Analysis and Synthesis of Textured Motion:
Particle, Wave and Cartoon Sketch

Yizhou Wang$^1$ and Song-Chun Zhu$^2$

Abstract

Natural scenes contain a wide range of textured motion patterns which are characterized by the movement of a large amount of particle and wave elements, such as falling snow, water waves, dancing grass, etc. In this paper, we present a generative method for modeling these motion phenomena and our method consists of four components: (1). A photometric model which represents an image as a linear superposition of image bases selected from a generic and over-complete 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 a raw image to a base (token) representation with $O(10^2)$-fold of dimension reduction. (2). A geometric model which groups adjacent bases and their motion trajectories into a number of basic moving elements – called “motons”. A moton is a deformable template in space-time representing a moving element, such as a snowflake. (3). A dynamic model which characterizes the motion of particles, waves, and their interactions, e.g., balls/leaves floating on water. The trajectories of these elements are coupled Markov chains. Given an input video sequence, a statistical learning algorithm computes a set of motons with their trajectories as hidden variables. It also learns the parameters that govern the geometric deformations and motion dynamics by maximum likelihood estimation (MLE). Consequently, novel sequences are synthesized easily from the learnt models. (4). A sketch model replaces the dictionary of Gabor/LoG and Fourier bases with symbolic sketches (token symbols). With the same generative model, the sketch model can render realistic and stylish cartoon animation. In this paper, cartoon sketch is viewed as a symbolic visualization and simplification of the inner representation in visual perception. The success of the cartoon animation, in turn, suggests that the generative model captures the essence of visual perception of textured motion.

Keywords

Textured motion, generative model, statistical learning, cartoon sketch, cartoon animation.

1,2 Department of Computer Science and 2. Department of Statistics, University of California, Los Angeles, Los Angeles, CA 90095. e-mail: wangyz@cs.ucla.edu, sczhu@stat.ucla.edu
I. 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, flock of birds, river waves, dancing grass, etc. It has been acknowledged [14] that such motion patterns fall beyond the scope of conventional optical flow field models [10] and a new framework has yet to be developed. In recent years, the study of such motion patterns has stimulated growing interests in both the vision and graphics communities, driven by a number of applications for synthesis and analysis.

Graphics methods. Computer graphics methods are concerned with rendering photorealistic video sequences or non-photorealistic and stylish cartoon animations. In the graphics literature, both physics-based and image-based methods are reported. The former uses partial differential equations, for example, creating animations of fire and gaseous phenomena with particles [19], [3]. The latter includes (1) video texture [21] which finds smooth transition points in a video sequence from which the video could be replayed with minimum discontinuity artifacts; (2) 3D volume texture [28] which generates motion through non-parametric sampling from an observed video motivated by recent work on texture synthesis [9], [31], [4]. 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. Consequently, the synthesized animations are less controllable.

Vision methods. In computer vision, the analysis of these motion patterns is important for video analysis, such as motion segmentation, annotation, recognition, retrieval and detecting abnormal motion in a crowd. 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. We briefly review these work to set the background for our method as follows.

Szummer and Picard [24] called the motion patterns temporal texture, and adopted a Spatial-Temporal Auto-Regression (STAR) model from Cliff and Ord [2]. In the STAR
model, a linear (or partial) order is imposed so that the intensity of each pixel only depends on its spatial and temporal neighbors for fast synthesis. Such model can be considered as an extension from a 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 [9], [31] to a tree structured multi-resolution representation, in a similar way to 3D volume texture method [28]. The dynamic texture work by Soatto et. al. [22] studied the motion dynamics explicitly using models and tools from control theory [13]. By Using principal component analysis (PCA), they represented an image $I(t)$ by a number of principal component images. The projections of $I(t)$ on these component images, denoted by $x(t)$, is modeled by a linear system model,

$$x(t + 1) = Ax(t) + Bv(t), \quad I(t) = Cx(t) + n(t),$$

where $v(t)$ is the noise driving the motion and $n(t)$ is the image noise for the reconstruction residues. The parameters $A, B, C$ are learnt by maximum likelihood estimation (MLE). This model can generate impressive synthesis for a variety of motion patterns especially when the moving objects can be represented well by PCA. And it is also shown to be useful for recognition [20]. Fitzgibbon [6] further studied the rigid camera motion in combination with the stochastic motion patterns, so that the motion is registered properly.

Despite reasonable success, the existing models need to be extended to address the following problems.

Firstly, the basic moving elements in the existing models are either pixels and points[19], [3], [24], [28], [1] or the entire image [21] and its principal components[22], [6]. Such representations usually do not identify the human perceived moving elements in the video, such as an individual bird or a snowflake.

Secondly, these models do not sufficiently characterize the dynamics of moving elements, such as trajectories, sources, sinks, and lifespan for the elements. It was considered very difficult to model the interactions among the elements, for instance, simulating balls or
leaves drifting in water waves. Consequently, these models have less localization in analysis and controllability in synthesis.

Furthermore, following a suggestion by Mumford in 1996, we call these patterns “textured motion” to emphasize the fact that the image sequences are fundamentally motion phenomena characterized by the dynamics, in contrast to referring them as texture phenomena, such as temporal texture [24], video texture [21], volume texture [28] and dynamic texture [22]. Textures correspond to status of systems with massive elements at thermodynamic equilibrium [31]. But motion patterns like fire, toilet flush, and gaseous turbulence are clearly not at equilibrium.

**Summary of Our method.** To solve the above problems, this paper presents a generative representation for textured motion which integrates the following four models.

1. **A photometric model.** An image is represented as a superposition of bases from an over-complete dictionary [5], including Fourier bases, LoG and Gabor bases at different scales, orientations. Such bases are known to be generic and effective for representing natural images including particle and wave patterns. This model transforms a raw image into a number of bases as a token representation and achieves $O(10^2)$-fold dimension reduction (see Table I).

2. **A geometric model.** We group 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 the “motons” in accordance with the notion of “textons” – the atomic perceptual elements in static images[12], [29]. 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 Figures 4, 3).

3. **A dynamic model.** We adopt a general motion equation which includes an auto-regression (AR) component for the trajectory of each moton, its source and sink maps, external driving forces and the interactions/coupling with other motons. The interactions among motons are always considered a challenge in both vision and graphics. In this paper, we assume
<table>
<thead>
<tr>
<th>Models</th>
<th>Parameters Stored in the Models</th>
<th>Compression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Sequence</td>
<td>$150 \times 200(I) \times 100(\text{frame number}) = 3 \times 10^6$</td>
<td>NA</td>
</tr>
<tr>
<td>Video Texture</td>
<td>$150 \times 200(I) \times 100(\text{frame number}) = 3 \times 10^6$</td>
<td>$1 : 1$</td>
</tr>
<tr>
<td>Dynamic Texture</td>
<td>$150 \times 200(I) + 150 \times 200 \times 20(\text{PCA}) + 20 \times$ $20(\text{dynamics}) + 20(\sigma) \approx 6.3 \times 10^5$</td>
<td>$1 : 5$</td>
</tr>
<tr>
<td>Textured Motion</td>
<td>$10^3 \times 3 + 10^4 \times 8 + 20 \times 8 \approx 10^4$</td>
<td>$1 : 300$</td>
</tr>
</tbody>
</table>

**TABLE I**

Comparison of the compression ratios of 4 typical models for a wavy river sequence.

