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 by integrating statistical models and algorithms from both texture and motion analysis. This generative model consists 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 bases for point/particle-elements and Fourier bases for wave-elements. These bases compete to explain the input images. The transform from a raw image to a base (token) representation leads to $O(10^2)$-fold dimension reduction. (2). A geometric model which groups the bases and their motion trejectories into a number of basic moving elements – called "motons". A moton is a a deformable template in space-time representing a moving element, such as a snow flake. (3). A unified dynamic model which characterize the motion of particles, waves, and their interactions, e.g. balls/leaves floating on water. 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 estimate (MLE). Consequently, novel sequences are synthesized easily from the learning models. (4). A sketch model which adopts the generative model above but replaces the dictionary of Gabor and Fourier bases with symbolic sketches (token symbols). With the same generative model, it can render realistic and stylish cartoon animation. In our view, cartoon sketch is a symbolic visualization of the inner representation for visual perception. The success of the cartoon animation, in turn, suggests that the generative model capture 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 particles 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 model[10] and new framework has yet to be developed. In recent years the study of such motion patterns have stimulated growing interests in both the vision and the 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 use 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]. Though some realistic animations can be rendered at fast speed, the video texture or 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 are important for video analysis, such as motion segmentation, annotation, recognition, retrieval, 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. In the following, we briefly review these work to set the background for our method.

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 2D texture synthesis work [9], [31] to a tree structured multi-resolution representation, in a similar spirit 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 a SVD analysis, they represent an image $\mathbf{I}(t)$ by a number of principal component images. The projections of $\mathbf{I}(t)$ on these component images, denoted by $x(t)$, is modeled by a linear system model,

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

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 learned 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 across the time sequence. The model was 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 above models need to be extended to address the following problems.

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

Secondly, these models do not fully characterize the dynamics of the moving elements. For example, the trajectories, sources, sinks, and lifespan for the elements, They also do not model the interactions between the elements, for example, simulating balls or leaves driven

by water waves. Thus they have less locality in analysis and controllability in synthesis.

Furthermore, following a suggestion by Mumford in 1996, we call these patterns as "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], dynamic texture [22]. Textures correspond to status of systems with massive elements at thermodynamic equilibrium [31]. But motion patterns like fire, toilet flash, and gaseous turbulence are clearly not at equilibrium.

Summary of Our method. To address these problems, this paper presents a generative representation for textured motion which includes 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, Gabor sin / cos 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 in a token representation and thus achieves a $O(10^2)$ folds dimension reduction (see Table I).

2. A geometric model. We group the bases and their motion trajectories into a number of basic moving elements. We call the basic moving elements the "motons" in accordance with the notion of "textons" – the atomic perceptual elements in static images[12], [**?**]. A moton is a a deformable template in space-time representing a moving element. For instance, each snowflake or bird is represented by a few Gabor bases moving 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 base, its source and sink maps, external driving forces and the interactions with other bases. The interactions among motons are always considered a challenge in both vision and graphics. In this paper, we assume that "waves have more influence on particles". For example, e.g. a ball (Gabor bases) floating on a river is driven by water waves (Fourier bases).

Models	Parameters Stored in the Models	Compression
Training Sequence	$150 \times 200(\text{I}) \times 100(\text{frame number}) = 3 \times 10^6$	NA
Video Texture	$150 \times 200(\text{I}) \times 100(\text{frame number}) = 3 \times 10^6$	1 : 1
Dynamic Texture	$150 \times 200(\text{I}) + 150 \times 200 \times 20(\text{PCA}) + 20 \times 20(\text{dynamics}) + 20(\sigma) \approx 6.3 \times 10^5$	1 : 5
Textured Motion	$10^3 \times 3 + 10^3 \times 8 + 20 \times 8 \approx 10^4$	1 : 300

TABLE I

COMPARISON OF THE COMPRESSION RATIOS OF 4 TYPICAL MODELS FOR A WAVY RIVER SEQUENCE. THE COMPRESSION RATIO IS NUMBER OF PIXELS DIVIDED BY THE NUMBER OF PARAMETERS IN MODELS.

4. A sketch model. For cartoon animation, we replace the dictionary of Gabor and Fourier bases with symbolic sketches. Together with the same motion model, we can render non-photorealistic and stylish cartoon animation. In our view, cartoon and sketch are simplified symbolic visualization of our inner representation and 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 inference of the hidden variables (bases, motons, and trajectories), and their parameters (source and sink maps, parameters of the dynamics). This generative model offers more controllability in rendering both motion sequences and cartoon animation. For example, in Figures 7 and 9, we can edit the number of motons, and sources of of motons etc.

In comparison with other models, our representation is much more parsimonious. Table. I compares the compression rates for a wavy river sequence. The training sequence is 100-frame long and each frame has 150×200-pixels. The video/volume texture method [21], [28] store 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 represent an image 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 sequences, 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 illustrate the models with experiments. Then in Section III, we present the learning and inference algorithm using Markov chain Monte Carlo methods for computing the three models from video. Then in Section IV we show how the generative model can be easily transfer into cartoon animation. A number of synthesized movies and cartoon animation are better evaluated from the supplementary video clips.

II. Textured motion representation

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

A. Photometric model– particles and waves

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

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

α_j is the coefficient of base ψ_j and β_j is the transforms (translation, rotation and scaling) on the base functions $\psi(u,v)$ which we shall specify shortly. The dictionary Δ is 100-fold over-complete [5] and it includes a dictionary of "particle bases" Δ_{pcl}, such as LoG and Gabor functions, and a dictionary of wave bases Δ_{wav}, such as Fourier functions. So

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

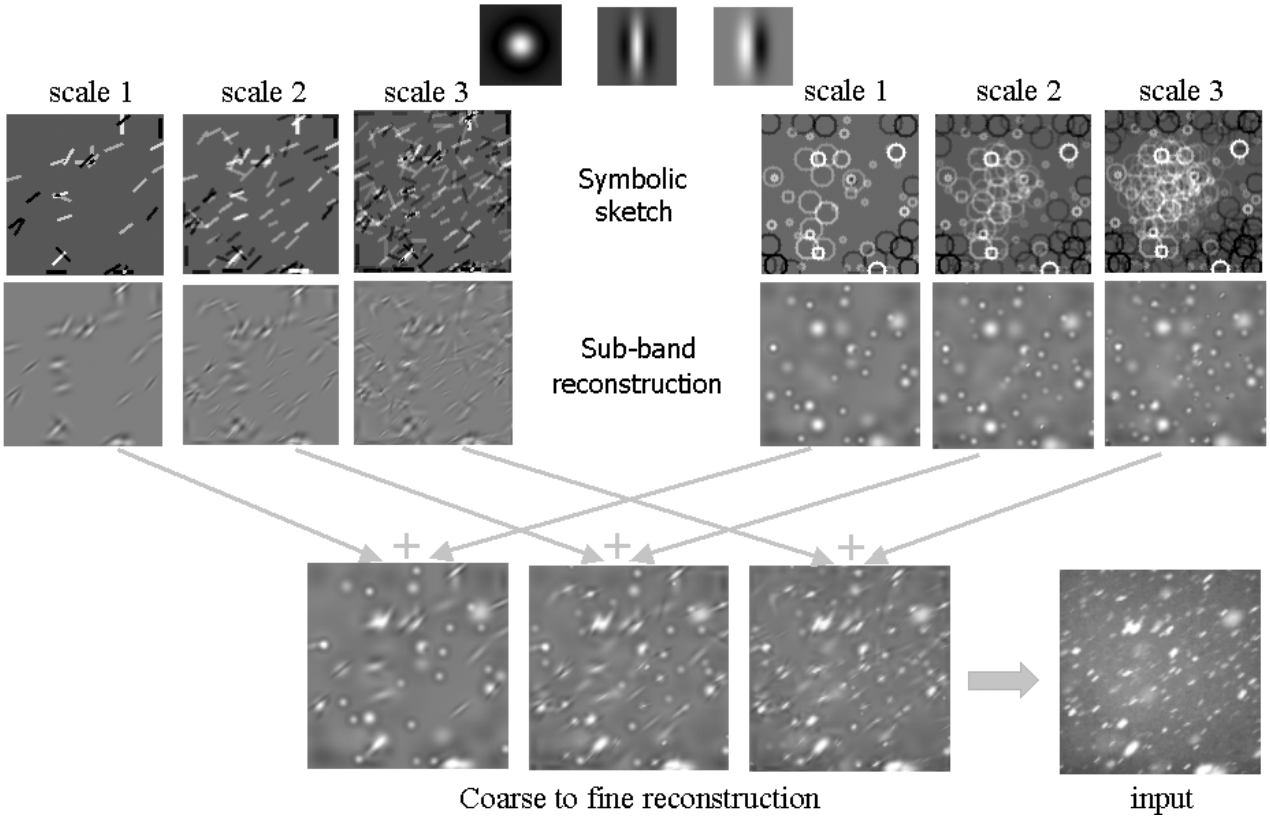


Fig. 1. Coarse to fine image reconstruction with Gabor and LoG bases. Top row: three prototypes of bases: LoG, Gabor cosine and Gabor sine. Mid-left: symbolic sketch maps for Gabor bases of the snow image. Mid-right: symbolic sketches for the Gabor bases (bars) and LoG bases (circles) and the images 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 $N_{\text{pcl}} = 800$ at scale 3.

