Modeling Complex Motion by Tracking and Editing Hidden Markov Graphs

Yizhou Wang
Computer Science Department
University of California, Los Angeles
$wangyz$@cs.ucla.edu

Song Chun Zhu
Department of Statistics
University of California, Los Angeles
$sczhu$@stat.ucla.edu

Abstract

In this paper, we propose a generative model for representing complex motion, such as wavy river, dancing fire and dangling cloth. Our generative method consists of four components: (1) A photometric model using primal sketch[8] which transfers an image into an attribute graph representation. Each vertex of the graph is a scaled and oriented image patch selected from a dictionary. The graph connects and aligns these patches. (2) A geometric model which characterizes the deformation of the attribute graph. (3) A dynamic model, which specifies the motion dynamics of these vertices (patches) and their interactions in the form of coupled Markov chains. (4) A topological model, which interprets the graph topological changes over time. We learn this generative model by a stochastic gradient algorithm implemented by Markov Chain Monte Carlo (MCMC) sampling. This method is shown to be effective in handling the topological changes of graphs. The correctness of the learned model is verified by the low-dimension reconstruction of the original image as well as by the realistic motion sequences it synthesized.

1. Introduction

In the literature, people from both graphic and vision communities always have keen interests in modeling complex motion patterns like dancing fire, wavy water and dangling cloth. A wide spectrum of models have been proposed to account for these motion phenomena. Fig.1 tries to map these models in a 1D axis according to the number of parameters that a model memorizes from the observed image sequence. At one extreme is the *physics-based models*, e.g. [16, 4]. This type of models are very parsimonious, and they explain the motion of the underlying systems by physics with few parameters. At the other extreme is *image-based models*, e.g. [17, 21], which remember every pixels of the system and reproduce new sequences by cut-and-paste techniques.

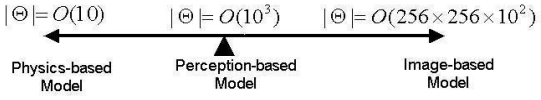
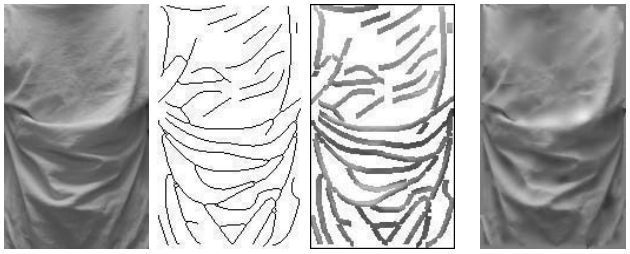


Figure 1: Three types of models used in the literature and numbers of their parameters. Physics-based models use the least number of parameters to specify a system. Image-based models use the most number of parameters. We believe the number of parameters for human beings to specify a system is in between the other two types of models.

Despite their success for realistic image synthesis, both models are not friendly or not suitable for image analysis and are perhaps pretty far from the mechanisms used in human visual perception. It is believed that a generic motion model, adopted by human vision, must lies somewhere between the two extremes. We vaguely call it the perception-based model.



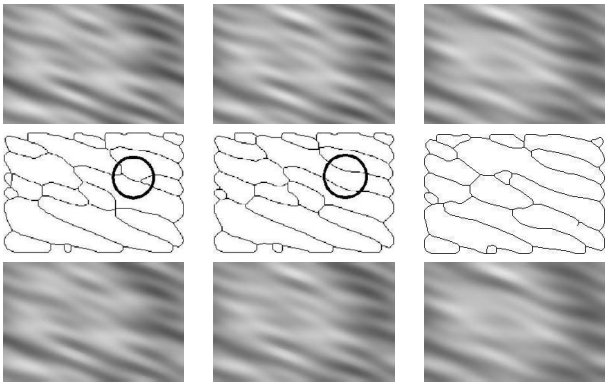
(a) Input (b) Sketch G (c) $\mathbf{I}_{\Lambda_{sk}}$ (d) Reconstructed

Figure 2: Representing a cloth image by primal sketch. (a) Input cloth image. (b) The primal sketch graph G. (c) The sketchable part $\mathbf{I}_{\Lambda_{sk}}$ Primal sketches of the cloth image. (d) Reconstructed image from (c) with heat diffusion.