The compression ratio is number of parameters in models divided by the number of pixels.

that "waves have more influence on particles", i.e., a ball (Gabor bases) floating on a river is driven by water waves (Fourier bases).

4. A sketch model. By replacing the dictionary of Gabor/LoG and Fourier bases with symbolic sketches, and together with the same motion model, we are able to render non-photorealistic and stylish cartoon animation. We believe that cartoon and sketch are simplified symbolic inner visualization of human perception. The success of the cartoon animation, in turn, suggests that our representation captures the essence of visual perception of textured motion.

We adopt an EM-like stochastic gradient algorithm [8] for the inference of hidden variables (bases, motons, and trajectories), and model parameters (deformable models, source and sink maps, parameters of the dynamics). This generative model offers more controllability in rendering both motion sequences and cartoon animation. Figures 9 and 11 show the synthesized results after we edit the number of motons, and sources of motons, respectively.

In comparison with other models, our representation is much more parsimonious. Table. I compares the compression rates based on a wavy river sequence. The training sequence is 100-frame long and each frame has $150 \times 200$-pixels. The video/volume texture method [21],
[28] stores the entire sequence, and synthesizes a new sequence by re-ordering the training frames or cut-and-paste. Dynamic texture model [22] remembers 1 mean image, 20 principal components of the frames, a dynamics matrix $A$ and 20 noise terms. The model achieves a compression rate of about 1 : 5. Our model represents an image with about 1000 Fourier bases without noticeable loss, and the dynamics are fitted by a 20th order AR model on the coefficients. See Section II-D. case 2 for a detailed account. The compression rate is about 1 : 300 due to the use of a generic dictionary. For the snowing or bird sequence, our model achieves even higher compression rates.

The paper is organized as follows. In Section II, we present a two-level generative representation with photometric, geometric, dynamic models, and some experiment results are shown along with the illustrations of the models. Section III presents the learning and inference algorithm by using Markov chain Monte Carlo methods to compute the three models from motion sequence. In Section IV, we show how the generative model can be easily applied to generate cartoon animations. A number of synthesized video clips and cartoon animation are better evaluated from the supplementary document.

II. TEXTURED MOTION REPRESENTATION

Let $I[0,\tau]$ denote an image sequence on a 2D lattice $\Lambda$ in a discrete time interval $[0,\tau] = \{0,1,...,\tau\}$. For $t \in [0,\tau]$, $I(t)$ is a frame and $I(u,v,t)$ denotes a pixel in the $t$th frame.

A. Photometric model – particles and waves

The photometric model assumes that an image $I$ is a superposition of $N$ image bases $\psi_j$, $j = 1,2,...,N$ selected from a dictionary $\Delta$ plus an iid Gaussian noise image $n$.

$$I(u,v) = \sum_{j=1}^{N} \alpha_j \psi_j(u,v;\beta_j) + n, \quad \psi_j \in \Delta, \quad n \sim N(0,\sigma_n^2).$$

(1)

$\alpha_j$ is the coefficient of base $\psi_j$ and $\beta_j$ denotes the transforms (translation, rotation and scaling) on the base functions $\psi(u,v)$ which we shall specify shortly. The dictionary $\Delta$ includes “particle bases”, such as LoG and Gabor functions, and wave bases, such as Fourier
Fig. 1. Coarse to fine image reconstruction with Gabor and LoG bases. Top row: three prototypes of bases: LoG, Gabor cosine and Gabor sine. Mid-row: symbolic sketch maps for the snow image with the Gabor bases (bars) and LoG bases (circles) and the images they reconstructed. Bottom row: combined images at three scales reconstructed by both the Gabor and LoG bases. Number of bases increases from left to right with $N_{pcl} = 800$ at scale 3.

functions. Its size is 100 times of the lattice $\Lambda$,

$$\Delta = \Delta_{pcl} \cup \Delta_{wav}, \text{ with } |\Delta| = O(100|\Lambda|)$$

In the following of this section, we briefly introduce the particle bases $\Delta_{pcl}$ and wave bases $\Delta_{wav}$, and discuss how the bases are selected for reconstructing the image.

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

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

Applying transformations denoted by variables $\beta$, we have a

$$\Delta_{pcl} = \{\text{Gcos}(u, v; \beta), \ Gsin(u, v; \beta), \ \text{LoG}(u, v; \beta) : \forall \beta\}. \quad (2)$$
For Gabor bases, $\beta = (x, y, \sigma, \theta)$ specify the centers, scales and orientations of the base function. For LoG bases, $\beta = (x, y, \sigma)$ specifies the centers and scales of the base function. If we represent each base by an attributed point (or token) $b_j = (\alpha_j, \beta_j)$, then the photometric model in Eqn. (1) transfers a raw image $I$ into a token representation, a layer of hidden variables – called the particle base map, which is denoted by

$$B_{pcl} = \{b_j = (\alpha_j, \beta_j), \ j = 1, 2, ..., N_{pcl}\}. \tag{3}$$

Figure 1 shows how a snow image being represented by particle bases. The three particle base functions LoG, Gcos, Gsin are shown on the top. In the middle of the figure, we show the base maps at three scales in a coarse-to-fine order with increasing $N_{pcl}$. At each scale, we divide the base map $B_{pcl}$ into 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 intensity means a positive coefficient). Each base map reconstructs a “sub-band” image. In the bottom row, the subband images of the two types of base maps are put together as the final reconstruction. Scale 3 has $N_{pcl} = 800$ bases, and we can see that the reconstructed image is a very good approximation to the input image. In general, $B_{pcl}$ is an effective representation with large dimension reduction. Furthermore, this representation employs a coarse-to-fine strategy which is efficient for computation and tracking tasks introduced in later section.

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

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

Let $\alpha_j$ be the Fourier coefficient, then the selected Fourier bases form a wave base map

$$B_{\text{wav}} = \{b_j = (\alpha_j, \beta_j), \ j = 1, 2, ..., N_{\text{wav}}\}. \tag{5}$$

Both Fourier bases and Gabor bases are generic vocabularies, which are well known to
Input image | Reconstruction by $\Delta_{\text{wav}}$ | Reconstruction by $\Delta_{\text{pcl}}$ | Coefficient plots
---|---|---|---
![Birds flying](image1) | ![300 Fourier Bases](image2) | ![216 Gabor Bases](image3) | ![Coef plots](image4)

(a) Birds flying | 300 Fourier Bases | 216 Gabor Bases

![Water waves](image5) | ![300 Fourier Bases](image6) | ![320 Gabor Bases](image7) | ![Coef plots](image8)

(b) Water waves | 300 Fourier Bases | 320 Gabor Bases

![Ball on water](image9) | ![400 Fourier Bases](image10) | ![291 Gabor Bases](image11) | ![80 Fouriers + 21 Gabors](image12) | ![Coef plots](image13)

(c) Ball on water | 400 Fourier Bases | 291 Gabor Bases | 80 Fouriers + 21 Gabors

**Fig. 2.** Comparison of image reconstructions by Fourier bases $\Delta_{\text{wav}}$ and Gabor/LoG bases $\Delta_{\text{pcl}}$ respectively. The curves plot the base coefficients obtained by projecting the images onto the image bases. The thick curve is for $\Delta_{\text{pcl}}$. The slopes of the curves reflect the coding efficiencies of the dictionary. (a) A typical particle image - flying birds. (b) A typical wave image - wavy river. (c) A typical image with mixed objects of particles and waves - floating ball.

be effective for representing different patterns in natural images [5]. This observation is verified in Figure 2, which compares the particle bases $\Delta_{\text{pcl}}$ and the wave bases $\Delta_{\text{wav}}$ in reconstructing different textured motion patterns. Here we select three typical images for illustration. From each image, we obtain two reconstructions: one by wave (Fourier) bases from $\Delta_{\text{wav}}$ and the other by particle (Gabor and LoG) bases from $\Delta_{\text{pcl}}$. For the third image, we select bases from both dictionaries.