In the following, we briefly introduce Δ_{pcl} and Δ_{wav}, and discuss how the bases are selected for reconstructing the image.

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

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

through transformation denoted by variables β. Thus,

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

For Gabor bases, $\beta = (x, y, \sigma, \theta)$ specify the centers, scales and orientations of the base function. For LoG bases, the variable $\beta = (x, y, \sigma)$ specify the centers and scales of the base

function. If we represent each base by an attributed point (or token) $\mathbf{b}_j = (\alpha_j, \beta_j)$, then the photometric model in eqn (1) transfers a raw image $\mathbf{I}$ into a token representation as a layer of hidden variables – called the *particle base map*. We denote it by

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

Figure 1 shows an example of representing a snow image 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 $\mathbf{B}_{\text{pcl}}$ into a Gabor map (left) and an LoG map (right). A Gabor base is sketched symbolically by a bar with the same size and orientation as the Gabor function and an LoG base is sketched by a circle with the size representing its scale. The brightness of the bars and circles represent the coefficients with white meaning positive coefficient. Each base map reconstructs a "sub-band" image. On the bottom row, the subband images by the two base maps are summed as the final reconstruction. Scale 3 has $N_{\text{pcl}} = 800$ bases and the reconstructed image is a very good approximation to the input image. $\mathbf{B}_{\text{pcl}}$ is an effective representation with large dimension reduction. The representation also introduces a *coarse-to-fine* strategy which is efficient for computation and tracking in later section.

Wave bases Δ_{wav} The wave dictionary is constructed from a single Fourier function $\text{FB}(u, v)$ by transforms $\beta = (\xi, \eta, \phi)$ for its spatial frequency (ξ, η) and phase ϕ

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

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

$$\mathbf{B}_{\text{wav}} = \{\mathbf{b}_j = (\alpha_j, \beta_j),\ j = 1, 2, ..., N_{\text{wav}}\}. \tag{5}$$

Both Fourier bases and Gabor bases are generic dictionaries, and it is well known that they are effective for various patterns in natural images [5]. This observation is well reflected in Figure 2 which compares the particle bases Δ_{pcl} and the wave bases Δ_{wav} for reconstructing

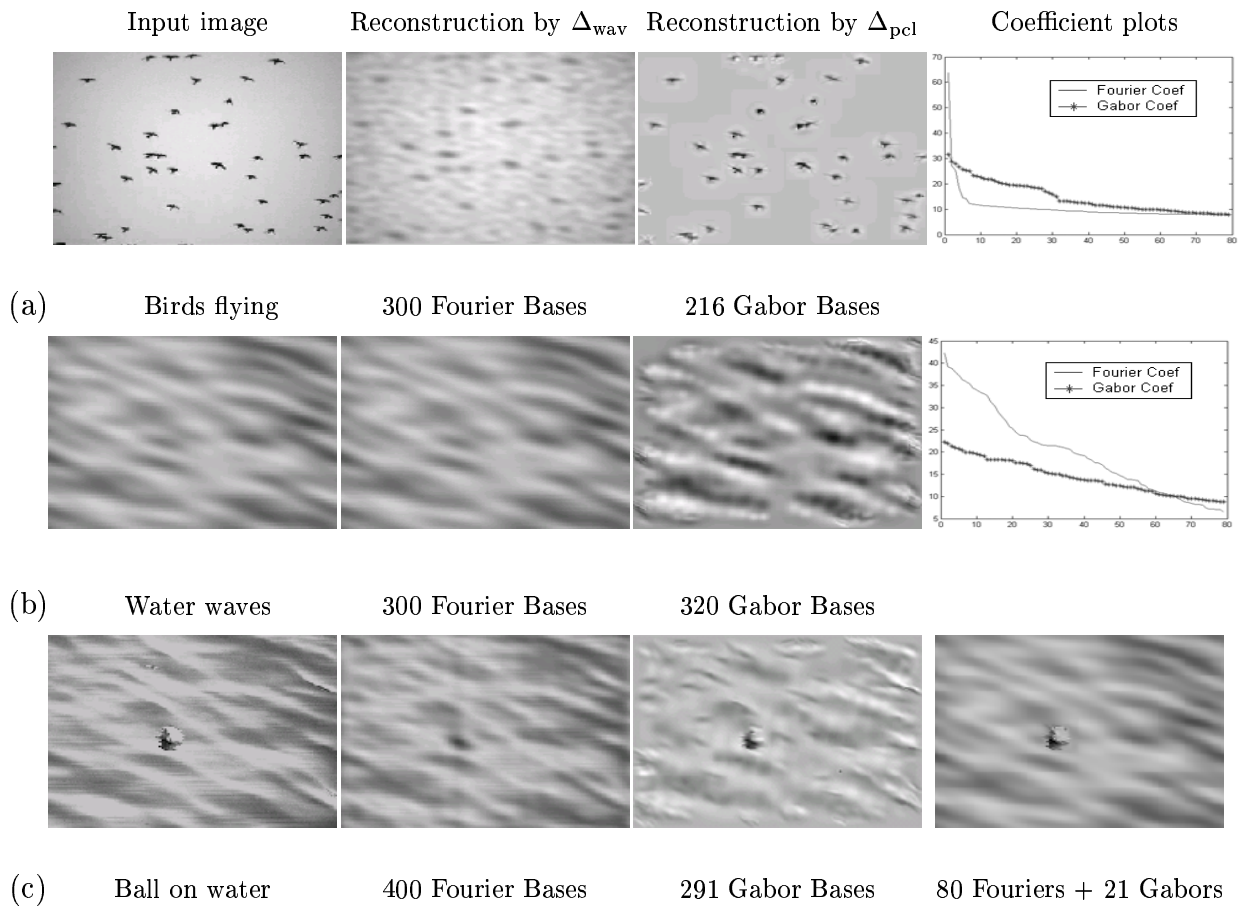


Fig. 2. Comparison of image reconstructions by Fourier bases Δ_{wav} and Gabor/LoG bases Δ_{pcl} respectively. The curves plot the base coefficients obtained by projecting the images onto the image bases. The thick curve is for Δ_{pcl}. The slopes of the curves reflect the coding efficiencies of the 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.

different textured motion patterns. We select three typical images for illustration. From each image, we obtain two reconstructions: one by wave (Fourier) bases from Δ_{wav} and the other by particle (Gabor and LoG) bases from Δ_{pcl}. For the third image, we select bases from both dictionaries.

Bottom-up computation and comparison of bases. The selection of Fourier bases from Δ_{wav} are easy, as they are orthonormal. We simply choose N_{wav} Fourier bases which have the highest coefficients. For the particle bases, we adopt a match pursuit procedure[16]. Given an input image $\mathbf{I}^{obs}$, it starts with a constant image $\mathbf{I}$ whose intensity is equal to the mean of $\mathbf{I}^{obs}$. At each step it selects a base ψ_j which has the highest response.

The response is the inner product between the base and the residue image.

$$\psi_j = \arg\max_{\psi \in \Delta_{\rm pcl}} < \psi, \mathbf{I}^{\rm obs} - \mathbf{I} >, \quad \alpha_j =< \psi_j, \mathbf{I}^{\rm obs} - \mathbf{I} > .$$

Once ψ_j is added to the base map, the response of a remaining base ψ_k in $\Delta_{\rm pcl}$ will be adjusted if ψ_k is not orthogonal to ψ_j, i.e. $\alpha_k \leftarrow \alpha_k - \alpha_j < \psi_j, \psi_k >$. The procedure stops at step N if the largest coefficient $\alpha_N \leq \epsilon$.

Figure 2.a is a flock of birds. The reconstruction with $N_{\rm wav} = 300$ Fourier bases (second column) is very blurred, in contrast the reconstruction with $N_{\rm pcl} = 216$ particle bases capture 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 well. 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 $\alpha_j, (j = 1, 2, ..., N_{\rm pcl}$ or $N_{\rm wav})$ of the bases in the order they are selected from $\Delta_{\rm wav}$ and $\Delta_{\rm pcl}$ 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 capturing the global lighting effects 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.

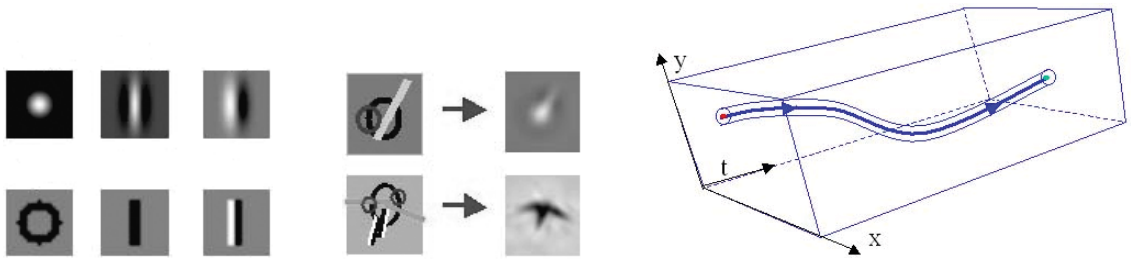
The two sets of bases are combined in general case to yield a base map

$$\mathbf{B} = \mathbf{B}_{\rm pcl} \ \cup \ \mathbf{B}_{\rm wav} = \{\mathbf{b}_j = (\alpha_j, \beta_j), \ j = 1, 2, ..., N\}, \quad N = N_{\rm pcl} + N_{\rm wav}. \tag{6}$$

These bases compete in a match pursuit procedure. In general, as Δ is over-complete with $|\Delta| = O(100\,|\Lambda|)$, $\mathbf{B}$ is a parsimonious token representation with $N = O(|\Lambda|/100)$.

