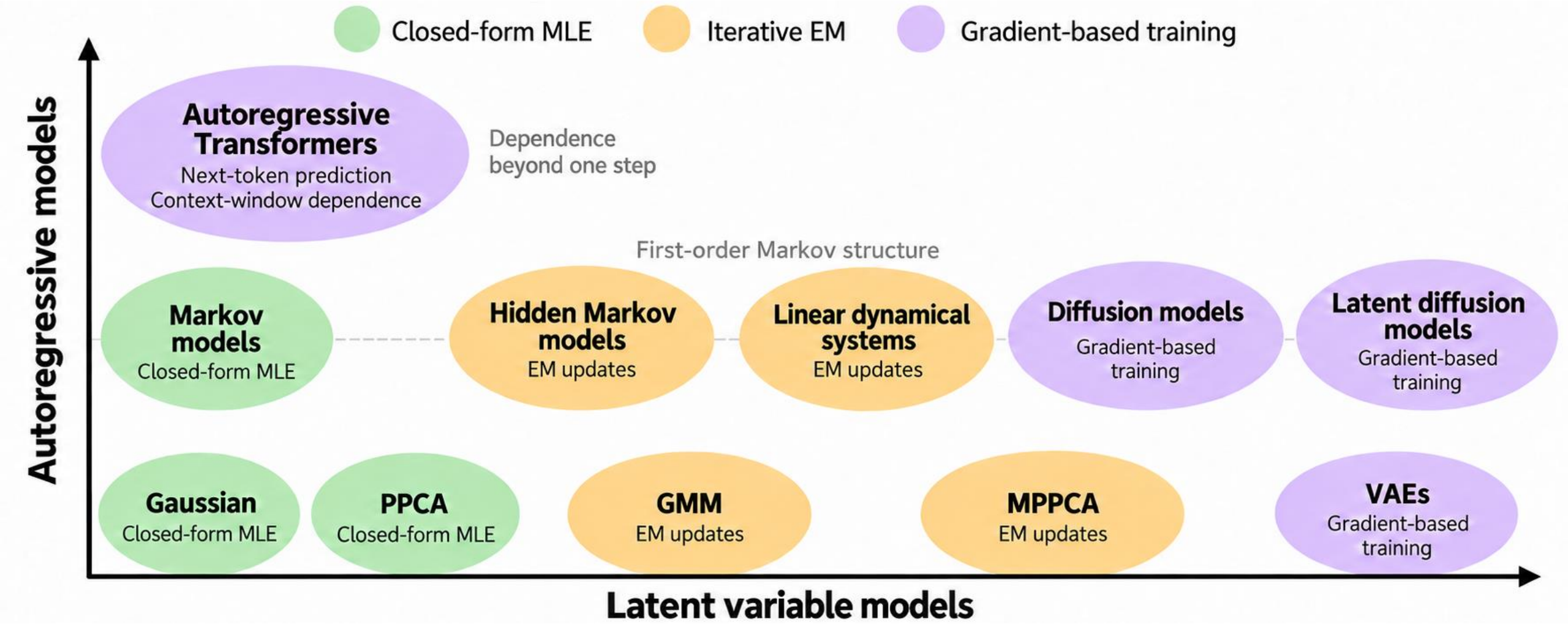
Deep Generative Models: Dynamic Textures

Fall Semester 2026

René Vidal
Director of the Center for Innovation in Data Engineering and Science (IDEAS),
Rachleff University Professor, University of Pennsylvania
Amazon Scholar & Chief Scientist at NORCE



Taxonomy of Generative Models



Motivation: Dynamic Textures

- Dynamic textures are videos of nonrigid deformable objects such as water, fire, smoke, flags moving with the wind, etc.





- Dynamic textures are characterized by appearance + dynamics




Need a model that captures the appearance and dynamics

Dynamic Texture Model

- A video is a sequence of T images $\{\boldsymbol{I}_t \in \mathbb{R}^D\}_{t=0}^{T}$, where D is the number of pixels.
- Suppose at each time instant we observe a noisy version of the image

$$\boldsymbol{x}_t = \boldsymbol{I}_t + \boldsymbol{v}_t$$

where $\boldsymbol{v}_t \sim \mathcal{N}(0, \boldsymbol{Q})$ is an i.i.d. sequence Gaussian noise.

