Analysis and Synthesis of Textured Motion: Particles and Waves

Yizhou Wang and Song-Chun Zhu

Abstract—Natural scenes contain a wide range of textured motion phenomena which are characterized by the movement of a large amount of particle and wave elements, such as falling snow, wavy water, and dancing grass. In this paper, we present a generative model for representing these motion patterns and study a Markov chain Monte Carlo algorithm for inferring the generative representation from observed video sequences. Our generative model consists of three components. The first is a photometric model which represents an image as a linear superposition of image bases selected from a generic and overcomplete dictionary. The dictionary contains Gabor and LoG bases for point/particle elements and Fourier bases for wave elements. These bases compete to explain the input images and transfer them to a token (base) representation with an $O(10^2)\text{-fold}$ dimension reduction. The second component is a geometric model which groups spatially adjacent tokens (bases) and their motion trajectories into a number of moving elements—called "motons." A moton is a deformable template in time-space representing a moving element, such as a falling snowflake or a flying bird. The third component is a dynamic model which characterizes the motion of particles, waves, and their interactions. For example, the motion of particle objects floating in a river, such as leaves and balls, should be coupled with the motion of waves. The trajectories of these moving elements are represented by coupled Markov chains. The dynamic model also includes probabilistic representations for the birth/death (source/sink) of the motons. We adopt a stochastic gradient algorithm for learning and inference. Given an input video sequence, the algorithm iterates two steps: 1) computing the motons and their trajectories by a number of reversible Markov chain jumps, and 2) learning the parameters that govern the geometric deformations and motion dynamics. Novel video sequences are synthesized from the learned models and, by editing the model parameters, we demonstrate the controllability of the generative model.

Index Terms—Textured motion, generative model, texton, statistical learning, object tracking, stochastic gradient.

1 Introduction

Natural scenes contain a wide variety of stochastic motion patterns which are characterized by the movement of a large amount of particle and wave elements, such as falling snow, a flock of flying birds, wavy river, and dancing grass. Such motion patterns, as is acknowledged in [15], fall beyond the scope of conventional optical flow field models [11] and a new framework has yet to be developed. In recent years, the study of such motion patterns has stimulated a growing interest in both the vision and the graphics communities, driven by a number of applications on synthesis and analysis.

Graphics methods. Computer graphics methods are concerned with rendering photorealistic video sequences or nonphotorealistic but stylish cartoon animations. In the graphics literature, both physics-based and image-based methods have been reported. The former uses partial differential equations, for example, creating animations of fire and gaseous phenomena with particles [20], [4]. The latter includes:

1. *Video texture* [22] and its extension [23], which replay a video by reordering the image frames to achieve smooth transition between frames.
2. Three-dimensional *volume texture* [31] which generates motion through nonparametric sampling from an observed video motivated by recent work on texture synthesis [10], [33], [5].
3. *Video rewrite* [2], which uses the hidden Markov model (HMM) to create a speech driven facial animation system.
4. *Motion texture* [24], which builds a HMM on a linear dynamic system for modeling realistic human motion.

Although some realistic animations can be rendered at fast speed, video texture and volume texture do not explicitly account for the dynamic and geometric properties of the moving elements, neither does video rewrite nor motion texture infer the moving elements from input video. Consequently, the synthesized animations are either less controllable or depend very much on the means of data acquisition.

Vision methods. In computer vision, the analysis of these motion patterns is important for video analysis, such as motion segmentation, annotation, recognition, retrieval, and abnormal motion detection. In the vision literature, as these motion patterns lie in the domains of both motion analysis and texture modeling, statistical models are proposed from both directions with a trend of merging the two. In the rest of this section, we briefly review this work to set the background of our method.

Szummer and Picard [27] called the motion patterns *temporal texture* and adopted a Spatial-Temporal Auto-Regression (STAR) model from Cliff and Ord [3]. The STAR

- *Y. Wang is with the Department of Computer Science, University of California, Los Angeles, 3532F Boelter Hall, Los Angeles, CA 90095. E-mail: wangyz@cs.ucla.edu.*
- *S.-C. Zhu is with the Departments of Statistics and Computer Science, University of California, Los Angeles, 8130 Math Science Bldg., Box 951554, Los Angeles, CA 90095. E-mail: sczhu@stat.ucla.edu.*

Manuscript received 20 May 2003; revised 3 Feb. 2004; accepted 8 Apr. 2004. Recommended for acceptance by S. Sclaroff.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0092-0503.

model represents the intensity of each pixel as a linear summation of intensities of its spatial and temporal neighbors. It can be considered as an extension to the causal Gaussian Markov random field model (GMRF) used in texture modeling by adding the time dimension. Bar-Joseph et al. [1] extended the 2D texture synthesis work [10], [33] to a tree structured multiresolution representation, in a way similar to the $3D$ volume texture method [31]. The *dynamic texture* work by Soatto et al. [25] studied the motion dynamics explicitly using models and tools from control theory [14]. It is also shown to be useful for recognition [21]. Fitzgibbon [7] added the rigid camera motion in combination with the stochastic motion patterns, so that the motion is registered properly.

Despite their reasonable success, the existing models need to be extended to address the following problems: First, the basic moving elements in the existing models are either pixels and points [20], [4], [27], [31], [1] or entire images [22] and their principal components [25], [7]. Such representations usually do not identify the human perceived moving elements in the video, such as an individual bird or a snowflake. Second, these models do not sufficiently characterize the dynamics of moving elements, such as trajectories, sources, sinks and lifespans of the elements. It is considered rather challenging to model the interactions and couplings between the elements. For instance, it is hard to simulate and control balls or leaves drifting in water waves. Consequently, these models have less localization in analysis and controllability (in terms of controlling the number of moving elements and death/birth of the moving elements at specified locations and time intervals) in synthesis.

In this paper, we call these motion patterns "textured motions," following a suggestion by Mumford in 1996, to emphasize the fact that the image sequences are fundamentally motion phenomena characterized by the dynamics of the moving elements rather than texture phenomena. The latter are often states of a large system at thermodynamic equilibrium (i.e., system with maximum entropy under constraints [33]).

Summary of our method. Motivated by the above observation, we present a generative representation for modeling textured motion that integrates the following three components.

1. *A photometric model*. An image is represented as a superposition of bases selected from an overcomplete dictionary, including Fourier bases, LoG, and Gabor bases at various scales, orientations. Such bases are known to be generic and effective for representing natural images [6] with particle and wave patterns. This model transforms a raw image into a number of bases as a token representation and achieves an $O(10^2)$-fold dimension reduction. We should trace these bases over the image frames and compute their trajectories.
2. *A geometric model*. We group the bases and their motion trajectories into a number of basic moving elements which are coherent in space and time. We call the basic moving elements "motons" in accordance with the notion of "textons"—the atomic perceptual elements in static images [13], [32]. A moton is a deformable template in space-time representing a moving element, for example, each snowflake or bird is represented by a few Gabor and LoG bases traveling together (see Figs. 3 and 4).
3. *A dynamic model*. We adopt a general motion equation which includes an autoregression (AR) component for the trajectory of each moton, its source and sink maps, the external driving forces, and the interaction/coupling with other motons. The interaction among motons is always considered a challenge in both vision and graphics. In this paper, we assume that "waves have more influence on particles," i.e., a ball (Gabor bases) floating on a river is driven by water waves (Fourier bases).

We adopt an EM-like stochastic gradient algorithm [9] for the inference and learning. It infers the hidden variables (bases, motons, and trajectories) by a number of reversible Markov chain jumps and estimates the model parameters (deformable models, source and sink maps, parameters of the dynamics). This generative model offers more controllability in rendering synthesized motion sequences. Figs. 9 and 11 show the synthesized results after we edit the number of motons and the sources of motons, respectively.

This paper is organized as follows: In Section 2, we present a two-level generative representation with photometric, geometric, dynamic models in detail, and some experimental results are shown together with the models' illustration. In Section 3, we present the learning and inference algorithm using Markov chain Monte Carlo methods. A number of synthesized video clips and a cartoon animation are presented. We conclude the paper with a discussion and future work in Section 4.

2 Textured Motion Representation

In this section, we present our generative representation with three components—a photometric model, a geometric model, and a dynamic model. We use $\mathbf{I}[0,\tau]$ to denote an image sequence on a 2D lattice Λ in a discrete time interval $[0,\tau]=\{0,1,\dots,\tau\}$. $\mathbf{I}(u,v,t)$ denotes the intensity of pixel $(u,v)\in\Lambda$ in frame $t\in[0,\tau]$.

2.1 Photometric Model—Particles and Waves

The photometric model represents an image $\mathbf{I}$ by a superposition of N image bases $\psi_j, j=1,2,\dots,N$ selected from a dictionary Δ plus a Gaussian residual image n.

$$\begin{aligned}&\mathbf{I}(u,v)=\sum_{j=1}^{N}\alpha_j\psi_{\gamma_j}(u,v;\beta_j)+n(u,v),\\&\psi_{\gamma_j}\in\Delta,\ \ n(u,v)\sim\Lambda(0,\sigma_o^2),\ \ \forall(u,v)\in\Lambda.\end{aligned}\tag{1}$$

α_j is the coefficient (or amplitude) of a base ψ_{γ_j}, β_j denotes the transforms (translation, rotation and scaling) on the base, and γ_j is the type of the base, such as Gabor Cosine, Gabor Sine, LoG, or Fourier functions. The dictionary Δ is divided into particle bases (Gabors and LoGs) and wave bases (Fouriers) and it is $O(10^2)$-fold overcomplete.

$$\Delta=\Delta_{\text{pcl}}\cup\Delta_{\text{wav}},\quad \text{with}\ \ |\Delta|=O(10^2|\Lambda|).$$

In the following, we briefly introduce the particle bases Δ_{pcl} and wave bases Δ_{wav} and a match pursuit algorithm for base selection.

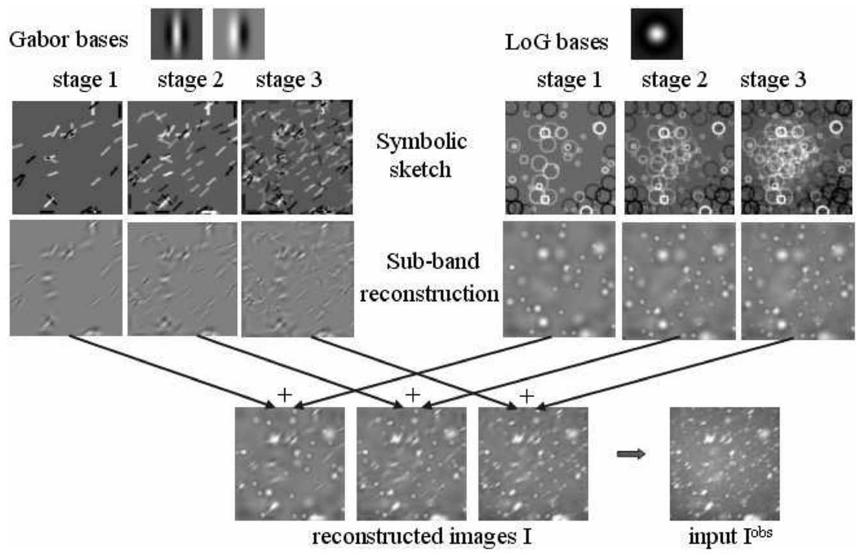


Fig. 1. Image reconstruction by match pursuit in three stages. Top row: Three particle bases. Middle row: Symbolic sketch for the Gabor bases (by bars) and LoG bases (by circles) and the images they reconstruct. Bottom row: Reconstructed images at three stages. It terminates with $N_{\rm pcl} = 800$ bases.