**Bottom-up computation and comparison of bases.** As the Fourier bases from $\Delta_{\text{wav}}$ are orthonormal, we can simply choose the first $N_{\text{wav}}$ Fourier bases with the largest coefficients. For the particle bases, we adopt a match pursuit procedure[16], which is de-
scribed in the following. Given an input image $I^{\text{obs}}$, it starts with a constant base map image $I$ whose intensity is equal to the mean of $I^{\text{obs}}$. Then the algorithm iterates the following step. At each step, a base $\psi_j$ is selected from $\Delta_{\text{pcl}}$. It has the highest response defined as the inner product between the base and the residue image.

$$\psi_j = \arg \max_{\psi \in \Delta_{\text{pcl}}} \langle \psi, I^{\text{obs}} - I \rangle, \quad \alpha_j = \langle \psi_j, I^{\text{obs}} - I \rangle.$$ 

Once $\psi_j$ is selected, it is added to the base map $B_{\text{pcl}}$ of $I$, and the response of a remaining base $\psi_k$ in $\Delta_{\text{pcl}}$ will be adjusted ($\alpha_k \leftarrow \alpha_k - \alpha_j \langle \psi_j, \psi_k \rangle$), if $\psi_k$ is not orthogonal to $\psi_j$. This iteration stops if the largest coefficient $|\alpha_N| \leq \epsilon$.

For example, Figure 2.a is a flock of birds. The reconstruction with $N_{\text{wav}} = 300$ Fourier bases is very blurry. Contrastively, the reconstruction with $N_{\text{pcl}} = 216$ particle bases captures the birds accurately. Figure 2.b is a water wave image where the Fourier bases are found to be better than the particle bases. Figure 2.c shows a ball floating on river. We can see that neither type of bases alone is able to effectively represent this image. However, using a combination of 80 Fourier bases and 21 Gabor bases exhibits a better reconstruction.

For the bird and water images, we plot the coefficients $(\alpha_j, j = 1, 2, \ldots, N_{\text{pcl}}$ or $N_{\text{wav}}$) of the bases in the order of being selected from $\Delta_{\text{wav}}$ and $\Delta_{\text{pcl}}$ 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.

In general, the two sets of bases are combined to yield a base map

$$B = B_{\text{pcl}} \cup B_{\text{wav}} = \{b_j = (\alpha_j, \beta_j), \ j = 1, 2, \ldots, N\}, \quad N = N_{\text{pcl}} + N_{\text{wav}}. \quad (6)$$

These bases compete to explain the image. Usually, as $\Delta$ is over-complete with $|\Delta| = O(100|\Lambda|)$, $B$ is a parsimonious token representation with $N = O(|\Lambda|/100)$. 

Fig. 3. Motons as the fundamental moving elements. (a) 3 base functions: LoG, Gcos, Gsin, and their symbolic sketches: circle, bar, edge. (b) Examples of learnt motons – a snowflake and a bird. (c) A graphic view of a moton trajectory – the cable model.

The photometric model in Eqn. (1) is rewritten as a conditional probability for image \( I \),

\[
p(I|B_{pcl}, B_{wav}; \sigma_o) = \frac{1}{(2\pi \sigma_o^2)^{|\Lambda|/2}} \exp\left\{ -\sum_{(u,v) \in \Lambda} (I(u,v) - \sum_{j=1}^N \alpha_j \psi_j(u, v; \beta_j))^2/(2\sigma_o^2) \right\}. \quad (7)
\]

B. Geometric model: identifying motons – the basic moving elements

The match pursuit procedure is only a bottom-up step in computing the base map \( B \) from a static image. As we proceed, \( B \) will be adjusted for spatial and temporal coherence, and tracked through an image sequence by an algorithm described in Section III. In this subsection, we discuss the geometric model for spatial coherence, and we shall present the dynamic model for temporal coherence afterwards.

The bases in \( B_{pcl} \) often form spatially coherent groups and each group is a moving element called “moton”. We partition \( B_{pcl} \) into some disjoint subsets

\[
B_{pcl} = S_1 \cup S_2 \cup \cdots \cup S_{M_{pcl}}, \quad \text{with} \quad M_{pcl} \ll N_{pcl}.
\]

Figure 3.b gives two moton examples. A snowflake image is a sum of three bases: 2 LoG bases and 1 Gcos base with certain space displacements. A bird image consists of 7 bases: 3 LoG bases, 2 Gcos bases and 2 Gsin bases. These subsets \( S_i, i = 1, 2, ..., M_{pcl} \) are further clustered into \( k \) typical configurations, represented by a set of deformable templates \( \Pi_\ell \)

\[
\Phi_\pi = \{\Pi_\ell : \ell = 1, 2, ..., k\}.
\]
Fig. 4. The computed motion elements: snowflakes and random examples. (a) A moton template in atomic structure. (b) 120 instance of snowflakes as motons π.

Each subset $S_i$ is an instance of one of the templates $\Pi_\ell$. For example, the snow sequence has one ($k = 1$) moton template shown in Figure 4.a. Figure 5 shows three ($k = 3$) templates for different poses of the bird sequence.

As Figure 4.a shows, each template $\Pi_\ell$ usually has a "heavy" base with relatively large coefficient $\alpha_i$ which is surrounded by several "light" bases with relatively small coefficients $\alpha_j$. By analogy to physical model of atoms, we call the heavy bases "nucleus bases" as they have heavy weights like protons and neutrons, and the light bases "electron bases". The atomic models are illustrated for the birds in Figure 6.b.

The dictionary of motons are formed from the templates in $\Phi_\pi$ through transforms $T$ denoted by $\beta = (x, y, \theta, \sigma)$ and deformations $D$ specified by variable $\zeta$. $\zeta$ includes binary variables for the presence or absence of a base component. Each moton instance is denoted by $\pi(\ell, \beta, \zeta)$. Thus we have a moton dictionary,

$$\Delta_\pi = \{ \pi(\ell, \beta, \zeta) = D_\zeta \circ T_{x,y,\theta,\sigma} \circ \Pi_\ell : , \forall \ell, \beta, \zeta \}. \quad (8)$$

Strictly speak, $\Delta_\pi$ is the "orbit" formed from $\Phi_\pi$ through some group operations. Figure 4.b shows 120 moton instances of the snowflakes and the deformable model captures the variations of snowflakes.