- Suppose each image $\boldsymbol{I}_t$ is generated via a latent representation $\boldsymbol{z}_t \in \mathbb{R}^d$ as follows

$$\boldsymbol{I}_t = \boldsymbol{C}_0 + \boldsymbol{C}\boldsymbol{z}_t \qquad \Rightarrow \qquad \boldsymbol{x}_t = \boldsymbol{C}_0 + \boldsymbol{C}\boldsymbol{z}_t + \boldsymbol{v}_t$$

- If $\boldsymbol{z}_t \sim^{i.i.d.} \mathcal{N}(0, \boldsymbol{I}_d)$ and $\boldsymbol{Q} = \sigma_v^2 \boldsymbol{I}_D$, then $\boldsymbol{x}_t$ is a PPCA model:
  - $\boldsymbol{C}_0$ is the mean image/average video frame and $\boldsymbol{C}_i$ is the ith principal direction/eigenimage.
- However, this is not a good model for video because all frames are i.i.d.

Dynamic Texture Model

- To capture the temporal evolution of a video, instead of assuming $\boldsymbol{z}_t$ i.i.d., we assume that $\boldsymbol{z}_t$ is a linear autoregressive model $\boldsymbol{z}_{t+1} = \boldsymbol{A}\boldsymbol{z}_t + \boldsymbol{w}_t$.

- We say the video sequence $\{\boldsymbol{x}_t \in \mathbb{R}^D\}_{t=0}^T$ is a **linear dynamic texture** if

$$\boldsymbol{z}_{t+1} = \boldsymbol{A}\boldsymbol{z}_t + \boldsymbol{w}_t$$
$$\boldsymbol{x}_t = \boldsymbol{C}_0 + \boldsymbol{C}\boldsymbol{z}_t + \boldsymbol{v}_t$$

where

- D is the number of pixels and T is the number of frames
- $\boldsymbol{z}_t \in \mathbb{R}^d$ is the hidden state, d is its dimension, and $\boldsymbol{z}_0$ is the initial state
- $\boldsymbol{w}_t \sim \mathcal{N}(0, \boldsymbol{Q})$ is the state noise, i.i.d. sequence, assumed to be independent from $\boldsymbol{z}_t$
- $\boldsymbol{v}_t \sim \mathcal{N}(0, \boldsymbol{R})$ is the output noise, i.i.d. sequence, assumed to be independent from $\boldsymbol{z}_t$
- $\boldsymbol{A} \in \mathbb{R}^{d \times d}$, $\boldsymbol{C}_0 \in \mathbb{R}^D$, $\boldsymbol{C} \in \mathbb{R}^{D \times d}$, $\boldsymbol{Q} \in \mathbb{R}^{d \times d}$ and $\boldsymbol{R} \in \mathbb{R}^{D \times D}$ are the model parameters

Dynamic Textures

$$z_{t+1} = Az_t + w_t$$
$$x_t = C_0 + Cz_t + v_t$$

(Soatto ICCV 01, Doretto IJCV 03)



$$C_0 \in \mathbb{R}^D$$
$$C \in \mathbb{R}^{D \times d}$$

APPEARANCE

$$A \in \mathbb{R}^{d \times d}$$
$$z_0 \in \mathbb{R}^d$$

DYNAMICS

$$Q \in \mathbb{R}^{d \times d}$$
$$R \in \mathbb{R}^{D \times D}$$

NOISE

Learning Dynamic Textures

- Let $\{\boldsymbol{x}_t\}_{t=0\ldots T}$, $\boldsymbol{x}_t \in \mathbb{R}^D$ be an observed video sequence. Our goal is to learn the model parameters $\theta = \{\boldsymbol{A}, \boldsymbol{C}_0, \boldsymbol{C}, \boldsymbol{Q}, \boldsymbol{R}\}$, a.k.a. the system identification problem.