1. **Particle bases** $\Delta_{\rm pcl}$. The dictionary of particle bases is constructed from three standard basis functions: Gabor cosine, Gabor sine, and Laplacian of Gaussian,

$$\Phi_{\rm pcl} = \{{\rm Gcos}(u,v),\ \ {\rm Gsin}(u,v),\ \ {\rm LoG}(u,v)\}.$$

Applying transforms denoted by variable β, we have

$$\Delta_{\rm pcl} = \{{\rm Gcos}(u,v;\beta), {\rm Gsin}(u,v;\beta), {\rm LoG}(u,v;\beta) : \forall\beta\}. \tag{2}$$

For Gabor bases, $\beta = (x, y, \sigma, \theta)$ specifies the center, scale, and orientation of the base function. For the isotropic LoG bases, $\beta = (x, y, \sigma)$ specifies its center and scale. Thus, each base is an attributed point (or token) $\mathbf{b}_j = (\alpha_j, \beta_j, \gamma_j)$, and the photometric model in (1) transfers a raw image $\mathbf{I}$ into a token representation, which is a layer of hidden variables—called the *particle base map*,

$$\mathbf{B}_{\rm pcl} = \{\mathbf{b}_j = (\alpha_j, \beta_j, \gamma_j) : j = 1, 2, \ldots, N_{\rm pcl}\}. \tag{3}$$

2. **Wave bases** $\Delta_{\rm wav}$. The wave dictionary is constructed from a single Fourier function ${\rm FB}(u,v)$ with transforms $\beta = (\xi, \eta, \phi)$ on its spatial frequency (ξ, η) and phase ϕ.

$$\begin{aligned}\Phi_{\rm wav} &= \{{\rm FB}(u,v)\},\\ \Delta_{\rm wav} &= \{{\rm FB}(u,v;\beta) = e^{-i(\xi u + \eta v + \phi)} : \forall\beta\}.\end{aligned} \tag{4}$$

Let α_j be the Fourier coefficient, then the selected Fourier bases form a *wave base map*,

$$\begin{aligned}\mathbf{B}_{\rm wav} = \{&\mathbf{b}_j = (\alpha_j, \beta_j, \gamma_j) :\\ &\gamma_j = {\rm FB}, j = 1, 2, \ldots, N_{\rm wav}\}.\end{aligned} \tag{5}$$

3. **Initializing the base map by match pursuit**. For Δ is not an orthogonal basis, we adopt a match pursuit [17] to compute the base map $\mathbf{B}$ at the initial stage.

Let $\mathbf{I}^{\rm obs}$ be an input image and $\mathbf{I}$ be its reconstruction by the bases. The match pursuit algorithm starts with $\mathbf{B}$ being an empty set and $\mathbf{I}$ being a constant image whose intensity equals the mean of $\mathbf{I}^{\rm obs}$. The match pursuit algorithm selects one base at each step so as to minimize the squared reconstruction error until the coefficient of the selected base falls below a threshold.

```
Repeat
    $\psi_{\gamma_j}(\beta_j) = \arg\max_{\psi\in\Delta} | < \psi, \mathbf{I}^{\rm obs} - \mathbf{I} > |^2,$
        $\alpha_j = < \psi_{\gamma_j}(\beta_j), \mathbf{I}^{\rm obs} - \mathbf{I} >,$
    $\mathbf{B} \leftarrow \mathbf{B} \cup \{(\alpha_j, \beta_j, \gamma_j)\}, \mathbf{I} \leftarrow \mathbf{I} + \alpha_j\psi_{\gamma_j}(\beta_j),\ j \leftarrow j+1,$
until $|\alpha_j| \leq \epsilon$.
```

It is a greedy algorithm and generates a sequence of bases with decreasing coefficients,

$$\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_N \geq \epsilon.$$

The rate of decrease reflects the efficiency of the dictionary Δ (see plots in Fig. 2). The computed base map $\mathbf{B}$ will be corrected by a Markov chain Monte Carlo algorithm integrating the motion cues in Section 3.3.

We show an example of the match pursuit procedure on a snow image in Fig. 1. We limit the dictionary to particle bases $\Delta_{\rm pcl}$ and show the reconstruction in three stages with decreasing thresholds $\epsilon_1 > \epsilon_2 > \epsilon_3$. At each stage, we visualize the base map $\mathbf{B}_{\rm pcl}$ in a Gabor map (left) and a LoG map (right). A Gabor base is sketched symbolically by a bar with the same size and orientation as the Gabor function and a LoG base is sketched by a circle with the size representing its scale. The brightness of the bars and circles represents the coefficients (bright for positive and dark for negative). Each base map reconstructs a "subband" image. In the bottom row, the subband images of the two types are added to yield the reconstructed image $\mathbf{I}$. We stop the algorithm with $N_{\rm pcl} = 800$ bases.

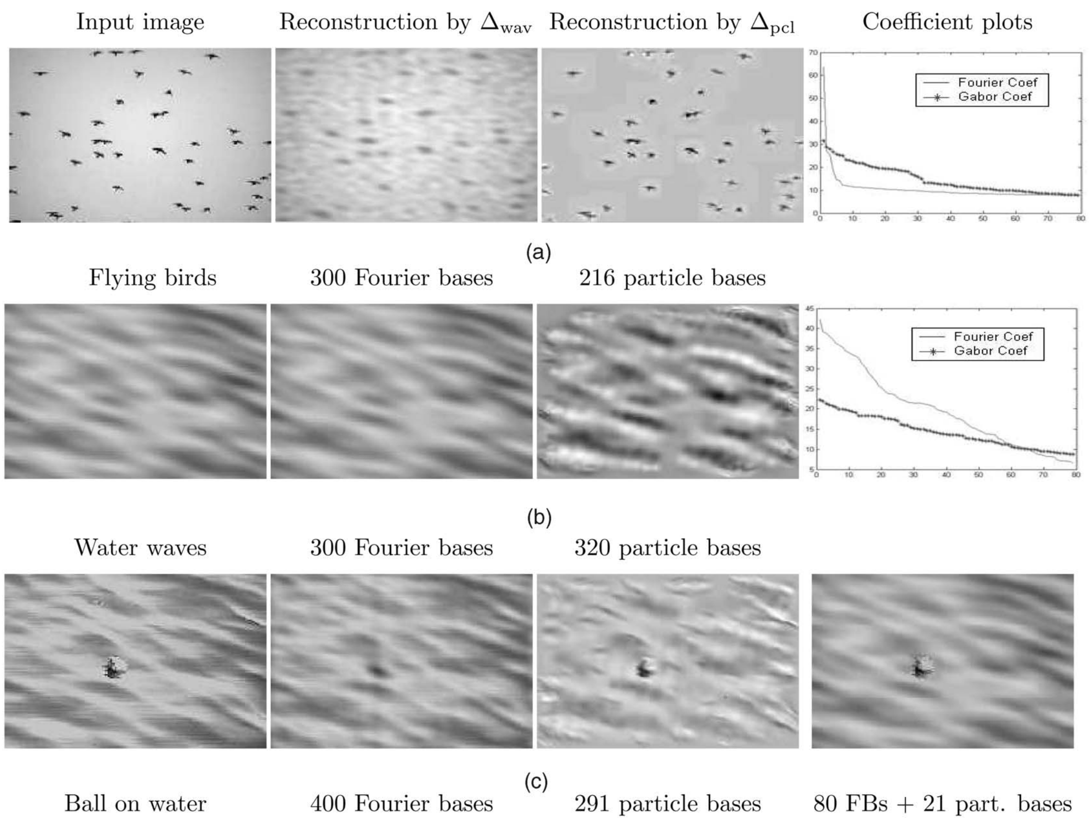


Fig. 2. Comparison of image reconstructions by wave bases Δ_{wav} and particle bases Δ_{pcl}, respectively. The curves plot the coefficients of the selected bases in decreasing order. The thick curve is for Δ_{pcl}. The slopes of the curves reflect the coding efficiency of the dictionary. (a) Flying birds. (b) Wavy river. (c) An image with both particles and waves—a floating ball in a river.

4. **Comparison of bases**. Since Δ is overcomplete, there are a combinatorial number of ways to reconstruct an input image. We compare the effectiveness of the particle bases Δ_{pcl} and the wave bases Δ_{wav} in Fig. 2 when reconstructing textured motion patterns. We select three typical images, and reconstruct each by the wave bases and the particle bases, respectively.

   Fig. 2a is a flock of birds. The reconstruction with $N_{\text{wav}} = 300$ Fourier bases is very blurry. In contrast, the reconstruction with $N_{\text{pcl}} = 216$ particle bases captures the birds accurately. Fig. 2b is a river image where the Fourier bases are found to be more effective than the particle bases. Fig. 2c shows a ball floating in a river. We can see that neither Δ_{pcl} nor Δ_{wav} alone is able to effectively represent this image. A combination of 80 Fourier bases and 21 Gabor bases exhibits better reconstruction.

   For the bird and water images, we plot the coefficients of the selected bases in Δ_{wav} and Δ_{pcl} in a decreasing order, respectively. A steep slope of the curve implies that the bases are effective in reconstructing the image, whereas a flat curve means the opposite. For the bird image, the curve plot shows that the first few Fourier bases have large responses in capturing the global lighting condition in the sky. Therefore, the best representation for this image is a few Fourier bases for lighting plus the particle bases for individual birds. For the water image, the dominance of Fourier bases is obvious.

Since, the particle and wave bases compete to explain the input image through the match pursuit algorithm, we obtain the base map by

$$\begin{aligned}\mathbf{B} &= \mathbf{B}_{\text{pcl}} \cup \mathbf{B}_{\text{wav}} = \{\mathbf{b}_j = (\alpha_j, \beta_j, \gamma_j),\ \ j = 1, 2, \ldots, N\},\\ N &= N_{\text{pcl}} + N_{\text{wav}} = O\left(\frac{|\Lambda|}{100}\right).\end{aligned} \tag{6}$$

After using the match pursuit algorithm to initialize the base map $\mathbf{B}$, the ambiguities in base selection can be resolved by a Monte Carlo algorithm in Section 3.3, which traces the bases over image frames, adjusts the bases for spatial-temporal coherence, and eliminates some false bases.

To summarize the photometric model, we rewrite (1) in a conditional probability. It specifies how a given image $\mathbf{I}^{obs}$ is generated by a hidden layer of bases $\mathbf{B}$,

$$\begin{aligned}&p(\mathbf{I}^{\text{obs}} | \mathbf{B}_{\text{pcl}}, \mathbf{B}_{\text{wav}}; \Delta, \sigma_o) = \\ &\frac{1}{(2\pi\sigma_o^2)^{|\Lambda|/2}} \exp\left\{ -\frac{1}{\sigma_o^2} \sum_{(u,v)\in\Lambda} (\mathbf{I}^{\text{obs}}(u, v) \right. \\ &\left. - \sum_{j=1}^{N} \alpha_j \psi_{\gamma_j}(u, v; \beta_j))^2 \right\}.\end{aligned} \tag{7}$$

2.2 Geometric Model: Identifying Motons—The Basic Moving Elements

In this section, we discuss a geometric model for grouping the N bases in $\mathbf{B}$ into a smaller number of moving elements

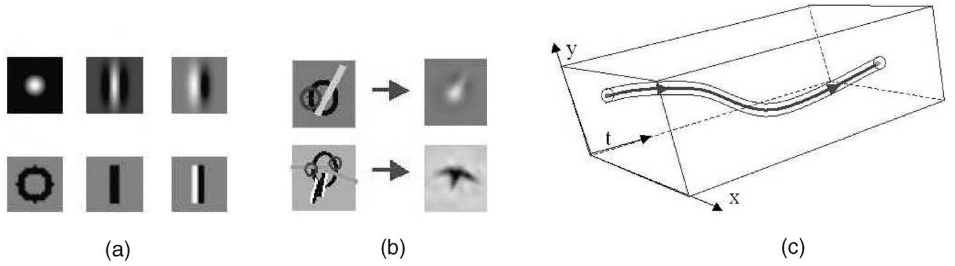


Fig. 3. Motons as basic moving elements. (a) Three base functions—LoG, Gcos, Gsin, and their symbolic sketches—circles, bars, and step edge. (b) Examples of learned motons—a snowflake and a bird. (c) A graphic view of a moton trajectory—the cable model.