With dictionary $\Delta_\pi$, the base map $B_{pcl}$ is generated by a moton map $M_{pcl}$ with each subset $S_i$ generated by a moton $\pi_i$. Thus we arrive at a more abstract and parsimonious
Fig. 5. Motons in a bird flying sequence. Left: input image. Right: three moton templates \( \Phi_\pi = \{ \Pi_i, i = 1, 2, 3 \} \) learnt in a clustering step for different poses. Two instances are shown for each template.

Fig. 6. (a) Input image. (b) 3D graphic illustration for the “atomic” model of bird motons \( \pi_j, j = 1, 2, ..., 9 \). (c) diagram of three-state transitions for birds flying.

representation.

\[
M_{pcl} = \{ \pi_j(\ell_j, \beta_j, \zeta_j), j = 1, 2, ..., M_{pcl}, 1 \leq \ell_j \leq k \},
\]

According to theory of ocean waves\[25\], the bases in \( \Delta_{wav} \) also travel in groups. For example, water flows travels as sinusoid waves caused by different sources of vibration, such as wind, boat, earthquake, etc. But such motions can only be seen from a large view scope. In our experiments, waves travel in a single group so that we do not need to group them. To conform the notation, we denote a wave base by \( \pi_j \).

\[
M_{wav} = \{ \pi_j = b_j; j = 1, 2, ..., N_{wav} \} = B_{wav}, \quad M = M_{pcl} \cup M_{wav}.
\]

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

\[
p(B|M; \Phi_\pi) = p(B_{pcl}|M_{pcl}; \Phi_\pi)p(B_{wav}|M_{wav})
= \left[ \prod_{i=1}^{M_{pcl}} \prod_{j \in S_i} p(b_j|\pi_i; \Pi_{\ell_i}) \right] \cdot \delta(B_{wav} - M_{wav}). \tag{9}
\]

Thus, we have a two-level generative model. The moton map \( M \) generates base map \( B \) with dictionary \( \Delta_\pi \), and the base map \( B \), in turn, generates image \( I \) with dictionary \( \Delta \).

\[
M = M_{pcl} \cup M_{wav} \xrightarrow{(\Phi_\pi, \Delta_\pi)} B = B_{wav} \cup B_{pcl} \xrightarrow{(\Phi_{pcl}, \Delta_{pcl}) \cup (\Phi_{wav}, \Delta_{wav})} I. \tag{10}
\]
In Section III, we shall discuss the algorithm that infers \( B \) and \( M \) from \( I \) as hidden variables and learns the moton templates \( \Phi_\pi \) as parameters.

C. The moton trajectories and representation of the sequence

The generative model in Eqn.(10) is for static images. For a sequence \( I[0, \tau] \), the motons and bases should be tracked frame by frame. As Figure 3.c shows, each moving element is represented by a trajectory in a time interval \([t^b, t^e]\). Let \( \pi(t) \) be the state of an element at time \( t \), then the trajectory of a moton is denoted as

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

(11)

For example, a snowflake enters our view at frame \( t^b \) and leaves our view at frame \( t^e \). Intuitively, a moton trajectory is like a cable. The trajectory of its nucleus base is 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.

We change the index from image frame \( t \) to moving element \( i \), and replace the two level hidden representation \( B[0, \tau] \) and \( M[0, \tau] \) by a number of \( K \) trajectories \( C_i, i = 1, 2, ..., K \), and denote them by

\[
W[0, \tau] = (M[0, \tau], B[0, \tau]) = (B_{\text{wav}}, K, \{C_i[t^b_i, t^e_i], i = 1, 2, ..., K\}).
\]  

(12)

The \( K \) trajectories represent the \( K \) moving objects over time. The number of motons and bases may change from frame to frame due to the birth and death events over time. This representation is not only low-dimensional and generic, but also captures the essence of visual perception of textured motion. In Section IV, we use this generative model to synthesize cartoon animation by replacing the bases \( B \) and motons \( \pi \) with a symbolic representation.
D. Dynamic model – sources, sinks and wave-particle interactions

In this subsection, we present the dynamic model for $C_i[t^b, t^e], i = 1, 2, ..., K$ – the moving particles, and $B_{\text{wav}}$ – the traveling waves. We are particularly interested in some interactions among the motons. The first type of coupling is the influence of waves on particles, e.g. balls drifting in a river, grass waving in the wind. This kind of effect cannot be simulated by the models in the literature [21], [22]. The second type of coupling is the interactions among wave components. Unlike particles such as birds and snow flakes, which move rather independently, the waves travel together with complex interactions. Therefore, the relative motion of different Fourier bases must be constrained to keep certain phase alignments. Other interactions, such as particle-particle collision and particle-wave collision (splash) are not considered in this paper.

The state of a moton at time $t$ is denoted by $\pi(t)$. For particles, $\pi = (\ell, \beta, \zeta)$ includes its type $\ell$, transforms $\beta = (x, y, \sigma, \theta)$, and deformation $\zeta$. For waves, $\pi = (\xi, \eta, \phi)$ is degenerated to a Fourier base with its frequencies and phase. The general motion equation for $\pi(t)$ is a $p$-th order AR model with coefficients $a = (a_1, ..., a_p)$, driven by three types of forces: (1) influence from the (other) waves $U(B_{\text{wav}}(t))$; (2) external force field $f(\pi(t))$, such as gravity, wind field, and external constraints, which may variation over space and time; (3) A Brownian motion $n$. So we have

$$\pi(t) = \sum_{j=1}^{p} a_j \pi(t-j) + U(B_{\text{wav}}(t)) + f(\pi(t)) + n, \quad n \sim \mathcal{N}(0, \sigma^2). \quad (13)$$

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

**Case 1:** Dynamic model for free moving particles – snow, birds and fireworks.

In this case, the particles 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, we obtain a simplified 2nd order Markov chain model,

$$\pi(t) = a_1 \pi(t-1) + a_2 \cdot \pi(t-2) + c + n, \quad n \sim \mathcal{N}(0, \sigma^2) \quad t \in [t^b + 1, t^e]$$
Fig. 7. Experiment on the falling snow sequence. (a) Observed sequence. (b) Synthesized sequence by sampling the generative model.

Fig. 8. Experiment on the falling snow sequence. Left: Observed sequence. Middle: Graphic view of the computed trajectories of the snowflakes (as hidden variables). Upper right: a probability map of the sources for snowflakes to enter the scene. Lower right: probability map of the sinks for the snowflakes to leave the view. Dark means high probability.

\[(\pi(t^b), t^b) \sim P_B(\pi, t), \quad (\pi(t^e), t^e - t^b) \sim P_D(\pi, t)\].

The birth of a moton \(\pi\) and its timing \(t^b\) follow a probability \(P_B(\pi, t)\). \(P_B\) specifies the “sources” of the motons, and its marginal on the location \(P_B(x, y)\) (summed over time and other attributes) is called 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 \(P_D(x, y)\) reveals the “sinks”, and is called death map. (\(\pi\) is a long vector, \(P_B\) and \(P_D\) are high dimensional.) In practice, we are most interested in the location \((x, y)\).

The probabilities \(P_B\) and \(P_D\) are represented in non-parametric form using Parzen win-
Fig. 9. Experiment on a flying-bird sequence. (a) Observed sequence. (b) Synthesized sequence with fewer flying birds by editing the number of motons $M$ when sampling the model.