- One method we have learned is Maximum Likelihood Estimation (using EM)
  - Given $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_T$, solve

$$\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{C}}_0, \widehat{\boldsymbol{C}}, \widehat{\boldsymbol{Q}}, \widehat{\boldsymbol{R}} = \underset{\boldsymbol{A}, \boldsymbol{C}_0, \boldsymbol{C}, \boldsymbol{Q}, \boldsymbol{R}}{\operatorname{argmax}} \log p(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_T)$$

  - subject to

$$\boldsymbol{z}_{t+1} = \boldsymbol{A}\boldsymbol{z}_t + \boldsymbol{w}_t \qquad t = 0, \ldots, T-1, \quad \boldsymbol{w}_t \sim^{i.i.d.} \mathcal{N}(0, \boldsymbol{Q}) \qquad t = 0, \ldots, T$$

$$\boldsymbol{x}_t = \boldsymbol{C}_0 + \boldsymbol{C}\boldsymbol{z}_t + \boldsymbol{v}_t \qquad t = 0, \ldots, T, \qquad \boldsymbol{v}_t \sim^{i.i.d.} \mathcal{N}(0, \boldsymbol{R}) \qquad t = 0, \ldots, T$$

- However, doing so is complicated and there is a simple approximate solution

Approximate Solutions ($\boldsymbol{C}_0$)

- Let $\{\boldsymbol{x}_t\}_{t=0\ldots T}$, $\boldsymbol{x}_t \in \mathbb{R}^D$ be an observed video sequence. Our goal is to learn the model parameters $\theta = \{\boldsymbol{A}, \boldsymbol{C}_0, \boldsymbol{C}, \boldsymbol{Q}, \boldsymbol{R}\}$, a.k.a. the system identification problem.
- One method we have learned is Maximum Likelihood Estimation (using EM)
  - Given $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_T$, solve

$$\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{C}}_0, \widehat{\boldsymbol{C}}, \widehat{\boldsymbol{Q}}, \widehat{\boldsymbol{R}} = \underset{\boldsymbol{A}, \boldsymbol{C}_0, \boldsymbol{C}, \boldsymbol{Q}, \boldsymbol{R}}{\operatorname{argmax}} \log p(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_T)$$

  - subject to

$$\boldsymbol{z}_{t+1} = \boldsymbol{A}\boldsymbol{z}_t + \boldsymbol{w}_t \qquad t = 0, \ldots, T-1, \quad \boldsymbol{w}_t \sim^{i.i.d.} \mathcal{N}(0, \boldsymbol{Q}) \qquad t = 0, \ldots, T$$

$$\boldsymbol{x}_t = \boldsymbol{C}_0 + \boldsymbol{C}\boldsymbol{z}_t + \boldsymbol{v}_t \qquad t = 0, \ldots, T, \qquad \boldsymbol{v}_t \sim^{i.i.d.} \mathcal{N}(0, \boldsymbol{R}) \qquad t = 0, \ldots, T$$

  - if $\mathbb{E}[\boldsymbol{z}_0] = 0$, then $\mathbb{E}[\boldsymbol{z}_t] = 0$ for all t
  - $\mathbb{E}[\boldsymbol{x}_t] = \boldsymbol{C}_0 + \boldsymbol{C}\mathbb{E}[\boldsymbol{z}_t] + \mathbb{E}[\boldsymbol{v}_t] = \boldsymbol{C}_0$.
  - so we can estimate $\boldsymbol{C}_0$ as the mean video $\widehat{\boldsymbol{C}}_0 = \frac{1}{T}\sum \boldsymbol{x}_t$
  - subtract the mean from each frame ($\boldsymbol{x}_t \leftarrow \boldsymbol{x}_t - \widehat{\boldsymbol{C}}_0$) and we obtain a different LDS

Reformulation without $\boldsymbol{C_0}$

- For our given process $\{\boldsymbol{z}_t\}, \boldsymbol{z}_t \in \mathbb{R}^d$, guided by

$$\boldsymbol{z}_{t+1} = \boldsymbol{A}\boldsymbol{z}_t + \boldsymbol{w}_t \qquad \boldsymbol{w}_t \sim \mathcal{N}(0, \boldsymbol{Q}) \qquad \boldsymbol{z}(0) = \boldsymbol{z}_0$$
$$\boldsymbol{x}_t = \boldsymbol{C}\boldsymbol{z}_t + \boldsymbol{v}_t \qquad \boldsymbol{v}_t \sim \mathcal{N}(0, \boldsymbol{R})$$

  with initial condition $\boldsymbol{z}(0) = \boldsymbol{z}_0$, positive definite matrices $\boldsymbol{R}$ and $\boldsymbol{Q}$.