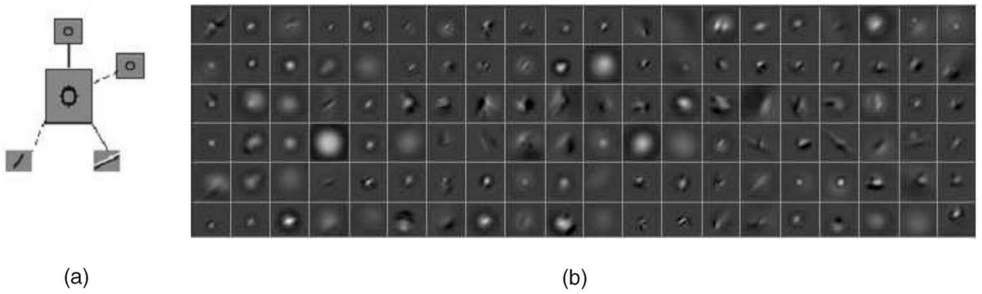


Fig. 4. The computed motion elements: snowflakes and random examples. (a) A moton template in atomic structure and (b) 120 instances of snowflakes as motons π.

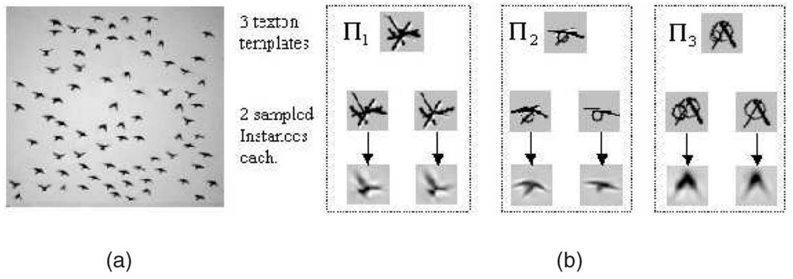


Fig. 5. Motons in a bird flying sequence. (a) Input image. (b) Three moton templates $\Phi_\pi = \{\Pi_i, i = 1, 2, 3\}$ learned in a clustering step for different poses. Two instances are shown for each template.

called "motons." First, we study the particle bases. In the base map $\mathbf{B}_{\text{pcl}}$, a moving element, such as the snowflake or bird in Fig. 3b, is often represented by 3-7 adjacent bases which travel together as a coherent group. For example, in Fig. 3b, a snowflake image is the sum of three bases, two LoG bases and one Gcos base with certain space displacements, and a bird image consists of seven bases, three LoG bases, two Gcos bases, and two Gsin bases. Therefore, $\mathbf{B}_{\text{pcl}}$ can be divided into M_{pcl} disjoint subsets, each corresponding to a moving element.

$$\mathbf{B}_{\text{pcl}} = S_1 \cup S_2 \cup \cdots \cup S_{M_{\text{pcl}}}.$$

In a textured motion scene, the M_{pcl} moving elements are instances of N_π-types of objects or spatial configurations. For example, we set $N_\pi = 1$ for the snowflakes in the snowing scene in Fig. 4 and $N_\pi = 3$ for the three poses of the flying birds in Fig. 5. Each of the configurations is represented by a deformable template Π_ℓ. Π_ℓ includes the following variables: the number of bases n_ℓ, the n_ℓ bases $b_{\ell,1}, \ldots, b_{\ell,n_\ell} \in \Delta_{\text{pcl}}$ with relative positions. The template is specified by three groups of parameters. 1) The global transform $\beta = (x, y, \sigma, \theta)$ for translation, scaling, and rotation. 2) A membership vector $\rho = (\rho_1, \ldots, \rho_{n_\ell}) \in \{0,1\}^{n_\ell}$ indicating the presence or absence of the base components due to occlusion or variation. 3) A vector ζ for the deformations of the template. We denote the set of templates by

$$\Phi_\pi = \{\Pi_\ell = (n_\ell, b_{\ell,1}, \ldots, b_{\ell,n_\ell}; \beta_\ell, \rho_\ell, \zeta_\ell) : \ell = 1, 2, \ldots, N_\pi\}.$$

A dictionary of motons is obtained from Φ_π by varying (β, ρ, ζ) and is denoted by

$$\Delta_\pi = \{\ \pi(\ell, \beta, \rho, \zeta) : , \forall \ell, \beta, \rho, \zeta\ \}. \tag{8}$$

Fig. 4a displays the snowflake template Π_1 and a variety of 120 snowflake instances $\{\pi_1, \ldots, \pi_{120}\}$ generated by this deformable template. Usually, a template has a "heavy" base with relatively large coefficient α_i which is surrounded by several "light" bases with relatively small coefficients. By analogy to the physical model of atoms, we call the heavy bases "nucleus" bases and the light bases "electron" bases. The atomic models for the birds are illustrated in Fig. 6b.

With dictionary Δ_π, the base map $\mathbf{B}_{\text{pcl}}$ is generated by a layer of moton map $\mathbf{M}_{\text{pcl}}$ with a subset $S_i \in \mathbf{B}$ for each moton π_i. Thus, we arrive at a more abstract and parsimonious representation,

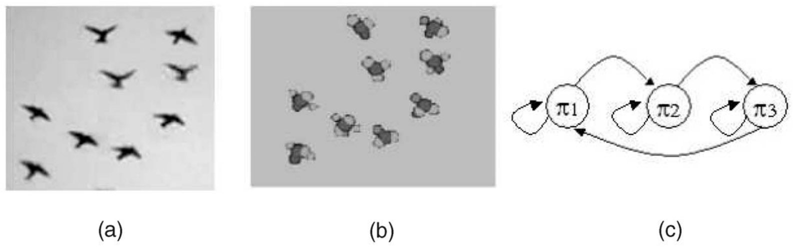


Fig. 6. (a) Input image. (b) Three-dimensional graphic illustration for the "atomic" model of bird motons $\pi_j, j = 1, 2, \ldots, 9$. (c) Diagram of three-state transitions for birds flying.

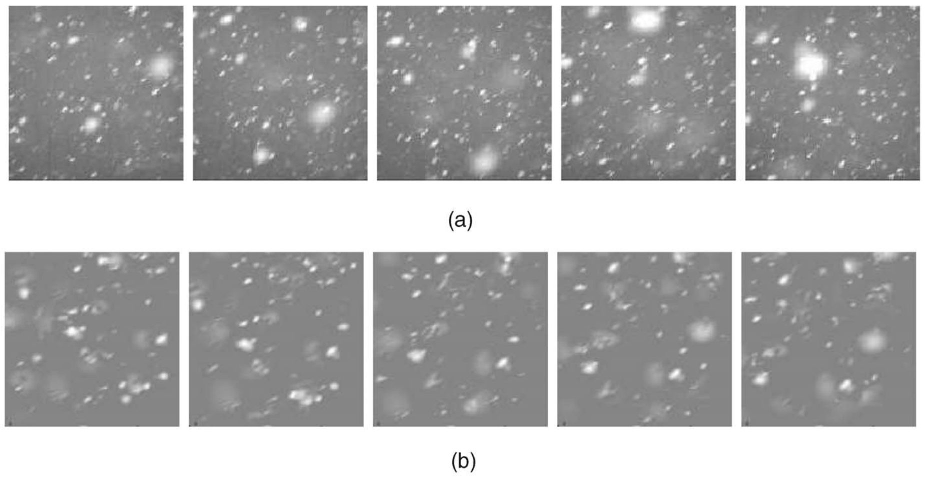


Fig. 7. Experiment on the falling snow sequence. (a) Observed sequence. (b) Synthesized sequence by sampling the generative model.

$$\mathbf{M}_{\text{pcl}} = \{\pi_j(\ell_j, \beta_j, \rho_j, \zeta_j), j = 1, 2, \ldots, M_{\text{pcl}},\ 1 \le \ell_j \le k\}.$$

Second, we study the wave base map. According to the theory of ocean waves [28], the bases in Δ_{wav} also travel in groups. For example, water flows travel as sinusoid waves caused by different sources of vibration, such as wind, boats, or earthquake. But, such motion groups can only be seen at a macroscopic scale. In our experiments, waves travel in a single group. To conform the notation, we denote a wave moton by $\pi_j = \mathbf{b}_j$ and the moton map for the waves is $\mathbf{M}_{\text{wav}} = \{\pi_j = \mathbf{b}_j; j = 1, 2, \ldots, N_{\text{wav}}\} = \mathbf{B}_{\text{wav}}$, then $\mathbf{M} = \mathbf{M}_{\text{pcl}} \cup \mathbf{M}_{\text{wav}}$. Thus, we have a two-level generative model; the moton map $\mathbf{M}$ generates the base map $\mathbf{B}$ with dictionary Δ_π; and the base map $\mathbf{B}$, in turn, generates the image $\mathbf{I}$ with dictionary Δ.

$$\begin{aligned} \mathbf{M} = \mathbf{M}_{\text{pcl}} \cup \mathbf{M}_{\text{wav}} &\overset{(\Phi_\pi, \Delta_\pi)}{\longrightarrow} \\ \mathbf{B} = \mathbf{B}_{\text{wav}} \cup \mathbf{B}_{\text{pcl}} &\overset{(\Phi_{\text{pcl}}, \Delta_{\text{pcl}}) \cup (\Phi_{\text{wav}}, \Delta_{\text{wav}})}{\longrightarrow} \mathbf{I}. \end{aligned} \tag{9}$$

In summary, the geometric model can be expressed as a conditional probability,

$$\begin{aligned} p(\mathbf{B}|\mathbf{M}; \Phi_\pi) &= p(\mathbf{B}_{\text{pcl}}|\mathbf{M}_{\text{pcl}}; \Phi_\pi) p(\mathbf{B}_{\text{wav}}|\mathbf{M}_{\text{wav}}) \\ &= \left[\prod_{i=1}^{M_{\text{pcl}}} \prod_{j \in S_i} p(\mathbf{b}_j | \pi_i; \Pi_{\ell_i}) \right] \cdot \delta(\mathbf{B}_{\text{wav}} - \mathbf{M}_{\text{wav}}). \end{aligned} \tag{10}$$

Combining (7) and (10), we obtain the generative model for a still image $\mathbf{I}^{\text{obs}}$—$p(\mathbf{I}^{\text{obs}}|\mathbf{B}; \sigma_o) p(\mathbf{B}|\mathbf{M}; \Phi_\pi)$ with $\mathbf{B}$ and $\mathbf{M}$ being the hidden variables. In Section 3, we shall discuss that the learning algorithm achieves several objectives:

1. dividing $\mathbf{B}$ into M_{pcl} groups,
2. clustering the M_{pcl} groups into N_π moton templates,
3. adjusting the initial $\mathbf{B}$ obtained by match pursuit for spatial coherence, and
4. learning the templates Φ_π.

All these steps must be integrated with the motion representation, which will be presented in the rest of the section.

2.3 The Moton Trajectories and Representation of the Video Sequence

For a video sequence $\mathbf{I}[0, \tau]$, the moton map $\mathbf{M}$ and the base map $\mathbf{B}$ shall be tracked over time and adjusted for motion coherence. Let $\pi(t)$ be the state of a moton at time t and $\mathcal{C}[t^b, t^e]$ be its trajectory,

$$\mathcal{C}[t^b, t^e] = (\pi(t^b), \pi(t^{b+1}), \ldots, \pi(t^e)), \quad [t^b, t^e] \subset [0, \tau]. \tag{11}$$

For example, a snowflake or a bird enters the image view at frame t^b and leaves the view at frame t^e (Figs. 8 and 10 show some snowflake and bird trajectories). Intuitively, a moton trajectory is like a cable where the trajectory of its "nucleus base" forms the *core* of the cable and the trajectories of its "electron" bases form the *coils* surrounding the core due to self-rotation. In a coarse-to-fine computation, we compute the trajectories of the cores first and then add the coils sequentially. In practice, the core of a moton is relatively consistent through its lifespan, but the number of coil bases may change over time due to self-occlusion, etc. Thus, we should use temporal coherence to regularize the coil trajectories while learning the deformable templates Φ_π.