Fig. 10. Experiment on the bird sequence. Left: Observed sequence. Middle: Graphic view of the computed trajectories of the birds. Upper right: a probability map of the sources for birds to enter the scene. Lower right: probability map of the sinks for birds to leave the view. Dark means high probability.

dows. During the learning process, suppose we have computed $K$ cables $C_i[t_i^b, t_i^e], i = 1, 2, ..., K$ from a sequence $I[0, \tau]$, we represent $p_B$ and $p_D$ as

$$p_B(\pi, t) = \frac{1}{K} \sum_{i=1}^{K} \delta(\pi - \pi_i(t_i^b), t - t_i^b), \quad p_D(\pi, t) = \frac{1}{K} \sum_{i=1}^{K} \delta(\pi - \pi_i(t_i^e), t - (t_i^e - t_i^b))$$ (14)

where $\delta()$ is a Parzen window centered at 0. When we project $p_B$ and $p_D$ to the $(x, y)$ dimensions, we get the death and birth maps.

Figure 8 (right column) displays the birth map $p_B(x, y)$ and death map $P_D(x, y)$ for a snow sequence, in which the darker the spot, the higher the probability. Thus the algorithm "understands" that the snowflakes mostly enter from the upper-right corner and disappear around the lower-left corner.
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.

Fig. 12. Experiment on the firework sequence. Left: observed sequence. Middle: graphic view of the
trajectories of the firework. Upper right: source map of the firework. Lower right: sink map of the firework.

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

\[
p(C[t^b, t^e]; \Gamma_{pcl}) = p_B(\pi(t^b)) p_D(\pi(t^e), t^e - t^b) \prod_{t=t^b+1}^{t^e} p(\pi(t)|\pi(t-1), \pi(t-2)).
\] (15)

In the above model, \(\Gamma_{pcl}\) denotes all the parameters in the dynamic models. For clarity of
notation, we omit the special treatment for the 2nd frame.

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

Remark 1: For the firework sequence in Figure 11, the death and birth of motons must
be synchronized, as a large number of particles come and go together. The death/birth
maps can also be manipulated, so that we are able to edit the number of objects and the
events happening at any time and places as we expect. For example, although we only
observe a single firework in the original sequence, but we can still generate several fireworks
at different places and time intervals as shown in Figure 11. By reducing the number of
birds, we observe fewer birds in the synthesized sequence in Figure 9.
Remark 2: For the bird sequence, the motion $\pi(t)$ comes from three possible templates $\Phi_\pi = \{\Pi_1, \Pi_2, \Pi_3\}$ and may change states over time. In order to have the birds flap wings, the Markov chain model $p(\pi(t)|\pi(t-1), \pi(t-2))$ includes a 1st 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 $3 \times 3$ matrix. Note that this is not necessary in the snow and firework sequences.

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

For pure wave sequences, e.g. Figures 13, 14, 15, each image is represented by a number of Fourier bases. The variables are

$$M = B = B_{\text{wav}} = \{(\alpha_j, \xi_j, \eta_j, \phi_j), \ j = 1, 2, ..., N\}, \ N = O(10^3).$$

$N$ is fixed and there is no birth or death events. Furthermore, if the camera does not move
Fig. 15. Experiment on a plastic foil sequence. (a) Observed sequence. (b) Synthesized sequence with 1500 Fourier bases.

Fig. 16. Experiment on a grassland sequence. (a) Observed sequence. (b) Synthesized sequence with 3000 Fourier bases for its spatial wave pattern.

and the motion is stationary, then the Fourier frequencies $\xi_j, \eta_j$ and amplitudes $\alpha_j$ are time-invariant. So only the phases $\phi_j, j = 1, ..., N$ change over the time and this is known as the phase motion [7]. The speed of phase motion $d\phi$ is related to the speed in the space $(dx, dy)$ by

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

(16)

A slight complication is that we have to wrap the phase into $[0, 2\pi)$ in computing $d\phi_j/dt$ and thus $(dx, dy)$ [7].

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

$$
\phi_j(t) = \sum_{i=1}^{P} a_{ji} \phi_j(t - i) + n_j, \quad n_j \sim \mathcal{N}(0, \sigma^2), \quad j = 1, 2, ..., N.
$$
With $p = 15 \sim 20$ accounting for low frequency components, this simple model can synthesize the river sequences reasonably well resulting from the observed sequence. However, the phases become mis-aligned after 30-50 frames. To solve this problem, we study a joint vector $\phi(t) = (\phi_1(t), ..., \phi_N(t))$, and reduce dimension by a standard PCA method over the training frames as follows. Let $e_i, i = 1, 2, ..., m$ be the eigenvectors with largest eigenvalues, then $\gamma_j(t) = \langle \phi(t), e_j \rangle, j = 1, ..., m$ are the projected coefficients. In our experiments $m = 8$, and the $m$ coefficients follow independently a $p$-th order AR model

$$\gamma_j(t) = \sum_{i=1}^{p} a_{ji} \gamma_j(t - i) + n, \quad n \sim \mathcal{N}(0, \sigma_j^2), \quad p = 20, \ j = 1, 2, ..., m = 8.$$  \hspace{1cm} (17)$$

The total number of variables used in the model is $3N$ for $(\xi_j, \eta_j, \alpha_j), j = 1, ..., N$, and $8N$ for the eigenvectors, plus $20 \times 8$ for dynamics AR coefficients. The compression rate is compared in Table I.

After transferring the wave sequence $B_{\text{wav}}[0, \tau]$ into a representation on the sequence of coefficients $\gamma[0, \tau]$, we write the probability as

$$p(B_{\text{wav}}[0, \tau]; \Gamma_{\text{wav}}) = \prod_{j=1}^{m} \prod_{t=0}^{\tau} p(\gamma_j(t)|\gamma_j(t-1), ..., \gamma_j(t-p)).$$  \hspace{1cm} (18)$$

We assume some initial conditions for the first $p$ frames and $\Gamma_{\text{wav}}$ denotes all the parameters in the dynamic model of waves. Some synthesis results for the water waves are shown in Figure 13, 14.

The same model is applied to the plastic foil in Figure 15 and the grass sequence in Figure 16 and we see that it successfully characterizes the spatial movement of the plastic foil and grass. In general the wavy plastic foil and grass are driven by invisible wind field which has wave properties. For the grass sequence, we need more Fourier bases $N = 3000$ to reconstruct the high frequency components.

**Case 3:** Dynamic model for particles-waves interactions: ball or foams on water.

Some motion sequences contain both particles and waves, e.g. Figures 17 and 18 show a ball and foams drifting on water, respectively. The coupling of the two types of elements is characterized by a driving force from wave to particles. As the particles are small, we
are only concerned about the position \((x, y)\) and fix other attributes in \(\pi\). To conform the notation, we write \(\pi\) for \((x, y)\).