To pursue a perception-based model, one shall first ask: "what do we see when we look at clothes and fire at a glance?" Some studies in psychology on this quest have led to the early vision theory including Julesz' texton concept[10] and Marr's primal sketch scheme[14]. They argued that we "see" fundamental image elements, called textons or image primitives, and tend to ignore details which are less structured. Most recently the texton and primal sketch concepts become more concrete due to the development of generative models[22, 8].

For example, Fig.2 illustrates how the primal sketch

Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)
1063-6919/04 $20.00 © 2004 **IEEE**



t-1 t t+1

Figure 3: River sequence. The first row contains three consecutive input river frames, the second row contains their corresponding primal sketch – graphs (The circled areas highlight the topological change.), and the third row contains the reconstructed images by diffusion from the sketches.

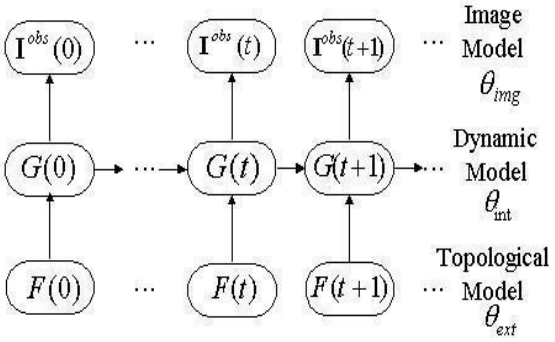


Figure 4: Graph model framework. Observed image sequence $I^{obs}_{[0,\tau]}$ is generated by hidden graph system $G_{[0,\tau]}$. The dynamics of the graph system is caused by internal interactions, which is controlled by parameter θ_{int}. Graph topological changes are caused by external topological operating forces $F_{[0,\tau]}$, which is controlled by parameter θ_{ext}.

model represents a cloth image. Given an input image in (a). The model first extracts the structured parts in (c) which correspond to the places with high image contrasts and changes (about 20% of the pixels). This part is then represented by a graph structure in (b), and (c) can reconstruct the original image with very little loss. To show this, we fill-in the remaining pixels by running a heat-diffusion equation which use the pixels in (c) as boundary conditions. Then we obtain the image in (d). Although (d) is not identical to (a), it captures the essential information.

Following the same model, Fig.3 displays the primal sketches for three frames of a water sequence. This time we show the graph representation explicitly. The primal sketch model remembers image patches about 5-pixel width along the curves in the graphs, and then reconstruct the water sequence from the sketch. As further examples, Figs.7 and 8 illustrates more meaningful structures as subgraph for the noticeable elements.

Therefore we propose a generative model in the context of hidden Markov model (HMM) for the complex motion. Our representation consists of four components: (1) A photometric model using primal sketch[8] which transfers an image into an attribute graph representation. Each vertex of the graph is a scaled and oriented image patch selected from a dictionary. The graph connects and align these patches. (2) A geometric model which characterizes the deformation of the attribute graph. (3) A dynamic model, which specifies the motion dynamics of these vertices (patches) and their interactions in the form of coupled Markov chains. (4) A topological model, which interprets the graph topological changes over time. For example, Fig.3 shows the graph structure change over time (see the circles).

Related work. In the vision literature, there have been two streams studying the complex motion patterns which are closely related to our work.

One stream is focused on modeling motion as a texture phenomenon. For example, *temporal texture* by Szummer and Picard [19] who used a Spatial-Temporal Auto-Regression (STAR) model on pixels. Bar-Joseph *et. al.* [1] extended the 2D texture synthesis work to a tree structured multi-resolution representation, in a similar way to $3D$ volume texture method [21]. The *dynamic texture* work by Soatto *et. al.* [18] studied the motion dynamics explicitly using models and tools from control theory. Fitzgibbon [6] further studied the rigid camera motion in combination with the stochastic motion patterns, so that the motion is registered properly. Wang and Zhu [20] represented image by additive image bases, such as Gabor and Fourier bases. selected from an over-complete image dictionary. But their methods failed on modeling some phenomena, e.g. fire or clothes, for two reasons: (1). fire or cloth images cannot be effectively represented by either Gabor-like bases or Fourier bases. (2). the fire and cloth motion exhibit clear topological changes which are noticeable to human vision. Our work in this paper can be viewed as extensions (generalizations) from these work.