We change the index from image frame t to moving element i and transform the two-level hidden representation

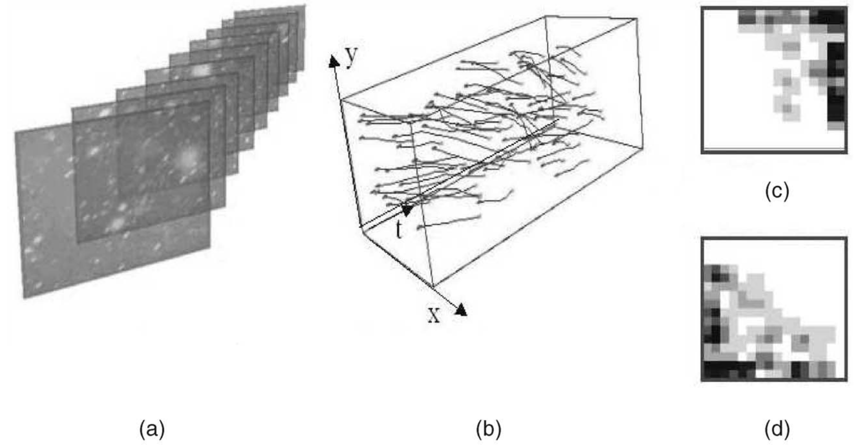


Fig. 8. Experiment on the falling snow sequence. (a) Observed sequence. (b) Graphic view of the computed trajectories of the snowflakes (as hidden variables). (c) A probability map of the sources for snowflakes to enter the scene. (d) Probability map of the sinks for the snowflakes to leave the view. Dark means high probability. Both the source and the sink maps are marginal probabilities over the locations (summed over time and other attributes).

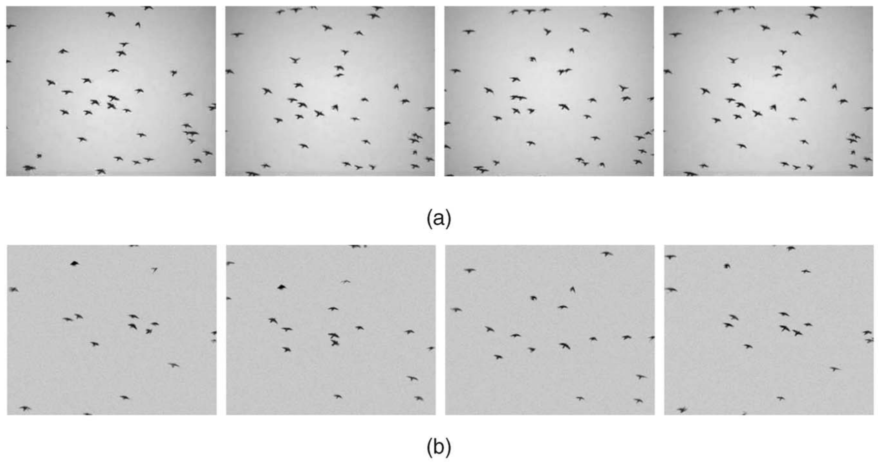


Fig. 9. Experiment on a flying-bird sequence. (a) Observed sequence. (b) Synthesized sequence with fewer flying birds by editing the number of motons M_{pcl} when sampling the model.

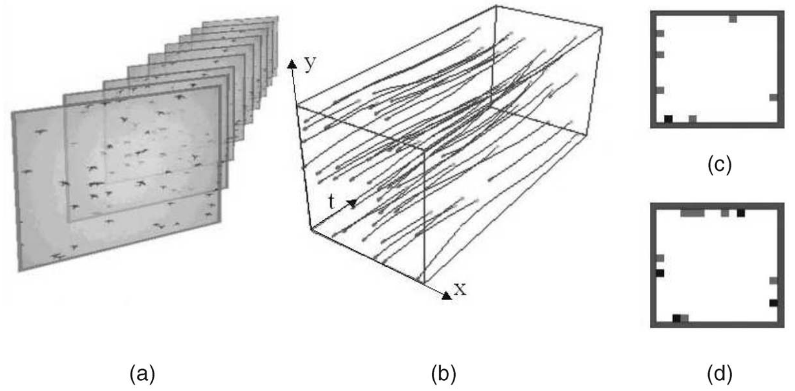


Fig. 10. Experiment on the bird sequence. (a) Observed sequence. (b) Graphic view of the computed trajectories of the birds. (c) A probability map of the sources for birds to enter the scene. (d) Probability map of the sinks for birds to leave the view. Dark means high probability.

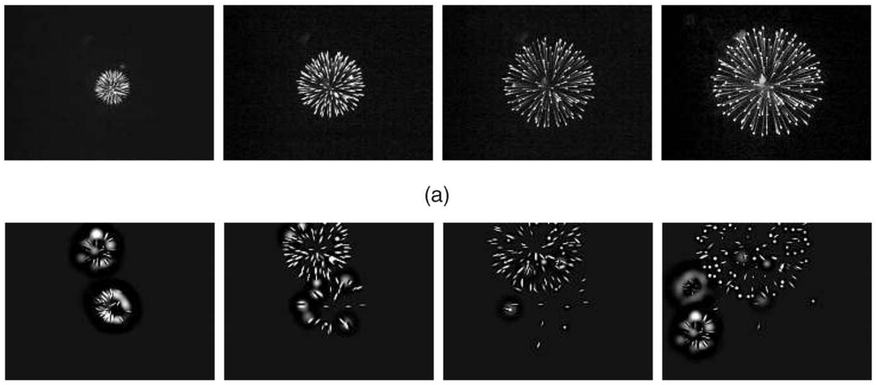
$\mathbf{B}[0,\tau]$ and $\mathbf{M}[0,\tau]$ to a representation with K trajectories $\mathcal{C}_i, i = 1,2,\ldots,K$, plus waves.

$$\begin{aligned} W[0,\tau] &= (\mathbf{M}[0,\tau], \mathbf{B}[0,\tau]) \\ &= (\mathbf{B}_{\text{wav}}[0,\tau], K, \{\mathcal{C}_i[t_i^b, t_i^e], i = 1,2,\ldots,K\}). \end{aligned} \tag{12}$$

K is the number of particle objects that appear in the video sequence. The number of motons and bases may change over time due to the birth and death events. $W[0,\tau]$ includes all the bases and motons. It is a low-dimensional generative representation for the video $\mathbf{I}[0,\tau]$.

2.4 Dynamic Model—Sources, Sinks, and Wave-Particle Interactions

The dynamic model characterizes the sources, sinks, and trajectories of the motons as well as their interactions. We



(a)

(b)

Fig. 11. Experiment on firework sequence. (a) Observed sequence with only one firework. (b) Synthesized sequence of multiple fireworks after editing its birth (source) map.

model two types of interactions. The first type is the interaction among wave components. Unlike particles such as birds and snow flakes, which move rather independently, waves travel together with complex interactions. Therefore, the relative motion of different Fourier bases must be constrained to keep certain phase alignments. The second type is the influence of waves on particles, e.g., balls drifting in a river, grass waving in the wind. Other interactions, such as particle-particle collision and particle-wave collision (splash) are not considered in this paper.

Let $\pi(t)$ denote the state of a moton or a wave base at time t, for particles $\pi = (\ell, \beta, \rho, \zeta)$, and for waves $\pi = (\alpha, \xi, \eta, \phi)$. The general motion equation for $\pi(t)$ is a pth order AR model with coefficients $a = (a_1, \ldots, a_p)$, driven by three types of forces: 1) influence from (other) waves $U(\mathbf{B}_{\text{wav}}(t))$, 2) external force field $f(\pi(t-1))$, such as gravity, wind field, and external constraints, which may vary over space and time, and 3) a Brownian motion n. Thus, we have a general motion equation for the textured motion patterns.

$$\begin{aligned} \pi(t) = & \sum_{j=1}^{p} a_j \pi(t-j) + U(\mathbf{B}_{\text{wav}}(t)) \\ & + f(\pi(t-1)) + n, \quad n \sim \Lambda(0, \sigma^2). \end{aligned} \tag{13}$$

In the rest of this section, we study three special cases that occur in our experiments.

Case 1. Dynamic model for independent moving particles—snow, birds, and fireworks.

The first case represents textured motion patterns with particles elements that move rather independently, such as snowing, birds flying, fireworks, etc. Though a few Fourier bases are used to model the global lighting effects, they are static and do not affect the motons. The external force $f(\pi) = c$ is a constant vector. Thus, the general motion equation (13) reduces to a second order Markov chain model,

$$\begin{aligned} \pi(t) = \; & a_1 \pi(t-1) + a_2 \cdot \pi(t-2) + c + n, \\ & n \sim \Lambda(0, \sigma^2) \;\; t \in [t^b + 1, t^e] (\pi(t^b), t^b) \sim P_B(\pi, t), \\ & (\pi(t^e), t^e - t^b) \sim P_D(\pi, t). \end{aligned}$$

The birth of a moton π and its timing t^b follow a probability $P_B(\pi, t)$. Its marginal probability on the location $P_B(x, y)$ (summed over time and other attributes) is called the "source map" or "birth map." The timing is important, for example, for controlling the fireworks. Similarly, the end of the trajectory $\pi(t^e)$ and its life span $t^e - t^b$ are governed by a probability $P_D(\pi, \lambda)$. Its marginal probability $P_D(x, y)$ reveals the "sinks" and is called the "death map." Note that π is a long vector and P_B and P_D are high-dimensional probabilities. In practice, we are most interested in the location (x, y).

The probabilities P_B and P_D are represented in a nonparametric form using Parzen windows. During the learning process, suppose we have computed K cables $\mathcal{C}_i[t_i^b, t_i^e], i = 1, 2, \ldots, K$ from a sequence $\mathbf{I}[0, \tau]$, we represent P_B and P_D as

$$\begin{aligned} P_B(\pi, t) &= \frac{1}{K} \sum_{i=1}^{K} \delta(\pi - \pi_i(t_i^b), t - t_i^b), \\ P_D(\pi, t) &= \frac{1}{K} \sum_{i=1}^{K} \delta(\pi - \pi_i(t_i^e), t - (t_i^e - t_i^b)), \end{aligned} \tag{14}$$

where $\delta()$ is a Parzen window centered at 0. When we project P_B and P_D to (x, y) plane, we get the death and birth maps. Fig. 8 (right column) displays the birth map $P_B(x, y)$ and the death map $P_D(x, y)$ for a snow sequence, where darker spots have higher probabilities. Thus, the algorithm "understands" that the snowflakes mostly enter from the upper-right corner and disappear around the lower-left corner.

In summary, the probability for a moton trajectory is in the form

$$\begin{aligned} & p(\mathcal{C}[t^b, t^e]; \Gamma_{\text{pcl}}) = \\ & P_B(\pi(t^b)) P_D(\pi(t^e), t^e - t^b) \prod_{t=t^b+1}^{t^e} p(\pi(t) | \pi(t-1), \pi(t-2)). \end{aligned} \tag{15}$$

In the above model, $\Gamma_{\text{pcl}} = (a_1, a_2, c, \sigma, P_B, P_D)$ denotes all the parameters in the dynamic models.

Due to space limit, we briefly remark on two details in the experiments of Case 1.

Remark 1. For the firework sequence in Fig. 11, the death and birth of motons must be synchronized, as a large number of particles come and go together. We manipulated the birth

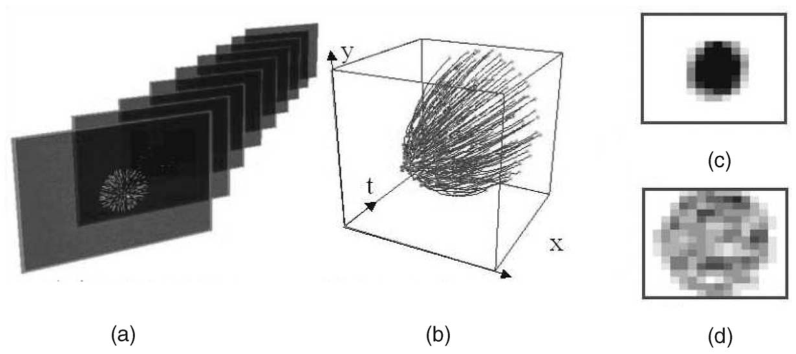


Fig. 12. Experiment on the firework sequence. (a) Observed sequence. (b) Graphic view of the trajectories of the firework. (c) Source map of the firework. (d) Sink map of the firework.