Let \(\phi(t) = (\phi_1(t), ..., \phi_{N_{\text{wav}}}(t))\) 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 equation (16). The motion of a particle is then influenced by the sum of the velocity at point \((x, y)\). In practice, we only need to choose \(q = 20 - 30\) Fourier bases \((\tilde{\xi}_k, \tilde{\eta}_k, \tilde{\phi}_k) \in B_{\text{wav}}, k = 1, 2, ..., q\) with lower frequencies to drive the particles. Thus the dynamics of the moton \(\pi\) is

\[
\pi_j(t) = \sum_{i=1}^{p=2} a_j \pi(t - i) + \sum_{k=1}^{q} b_k (\tilde{\xi}_k, \tilde{\eta}_k)' d\tilde{\phi}_k(t) + c + n, \quad n \sim \mathcal{N}(0, \sigma_n^2), \quad \forall j.
\]

The second term in the above equation accounts for the coupling of the particle motion with waves. \(a_j, b_k\) are the coefficients that can be independent of the individual particles. The death and birth of particles follow the same model in case 1. This model is still a Markov Chain model and the trajectory of a moton follows the following probability,

\[
p(C[t^b, t^e]; \Gamma_{\text{pcl}}) = p_B(\pi(t^b)) p_D(\pi(t^e), t^e - t^b) \prod_{t=t^b+1}^{t^e} p(\pi(t)|\pi(t-1), \pi_i(t-2), \tilde{\phi}_1(t), ..., \tilde{\phi}_q(t)).
\]

The wave bases follow the dynamics in equation (17).

The synthesized floating ball and floating foams are shown in Figure 17 18. The coupling of the particles with waves appears realistic in the video sequence (see supplementary file).
Fig. 18. Experiment on a sequence with many foam particles drifting in a river. (a) Observed sequence. (b) Synthesized sequence. (c) Learnt motons: foams and their trajectories. (d) Sources and sinks of the floating foams.

Now we conclude this section by integrating the photometric model Eqn. (7), geometric model Eqn. (9), and dynamic models in Eqns. (15),(18), and (20) into a joint probability for an image sequence $I^{\text{obs}}[0, \tau]$ and the hidden representation $W[0, \tau]$,

$$p(I^{\text{obs}}[0, \tau], W[0, \tau]; \Theta) = \prod_{t=1}^{\tau} p(I^{\text{obs}}(t)|B_{\text{pcl}}(t), B_{\text{wav}}(t)) \cdot p(B_{\text{pcl}}(t)|M_{\text{pcl}}(t); \Phi_{\pi}) \cdot p(B_{\text{wav}}[0, \tau]; \Gamma_{\text{wav}}) \cdot p(K) \prod_{k=1}^{K} p(C_k[0, \tau]; \Gamma_{\text{pcl}}).$$

In the above representation, $W[0, \tau]$ is the hidden variable

$$W[0, \tau] = (M[0, \tau], B[0, \tau]) = (B_{\text{wav}}, K, \{C_i[t_i^b, t_i^e], i = 1, 2, ..., K\}).$$

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

In this section, we study the algorithm that infers the hidden variables $W[0, \tau]$ and learns the parameters $\Theta$ in the models. With the learnt parameters $\Theta$, one can easily synthesize sequences following the two level generative model. This algorithm produces all the results presented in the previous Section (Figures 7-18).

A. 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 $\Gamma_{\text{obs}}[0, \tau]$,

$$\Theta^* = \arg \max p(\Gamma_{\text{obs}}[0, \tau]; \Theta) = \arg \max \int p(\Gamma_{\text{obs}}[0, \tau], W[0, \tau]; \Theta) dW[0, \tau] \quad (21)$$

By taking the derivative with respect to $\Theta$, and setting it to zero, we have

$$\frac{1}{p(\Gamma_{\text{obs}}[0, \tau]; \Theta)} \int \frac{\partial}{\partial \Theta} \log p(\Gamma_{\text{obs}}[0, \tau], W[0, \tau]; \Theta) p(\Gamma_{\text{obs}}[0, \tau], W[0, \tau]; \Theta) dW[0, \tau] = 0,$$

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

The MLE is solved by iterating the following two steps.

Firstly, under the current parameter $\Theta$, we simulate samples from the posterior

$$W_i[0, \tau] \sim p(W[0, \tau] | \Gamma_{\text{obs}}[0, \tau]; \Theta), i = 1, 2, ..., M. \quad (22)$$

and then estimate the above expectation by importance sampling.

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

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

Secondly, by plugging in equation (21), we have the following equations for learning the parameters $\Theta = (\Phi_\pi, \Gamma_{\text{wav}}, \Gamma_{\text{pcl}})$,

$$\sum_{t=1}^{\tau} \frac{\partial}{\partial \Phi_\pi} \log p(B_{\text{pcl}}(t) | M_{\text{pcl}}(t); \Phi_\pi) = 0, \quad \text{(learning motons)} \quad (23)$$
\[
\frac{\partial \log p(B_{\text{wav}}[0, \tau]; \Gamma_{\text{wav}})}{\partial \Gamma_{\text{wav}}} = 0 \quad \text{(learning wave dynamics)}
\]  

(24)

\[
\sum_{k=1}^{K} \frac{\partial \log p(C_{k}[0, \tau]|B_{\text{wav}}[0, \tau]; \Gamma_{\text{pcl}})}{\partial \Gamma_{\text{pcl}}} = 0, \quad \text{(learning particle dynamics)}.
\]  

(25)

We update \( \Theta = (\Phi_{\pi}, \Gamma_{\text{wav}}, \Gamma_{\text{pcl}}) \) by gradient ascent with a small stepsize.

This algorithm is a stochastic version of EM-algorithm. The two iterative steps are said to converge to a globally optimal \( \Theta^* \) even with \( M = 1 \) [8], provided that the stepsize in learning parameters \( \Theta \) is small enough so that the importance sampling makes a good approximation at the current \( \Theta \). Intuitively, with small stepsize, samples obtained over time are used to estimate the expectation.

In the following three subsections we present some details of the algorithm.

B. Initialization by bottom-up methods

Given \( \mathbf{I}_{\text{obs}}[0, \tau] \), we initialize \( W[0, \tau] \) by a sequence of ”bottom-up” steps in a coarse-to-fine manner. Then we refine \( W[0, \tau] \) by carefully designed MCMC steps.

Firstly, we adopt a match pursuit method [16] which selects a number of particle and wave bases whose coefficients are large, say \( |\alpha_j| \geq \epsilon = 3.0 \). The particle bases with such high coefficients are treated as the “nuclei” of the motons. Then we lower the threshold, say \( \epsilon = 1.0, 0.5 \), 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 with partitions

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

Secondly, we classify \( S_1, \ldots, S_{M_{\text{pcl}}} \) into a smaller number of \( k \) clusters. The mean of each cluster is then a deformable template for motons, and we denote them by \( \Phi_{\pi} = \{\Pi_1, \ldots, \Pi_k\} \). Usually we have to pre-define the number \( k \), e.g. \( 1 \leq k \leq 3 \) for a sequence. This will force each set \( S_j, j = 1, \ldots, M_{\text{pcl}} \) to fit to one of the templates. This clustering process is easily implemented by a k-mean method. The distance between a set \( S_j \) and a deformable model \( \Pi_i \) is defined as the difference between images generated by the bases in \( S_j \) and in \( \Pi_i \) plus the structural divergence. \( S_j \) is registered to \( \Pi_i \) by a similarity transform and a simple
graph matching in structure. We refer to a previous paper for more details [29].

Thirdly, we track the nuclei bases in the video and compute trajectories $C_i, i = 1, 2, ..., K$ by a simplified CONDENSATION algorithm [11]. When the motons are dense and move fast, such as the snowflakes, the tracking result is pretty rough, and consists of an excessive number of $K$ short fragments of trajectories. Such fragments must be further computed using the MCMC steps (death/birth, extending/shrinking, group/ungroup) to achieve good results. Then the light bases are added to these trajectories to form $K$ “cables”.