Another stream of work is the HMM models for modeling realistic human motion[3], such as motion texture[12] and style machines [2]. The underlying human articulation is represented by deformable graphs as in our model. But these work obtain the graphs by direct motion capture and also the graphs have fixed number of vertices (markers). While in our representation, the graph is generic, inferred from images, and changes structures over time.

The paper is arranged as follows. In the next section, we introduce four components of this generative representation: photometric model, geometric model, dynamic model and topological model. Then we describe model learning, inference, graph matching with editing, and synthesis. The paper is concluded with a discussion of the model limitations and future work.

2. Generative Graph Representation

Fig.4 illustrates the generative representation in three layers. We assume the underlying system has a varying number of perceptual elements – which are the image patches in the primal sketch. These elements are coupled spatially in a graph structure $G(t), t \in [0, \tau]$. The coupling between adjacent elements will be represented by potential energies. The image sequence $\mathbf{I}^{\text{obs}}(t), t \in [0, \tau]$ is the observables and is generated by $G(t)$ with the primal sketch model (see Figs.2 and 3). The system $G(t)$ is driven by external forces $F(t)$. We consider two types of forces: (1) drifting forces acting on each patch which are Brownian motion; (2) topological operators acting on subgraphs and thus change the graph structure (topology). The system is then specified by three sets of parameters $\Theta = (\theta_{img}, \theta_{int}, \theta_{ext})$ for the three layers as Fig.4 shows.

Therefore we have a joint probability for an image sequence $\mathbf{I}^{obs}[0, \tau]$, the hidden graph representation $G[0, \tau]$ and the external force field $F[0, \tau]$,

$$\begin{aligned} & p(\mathbf{I}^{obs}[0,\tau], G[0,\tau], F[0,\tau]; \Theta) \\ = & \prod_{t=0}^{\tau} p(\mathbf{I}^{obs}(t)|G(t);\theta_{img}) \cdot \prod_{t=0}^{\tau} p(F(t);\theta_{ext}), \\ \cdot & \quad p(G(0)) \cdot \prod_{t=1}^{\tau} p(G(t)|G(t-1), F(t);\theta_{int}) \end{aligned} \quad (1)$$

$p(\mathbf{I}(t)|G(t);\theta_{img})$ is the image model (primal sketch) with θ_{img} being the dictionary of image patches. $p(G(t)|G(t-1), F(t);\theta_{int})$ is the probability model for graph dynamics, and also include the coupling of elements in terms of Gibbs potentials which are expressed by the Gestalt properties in the graph. θ_{int} includes the parameters for the kinetic and potential energies. $p(F(t);\theta_{ext})$ is the probability model for independent drifting force and the probability for the events of graph editing operators to occur over time.

In the following subsections, we will introduce the models in detail.

2.1 Photometric model by primal sketch

The generative model for primal sketch is proposed in [8] for representing natural images. It divides the image lattice Λ into two parts: the "sketchable" part for noticeable intensity changes and the "non-sketchable" part for relatively structureless areas.

$$\Lambda = \Lambda_{sk} \cup \Lambda_{nsk}.$$

For clothes, fire and water images, the sketchable part usually corresponds to pixels around the ridges and valleys (creases)[9, 13]. These pixels are covered by a number of image patches which are vertices in the graph G. These patches comes from a learned dictionary. They are aligned by the graph G and are non-overlapping. The pixels not covered by these patches are considered the non-sketchable area and will be filled in by sampling a texture model[24] which matches local filter histograms in the observed images or simple method[5]. But for fire, water and clothes, we can simply fill the non-sketchable pixels by heat-diffusion. In summary, the model is

$$p(\mathbf{I}|G;\theta_{img}) = p(\mathbf{I}_{\Lambda_{nsk}}|\mathbf{I}_{\Lambda_{sk}}) p(\mathbf{I}_{\Lambda_{sk}}|G;\theta_{img}).$$

We refer to [8] for the detailed formulation and the inference of this graph G from image $\mathbf{I}$.