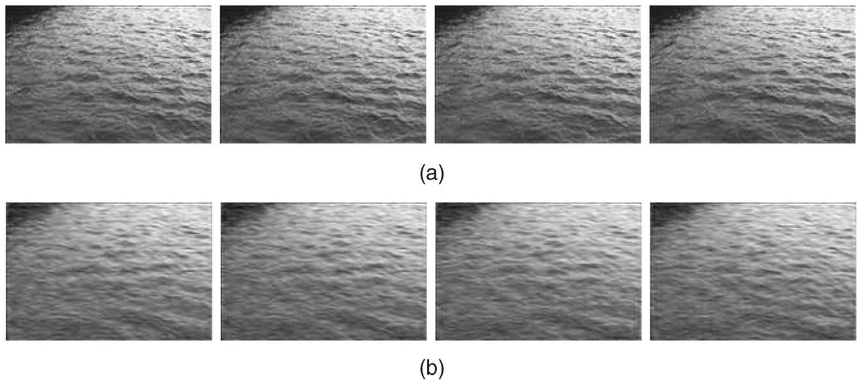


Fig. 13. Experiment on a river sequence. (a) Observed sequence of wavy river. (b) Synthesized sequence with $1,000$ Fourier bases.

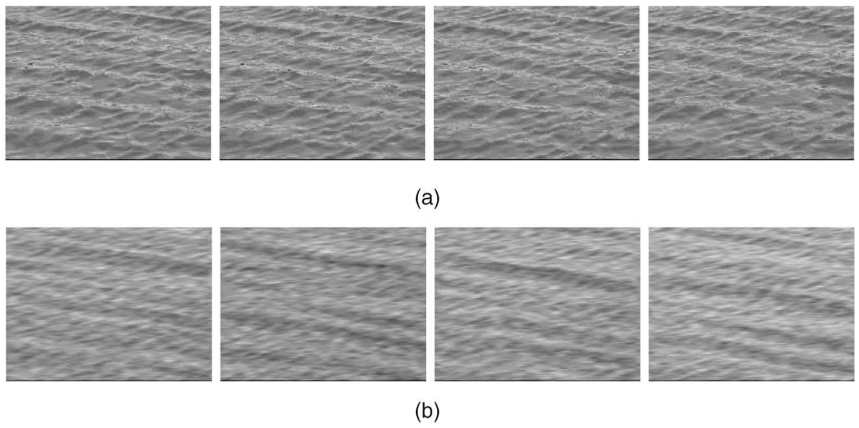


Fig. 14. Experiment on a wavy pond sequence. (a) Observed sequence. (b) Synthesized sequence with $1,200$ Fourier bases.

map to illustrate controllability. For example, we observe a single firework at the center of the original sequence. By editing the birth probability P_B, we can easily render many fireworks at all places and time intervals (See Fig. 11b). Similarly, by reducing the motons' number ($M_{\rm pcl}$), the model generates fewer birds in the synthesized sequence in Fig. 9b.

Remark 2. For the bird sequence, the moton $\pi(t)$ comes from three possible templates, $\Phi_\pi = \{\Pi_1, \Pi_2, \Pi_3\}$, and may change states over time (see Fig. 6c). In order to have the birds flap wings properly, the Markov chain model $p(\pi(t)|\pi(t-1), \pi(t-2))$ includes a first order transition probability $p(\ell(t)|\ell(t-1))$ as $\ell(t) \in \{1,2,3\}$ is a variable in $\pi(t)$. The transition probability is represented by a 3×3 matrix. Note that this is not necessary in the snow and firework sequences.

Case 2. Dynamic model for waves—river, pond, and plastics.

The second case represents pure wave sequences with only Fourier bases, e.g., Figs. 13, 14, and 15. There is no birth/death event. The variables are

$$\begin{aligned} W &= \mathbf{M} = \mathbf{B} = \mathbf{B}_{\rm wav} \\ &= \{(\alpha_j, \xi_j, \eta_j, \phi_j),\ j = 1, 2, \ldots, N_{\rm wav}\},\ \ N_{\rm wav} = O(10^3). \end{aligned}$$

If the camera does not move and the motion is stationary, then the Fourier frequencies ξ_j, η_j and amplitudes α_j are time-invariant. Only the phases $\phi_j, j = 1, \ldots, N$ change and this is known as the phase motion [8]. The speed of phase $d\phi$ is related to the speed (dx, dy) by

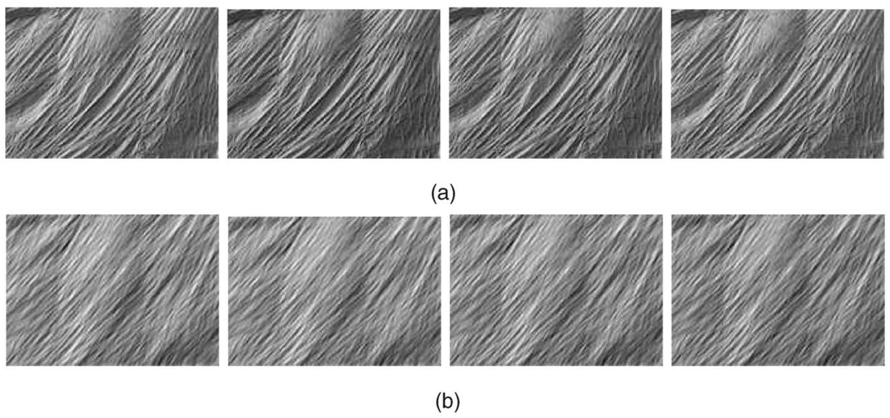


Fig. 15. Experiment on a plastic foil sequence. (a) Observed sequence. (b) Synthesized sequence with $1,500$ Fourier bases.

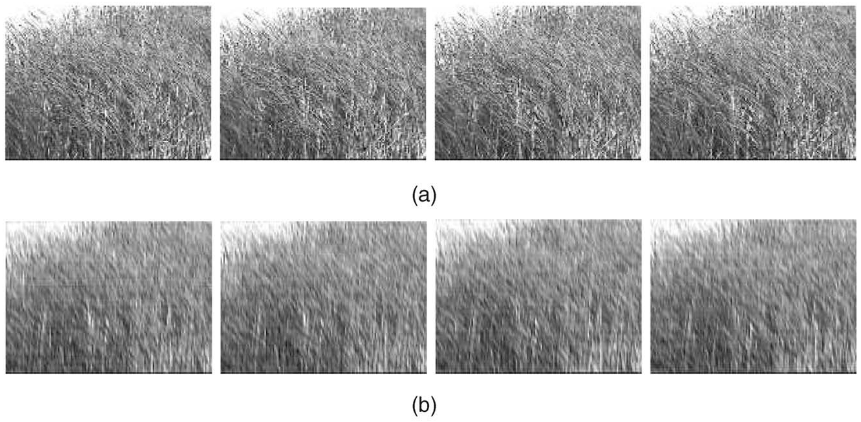


Fig. 16. Experiment on a grassland sequence. (a) Observed sequence. (b) Synthesized sequence with $3,000$ Fourier bases for its spatial wave pattern and high-frequency structures.

$$\frac{d\phi_j(t)}{dt} = \xi_j \frac{dx}{dt} + \eta_j \frac{dy}{dt}, \quad \text{or} \quad \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} \xi_j \\ \eta_j \end{pmatrix} d\phi_j/(\xi_j^2 + \eta_j^2). \tag{16}$$

A slight complication is that we have to wrap the phase into a periodical interval $[0, 2\pi)$ in computing $d\phi_j$ and (dx, dy) [8].

Our first attempt is to let each Fourier base move independently in an AR model, as it is for the particles in Case 1.

$$\begin{aligned} \phi_j(t) &= \sum_{i=1}^{p} a_{ji}\phi_j(t-i) + n_j, \\ n_j &\sim \mathcal{N}(0, \sigma^2), \quad j = 1, 2, \ldots, N. \end{aligned}$$

With $p = 15 \sim 20$ accounting for low frequency components, this simple model can synthesize the river sequences reasonably well. However, the phases become misaligned after 30-50 frames. To solve this problem, we study a joint vector $\phi(t) = (\phi_1(t), \ldots, \phi_{N_{\rm wav}}(t)), t \in [0, \tau]$. To reduce the dimension, we employ a standard PCA on $\{\phi(0),, \phi(\tau))$ and obtain $e_i, i = 1, 2, \ldots, m$ as the eigenvectors with the largest eigenvalues. Then, we project $\phi(t)$ to an m-vector $\chi(t) = (\chi_1(t), \ldots, \chi_m(t))$, where $\chi_i(t) =< \phi(t), e_i >$. In our experiments, $m = 8$ and the m coefficients follow a pth order AR model independently as in the particle sequences,

$$\begin{aligned} \chi_j(t) &= \sum_{i=1}^{p} a_{ji}\chi_j(t-i) + n, \\ n &\sim \mathcal{N}(0, \sigma_j^2), \ \ p = 20, \ j = 1, 2, \ldots, m = 8. \end{aligned} \tag{17}$$

The total number of variables used in the model is $3N_{\rm wav}$ for $(\xi_j, \eta_j, \alpha_j), j = 1, \ldots, N_{\rm wav}$, and $8N_{\rm wav}$ for the eigenvectors, plus 20×8 for the AR coefficients.

In summary, we write the wave dynamics in the following probability model,

$$p(\mathbf{B}_{\rm wav}[0, \tau]; \Gamma_{\rm wav}) = \prod_{j=1}^{m} \prod_{t=0}^{\tau} p(\chi_j(t) | \chi_j(t-1), \ldots, \chi_j(t-p)). \tag{18}$$

We assume some initial conditions for the first p frames and $\Gamma_{\rm wav} = (a_1, \ldots, a_p)$ denotes all the parameters.

Figs. 13 and 14 show the synthesis results for the river and pond waves. The same model is applied to the plastic foil in Fig. 15 and the grass sequence in Fig. 16. In general, the wavy plastic foil and grass are driven by an invisible wind field which has wave properties. For the grass sequence, we need more Fourier bases, $N_{\rm wav} = 3,000$, to reconstruct the high-frequency components.

Case 3. Dynamic model for particles—waves interactions: a ball or foam on water.

Our third case concerns motion sequences with both particles and waves, such as a ball or foam floating on water as

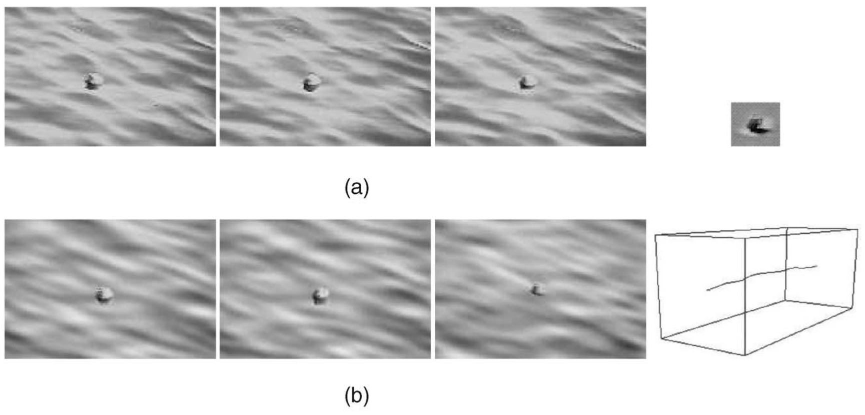


Fig. 17. Experiment on a floating-ball sequence. (a) Observed sequence and the learned moton: the ball. (b) Synthesized sequence and the trajectory of the ball.