C. Sampling $W[0, \tau]$ from the posterior by Markov chain Monte Carlo (MCMC)

As the Fourier bases are consistent through the sequence, the MCMC steps are mainly designed to adjust the trajectories of the motons $C_j[t^b_j, t^e_j], j = 1, 2, ..., K$, so that some trajectories are grouped, extended, and mutated to achieve a high posterior probability

$$\left( K, \{C_k[t^b, t^e] : k = 1, 2, ..., K \} \right) \sim p(K) \prod_{k=1}^{K} p(C_k[0, \tau] | I^\text{obs}[0, \tau]; \Gamma_{pcl}).$$

Our MCMC inference is different from the sequential Monte Carlo algorithm, e.g. CONDENSATION [11] for object tracking in the following two aspects. Firstly, we have a full generative model for images. In contrast, object tracking algorithms often have partial model of the image, thus its likelihood can only be evaluated relatively. The advantage of a full generative model is the explain-away mechanism, so that we do not have to preserve a large number of hypotheses for each moton. Secondly, we optimize the whole trajectories over the image sequence and trace back in time during the computation. In contrast, object tracking methods like CONDENSATION always propagates hypotheses forward from $t$ to $t + 1$. But our algorithm could use information in later frames to resolve ambiguities in earlier frames.

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. We design a pair of proposal probabilities for moving from $A$ to $B$ $q(A \rightarrow dB) = q(B|A)dB$ and back with $q(B \rightarrow dA) = q(A|B)dA$.

The proposed move is accepted with probability

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

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 Eqn. 26.

Our Markov chain consists of the 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. Due to space limit, we briefly specify the four moves in as follows.

**Move Type 1: Extending/shrinking a trajectory $C_i$.** This move is illustrated in Fig. 19.a and it is a jump process between two states $A$ and $B$,

$$
A = (K, C_i[t^b, t^e], W_-) \Leftrightarrow (K, C_i[t^b - 1, t^e] \text{ or } C_i[t^b, t^e + 1], W_-) = B,
$$

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

\[ q(A \rightarrow B) = q_1 q(i) q_{\text{tail}} q_{\text{ext}} q(\pi(t^e + 1)|C_i[t^b, t^e]; \Gamma_{\text{pcl}}), \]

\[ q(B \rightarrow A) = q_1 q(i) q_{\text{tail}} q_{\text{shrk}}. \]

\( q_1 \) is a probability for choosing type 1 move, \( q(i) \) is the probability for picking \( C_i \), and with \( q_{\text{tail}} \) 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 \( C_i \) predicted by dynamics \( \Gamma_{\text{pcl}} \). This prediction is expressed as probability \( q(\pi(t^e + 1)|C_i[t^b, t^e]; \Gamma_{\text{pcl}}) \). Similarly one can predict the extension at the head of the trajectory.

**Move Type 2: Group/ungroup a trajectory**

This move is illustrated in Fig. 19.b. Let \( C_k \) be a short trajectory of a base, usually an “electronic” base with small coefficient. It is desirable to group it with a nearby trajectory \( C_i \) or \( C_j \). The length of \( C_k \) could be different from those of \( C_i \) and \( C_j \).

This move is a jump between two states \( A \) and \( B \),

\[ A = (K, C_i, C_k, W_-) \rightleftharpoons (K - 1, C'_i, W_-) = B. \]

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

\[ q(A \rightarrow B) = q_2 q_{\text{grp}} q(k) q(C_i|C_k), \quad q(B \rightarrow A) = q_2 q_{\text{ugrp}} q(i) q(C_k, C_i|C'_i). \]

We firstly choose move type 2, and then choose to group or ungroup an existing trajectory with probabilities \( q_{\text{grp}} \) or \( q_{\text{ugrp}} \) respectively. Next, we choose a single-base trajectory \( C_k \) to be grouped with probability \( q(k) \) or a multi-trajectory \( C'_i \) to be ungrouped, and so on. The probabilities, like \( q_{\text{grp}}, q_{\text{ugrp}}, q(i), \) and \( q(k) \) are computed based on the current queues for grouping and ungrouping.

**Move Type 3: Mutation, split/merge of trajectories**

This move is illustrated in Fig. 19.c. It mutates two trajectories \( C_i[t^b_i, t^e_i], C_j[t^b_j, t^e_j] \) into two new trajectories \( C'_i \) and \( C'_j \), by exchang-
ing some portions of the trajectories at a certain time $t$,

$$C'_i = C_i[t^b_i, t] \otimes C_j[t + 1, t^e_j], \quad C'_j = C_j[t^b_j, t] \otimes C_i[t + 1, t^e_i]$$

In a special case when $t^e_i = t = t^b_j - 1$, it becomes a split and merge move.

$$A = (K, C_i, C_j, W) \xleftarrow{} (K, C'_i, C'_j, W) = B.$$  

The proposal probabilities are

$$q(A \rightarrow B) = q_3 q(i, j) q(t | C_i, C_j), \quad q(B \rightarrow A) = q_3 q(i, j) q(t | C'_i, C'_j).$$

It first proposes move type 3 with $q_3$, then proposes a pair of trajectories in a queue by probability $q(i, j)$. Then based on the two trajectories, it proposes a site $t$ for mutation.

**Move Type 4: Death and birth of a single-base trajectory**

This move eliminates some degenerated trajectories with length 1, or reversely creates new bases. For example, in snow or bird sequences, a particle may enter at certain time frame, and thus new bases will be created at that time frame.

$$A = (K, W) \xleftarrow{} (K, b_j, W) = B, \quad b_j \in \Delta_{pcl}.$$  

So the proposal probabilities are very simple,

$$q(A \rightarrow B) = q_4 q(b_j), \quad q(B \rightarrow A) = q_4 q(j).$$

It proposes to use type 4 with probability $q_4$, and then creates a base with $q(b_j)$ for birth move, and select $b_j$ with $q(j)$ for the death move.

**D. Learning the parameters $\Theta$**

Given the sampled hidden variables $W[0, \tau] = (B_{wav}, K, C_1, ..., C_K)$, we update the parameters $\Theta$ in the second step for both the deformable moton template $\Phi$ and the dynamics $\Gamma_{wav}, \Gamma_{pcl}$, following the equations (23), (24), and (25).

$$\Gamma_{wav} \leftarrow (1 - \rho)\Gamma_{wav} + \rho \frac{\partial \log p(B_{wav}[0, \tau]; \Gamma_{wav})}{\partial \Gamma_{wav}},$$
Unlike the EM-algorithm, which maximizes the likelihood at each step, our algorithm only update $\Theta$ with a small stepsize for global convergence\[8\]. The birth/death maps, $p_B$ and $p_D$, of particles are learnt by counting the heads and tails of the cables at their locations and time (see equation (14)).

E. Experiments

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

$$(I_{\text{syn}}[0, \tau], W_{\text{syn}}[0, \tau]) \sim p(I[0, \tau], W[0, \tau]; \Theta), \forall \tau > 0.$$ Figure 7-18 show some results of the analysis and synthesis (with edit) for a number of textured motion patterns. We have discussed them in Section II and these results can also be seen from the supplementary video clips. Here we would remark on two problems.