- We seek the estimate parameters $\boldsymbol{A} \in \mathbb{R}^{d\times d}, \boldsymbol{C} \in \mathbb{R}^{D\times D}, \boldsymbol{Q} \in \mathbb{S}_+^{d\times d}, \boldsymbol{R} \in \mathbb{S}_+^{D\times D}$ from the given samples $\boldsymbol{x}_1, \dots, \boldsymbol{x}_T$ that maximizes the log-likelihood

$$\widehat{\boldsymbol{A}}_T, \widehat{\boldsymbol{C}}_T, \widehat{\boldsymbol{Q}}_T, \widehat{\boldsymbol{R}}_T, = \underset{\boldsymbol{A},\boldsymbol{C},\boldsymbol{Q},\boldsymbol{R}}{\operatorname{argmax}} \log p(\boldsymbol{x}_0, \dots, \boldsymbol{x}_T)$$

Approximate Solutions ($\boldsymbol{C}$ and $\boldsymbol{Z}_{0:T}$)

- Let
  - $\boldsymbol{X}_{0:T} = [\boldsymbol{x}_0, \dots, \boldsymbol{x}_T] \in \mathbb{R}^{D\times(T+1)}$,
  - $\boldsymbol{Z}_{0:T} = [\boldsymbol{z}_0, \dots, \boldsymbol{z}_T] \in \mathbb{R}^{d\times(T+1)}$ with $T+1 > d$,
  - $\boldsymbol{V}_{0:T} = [\boldsymbol{v}_0, \dots, \boldsymbol{v}_T] \in \mathbb{R}^{d\times(T+1)}$
- Assuming $D \gg d, \text{rank}(\boldsymbol{C}) = d$ and $\boldsymbol{C}^\top\boldsymbol{C} = \boldsymbol{I}_d$, we can rewrite the output as

$$\boldsymbol{X}_{0:T} = \boldsymbol{C}\boldsymbol{Z}_{0:T} + \boldsymbol{V}_{0:T};\ \boldsymbol{C} \in \mathbb{R}^{D\times d};\boldsymbol{C}^\top\boldsymbol{C} = \boldsymbol{I}_d$$

- We can estimate $\boldsymbol{C}$ and $\boldsymbol{Z}_{0:T}$ via solving

$$\widehat{\boldsymbol{C}}_T, \widehat{\boldsymbol{Z}}_{0:T} = \text{argmin}_{\boldsymbol{C},\boldsymbol{Z}_{0:T}} \left|\left|\boldsymbol{X}_{0:T} - \boldsymbol{C}\boldsymbol{Z}_{0:T}\right|\right|_F^2 \text{ subject to } \boldsymbol{C}^\top\boldsymbol{C} = \boldsymbol{I}_d$$

- The solution can be found using SVD. Let $\boldsymbol{X}_{0:T} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, then

$$\widehat{\boldsymbol{C}}_T = \boldsymbol{U} \qquad \widehat{\boldsymbol{Z}}_{0:T} = \boldsymbol{\Sigma}\boldsymbol{V}^\top$$

- This solution is very close to learning a PPCA model for $\boldsymbol{Y}_{0:T}$.

Approximate Solutions ($\boldsymbol{A}$, $\boldsymbol{Q}$, $\boldsymbol{R}$)

$$\boldsymbol{X}_{0:T} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$$
$$\widehat{\boldsymbol{Z}}_{0:T} = \boldsymbol{\Sigma}\boldsymbol{V}^\top$$

- We can estimate $\boldsymbol{A}$ by the following

$$\widehat{\boldsymbol{A}}_T = \operatorname{argmin}_{\boldsymbol{A}} \left|\left|\widehat{\boldsymbol{Z}}_{1:T} - \boldsymbol{A}\widehat{\boldsymbol{Z}}_{0:T-1}\right|\right|_F = \widehat{\boldsymbol{Z}}_{1:T}\widehat{\boldsymbol{Z}}_{1:T}^\top\left(\widehat{\boldsymbol{Z}}_{0:T}\widehat{\boldsymbol{Z}}_{0:T}^\top\right)^{-1}$$

- We can estimate the state and output covariances via

$$\widehat{\boldsymbol{Q}}_T = \frac{1}{T}\sum_{t=0}^{T-1}\widehat{\boldsymbol{w}}_t\widehat{\boldsymbol{w}}_t^\top \qquad \widehat{\boldsymbol{R}}_T = \frac{1}{T}\sum_{t=0}^{T-1}\widehat{\boldsymbol{v}}_t\widehat{\boldsymbol{v}}_t^\top$$

where $\widehat{\boldsymbol{w}}_t \doteq \boldsymbol{x}_t - \widehat{\boldsymbol{C}}_T\hat{\boldsymbol{z}}_t$ and $\widehat{\boldsymbol{v}}_t \doteq \hat{\boldsymbol{z}}_{t+1} - \widehat{\boldsymbol{A}}_T\hat{\boldsymbol{z}}_t$.