From Fig.2 and Fig.3, we can see that the sketch representation is not only sparse , but also quite realistic. It is worth mentioning that this image model is not a linear additive model as in [20]. The linear additive model is found to be difficult to capture sharp features as it has to count on the alignment of several image bases to generate sharp boundaries. In contrast, each image patch in the primal sketch model can be very sharp and they are aligned by the graph.

2.2 The graph representation

At each frame t, $G =< V, E >$ is an attribute graph representation with N vertices.

$$V = \{\pi_i = (\ell_i, \alpha_i, \beta_i, \gamma_i), i = 1, 2, ..., N\}$$

Each vertex is an image patch π_i specified by four set of attributes

1. A label ℓ_i indexing the type of the image patch in the dictionary, e.g. ridge, valley, bar, step edge, etc.
2. Image attributes α_i for the contrast of the patch.
3. Geometric transforms β_i for location, orientation and scale (size) of the patch.
4. Each patch has a degree of γ connections : $\gamma = 0$ means it is an isolated patch, $\gamma = 1$ means a terminator, and $\gamma = 2$ means an edge segment.

Fig.5 show a subgraph with a number of patches.
The neighboring structure is specified by the edge set

$$E = \{e = (p, q) : \pi_p, \pi_q \in V\}$$

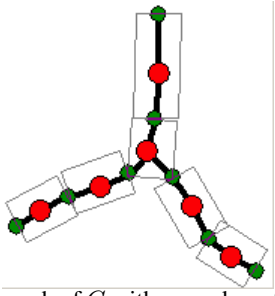


Figure 5: A subgraph of G with a number of image patches.

Then an inhomogeneous Gibbs model is defined on this attribute graph to enforce some Gestalt properties, such smoothness and continuity:

$$p(G) \propto \exp\{-\lambda_o N - \sum_{(p,q)\in E} \psi(\pi_p, \pi_q)\},$$

where λ_o is the parameter that controls the number of primitives N and thus the density, and $\psi_{l_j}(\pi_{p,j}, \pi_{q,j})$ is the potential function of the relationship between two vertices.

Figure 7 shows examples of such as subgraphs for the the fire strokes.

2.3 Graph motion and structure editing

While travelling in spatial-temporal domain, the graphs of fire, water, or cloth evolve with both continuous movement and abrupt structure changes. Our model deals with both.

Dynamic model for continuous graph motion. This part characterizes continuous motion of the image patches caused by three sources

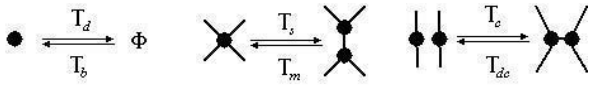
1. The Brownian motion with Gaussian noise.
2. Drifting by inertia. We model this kinetic term $K(G(t), \dot{G}(t), t)$ by an auto-regression (AR) model.
3. Coupling by internal interactions among the adjacent image patches. This is derived from the potentials in the Gibbs energy

$$U_{int}(G) = \sum_{(p,q)\in E} \psi(\pi_p, \pi_q).$$

Topological model for graph editing. Graph topological changes are assumed to be caused by external graph operators

$$S_{\mathcal{T}} = \{\mathcal{T}_\emptyset, \mathcal{T}_d, \mathcal{T}_b, \mathcal{T}_s, \mathcal{T}_m, \mathcal{T}_c, \mathcal{T}_{dc}\}$$

They stands respectively for null operation (no change), death of a vertex, birth of a vertex, split of a vertex, merging of two vertices, connecting two vertices with an edge, disconnecting two vertices.



a) b) c)

Figure 6: Graph operators. a) Death/Birth operators $\mathcal{T}_d/\mathcal{T}_b$. b) Split/Merge operators $\mathcal{T}_s/\mathcal{T}_m$. c) Connect/Dis-connect operators $\mathcal{T}_c/\mathcal{T}_{dc}$.

Fig.6 illustrates the graph topological changes caused by these operators. The occurrences of these graph operating events depend on the neighborhood relationship of subgraphs.

To summarize, the motion model of graphs from t to $t+1$ is denoted as:

$$g_i|G_{\partial i} \xrightarrow{\mathcal{T}} g_i'|G_{\partial i}, \quad \mathcal{T} \in S_{\mathcal{T}}. \tag{2}$$

In the above formula, g_i is a subgraph in G, and $G_{\partial i}$ are the neighbors or environment of g_i. g_i' is the new subgraph after editing.