The photometric model in eqn. (1) is rewritten as a conditional probability for image $\mathbf{I}$,

$$p(\mathbf{I}|\mathbf{B}_{\rm pcl}, \mathbf{B}_{\rm wav}; \sigma_o) = \frac{1}{(2\pi\sigma_o^2)^{|\Lambda|}} \exp\{-\sum_{(u,v)\in\Lambda} (\mathbf{I}(u,v) - \sum_{j=1}^{N} \alpha_j \psi_j(u,v;\beta_j))^2/2\sigma_o^2.\} \tag{7}$$



(a) (b) (c)

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 learned motons – snowflake and bird. (c) A graphic view of moton trajectory – the cable model.

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

The match pursuit procedure is only a bottom-up step in computing the base map $\mathbf{B}$ from a static image. As we proceed, $\mathbf{B}$ will be adjusted for spatial and temporal coherence, and tracked in the image sequence by an algorithm 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.

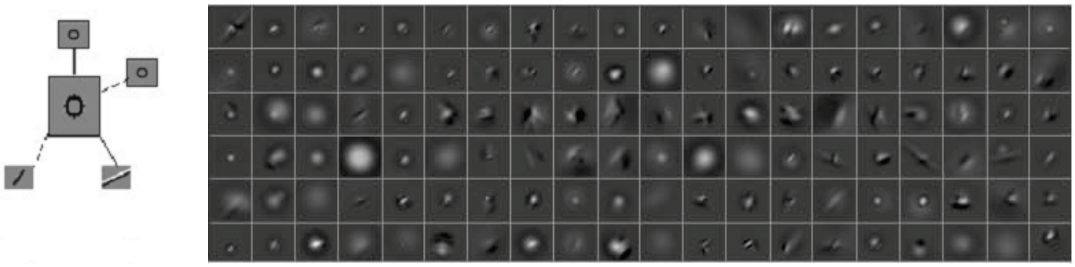
The bases in $\mathbf{B}_{\text{pcl}}$ often form spatially coherent groups and each group is a moving object called "motons". Thus $\mathbf{B}_{\text{pcl}}$ is partitioned into disjoint subsets

$$\mathbf{B}_{\text{pcl}} = S_1 \cup S_2 \cup \cdots \cup S_{M_{\text{pcl}}}, \quad \text{with } M_{\text{pcl}} \ll N_{\text{pcl}}.$$

Figure 3.b shows two examples. The image of a snowflake is the sum of three bases: 2 LoG bases and 1 Gcos base with some space displacements. A bird consists of 7 bases: 3 LoG bases, 2 Gcos bases 2 Gsin bases. These subset $S_i, i = 1, 2, ..., M_{\text{pcl}}$ are further clustered into a few typical configurations, represented by a set of deformable templates

$$\Phi_\pi = \{\Pi_\ell : \ell = 1, 2, ..., k\}.$$

Each subset S_i is an instance of one of the templates Π_ℓ. For example, the snow sequence has one ($k = 1$) moton template shown in Figure 4.a. Figure 11 shows three ($k = 3$) templates for different gestures for the bird sequence. As Figure 4.a shows, each template



(a) (b)

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.

usually has a "heavy" base with relatively large coefficient α_j surrounded by several "light" bases with relatively small coefficients α_j. By analogy to physical model of atoms, we call the heavy bases the "nucleus bases" as they have heavy weights like protons and neutrons, and the light bases the "electron bases". The atomic models are illustrated for the birds in Figure 12.b.

The dictionary of motons are formed from the templates in Φ_π through transformations T denoted by $\beta = (x, y, \theta, \sigma)$ and deformations D specified by variables ζ. ζ also includes the binary variables for presence and absence of a base.

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

Each moton instance is denoted by $\pi(\ell, \beta, \zeta)$. In a formal language, Δ_π is the "orbit" formed from Φ_π through some group operations. Figure 4.b shows 120 moton instances for the snowflakes and the deformable model captures the variations of snowflakes.

With dictionary Δ_π we represent the base map $\mathbf{B}_{\text{pcl}}$ by a moton map $\mathbf{M}_{\text{pcl}}$ with each subset S_i denoted by a moton π_i,

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

Thus we arrive at a more abstract and parsimonious representation.

The bases in Δ_{wav}, in theory [25], also travel in groups. For example, water flows are traveling sinusoid waves caused by different sources of vibration, such as wind, boat, earth-

quake, etc. But such motion can only be seen in images with large view scope. In our experiments, we only see waves in a single group and thus we don't need to group them. For unification of notation, we use π_j to denote a wave base, thus

$$\mathbf{M}_{\text{wav}} = \{\pi_j = \mathbf{b}_j; j = 1, 2, ..., N_{\text{wav}}\} = \mathbf{B}_{\text{wav}}.$$

Thus the geometric model can be expressed as a conditional probability

$$p(\mathbf{B}|\mathbf{M}; \Phi_p i) = p(\mathbf{B}_{\text{pcl}}|\mathbf{M}_{\text{pcl}}; \Phi_\pi) p(\mathbf{B}_{\text{wav}}|\mathbf{M}_{\text{wav}}) = \prod_{i=1}^{M_{\text{pcl}}} \prod_{j \in S_i} p(\mathbf{b}_j|\pi_i; \Pi_{\ell_i}) \cdot \delta(\mathbf{B}_{\text{wav}} - \mathbf{M}_{\text{wav}}). \quad (9)$$

In summary, we have a two-level generative model. The moton map $\mathbf{M}$ generates base map $\mathbf{B}$ with dictionary Δ_π, and the base map bB in turn generate image $\mathbf{I}$ with dictionary Δ.

$$\mathbf{M} = (\mathbf{M}_{\text{pcl}}, \mathbf{M}_{\text{wav}}) \stackrel{(\Phi_\pi, \Delta_\pi)}{\longrightarrow} \mathbf{B} = \mathbf{B}_{\text{wav}} \cup \mathbf{B}_{\text{pcl}} \stackrel{(\Phi_{\text{pcl}}, \Delta_{\text{pcl}}) \cup (\Phi_{\text{wav}}, \Delta_{\text{wav}})}{\longrightarrow} \mathbf{I}. \quad (10)$$

In Section (III), we shall discuss the algorithm that infers $\mathbf{B}$ and $\mathbf{M}$ from $\mathbf{I}$ as hidden variables and learn the moton templates Φ_π as parameters.

C. The moton trajectories and representation of the sequence

The generative model in eqn.(10) is for static image. For a sequence $\mathbf{I}[0, \tau]$, the motons and bases should be tracked from frame to frame. As Figure 3.c shows, each 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, the trajectory of a moton is denoted by

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

For example, a snowflake enters our view at frame t^b and leaves our view at frame t^e. Intuitively, the moton trajectory is like a cable. Its nucleus base forms the *core* of the cable, and the trajectories of its "electron bases" form the *coil*, due to self-rotation, surrounding the cable's core. In a coarse-to-fine computation, we can 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, and 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 thus rewrite the two level hidden representation $\mathbf{B}[0,\tau]$ and $\mathbf{M}[0,\tau]$ as a number of K trajectories $\mathcal{C}_i, i = 1, 2, ..., K$, and denote it by

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

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 of motons and bases 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 $\mathbf{B}$ and motons π with symbolic representation.