Approximate Solution: Pseudocode

- Therefore, everything can be computed in succession and in closed-form

- In the original work, the algorithm is implemented in fewer than 20 lines

```
function [x0,Ymean,Ahat,Bhat,Chat] = dytex(Y,n,nv)

% Suboptimal Learning of Dynamic Textures;
% (c) UCLA, March 2001.

  tau = size(Y,2); Ymean = mean(Y,2);
  [U,S,V] = svd(Y-Ymean*ones(1,tau),0);
  Chat=U(:,1:n); Xhat = S(1:n,1:n)*V(:,1:n)';
  x0=Xhat(:,1);
  Ahat = Xhat(:,2:tau)*pinv(Xhat(:,1:(tau-1)));
  Vhat = Xhat(:,2:tau)-Ahat*Xhat(:,1:(tau-1));
  [Uv,Sv,Vv] = svd(Vhat,0);
  Bhat  = Uv(:,1:nv)*Sv(1:nv,1:nv)/sqrt(tau-1);

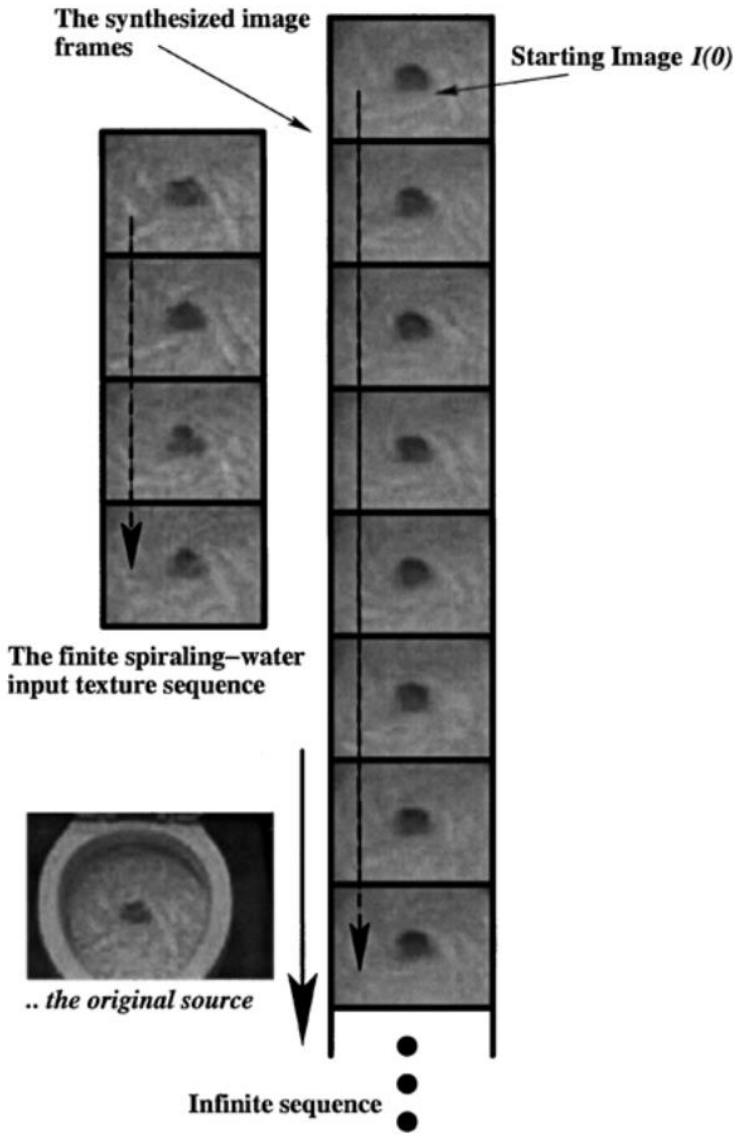
function [I] = synth(x0,Ymean,Ahat,Bhat,Chat,tau)

% Synthesis of Dynamic Textures;
% (c) UCLA, March 2001.

  [n,nv] = size(Bhat);
  X(:,1) = x0;
  for t = 1:tau,
    X(:,t+1) = Ahat*X(:,t)+Bhat*randn(k,1);
    I(:,t) = Chat*X(:,t)+Ymean;
  end;
```