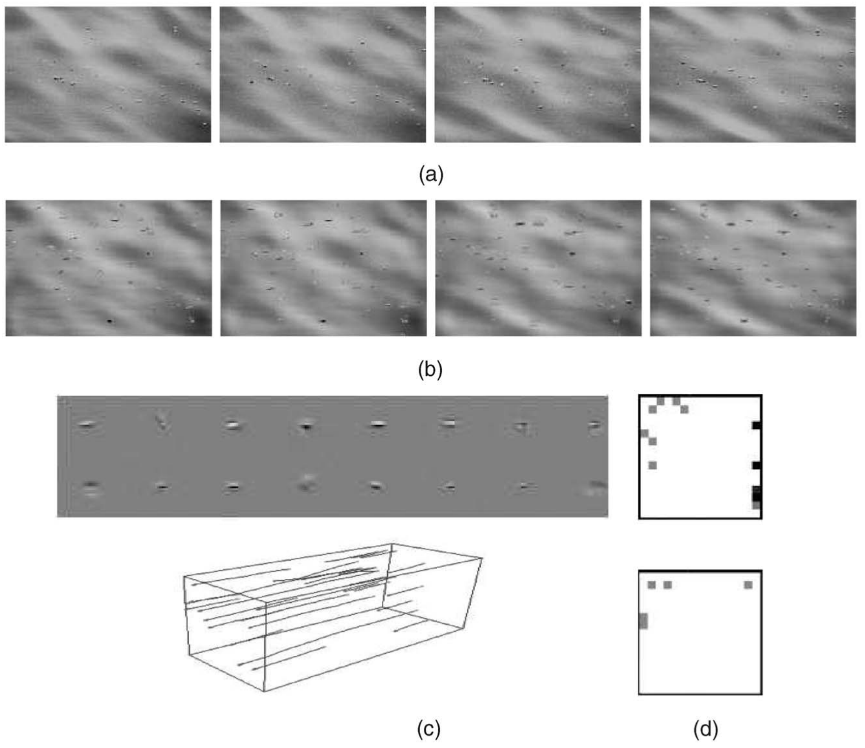


Fig. 18. Experiment on a sequence with many foam particles drifting in a river. (a) Observed sequence. (b) Synthesized sequence. (c) Learned motons: foams and their trajectories. (d) Sources and sinks of the floating foam.

shown in Figs. 17 and 18, respectively. The coupling of the two types of elements is characterized by a driving force from waves to particles. As the particles are small, we are only concerned about their positions (x, y) and fix other attributes in π. To conform the notation, we write π for (x, y).

Let $\phi(t) = (\phi_1(t), \dots, \phi_{N_{\text{wav}}}(t))$ be the phases of all Fourier bases whose motion follows the dynamic model in Case 2. Given the phase motion $d\phi$ in Case 2, we transfer it into motion velocity in spatial domain (dx, dy) by (16). The motion of a particle is then influenced by the sum of all wave velocities at point (x, y). In practice, we only need to choose $q = 20 \sim 30$ lower frequency Fourier bases $\{(\tilde{\xi}_k, \tilde{\eta}_k, \tilde{\phi}_k) : k = 1, 2, \dots, q\}$ to drive the particles. Thus, the motion equation of a particle $\pi(t) = (x(t), y(t))$ is:

$$\begin{aligned}\pi_j(t) = &\sum_{i=1}^{p} a_j \pi(t-i) + \\ &\sum_{k=1}^{q} b_k(\tilde{\xi}_k, \tilde{\eta}_k) d\tilde{\phi}_k(t) + c + n, \quad n \sim \mathcal{N}(0, \sigma_o^2), \ \forall j.\end{aligned} \tag{19}$$

The second term in the above equation accounts for the coupling of the particle motion with waves and a, b are the coefficients. The death and birth of particles follow the same model in Case 1. This Markov chain model follows the probability below:

$$\begin{aligned}&p(\mathcal{C}[t^b, t^e]; \Gamma_{\text{pcl}}) = p_B(\pi(t^b)) p_D(\pi(t^e), t^e - t^b) \\ &\prod_{t=t^b+1}^{t^e} p(\pi(t) | \pi(t-1), \pi(t-2), \tilde{\phi}_1(t), \dots, \tilde{\phi}_q(t)).\end{aligned} \tag{20}$$

The wave bases follow the dynamics in (17).

Figs. 17 and 18 show the experiment results on the ball and foam sequences, respectively. The coupling of the particles with waves appears realistic in the video sequence.

To conclude Section 2, we integrate the photometric model (7), the geometric model (10), and the dynamic models in (15), (18), and (20) into a joint probability for an image sequence $\mathbf{I}^{\text{obs}}[0,\tau]$ and the hidden representation $W[0,\tau]$,

$$\begin{aligned} &p(\mathbf{I}^{\text{obs}}[0,\tau], W[0,\tau];\Theta) = \\ &\left[\prod_{t=1}^{\tau} p(\mathbf{I}^{\text{obs}}(t)|\mathbf{B}_{\text{pcl}}(t),\mathbf{B}_{\text{wav}}(t))\cdot p(\mathbf{B}_{\text{pcl}}(t)|\mathbf{M}_{\text{pcl}}(t);\Phi_\pi)\right] \\ &\quad\cdot p(\mathbf{B}_{\text{wav}}[0,\tau];\Gamma_{\text{wav}})p(K)\prod_{k=1}^{K} p(\mathcal{C}_k[0,\tau];\Gamma_{\text{pcl}}). \end{aligned} \tag{21}$$

In the above representation, $W[0,\tau]$ includes all the hidden variables,

$$\begin{aligned} W[0,\tau] &= (\mathbf{M}[0,\tau],\mathbf{B}[0,\tau]) \\ &= (\mathbf{B}_{\text{wav}}, K, \{\mathcal{C}_i[t_i^b, t_i^e], i=1,2,\ldots,K\}), \end{aligned}$$

and $\Theta = (\Phi_\pi, \Gamma_{\text{wav}}, \Gamma_{\text{pcl}})$ includes the parameters of the deformable templates for motons and parameters of the dynamics of waves and particles.

3 Learning and Inference

In this section, we present the algorithm that infers the hidden variables $W[0,\tau]$ and learns the parameters Θ in (21). With learned parameters Θ, one can easily synthesize sequences following the two-level generative model. This algorithm produces all the results presented in the previous section (Figs. 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, and 18).

3.1 Problem Formulation and Stochastic Gradient

The problem is posed as statistical learning by maximum likelihood estimation (MLE). The objective is to compute the optimal parameters that maximize the log-likelihood for an observed sequence $\mathbf{I}^{\text{obs}}[0,\tau]$,

$$\begin{aligned} \Theta^* &= \arg\max \log p(\mathbf{I}^{\text{obs}}[0,\tau];\Theta) \\ &= \arg\max \log \int p(\mathbf{I}^{\text{obs}}[0,\tau], W[0,\tau];\Theta) dW[0,\tau]. \end{aligned} \tag{22}$$

By taking the derivative with respect to Θ and setting it to zero, we have

$$\begin{aligned} 0 &= \frac{1}{p(\mathbf{I}^{\text{obs}}[0,\tau];\Theta)} \frac{\partial \int p(\mathbf{I}^{\text{obs}}[0,\tau], W[0,\tau];\Theta) dW[0,\tau]}{\partial\Theta}, \\ &= \int \frac{\partial \log p(\mathbf{I}^{\text{obs}}[0,\tau], W[0,\tau];\Theta)}{\partial\Theta} \\ &\quad p(W[0,\tau]\,|\,\mathbf{I}^{\text{obs}}[0,\tau];\Theta) dW[0,\tau], \\ &= E_{p(W[0,\tau]\,|\,\mathbf{I}^{obs}[0,\tau];\Theta)}\left[\frac{\partial \log p(\mathbf{I}^{\text{obs}}[0,\tau], W[0,\tau];\Theta)}{\partial\Theta}\right]. \end{aligned} \tag{23}$$

This MLE problem is solved by iterating the following two steps.

First, given current parameter Θ, we simulate samples from the posterior by Markov chain Monte Carlo (see Section 3.3),

$$W_i^{\text{smp}}[0,\tau] \sim p(W[0,\tau]\,|\,\mathbf{I}^{\text{obs}}[0,\tau];\Theta), i = 1,2,\ldots,M. \tag{24}$$

The expectation in (23) is then approximated by the sample mean,

$$\frac{1}{M}\sum_{i=1}^{M} \frac{\partial \log p(\mathbf{I}^{\text{obs}}[0,\tau], W_i^{\text{smp}}[0,\tau];\Theta)}{\partial\Theta} = 0. \tag{25}$$

Without loss of generality, we set $M = 1$ for easy discussion.

Second, given the sampled hidden variables $W^{\text{smp}}[0,\tau] = (\mathbf{B}_{\text{wav}}, K, \mathcal{C}_1,\ldots,\mathcal{C}_K)$, we update the parameters $\Theta = (\Phi_\pi, \Gamma_{\text{pcl}}, \Gamma_{\text{wav}})$ by gradient ascent with a step size ε.

$$\begin{aligned} \Gamma_{\text{wav}} &\leftarrow (1-\varepsilon)\Gamma_{\text{wav}} + \varepsilon\frac{\partial \log p(\mathbf{B}_{\text{wav}}[0,\tau];\Gamma_{\text{wav}})}{\partial\Gamma_{\text{wav}}}, && \text{(learning motons)} \\ \Gamma_{\text{pcl}} &\leftarrow (1-\varepsilon)\Gamma_{\text{pcl}} + \varepsilon\sum_{k=1}^{K}\frac{\partial \log p(\mathcal{C}_k[0,\tau]|\mathbf{B}_{\text{wav}}[0,\tau];\Gamma_{\text{pcl}})}{\partial\Gamma_{\text{pcl}}}, && \text{(learning particle dynamics)} \\ \Phi_\pi &\leftarrow (1-\varepsilon)\Phi_\pi + \varepsilon\sum_{t=1}^{\tau}\frac{\partial \log p(\mathbf{B}_{\text{pcl}}(t)|\mathbf{M}_{\text{pcl}}(t);\Phi_\pi)}{\partial\Phi_\pi} && \text{(learning wave dynamics).} \end{aligned}$$

This algorithm is a stochastic version of the EM-algorithm. Unlike the EM-algorithm, which maximizes the likelihood at each step, our algorithm only updates Θ with a small step size. The two iterative steps are shown to converge to a globally optimal Θ^* even with $M = 1$ [9], provided that the step size in learning the parameters Θ is small enough so that the sample mean in (25) makes a good approximation to the expectation at the current Θ. Intuitively, with a small step size, samples collected over iterations are used to estimate the expectation.

In the following two sections, we present the algorithm for computing $W[0,\tau]$ from $\mathbf{I}^{\text{obs}}[0,\tau]$.

3.2 Initializing $W[0,\tau]$ by Bottom-Up Methods

Given $\mathbf{I}^{\text{obs}}[0,\tau]$, we initialize the hidden variables $W[0,\tau]$ using deterministic methods in a bottom-up phase. The errors and ambiguities in the initial solution will be resolved by a Markov chain Monte Carlo method (next section) which generates samples from the posterior $p(W[0,\tau]|\mathbf{I}^{\text{obs}}[0,\tau];\Theta)$.

Initial Step 1. Computing the base map $\mathbf{B}$ *by match pursuit.* We run the match pursuit algorithm on each frame independently. The match pursuit algorithm in Section 2.1 is a greedy method for selecting both particle and wave bases from dictionary Δ according to their coefficients. For the particle bases, we first set a high threshold (say $\epsilon = 3.0$) to obtain the "nucleus" bases. Then, we lower the threshold to $\epsilon = 1.0$ or 0.5 so that some new "electron" bases are added and assigned to one of the existing "nucleus" bases in a neighborhood. Thus, we have an initial base map which is partitioned in M_{pcl} subsets.

$$\mathbf{B} = (\mathbf{B}_{\text{wav}}, \mathbf{B}_{\text{pcl}}), \quad \mathbf{B}_{\text{pcl}} = S_1 \cup \cdots \cup S_{M_{\text{pcl}}}.$$

Initial Step 2. Computing the moton map $\mathbf{M}$ *and templets* Φ_π *by* K*-means clustering*. Each base map $\mathbf{B}_{\text{pcl}}$ has $M_{\text{pcl}} = O(10^2)$ subsets $S_1,\ldots,S_{M_{\text{pcl}}}$. We collect them over a number of frames and cluster these subsets into a smaller number of $N_\pi = 1 \sim 3$ clusters by K-means clustering. The mean of each cluster is then a deformable template for motons and we denote them by $\Phi_\pi = \{\Pi_1,\ldots,\Pi_{N_\pi}\}$. We have to predefine a threshold for the clustering or equivalently deciding the number N_π. This will force each subset $S_j, j = 1,\ldots,M_{\text{pcl}}$ to fit to one of the templates. S_j is registered to Π_i by a similarity transform and a simple graph matching in structure. The distance between a set S_j and a deformable model Π_i is defined as the difference (sum of squared errors) between the image patch generated by the bases in

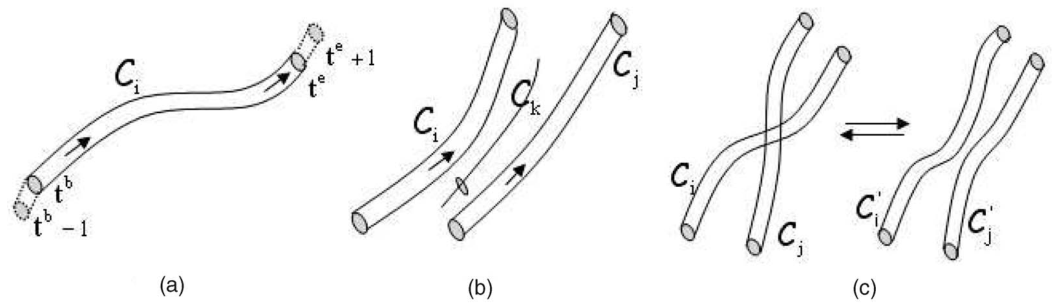


Fig. 19. Three pairs of typical reversible jumps. (a) Extend/shrink a trajectory. (b) Group/ungroup a trajectory. (c) Mutation and split/merge of trajectories.