If the operator $\mathcal{T}_\emptyset$ works alone on the $G(t)$, no topological change will occur. Therefore, the dynamics of each element in the system is reduced to

$$\begin{aligned} \pi_i(t)|G_{\partial i}(t) &= A\pi_i(t-1) + B + d\omega_i(t) \\ d\omega_i(t) &\sim \mathcal{N}(0, |dt|). \end{aligned} \tag{3}$$

where A, B are the AR coefficients, and $\omega_i(t)$ is the Brownian motion of the elements in graph.

If other operators edit the graph, there will be topological changes. The graph motion in Eq.2 is reduced to the following cases corresponding to each graph operator.

$$\begin{aligned} \mathcal{T}_d/\mathcal{T}_b: &\quad \pi_i \rightleftharpoons \phi \\ \mathcal{T}_s/\mathcal{T}_m: &\quad \{\pi_i\} \rightleftharpoons \{(\pi_{i1}, \pi_{i2}), e_{j(i1,i2)}\} \\ \mathcal{T}_c/\mathcal{T}_{dc}: &\quad \{\pi_i, \pi_j\} \rightleftharpoons \{(\pi_i, \pi_j), e_{k(i,j)}\} \end{aligned}$$

In summary, the probability model for the graph motion at each time step t is fully specified by the Brownian motion and the occurrence of the graph operators. The latter are often rare events.

$$\begin{aligned} & p(G(t+1)|G(t), F(t)) \\ = & \prod_{i=1}^{N(t)} \{p(\omega_i(t)) \cdot \prod_{j=1}^{M(t)} p(\mathcal{T}_j(t)|G_{\partial i}(t))\}, \end{aligned} \tag{4}$$

where $M(t)$ is the number of operating events occurred in the time interval $[t, t+1]$. Assuming a time-invariant system, $p(\mathcal{T}_j|G_{\partial i})$ is the operating events occurred under certain given graph configuration, which can be learned over time by accumulation.

Figure 7: A trajectory of an evolving fire stroke. The square boxes on the fire strokes are image patches along the sketches. To the right of each fire stroke image is the symbolic graph. The vertices in the symbolic graphs are control points. The dotted line in the first symbolic graph denotes a link between two subgraphs.

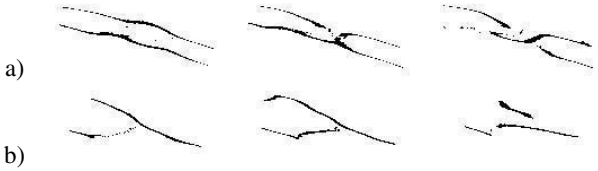


Figure 8: Two learned river vertices trajectories.

3. Learning and Inference

In this section, we briefly study the algorithm that infers the hidden variables $G[0,\tau]$ and learns the parameters $\Theta = (\theta_{int}, \theta_{ext}, \theta_{img})$ in the model. With the learned parameters Θ, one can synthesize sequences following the generative method.

3.1 Problem formulation and stochastic gradient

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

$$\begin{aligned} \Theta^* &= \arg\max \log p(\mathbf{I}^{\text{obs}}[0,\tau];\Theta) \qquad (5) \\ &= \arg\max \log \int p(\mathbf{I}^{\text{obs}}[0,\tau], G[0,\tau];\Theta) dG[0,\tau] \end{aligned}$$

To solve the MLE in the above equation, we set $\frac{\partial \mathcal{L}(\Theta)}{\partial \Theta} = 0$. Thus,

$$\frac{1}{p(\mathbf{I}^{\text{obs}}[0,\tau];\Theta)} \frac{\partial \int p(\mathbf{I}^{\text{obs}}[0,\tau], G[0,\tau];\Theta) dG[0,\tau]}{\partial \Theta} = 0,$$

$$E_{p(G[0,\tau]\,|\,\mathbf{I}^{obs}[0,\tau];\Theta)}[\frac{\partial \log p(\mathbf{I}^{\text{obs}}[0,\tau], G[0,\tau];\Theta)}{\partial \Theta}] = 0.$$

We adopt the stochastic gradient algorithm used in [7] to solve this MLE problem. The learning process iterates in three steps.