Experimental Results

- After learning the parameters of a linear dynamical systems from a finite length sequence, the system can synthesize *infinite length sequences*



Experimental Results

- Top: The original sequence of water waves
- Left: As you increase the order of the dynamical system (dimension of $\boldsymbol{z}$), the reconstruction error decreases
- Right: extrapolation error reduces as number of time steps increases

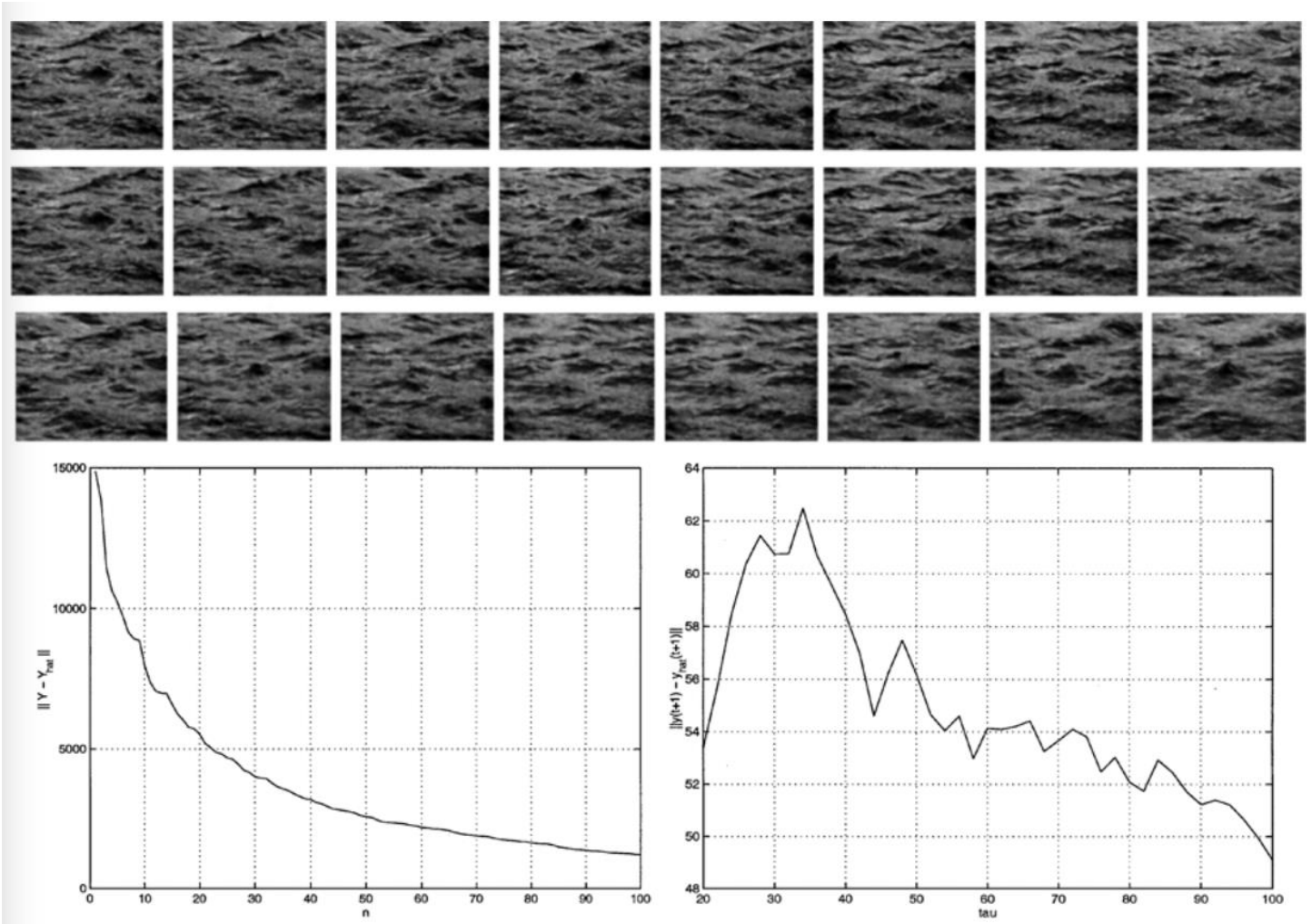
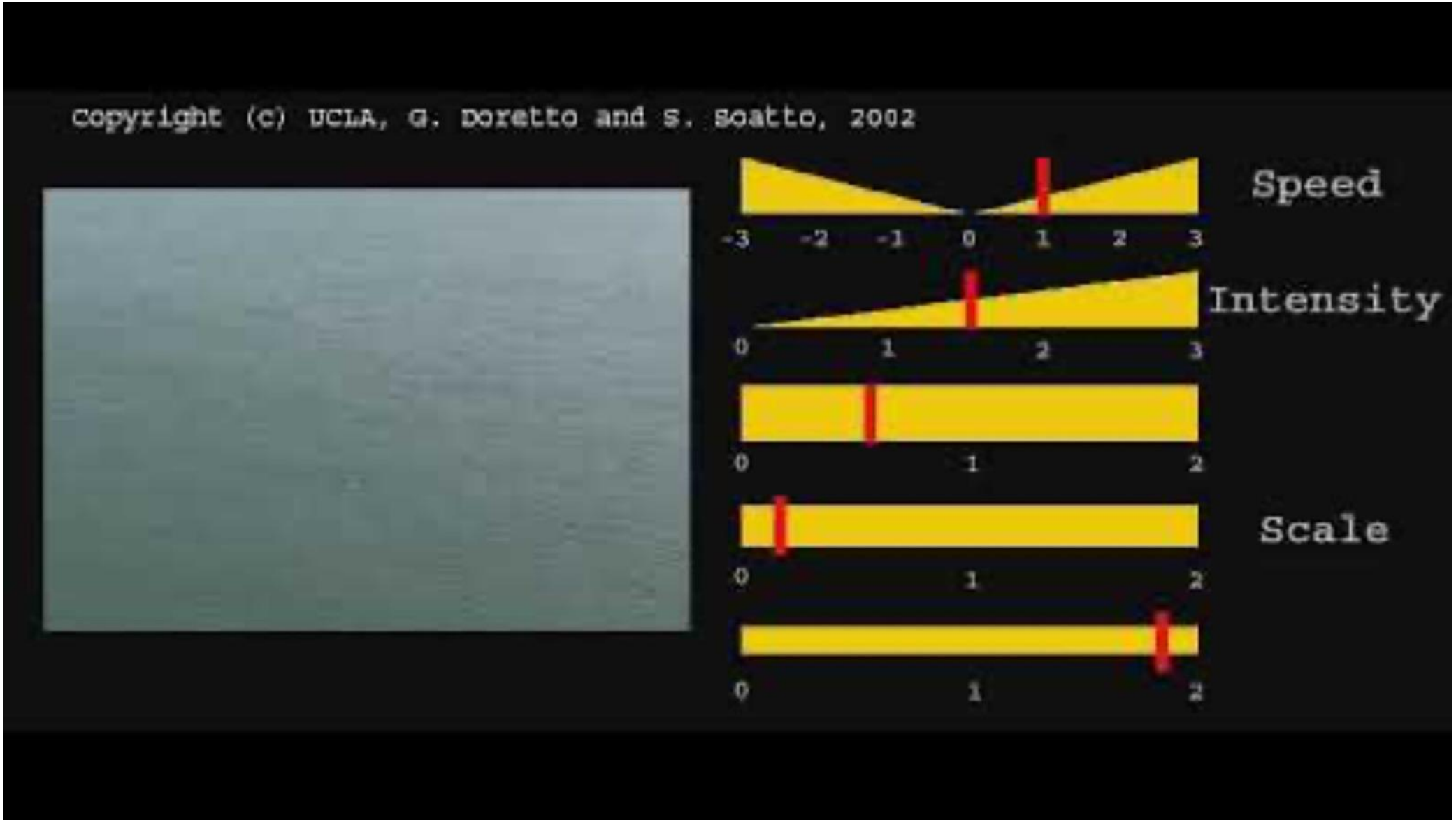


Figure 3. River. From top to bottom: Samples of the original sequence, corresponding samples of the compressed sequence (compression ratio: 2.53), samples of extrapolated sequence (using $n = 50$ components, $\tau = 120$, $m = 170 \times 115$), compression error as a function of the dimension of the state space n, and extrapolation error as a function of the length of the training set τ. The data set used comes from the *MIT Temporal Texture database*.

https://www.youtube.com/watch?v=bXHpOodkUv0



Application to Modeling Human Motion