(i). The Fourier representation can synthesize some wave patterns, but some blurry effects are noticeable in Figure 13 and 16.

(ii). 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/frame on an Intel Pentium 4 1.5GHz computer, depending on the complexity of the scene. The analysis and synthesis of wave patterns usually take about $2 \sim 3$ minutes for $50 \sim 100$ frames.

IV. Sketch model

In this section, we present the fourth component – a sketch model which renders a cartoon animation $J[0, \tau]$ from either an observed or a synthesized sequence $I[0, \tau]$.

In our view, a cartoon $J[0, \tau]$ is a symbolic visualization of our inner representation $W[0, \tau]$
Fig. 20. From video to cartoon sketch. The cartoon representation $W' \subset W$ is a simplified version of $W$ with some details selectively removed. By replacing the particle and wave bases with symbolic sketches, we can easily synthesize cartoon animation with the same or edited generative model.

with some “unnecessary” details selectively removed. It is rendered in two simple steps as Figure 20 illustrates.

Firstly, we extract a subset of hidden variables $W'[0, \tau]$ from $W[0, \tau]$ to simplify the description. $W'[0, \tau]$ is supposed to capture the essential semantics. For example, we may keep the geometric and dynamic properties of a moton but ignore its photometric attributes.

Secondly, we replace the photometric dictionaries $\Delta_{p\ell1}$, $\Phi_{\pi}$ and $\Delta_{w\ell}$ by symbolic sketches $\Delta'_{p\ell1}$ and $\Delta'_{w\ell}$ respectively. Then the cartoon $J[0, \tau]$ is rendered with the generative model in the same way as we synthesize the photorealistic sequence. The selection of the sketch dictionaries reflects the style of a cartoon.

In the following we briefly explain how we choose the symbolic representation for particles and waves.

(1). Rendering particles. Each particle object $\pi$, such as a bird or a snowflake, is rendered by a contour outlining the deformable template. Its motion follows the same dynamic model. Obviously one can choose other symbolic representations.

(2). Rendering waves. As the Fourier bases is indistinguishable, we sketch all Fourier bases as a whole. Let $I_{\text{wav}}$ be an image reconstructed from $B_{\text{wav}}$,

$$I_{\text{wav}}(u, v) = \sum_{j=1}^{N_{\text{wav}}} \alpha_j \psi_j(u, v; \beta_j), \quad \psi_j \in \Delta_{\text{wav}}.$$  

When we view $I_{\text{wav}}(u, v)$, we not only perceive the global periodic wave pattern, but also
Fig. 21. Image diffusion from extracted sketches. (a) Observed image. (b) Extracted curves for ridges and valleys. (c) Reconstructed image from the information on the curves.

notice the individual peaks and valleys. This dual representation was noticed in Marr’s primal sketch[17]. Marr cited a theorem[15] that for a bandpass signal the positions of its zero-crossings alone is sufficient for reconstructing the original signal up to a multiplicative factor.

Figure 21.a shows an example for the river image. Figure 21.b is a collection of points for the ridge and valleys, denoted by

$$SK = \{(u, v) : |\nabla^2 I_{\text{wav}}(u, v)| \leq \epsilon\}$$

For each point $(u, v)$ we remember its pixel intensity $I_{\text{wav}}(u, v)$ and its slope $\nabla I_{\text{wav}}(u, v)$. Then we can reconstruct the rest of the image by heat diffusion using the curves as boundary condition. That is, we run the following equations,

$$\frac{dI}{dt} = \frac{\partial^2 I}{\partial u^2} + \frac{\partial^2 I}{\partial v^2}, \quad \text{for } (u, v) \notin SK,$$

$$I(u, v) = I_{\text{wav}}(u, v), \quad \nabla I(u, v) = \nabla I_{\text{wav}}(u, v), \quad \forall (u, v) \in SK.$$  

Figure 21.c shows the diffused result for a river image. We can see that the original image is well recovered from the information at the sketch points $SK$.

For clarity, we choose to show a subset of the curves in $SK$ which have relatively high contrast, i.e. their accumulated intensity gradients along the curve is larger than a certain threshold. Other weak and short curves are removed for simplicity of the cartoon.

(3). Rendering particles driven by waves. The particles and waves are sketched by the methods above. For particle objects floating on water, their dynamics follows case 3.
Fig. 22. Synthesized cartoon animation based on learnt textured motions. (a) The static background image drawn manually. Shaded area will be fill by three learnt textured motion sequences showed in the previous sections, flying birds, dancing grass, and floating ball. The floating ball is replaced by a boat sketch. And the dancing flower is one of the grasses from the grassland. (b-d) Synthesized frames at $t = 1, 10, 20$.

Figure 22 shows a combined cartoon animation. We choose three natural sequences: flying birds, floating ball on a river and wavy grassland, and learn the geometric and dynamic models for the objects in each of the three sequences by using the algorithms described in the previous sections. Then we render synthesized sequences and generate their cartoons using the sketch model. The floating ball is replaced by a boat. And the dancing flower is one of the grasses from the grassland. A static background – mountain, sun, and river bank is drawn manually in Figure 22.(a). We fill the three cartoons into the blank areas of the background image to render the animation.

There is a slight detail in the animation of grass. The tip of a grass is treated as a particle, whose motion is driven by the learnt Fourier waves from the grass sequence (case 2, Fig.16). The bottom point of the grass is fixed, and the curve between the two points is interpolated by a spline. The movement of the tips are similar to the motion of floating particles in water. Let $(x, y)$ be a tip of the grass, its motion follows

$$
(x(t), y(t)) = \sum_{i=1}^{p=2} a_i(x(t - i), y(t - i)) + \sum_{k=1}^{q} b_k(\tilde{\xi}_k, \tilde{\eta}_k)'d\hat{\phi}_k(t) + \kappa(x - x_0, y - y_0) + n. \tag{27}
$$

which is almost the same as case 3 in Eqn. (19), except that we add an extra term of the
force. Each tip is assumed to have a resting position \((x_0, y_0)\), and a spring is attached between \((x, y)\) and \((x_0, y_0)\). This generates the nodding effects for the grass.

V. Summary and future work

In this paper, we present a generative method for modeling textured motion patterns. Our representation includes photometric, geometric, dynamic and sketch models, built on a generic and over-complete base representation. This representation identifies the fundamental moving elements, their trajectories, sources, sinks, and coupling in motion. A Markov chain Monte Carlo method is adopted for learning and inference.

In the future, we would extend this current model in the following aspects. (1) Modeling the interaction among particles, e.g. collision. (2) Studying the influence of particles on waves, e.g. splash effect of a stone dropped into water. (3) Eliminating the blurry effect of water waves. (4) Developing effective representation for transient elements, such as fire flame, etc.

ACKNOWLEDGMENTS

We’d like to thank for the support of NSF grants IIS-02-44763 and IIS-02-22967. We thank Adrian Barbu, Cheng-en Guo, and Yingnian Wu for extensive discussions and Barbu and Guo assisted the experiments. Two test sequences used in the experiments (river in Fig.13 and plastic in Fig.15) are from the MIT database collected by Dr. Picard’s group. And the Cartoon background drawing is from Hanning Chen.

REFERENCES