D. Dynamic model – sources, sinks and wave-particle interactions

In this subsection, we present the dynamic model for $\mathcal{C}_i[t^b, t^e], i = 1, 2, ..., K$ – the moving elements. We are particularly interested in some interactions between the elements and thus the coupling of the trajectories $\mathcal{C}_i[t^b, t^e], i = 1, 2, ..., K$. The first type of coupling is the influence of waves on particles. For example, balls drifting in a river, grass waving in the wind. This kind of effect cannot be simulated by previous models [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 travels together with complex interactions. The relative motion of different Fourier bases must be constrained to keep certain phase alignments. Other interactions, such as particle-particle collision, 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 ℓ, transforms $\beta = (x, y, \sigma, \theta)$, and deformation ζ. For waves $\pi = (\xi, \eta, \phi)$ is degenerated to a Fourier base with its frequency and phase. The general motion equation for $\pi(t)$ is a

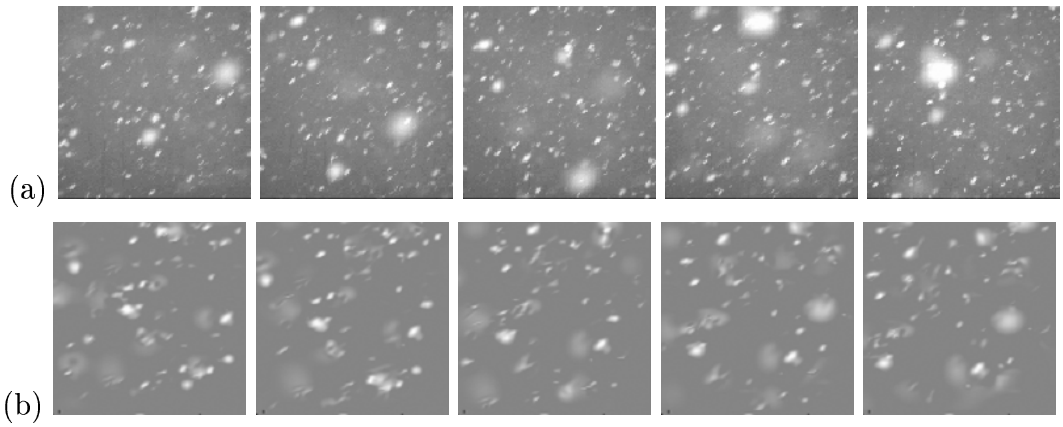


Fig. 5. Experiment on the falling snow sequence. (a) Observed sequence of falling snow. (b) Synthesized sequence of falling snow by sampling the generative model.

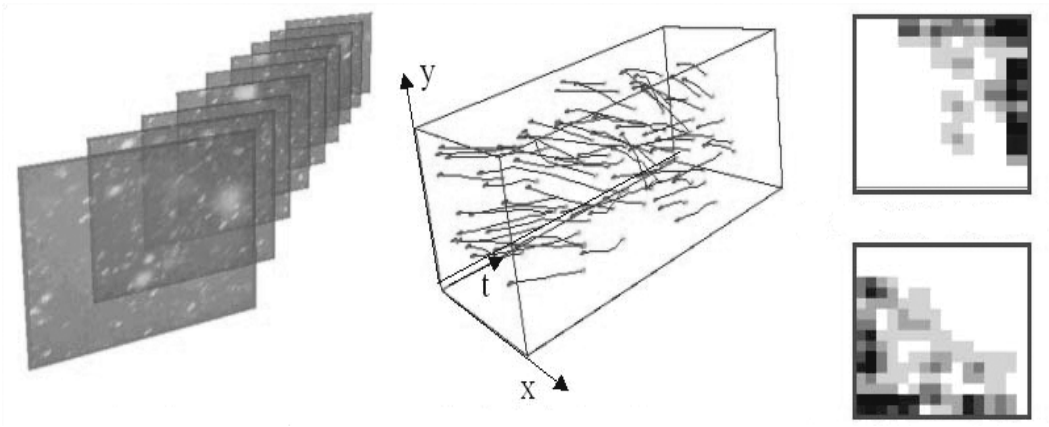


Fig. 6. Experiment on the falling snow sequence. Left: Observed sequence. Middle: Graphic view of the computed trajectories of the snowflakes (as hidden variables). Upper right: a probability map of the sources for snowflakes to enter the scene. Lower right: probability map of the sinks for the snowflakes to leave the view. Dark means high probability.

p-th order AR model with coefficients $a = (a_1, ..., a_p)$, driven by three sources of forces: (1) the influence from the other waves $U(\mathbf{B}_{\text{wav}}(t))$; (2) an external force $f(\pi(t))$ from objects outside the system, such as gravity, wind field, and external constraints, which may variation over space and time; (3). A Brownian motion n. So we have

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

In the following, we study three special cases that occur in our experiments.

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

We start with a case where we assume the particles move independently, such as snowing, bird flying, fireworks etc. Though a few Fourier bases is used to model the global lighting

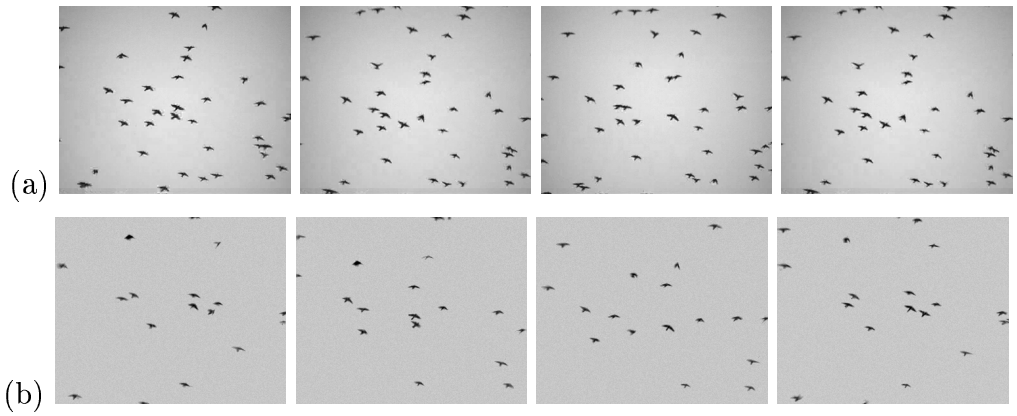


Fig. 7. Experiment on a flying-bird sequence. (a) Observed sequence of flying birds. (b) Synthesized sequence with fewer flying birds by editing the number of motons M when sampling the model.

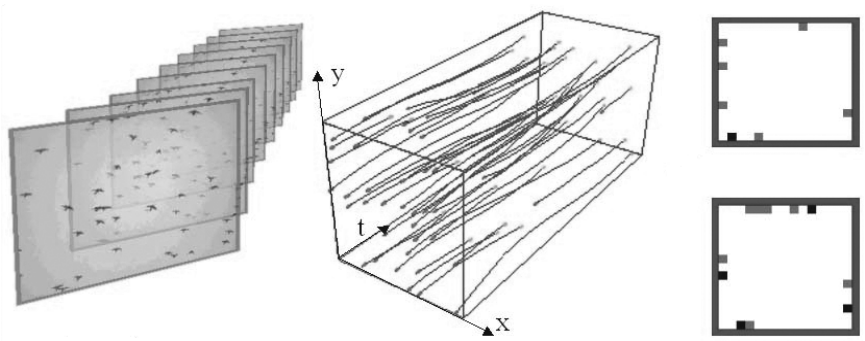


Fig. 8. Experiment on the bird sequence. Left: Observed sequence. Middle: Graphic view of the computed trajectories of the birds. Upper right: a probability map of the sources for birds to enter the scene. Lower right: probability map of the sinks for birds to leave the view. Dark means high probability.

effects, they are static and do not affect the motons. The external force $f(\pi) = c$ is a constant vector. Thus we obtain a simplified 2nd order Markov chain model,

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

The birth of a moton π and its timing t^b follows 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 the source map or birth map. The timing is important 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 the death map. π is a long vector, P_B and P_D are high dimensional, we are most

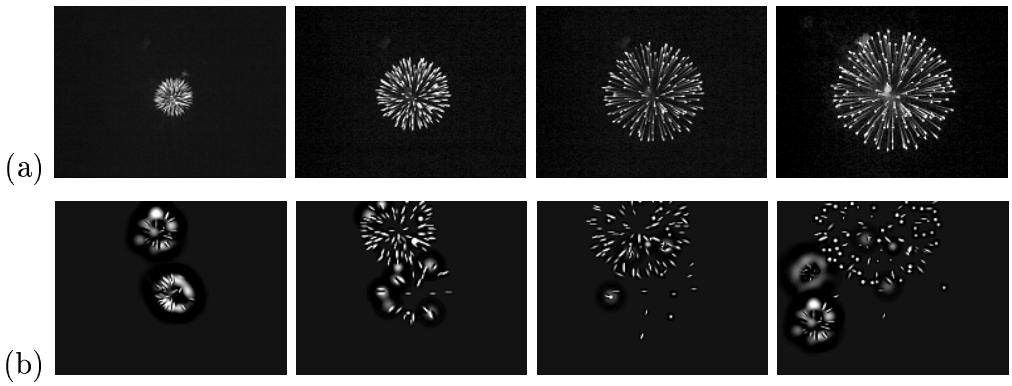


Fig. 9. Experiment on firework sequence. (a) Observed sequence with only one firework. (b) Synthesized sequence of multiple fireworks after editing its birth (source) map.

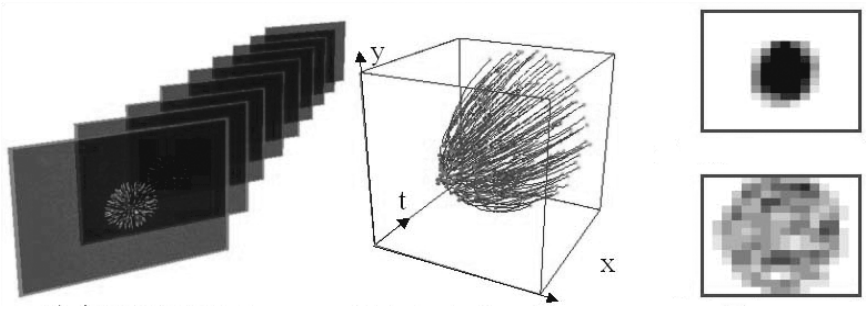


Fig. 10. Experiment on the firework sequence. Left: observed sequence. Middle: graphic view of the trajectories of the firework. Upper right: source map of the firework. Lower right: sink map of the firework.

interested in the location (x, y) in practice.