S_j and the patch generated by bases in Π_i plus the structural divergence. The latter includes the differences of the total number of bases in S_j and in Π_i, and the type of bases in S_j and in Π_i. We refer to a texton paper for more details in learning the deformable templates [32].

Initial Step 3. Tracing the particle trajectories. At each frame $\mathbf{I}(t)$, we have a base map $\mathbf{B}_{\text{pcl}}$ by the match pursuit method. We connect the "nucleus" bases at frame $\mathbf{B}(t)$ to the existing trajectories at frames $\mathbf{B}_{\text{pcl}}(t-1)$ and $\mathbf{B}_{\text{pcl}}(t-2)$ as long as they fit to a smooth spline. Recall that the motion of particles follows second order AR models, their trajectories form smooth curves. Any "nucleus" base that cannot find a good fit starts a new trajectory. Once we have traced the heavy bases and obtain the "core" of the cables, we trace the "electron" bases as "coils." We tried an alternative method by projecting the bases at $\mathbf{B}_{\text{pcl}}(t)$ to $\mathbf{B}_{\text{pcl}}(t+1)$ and doing the match pursuit only on the residue image at frame $t+1$, but we got less satisfactory initial results in this way.

3.3 Sampling $W[0,\tau]$ from the Posterior by Markov Chain Monte Carlo (MCMC)

The deterministic methods in the initial stage are fast and often produce satisfactory results, but we need a nongreedy phase to fix the following problems.

First, the base maps $\mathbf{B}_{\text{pcl}}(t), t=1,2,\ldots,\tau$ are computed independently by a greedy match pursuit algorithm. Ideally, the base should be selected in its spatial and temporal context. For example, since Δ (or even Δ_{pcl}) is overcomplete, there are many combinations of bases that can reconstruct a moving element (say a bird or a snowflake) equally well. It is then desirable to select a common set of bases so that 1) the same type of moving elements in the image are reconstructed in the same way for better clustering and registration (i.e., spatial context) and 2) the same moving object is reconstructed by the same set of bases across the frames for better tracking (i.e., temporal context).

Second, when the motons are dense and move fast, such as the snowflakes, the tracking results are very rough, which produces an excessive number of short fragments of trajectories. Also, sometimes the edgy waves could generate some particle bases which have very short trajectories.

To resolve those problems above and to draw fair samples from the posterior (24), as is required for computing the sample mean in (25), we resort to the Markov chain Monte Carlo algorithm and design a number of reversible jumps (death/birth, extending/shrinking, group/ungroup) operating on the trajectories of the motons. As Fourier bases are consistent through the sequence, these reversible jumps are mainly designed to adjust the trajectories of the motons $\mathcal{C}_j[t^b_j, t^e_j], j=1,2,\ldots,K$, so that some trajectories are born, removed, grouped, extended, and mutated to achieve a high posterior probability. Along this process, the base maps and the moton maps are adjusted.

Our MCMC inference is different from the sequential Monte Carlo algorithms, e.g., particle filtering or condensation [12] for object tracking in the following two aspects: First, we have a full generative model for images. In contrast, object tracking algorithms often utilize discriminative models, such as intensity contrast, along object contours. As we have to track hundreds of moving elements in the sequence, a condensation method will need to keep a huge number of hypotheses for these elements and can easily lose the correspondences. A full generative model has the advantage in pruning the number of hypotheses because the image is represented by a fixed number of bases, each belonging to only one moton. This is often called the "explain-away" mechanism, in other words, the motons and the bases are exclusive in explaining the image while discriminative hypotheses are not (e.g., an edge may belong to many hypotheses in particle filtering). Second, we optimize the whole trajectories over the image sequence and they can be traced back in time during the computation. In contrast, object tracking methods always propagate hypotheses forward from t to $t+1$ without an explicit notion of trajectory.

The essence of the Markov chain design is to form an ergodic process in the space of all possible combinations of the "cables" and the Markov chain should observe some basic conditions such as detailed balance to ensure that it follows the posterior probability as it converges.

Each move in our Markov chain design is a reversible jump between two states A and B realized by a Metropolis-Hastings method [19]. We design a pair of proposal probabilities for moving from A to B $q(A \to dB) = q(B|A)dB$ and back with $q(B \to dA) = q(A|B)dA$. The proposed move is accepted with probability

$$\alpha(A \to B) = \min\left(1, \frac{q(A|B)dA \cdot p(B|\mathbf{I}^{\text{obs}}[0,\tau])dB}{q(B|A)dB \cdot p(A|\mathbf{I}^{\text{obs}}[0,\tau])dA}\right). \quad (26)$$

The move between A and B may involve a dimension change so that the number of variables in A is different from that in B. Thus, the proposal probabilities should match the dimension difference. For example, $dAdB$ is matched in both the denominator and nominator in (26).

Our Markov chain consists of four pairs of moves. Each type of move is randomly selected with probability $q_1+q_2+q_3+q_4=1$. Each pair involves designing a number of proposal probabilities. Thus, we need to maintain some queues which list a number of candidate trajectories to be grouped, ungrouped, extended, and shrunk, respectively, in a order according to some fitness measurement. Similar MCMC designs were reported in our previous work [29]. Due to space limit, we briefly specify the four pairs of reversible jumps in the rest of this section.

Move Type 1. Extending/shrinking a trajectory $\mathcal{C}_i$. This pair of moves are illustrated in Fig. 19a. The purpose of the moves are to extend a short fragment of trajectories or to kill the over extended trajectories. They implement a pair of reversible moves between two states A and B,

$$\begin{aligned} A &= (K, \mathcal{C}_i[t^b, t^e],\ W_-) \\ &\rightleftharpoons (K, \mathcal{C}_i[t^b - 1, t^e] \text{ or } \mathcal{C}_i[t^b, t^e + 1],\ W_-) = B, \end{aligned}$$

where W_- denotes all other variables which are unchanged during this move. The proposal probabilities are:

$$\begin{aligned} q(A \to B) &= q_1 \cdot q(i) \cdot q_{\text{tail}} \cdot q_{\text{ext}} q(\pi(t^e + 1) | \mathcal{C}_i[t^b, t^e]; \Gamma_{\text{pcl}}), \\ q(B \to A) &= q_1 \cdot q(i) q_{\text{tail}} \cdot q_{\text{shrk}}. \end{aligned}$$

q_1 is a probability for choosing type 1 jump, $q(i)$ is the probability for picking $\mathcal{C}_i$, and, with q_{tail}, we choose to operate at the tail. $q_{\text{ext}} + q_{\text{shrk}} = 1$ are probabilities for extending or shrinking the trajectory, respectively. Then, the new element $\pi(t^e + 1)$ is proposed based on the current cable $\mathcal{C}_i$ predicted by dynamics Γ_{pcl}. This prediction is expressed as probability $q(\pi(t^e + 1)|\mathcal{C}_i[t^b, t^e]; \Gamma_{\text{pcl}})$. Similarly, one can predict the extension at the head of the trajectory.

Move Type 2. Group/ungroup a trajectory. This pair of moves are illustrated in Fig. 19b. Let $\mathcal{C}_k$ be a short trajectory of a base, usually an "electron" base with a small coefficient. It is desirable to group it with a nearby trajectory $\mathcal{C}_i$ or $\mathcal{C}_j$. The ungroup proposal will remove those extra parts not belonging to a moton. In this jump, the length of $\mathcal{C}_k$ could be different from those of $\mathcal{C}_i$ and $\mathcal{C}_j$.

This jump implements a move between two states A and B,

$$A = (K, \mathcal{C}_i, \mathcal{C}_k, W_-) \rightleftharpoons (K - 1, \mathcal{C}'_i, W_-) = B.$$

Again, W_- denotes the remaining variables that are unchanged during the move. The proposal probabilities are

$$\begin{aligned} q(A \to B) &= q_2 q_{\text{grp}} q(k) q(\mathcal{C}_i | \mathcal{C}_k), \\ q(B \to A) &= q_2 q_{\text{ugrp}} q(i) q(\mathcal{C}_k, \mathcal{C}_i | \mathcal{C}'_i). \end{aligned}$$

We first choose move type 2 and then choose to group or ungroup an existing trajectory with probabilities q_{grp} or q_{ugrp}, respectively, where $q_{\text{grp}} + q_{\text{ugrp}} = 1$. Next, we choose a single-base trajectory $\mathcal{C}_k$ to be grouped with probability $q(k)$ or a cable $\mathcal{C}'_i$ to be ungrouped. The probabilities, q_{grp}, q_{ugrp}, $q(i)$, and $q(k)$, are computed based on the current queues for grouping and ungrouping. For example, $q(k)$ can be computed according to the distance and trajectory similarity between the $\mathcal{C}_k$ and $\mathcal{C}_i$. The closer and more similar, the larger the chance to be grouped together.

Move Type 3. Mutation, split/merge of trajectories. Thispair of moves are illustrated in Fig. 19c. They aim to correct those wrongly tracked trajectories. It mutates two trajectories $\mathcal{C}_i[t_i^b, t_i^e], \mathcal{C}_j[t_j^b, t_j^e]$ into two new trajectories $\mathcal{C}'_i$ and $\mathcal{C}'_j$ by exchanging some portions of the trajectories at a certain time t,

$$\mathcal{C}'_i = \mathcal{C}_i[t_i^b, t] \otimes \mathcal{C}_j[t + 1, t_j^e], \quad \mathcal{C}'_j = \mathcal{C}_j[t_j^b, t] \otimes \mathcal{C}_i[t + 1, t_i^e].$$

In a special case when $t_i^e = t = t_j^b - 1$, it becomes a split and merge move.

$$A = (K, \mathcal{C}_i, \mathcal{C}_j, W_-) \rightleftharpoons (K, \mathcal{C}'_i, \mathcal{C}'_j, W_-) = B.$$

The proposal probabilities are

$$\begin{aligned} q(A \to B) &= q_3 q(i, j) q(t | \mathcal{C}_i, \mathcal{C}_j), \\ q(B \to A) &= q_3 q(i, j) q(t | \mathcal{C}'_i, \mathcal{C}'_j). \end{aligned}$$

It first proposes move type 3 with q_3, then proposes a pair of trajectories in a queue by probability $q(i, j)$ to be mutated. Then, based on the two trajectories' dynamics (shapes), it proposes a site t for mutation. As a result of this move, both trajectories will fit their dynamics better after mutation.

Move Type 4: Death and birth of a single-base trajectory. This pair of moves eliminate some degenerated trajectories with length 1 or, conversely, create new bases and, thus, is a necessary step to adjust the base maps created by match pursuit. For example, in the snow or bird sequences, a particle may enter at certain time frame and, thus, new bases will be created at that time frame.

$$A = (K, W_-) \rightleftharpoons (K, \mathbf{b}_j, W_-) = B, \quad \mathbf{b}_j \in \Delta_{\text{pcl}}.$$

So, the proposal probabilities are very simple,

$$q(A \to B) = q_4 q(\mathbf{b}_j), \quad q(B \to A) = q_4 q(j).$$

It proposes to use type 4 with probability q_4 and then creates a base with $q(\mathbf{b}_j)$ for a birth move or select $\mathbf{b}_j$ with $q(j)$ for a death move according to the base map configuration.