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

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

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

For example, Figure 6 (right size) displays birth map $p_B(x, y)$ and and death map $P_D(x, y)$ for the snow sequence. The dark means high probability. Thus the algorithm "understands" that the snowflakes enter mostly from the upper-right corner and disappear around the lower-left corner.

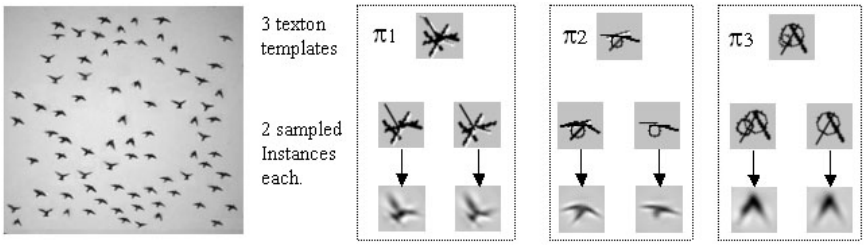


Fig. 11. Motons in a bird flying sequence. Left: input image. Right: three moton templates $\Phi_\pi = \{\Pi_i, i = 1, 2, 3\}$ learned in a clustering step for different poses. Two instances are shown for each template.

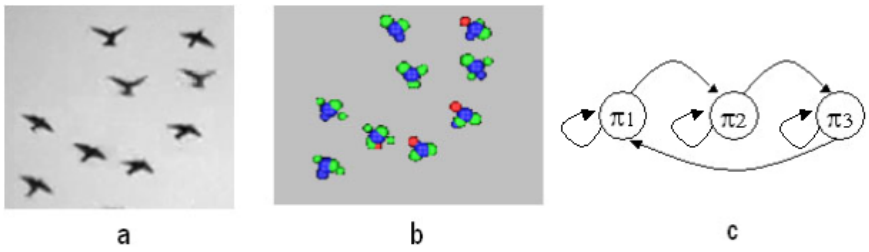


Fig. 12. (a) Input image. (b) 3D graphic illustration for the "atomic" model of bird motons $\pi_j, j = 1, 2, ..., 9$. (c) diagram of three states transitions for birds flying.

In summary, we can write the probability for a moton trajectory as

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

Γ denotes all the parameters in the dynamic models.

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

Remark 1: For the firework sequence in Figure 9, the death and birth of motons must be synchronized in timing, as a large number of particles come and go together. This is coded by the probabilities P_B and P_D. The death/birth maps can also be manipulated, so that we edit the number of objects and the events happening at any time and places we expect. For example, we only observe a single firework in the original sequence, but we can generate several fireworks at different places and time intervals as shown in Figure 9. By reducing the number of birds, we observe fewer birds in the synthesized sequence in Figure 7.

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. Thus to have the birds flap wings, the Markov chain model $p(\pi(t)|\pi(t-1), \pi_i(t-2))$ includes a 1st order transition probability $p(\ell(t)|\ell(t-1))$ with $\ell(t) \in \{1, 2, 3\}$ being a variable in $\pi(t)$. It is represented by a 3×3

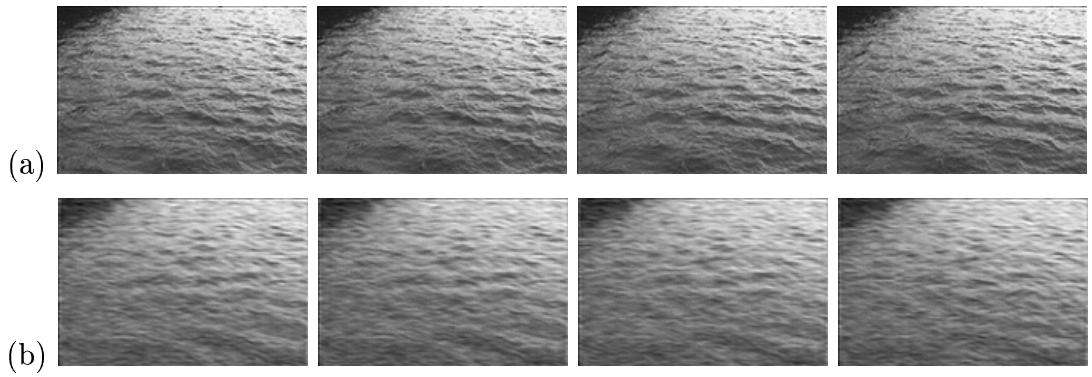


Fig. 13. Experiment on a river sequence. (a) Observed sequence of wavy river. (b) Synthesized sequence with 1000 Fourier bases.

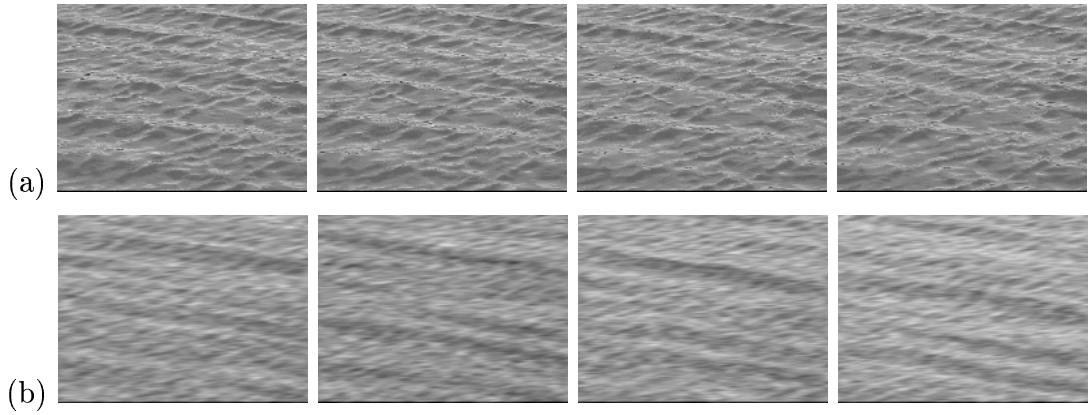


Fig. 14. Experiment on a wavy pond sequence. (a) Observed sequence. (b) Synthesized sequence with 1200 Fourier bases.

matrix. This is not necessary in the snow and firework sequences.

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

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

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

N is fixed and there is no birth or death events. Furthermore, 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 $\phi_j, j = 1, ..., N$ change and it is known as the phase motion [7].

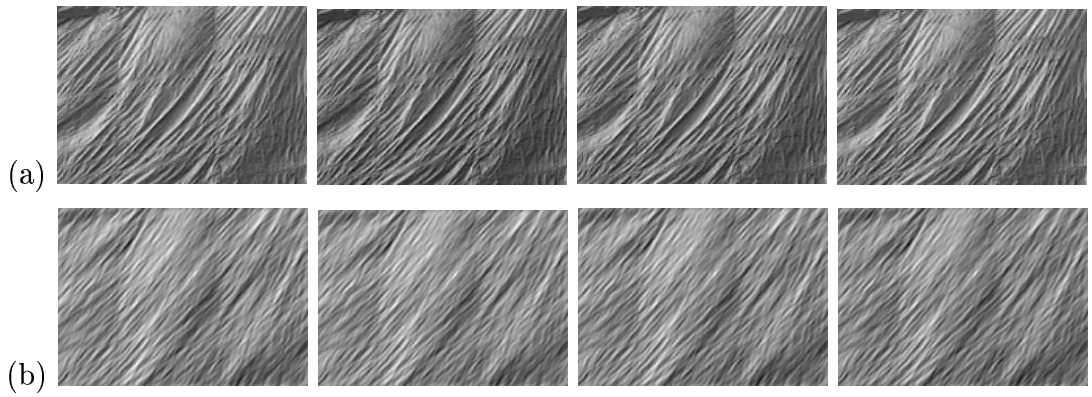


Fig. 15. Experiment on a plastic foil sequence. (a) Observed sequence. (b) Synthesized sequence with 1500 Fourier bases.

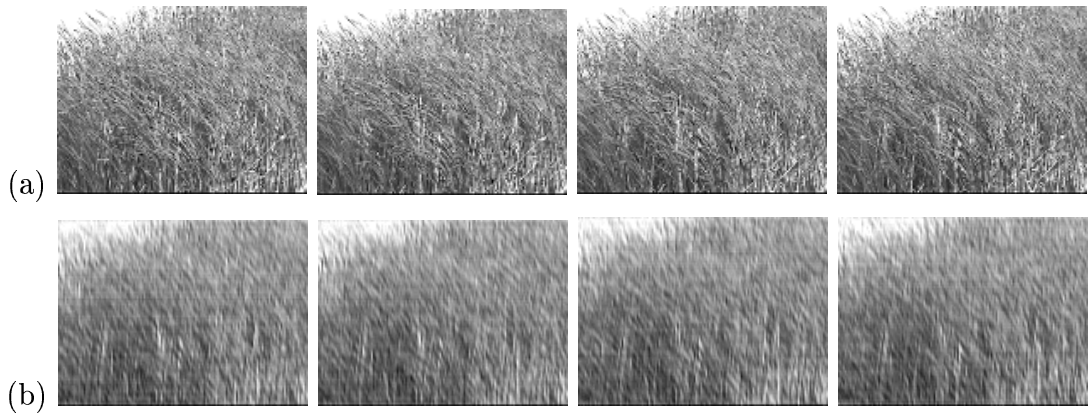


Fig. 16. Experiment on a grassland sequence with 2000 Fourier bases for its spatial wave pattern. (a) Observed sequence. (b) Synthesized sequence.

The speed of phase motion is related to the speed in the space by

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

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.

$$\phi_j(t) = \sum_{i=1}^{p} a_{ji}\phi_j(t-i) + n_j, n_j \ N(0, \sigma^2), \quad j = 1, 2, ..., N.$$

With $p = 15 \sim 20$ to account for low frequency components, this simple model can synthesize the river sequences reasonably well, continuing from the observed sequence. But the phases

become misaligned-aligned after 30-50 frames. To align the phases, 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. Let $e_i, i = 1, 2, ..., m$ be the eigen-vectors with largest eigen values, then $\gamma_j(t) =< \phi(t), e_j >, j = 1, ..., m$ are the projected coefficients. In our experiments $m = 8$ and the m coefficients follow independently a p-th order AR model

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

The total number of variables used in the model is $3N$ for $(\xi_j, \eta_j, \alpha_j), j = 1, ..., N$, $8N$ for the eigen vectors, plus 20×88 for dynamics AR coefficients. The compression rate is compared in Table I.

Since we transfer the wave sequence $\mathbf{B}_{\text{wav}}[0, \tau]$ into a representation on the sequence of coefficients $\gamma[0, \tau]$, we can write the probability as

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

We assume some initial conditions for the first p frames and $\Gamma_w ave$ 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 it successfully characterize 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 $N = 2000$ to reconstruct the high frequency components.

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

Some motion sequences have both particles and waves, for example, Figures 17 and 18 show the ball and foams drifting on water. The coupling of the two types of elements is characterized by a driving force from wave to particles.

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. The movement of the particles are driven by the waves. As

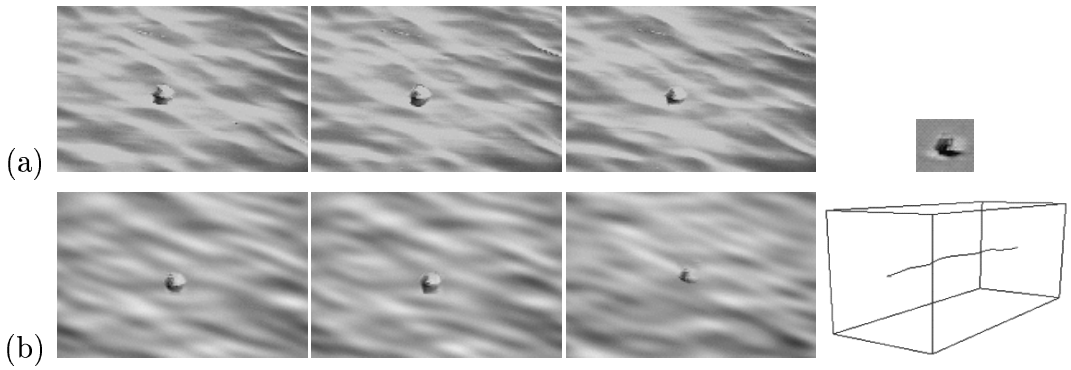


Fig. 17. Experiment on a floating-ball sequence. (a) Observed sequence. (b) Synthesized sequence and the trajectory of the ball.

the particles are small, we are only concerned with the position (x, y) and other attributes in π can be fixed. For unification of notation, we write π for (x, y).

Given the phase motion $d\phi$ in case 2, we transfer it to motion speed in spatial domain (dx, dy) through equation (16). The motion of a particle is then influenced by the sum of the speed at point (x, y). Thus the dynamics of the moton π is

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

The second term in the above equation accounts for the coupling of the particle motion with waves. In practice, we only need to choose $q = 20 - 30$ Fourier bases $(\tilde{\xi}_k, \tilde{\eta}_k, \tilde{\phi}_k) \in \mathbf{B}_{\text{wav}}, k = 1, 2, ..., q$ with lower frequencies to drive the particles. 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. The trajectory of a moton follows the following probability,

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

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

The synthesized floating ball and floating foams results are shown in Figure 17 and 18. The coupling of the particles with waves appears realistic in the video sequence (see supplementary file).

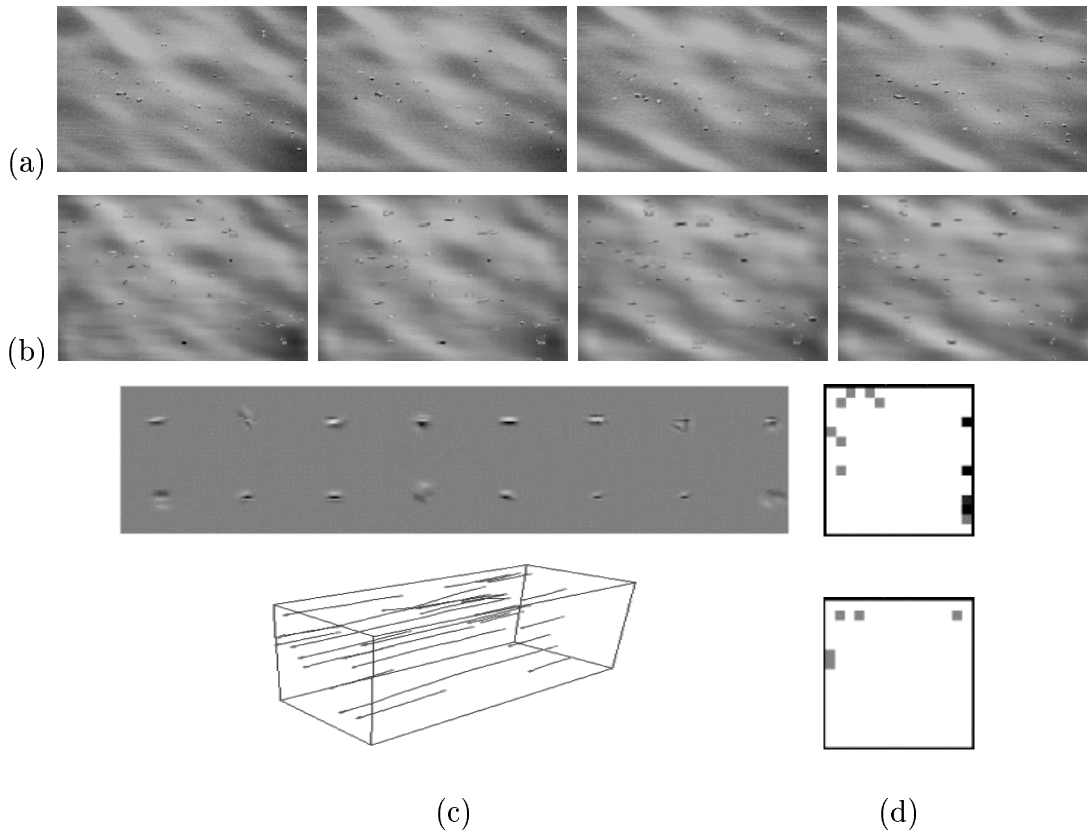


Fig. 18. Experiment on a sequence with many foam particles drifting in a river. (a) Observed sequence. (b) Synthesized sequence. (c) Learned motons: foams and their trajectories. (d) Sources and sinks of the floating foams.

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 $\mathbf{I}^{\text{obs}}[0,\tau]$ and hidden representation $W[0,\tau]$,

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

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

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

and $\Theta = (\Phi, \Gamma_{\text{wav}}, \Gamma_{\text{pcl}})$ includes the parameters in the deformable templates for moton, and parameters in 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 Θ in the models. With the 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 (Figures 5-18).

A. Problem formulation and stochastic gradient

The problem is posed as statistical learning by maximum likelihood estimate (MLE). The objective is to compute the optimal parameters that maximize the log-likelihood,

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

Take the derivative with respect to Θ, and set it to zero, we have,

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

The MLE is solved by iterating two steps.

Firstly, under the current parameter Θ, we simulate samples for the posterior

$$W_i[0,\tau] \ \sim \ p(W[0,\tau] \mid \mathbf{I}^{\text{obs}}_{[0,\tau]};\Theta), i = 1,2,...,M. \tag{22}$$

Then we estimate the above expectation by importance sampling.

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

For ease of discussion we set $M = 1$ without loss of generality.