3.4 Experiments

Once we have learned the parameters in Θ, we can synthesize new sequences from the joint probability following the two-level generative model in a straightforward manner.

$$(\mathbf{I}^{\text{syn}}[0, \tau], W^{\text{syn}}[0, \tau]) \sim p(\mathbf{I}[0, \tau], W[0, \tau]; \Theta), \quad \forall \tau > 0.$$

Figs. 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, and 18 show some results of the analysis and synthesis for a number of textured motion patterns. We edit the birth maps of the fireworks sequence and density of the bird sequence to show controllability of the model. Besides controllability, we replace the dictionary of Gabor and Fourier bases with symbolic sketches (contours for particles and ridges for waves), thus we easily render cartoon animations in Fig. 20. In our view, a cartoon sketch is a symbolic visualization of our inner representation of visual perception. The success of the cartoon animation in turn suggests that the generative model captures the essence of visual perception of textured motion.

However, despite the success of the generative model, there are two problems with the current representation.

1. The Fourier representation can synthesize some wave patterns, but some blurry effects are noticeable in Figs. 13 and 16.
2. The inference of $W[0, \tau]$ with MCMC is computationally intensive. The time complexity for learning a textured motion sequence containing particles is usually about $1 \sim 6$ minutes per frame on an Intel Pentium 4 1.5GHz computer, depending on the

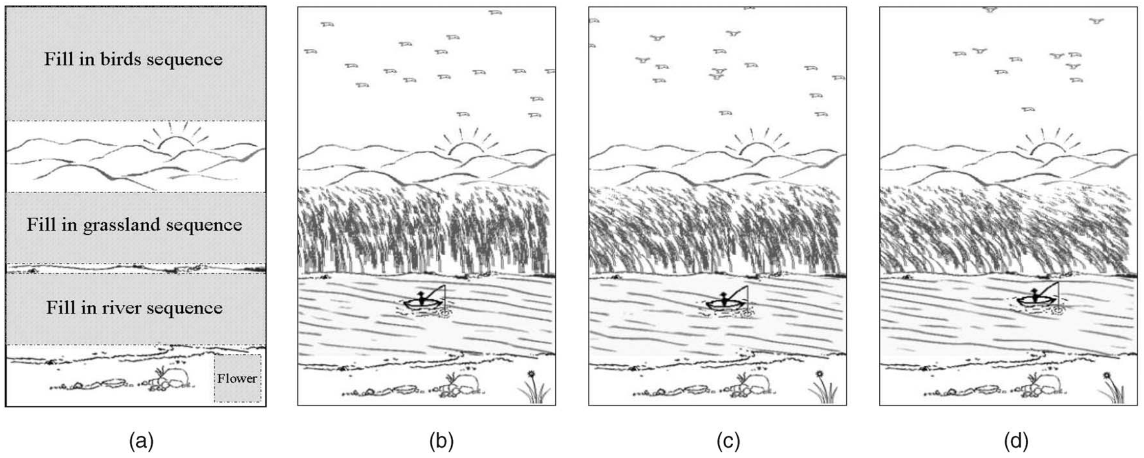


Fig. 20. Synthesized cartoon animation based on learned textured motions. (a) The static background image is drawn manually and the shaded area will be filled by the three learned textured motion sequences shown in the previous sections, flying birds, dancing grass, and the floating ball. The floating ball is replaced by a boat sketch. The dancing flower is one of the grasses from the grassland. (b), (c), and (d) Synthesized frames at $t = 1, 10, 20$.

complexity of the scene. This includes computing the trajectory and learning the parameters. The analysis and synthesis of wave patterns usually take about $2 \sim 3$ minutes for 50-100 frames.

4 Discussion and Future Work

In this paper, we present a generative method for modeling textured motion patterns. Our representation includes photometric, geometric, and dynamic models built on a generic and overcomplete base representation. This representation identifies the fundamental moving elements, their trajectories, sources, sinks, and couplings in motion. A Markov chain Monte Carlo method is adopted for learning and inference. When analyzing these textured motion patterns, we usually have to track hundreds of moving elements. Thus, the full generative model plays a very important role in terms of dimension reduction and hypothesis pruning.

However, the choice of generative models (e.g., the Gabor or Fourier bases) is still a matter of art. There is no rigorous model complexity criterion for generative models in deciding whether one set of bases (vocabulary) is better than the other. The criterion should not only account for properties such as MLE or MDL, but should also capture human perception so as to reflect the purpose of vision. For example, when we see wavy water, psychologically, it may be unclear what we perceive exactly and, thus, we don't have a quantitative measure for the correctness of the model. In generating the cartoon sequence, it is true that human observers are sensitive to some aspects, e.g., whether the birds flap wings properly, but are less sensitive to other details.

In the future, we would extend this current model in the following aspects. 1) Learning a large base dictionary including image primitives and textons [32] with heavy lighting changes. 2) In this paper, we only deal with simple topology changes, such as death/birth of bases. In practice, the bases (including image primitives) or moving elements may form spatial structures which can be represented with a Gestalt graph representation. Thus, we need to track the whole graph over time. The tracking procedure must incorporate topological changes using graph grammars. Only with such sophisticated representation can the complex nonstationary motion patterns, e.g., dancing fire and big surges, be properly modeled.

Acknowledgments

The authors would like to thank the support of US National Science Foundation IIS-02-44763 and IIS-02-22967. They also thank Adrian Barbu, Cheng-en Guo, and Yingnian Wu for discussions. Two test sequences used in the experiments (Figs. 13 and 15) are from the MIT temporal texture database. The cartoon background is drawn by Hanning Chen.

References

[1] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman, "Texture Mixing and Texture Movie Synthesis Using Statistical Learning," *IEEE Trans. Visualization and Computer Graphics,* vol. 7, 2001.
[2] C. Bregler, M. Covell, and M. Slaney, "Video Rewrite: Driving Visual Speech with Audio," *Proc. SIGGRAPH,* 2000.
[3] A. Cliff and J. Ord, "Space-Time Modeling with an Application to Regional Forecasting," *Trans. Inst. British Geographers,* vol. 66, pp. 119-128, 1975.
[4] D. Ebert and R. Parent, "Rendering and Animation of Gaseous Phenomena by Combining Fast Volume and Scaleline A-Buffer Techniques," *Proc. SIGGRAPH,* 1990.
[5] A. Efros and T. Leung, "Texture Synthesis by Non-Parametric Sampling," *Proc. Seventh Int'l Conf. Computer Vision,* vol. 2, pp. 1033-1038, Sept. 1999.
[6] D. Field, "What Is the Goal of Sensory Coding?" *Neural Computation,* vol. 6, pp. 559-601, 1994.
[7] A. Fitzgibbon, "Stochastic Ridigity: Image Registration for Nowhere-Static Senes," *Proc. Ninth Int'l Conf. Computer Vision,* pp 662-669, July 2001.
[8] D. Fleet and A. Jepson, "Stability of Phase Information," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 12, pp. 1253-1268, Dec. 1993.
[9] M.G. Gu and F.H. Kong, "A Stochastic Approximation Algorithm with Markov Chain Monte-Carlo Method for Incomplete Data Estimation Problems," *Proc. Nat'l Academy of Sciences,* vol. 95, pp. 7270-74, 1998.
[10] D. Heeger and J. Bergen, "Pyramid-Based Texture Analysis and Synthesis," *Proc. SIGGRAPH,* 1995.
[11] B. Horn and B. Schunck, "Determining Optical Flow," *Aritificial Intelligence,* vol. 17, pp. 185-203, 1981.
[12] M. Isard and A. Blake, "Contour Tracking by Stochastic Propagation of Conditional Density," *Proc. European Conf. Computer Vision,* 1996.

[13] B. Julesz, "Textons, the Elements of Texture Perception and Their Interactions," *Nature,* vol. 290, pp. 91-97, 1981.

[14] R. Kailath, *Linear Systems.* Englewood Cliffs, N.J.: Prentice Hall, 1980.

[15] R. Mann and M.S. Langer, "Optical Snow and the Aperture Problem," *Proc. Int'l Conf. Pattern Recognition,* vol. IV, pp. 264-267, Aug. 2002.

[16] B.F. Logan Jr., "Information in the Zero-Crossings of Band Pass Signals," *Bell Systems Technical J.* , vol. 56, pp. 487-510, 1977.

[17] S. Mallat and Z. Zhang, "Matching Pursuit in a Time-Frequency Dictionary," *IEEE Trans. Signal Processing,* vol. 41, pp. 397-415, 1993.

[18] D. Marr, *Vision.* I.W.H. Freeman, 1983.

[19] N. Metropolis, M. Rosenbluth, A. Rosenbluth, A. Teller, and E. Teller, "Equations of State Calculations by Fast Computing Machines," *J. Chemical Physics,* vol. 21, pp. 1087-1092, 1953.

[20] W.T. Reeves and R. Blau, "Approximate and Probabilistic Algorithms for Shading and Rendering Structured Particle Systems," *Proc. SIGGRAPH,* 1985.

[21] P. Saisan, G. Doretto, Y. Wu, and S. Soatto, "Dynamic Texture Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2001.

[22] A. Schodl, R. Szeliski, D. Salesin, and I. Essa, "Video Textures," *Proc. SIGGRAPH,* 2000.

[23] A. Schodl and I. Essa, "Controlled Animation of Video Sprites," *Proc. ACM Symp. Computer Animation,* 2002.

[24] Y. Li, T. Wang, and H.Y. Shum, "Motion Texture: A Two-Level Statistical Model for Character Motion Synthesis," *Proc. SIGGRAPH,* 2002.

[25] S. Soatto, G. Doretto, and Y. Wu, "Dynamic Texture," *Proc. Int'l Conf. Computer Vision,* 2001.

[26] P. Stoica and R. Moses, *Introduction to Spectral Analysis.* Prentice Hall, 1997.

[27] M. Szummer and R.W. Picard, "Temporal Texture Modeling," *Proc. IEEE Conf. Image Processing,* vol. 3, pp. 823-826, 1996.

[28] R.A.R. Tricker, *Bores, Breakers, Waves and Wakes.* New York: Am. Elsevier, 1965.

[29] Z. Tu and S. Zhu, "Image Segmentation by DDMCMC," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 5, pp. 657-673, May 2002.

[30] Y. Wang and S. Zhu, "A Generative Method for Textured Motion: Analysis and Synthesis," *Proc. European Conf. Computer Vision,* 2002.

[31] L. Wei and M. Levoy, "Fast Texture Synthesis Using Tree-Structured Vector Quantization," *Proc. SIGGRAPH,* 2000.

[32] S. Zhu, C. Guo, Y. Wang, and Z.J. Xu, "What Are Textons?" *Int'l J. Computer Vision,* (to appear). A short version appeared in ECCV, 2002.

[33] S. Zhu, Y. Wu, and D. Mumford, "Minimax Entropy Principle and Its Applications to Texture Modeling," *Neural Computation,* vol. 9, pp. 1627-1660, 1997.

Yizhou Wang received the BE degree from the Electrical Engineering Department of Tsinghua University, Beijing, China, in 1996, and the ME degree from the National University of Singapore in 2000. He worked as a computer hardware consultant for Hewlett-Packard, Singapore, from 1996 to 1998. Currently, he is working toward the PhD degree in computer science at the University of California, Los Angeles (UCLA). His research interests include motion analysis and modeling, vision for animation, pattern recognition, and statistical learning and modeling.

Song-Chun Zhu received the BS degree from the University of Science and Technology of China in 1991, and the MS and PhD degrees from Harvard University in 1994 and 1996, respectively. All degrees are in computer science. He is currently an associate professor jointly with the Departments of Statistics and Computer Science at the University of California, Los Angeles (UCLA). He is a codirector of the UCLA Center for Image and Vision Science. Before joining UCLA, he worked at Brown University (applied math), Stanford University (computer science), and The Ohio State University (computer science). His research is focused on computer vision and learning, statistical modeling, and stochastic computing. He has published more than 70 articles and received a number of honors, including the David Marr prize honorary nomination, the David Marr prize, the Sloan fellow in computer science, the US National Science Foundation Career Award, and a US Office of Naval Research Young Investigator Award.