Secondly, plug in equation (21), we have the following equations for learning the parameters $\Theta = (\Phi, \Gamma_w ave, \Gamma_{\text{pcl}})$,

$$\sum_{i=1}^{\tau} \frac{\partial \log p(\mathbf{B}_{\text{pcl}}(t)|\mathbf{M}_{\text{pcl}}(t);\Phi)}{\partial \Phi} = 0, \quad \text{(learning motons)} \tag{23}$$

$$\frac{\partial \log p(\mathbf{B}_{\text{wav}}[0,\tau];\Gamma_{\text{wav}})}{\partial \Gamma_{\text{wav}}} = 0 \quad \text{(learning wave dynamics)} \tag{24}$$

$$\sum_{k=1}^{K} \frac{\partial \log p(\mathcal{C}_k[0,\tau]|\mathbf{B}_{\text{wav}}[0,\tau];\Gamma_{\text{pcl}})}{\partial \Gamma_{\text{pcl}}} = 0, \quad \text{(learning particle dynamics)}. \tag{25}$$

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 [8] to converge to a globally optimal Θ^* even with $M = 1$, provided that the stepsize in learning of parameters Θ is slow enough so that the importance sampling makes a good approximation at the current Θ. Intuitively, with small stepsize it uses samples obtained over time 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 have a large value, say $\alpha_j \geq \epsilon = 3.0$. The particle bases with such high coefficients are treated as the "nuclei" for the motons. Then we lower the threshold, say $\epsilon = 1.0, 0.5$. Thus some new "electron" bases are added and are 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 $S_1, ..., S_{M_{\text{pcl}}}$ into a smaller number of k clusters. The mean of each cluster is then a deformable template for motons, and we denote them by $\Phi_\pi = \{\Pi_1, ..., \Pi_k\}$. Usually we have to come up with a pre-defined number k with $1 \leq k \leq 3$ for a sequence. This will force each set $S_j, j = 1, ..., M_{\text{pcl}}$ to fit to one of the template. This clustering process is easily implemented by a k-mean method. We define the distance between a set S_j and a deformable model Π_i to be the difference of image generated by the bases in S_j and

in Π_i plus the structural divergence. S_j is registered to Π_j by a similarity transform and a simple graph matching in structures. We refer to a previous paper for more details [29].

Thirdly, we track the nuclei bases in the video and compute trajectories $\mathcal{C}_i, i = 1, 2, ..., K$ by a simplified Condensation algorithm [11]. When the motons 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 $\mathcal{C}_j[t_j^b, t_j^e], j = 1, 2, ..., K$, so that some trajectories are grouped, extended, and mutated to achieve a high posterior probability

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

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

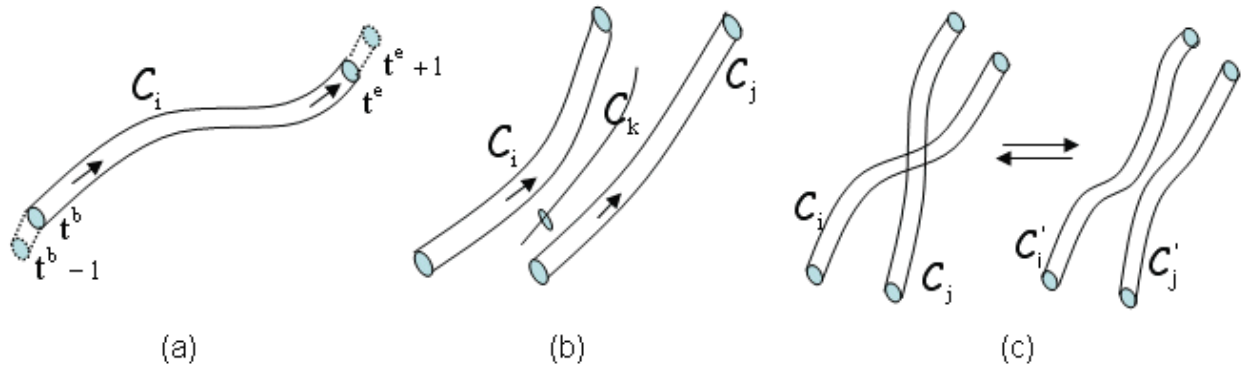


Fig. 19. Three typical reversible jumps. (a) Extend/shrink a trajectory, (b) Group/ungroup a trajectory. (c) Mutation and split/merge of trajectories.

Each move in our Markov chain design is a reversible jump between two states A and B realized by a Metropolis-Hastings method[18]. 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

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

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 $dAdB$ are 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 selected at random 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, and each queue lists a number of candidate trajectories that need 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[26], [30]. Due to space limit, we only briefly specify the four moves in the following.

Move Type 1: Extending/shinking a trajectory $\mathcal{C}_i$. This move is illustrated in Fig. 19.a

and is a jump between two states A and B,

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

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

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

q_1 is a probability for choosing type 1 move, $q(i)$ is the probability for picking $\mathcal{C}_i$, and with q_{tail} it chooses 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 move is illustrated in Fig. 19.b. Let $\mathcal{C}_k$ be a short trajectory of a base, usually an "electronic" base with small coefficient, it is desirable to group it with a nearby trajectory $\mathcal{C}_i$ or $\mathcal{C}_j$. The length of $\mathcal{C}_k$ could be different from those of $\mathcal{C}_i$ and $\mathcal{C}_j$.

The move is a jump 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

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

We first choose move type 2, and then choose to group or ungroup an existing trajectory with probabilities q_{grp} or q_{ugrp} respectively. Then we choose a trajectory $\mathcal{C}_k$ with single base to group with probability $q(k)$ or a composed trajectory $\mathcal{C}_i'$ to ungroup, and so on. The

probabilities, like q_{grp}, q_{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 $\mathcal{C}_i[t_i^b, t_i^e], \mathcal{C}_j[t_j^b, t_j^e]$ 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

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

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.

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

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 birth move, and select $\mathbf{b}_j$ with $q(j)$ for the death move.

D. Learning the parameters Θ

Given the sampled hidden variables $W[0,\tau] = (\mathbf{B}_{\text{wav}}, K, \mathcal{C}_1, ..., \mathcal{C}_K)$, we update the parameters Θ in a second step for both the deformable moton template Φ and the dynamics $\Gamma_{\text{wav}}, \Gamma_{\text{pcl}}$, following the equations (23), (24), and (25).

$$\begin{aligned}
\Gamma_{\text{wav}} &\leftarrow (1-\rho)\Gamma_{\text{wav}} + \rho \frac{\partial \log p(\mathbf{B}_{\text{wav}}[0,\tau]; \Gamma_{\text{wav}})}{\partial \Gamma_{\text{wav}}}, \\
\Gamma_{\text{pcl}} &\leftarrow (1-\rho)\Gamma_{\text{pcl}} + \rho \sum_{k=1}^{K} \frac{\partial \log p(\mathcal{C}_k[0,\tau] | \mathbf{B}_{\text{wav}}[0,\tau]; \Gamma_{\text{pcl}})}{\partial \Gamma_{\text{pcl}}}, \\
\Phi &\leftarrow (1-\rho)\Phi + \rho \sum_{i=1}^{\tau} \frac{\partial \log p(\mathbf{B}_{\text{pcl}}(t) | \mathbf{M}_{\text{pcl}}(t); \Phi)}{\partial \Phi}
\end{aligned}$$

Unlike the EM-algorithm, which maximizes the likelihood at each step, our algorithm only update Θ with a small stepsize for global convergence[8].

The birth/death maps, p_B and p_D, of particles are learned by counting the the head and tail of cable at their locations and time (see equation (14)).

E. Experiments

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

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

Figures 5- 16 show some results on the analysis and synthesis (with editing) for a number of texture motion patterns. We have discussed them in Section II and these results can be seen from the supplementary video clips. Here we mention a few problem.

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

(ii). The inference of $W[0,\tau]$ with MCMC is computationally intensive. The time complexity for learning a textured motion sequence containing particles usually is 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 $50 - 100$ frames.

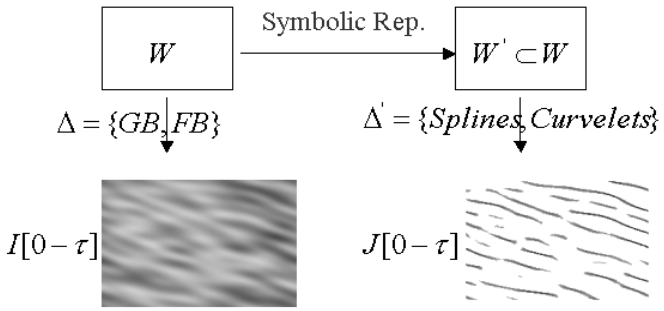


Fig. 20. From video to cartoon sketch. The cartoon representation $W\prime \subset W$ is a simplified version of W with some details removed selectively. By replacing the particle and wave bases with symbolic sketches, we can easily synthesize cartoon with the same or edited generative model.

IV. Sketch model

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

In our view, a cartoon $\mathbf{J}[0,\tau]$ is a symbolic visualization of our inner representation $W[0,\tau]$ with some "unnecessary" details remove. 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 Δ_{pcl}, Φ_π and Δ_{wav} by symbolic sketches Δ'_{pcl} and Δ'_{wav} respectively. Then the cartoon $\mathbf{J}_{[0,\tau]}$ 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 the cartoon.

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

(1). Rendering particles. Each particle object π, such as birds or snowflakes, is rendered by a contour outlining the deformable template. Its motion follows the same dynamic models. Obviously one can choose other symbolic representations.

(a) (b) (c)

Fig. 21. Image interpolation from extracted sketches. (a) Observed image. (b) Extracted edges for ridges and valleys. (c) Reconstructed image from the information on the edges.

(2). Rendering waves. As the Fourier bases cannot be distinguishable, we sketch all Fourier bases as a whole. Let $\mathbf{I}_{\text{wav}}$ be an image reconstructed from $\mathbf{B}_{\text{wav}}$,

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

When we view $\mathbf{I}_{\text{wav}}(u,v)$, we not only perceive the global periodic waves as a whole, but also notice the individual peaks and valleys. There is a dual representation 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.

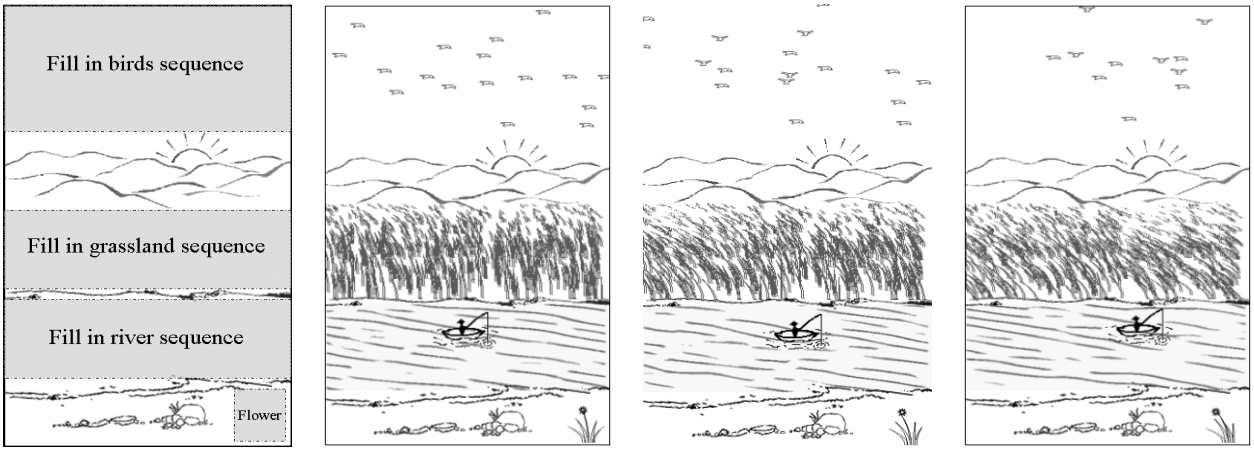
Figure 21.a shows an example for the river image. Figure 21.b is a collection of points for the ridge and valleys, denoted by

$$SK = \{(u,v) : \ \nabla^2 \mathbf{I}_{\text{wav}}(u,v) = 0\}$$

For each point (u,v) we remember its pixel intensity $\mathbf{I}_{\text{wav}}(u,v)$ 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.

$$\begin{aligned} \frac{dI(u,v)}{dt} &= \frac{\partial^2 \mathbf{I}}{\partial u^2} + \frac{\partial^2 \mathbf{I}}{\partial v^2}, \quad \text{for } (u,v) \notin SK, \\ \mathbf{I}(u,v) &= \mathbf{I}_{\text{wav}}(u,v), \ \nabla \mathbf{I}(u,v) = \nabla \mathbf{I}_{\text{wav}}(u,v), \ \forall (u,v) \in SK. \end{aligned}$$

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



(a) (b) (c) (d)

Fig. 22. Synthesized cartoon sequence based on learned textured motions. (a) The static background image drawn manually. Shaded area will be fill by three learned textured motion sequences showed in the previous sections, flying birds, dancing grass, and floating ball. The floating ball is replaced by a boat sketch. (b-d) Synthesized frames at $t = 1, 10, 20$.

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.

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. 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 learned 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

$$(x(t), y(t)) = \sum_{i=1}^{p=2} a_i(x(t-i), y(t-i)) + \sum_{k=1}^{q} b_k(\tilde{\xi}_k, \tilde{\eta}_k) d\tilde{\phi}_k(t) + \kappa(x - x_0, y - y_0) + n. \quad (27)$$

It is the same as case 3 in eqn. (19), except that we add an extra term in the force. Each tip is assumed to have a resting position (x_0, y_0), a spring is attached from (x, y) to (x_0, y_0).

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, source, sinks, and coupling in motion. A Markov chain Monte Carlo method is adopted for learning and inference.

In future work, 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 blur effect in 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 grant IIS-02-44763 and ONR grant N00014-02-10952. We thank Adrian Barbu, Cheng-en Guo, and Yingnian Wu for extensive discussions and especially 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.

References

[1] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman. "Texture mixing and texture movie synthesis using statistical learning", *IEEE Trans. on Vis. & Comp. Graph.*, 7, 2001.

[2] A. Cliff and J. Ord, "Space-time modeling with an application to regional forecasting", *Trans. Inst. British Geographers*, 66:119-128, 1975.

[3] D. Ebert and R. Parent, "Rendering and animation of gaseous phenomena by combining fast volume and scaleline A-buffer techniques", *SIGGRAPH*, 1990.

[4] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling.",*ICCV, 2:10338*,

[5] D. Field, "What is the goal of sensory coding?", *Neural Computation, 6:559-601, 1994*

[6] A. Fitzgibbon, "Stochastic ridigity: image registration for nowhere-static senes", *ICCV*, pp 662-669, July 2001.

[7] D. Fleet, A. Jepson, "Stability of phase information", *IEEE Trans. on PAMI*, 15(12):1253-1268, 1993.

[8] M.G. Gu, and F.H. Kong, "A stochastic approximation algorithm with Markov chain Monte-Carlo method for incomplete data estimation problems", *Proc. of the National Academy of Sciences*, 95, pp. 7270-74, 1998.

[9] D. Heeger and J. Bergen, "Pyramid-based texture analysis and synthesis", *SIGGRAPH, 1995.*

[10] B. Horn and B. Schunck, "Determining optical flow", *Aritificial Intelligence*, 17:185-203, 1981.

[11] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density", *ECCV*, 1996.

[12] B. Julesz, "textons, the elements of texture perception and their interactions", *Nature*, 290:91-97, 1981.

[13] R. Kailath, *Linear Systems*, Prentice Hall, Englewood Cliffs, NJ, 1980.

[14] R. Mann and M. S. Langer, "Optical snow and the aperture problem", *ICPR, Vol. IV, pp.264-7, Aug. 2002.*

[15] B.F. Logan Jr, "Information in the zero-crossings of band pass signals", *Bell Sys. Tech. J.*, 56, 487-510, 1977.

[16] S. Mallat and Z. Zhang, "Matching Pursuit in a Time-Frequency Dictionary", *IEEE Trans. on Sigmal Processing, 41:3397-415, 1993.*

[17] D. Marr, "Vision", *IW.H. Freeman, 1983*

[18] N. Metropolis, M. Rosenbluth, A. Rosenbluth, A. Teller, and E. Teller, "Equations of state calculations by fast computing machines," *J. Chemical Physics, 21, 1087-92, 1953.*

[19] W. T. Reeves and R. Blau, "Approximate and probabilistic algorithms for shading and rendering structured particle systems", *SIGGRAPH, 1985.*

[20] P. Saisan, G. Doretto, Y. Wu, and S. Soatto, "Dynamic Texture Recognition," *CVPR, 2001.*

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

[22] S. Soatto, G. Doretto, and Y. Wu, "Dynamic Texture", *ICCV, 2001*

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

[24] M. Szummer and R. W. Picard, "Temporal texture modeling" , *ICIP, 3, 823-6, 1996.*

[25] R.A.R. Tricker, "Bores, Breakers, Waves and Wakes", *American Elsevier, New York, 1965.*

[26] Z. Tu and S. Zhu, "Image Segmentation by DDMCMC", *PAMI, vol.24, pp. 657-673, May, 2002.*

[27] Y. Wang, S. Zhu, "A Generative Method for Textured Motion: Analysis and Synthesis", *ECCV, 2002.*

[28] L. Wei and M. Levoy, "Fast Texture Synthesis using Tree-structured Vector Quantization", *SIGGRAPH, 2000.*

[29] S. Zhu, C. Guo, Y. Wu, and Y. Wang, "What are Textons?", *ECCV, 2002.*

[30] S.C. Zhu, R. Zhang, and Z.W. Tu, "Integrating top-down/bottom-up for object recognition by data-driven Markov chain Monet Carlo", *CVPR*, Hilton Head, SC. 2000.

[31] S. Zhu, Y. Wu, and D. Mumford, "Minimax entropy principle and its applications to texture modeling", *Neural Computation*, 9:1627-60, 1997.