1. Sampling $G^{syn}_{[0,\tau]} \sim p(G|\mathbf{I}^{obs};\Theta)$ under the current estimated Θ.

   The sampling procedure is realized by Markov Chain Monte Carlo (MCMC) techniques. It is based on the results computed from bottom-up process introduced in Section 3.2. The MCMC steps are mainly designed to adjust the matching of adjacent graphs, so as to achieve a high posterior probability $p(G[0,\tau]|\mathbf{I}^{obs}[0,\tau])$. We define seven types of MCMC moves as follows.

   (a) Switch the matching correspondence of one vertex in a graph to another vertex in neighbor graph.

   (b) Connect two vertices in the same graph by adding an edge.

   (c) Cut an edge between two vertices in the same graph.

   (d) Split one vertex into two.

   (e) Merge two vertices into one.

   (f) Add a new vertex.

   (g) Delete an existing vertex, together with its edges.

   The graphic illustration of some operations are shown in Fig.6 and Fig.9. Some details of Markov Chain move design will be made available in a technical report.

2. Updating the motion parameters $\theta_{int,ext}$.

$$\theta_{int,ext} \leftarrow (1-\rho)\theta_{int,ext} + \rho \frac{\partial \log p(G(t);\theta_{int,ext})}{\partial \theta_{int,ext}}$$

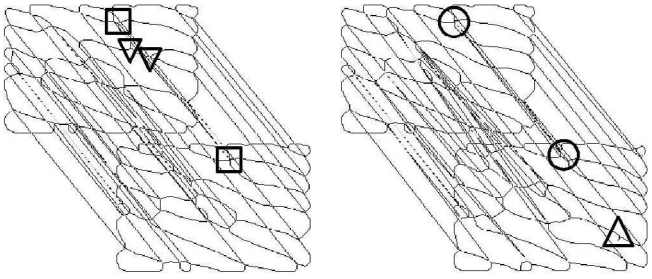


Figure 9: River sequence graph matching. a) Graph matching between frames 1 & 2. b) Graph matching between frames 2 & 3. □ highlights a merge operation, ◯ highlights a split operation, ▽ highlights a death operation, and △ highlights a birth operation.

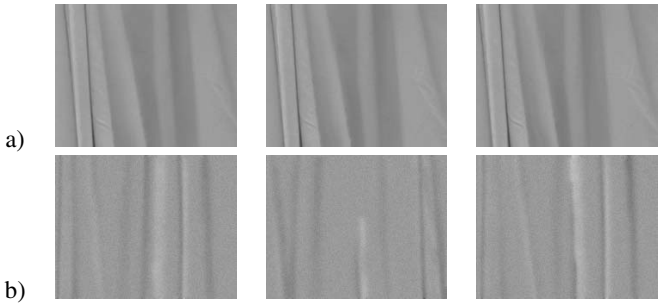


Figure 10: Cloth sequence. a) Input sequence. b) Synthesized sequence.

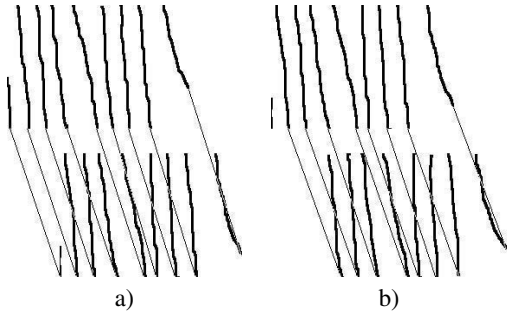


Figure 11: Cloth sequence graph matching. a) Graph matching between frames 10 & 11. b) Graph matching between frames 11 & 12.

Some learned results are shown in Fig.7 - Fig.14. For more details of the automatic learning process, please refer to our technical report.

3.2 Bottom-up graph matching process

Among all the inference steps in the previous subsection, the correct matching of graph from $G(t)$ to $G(t+1)$ deserves most attention. In the paper, we assume the motion is not very large, and thus $G(t)$ to $G(t+1)$ will not have many large structure changes. In this section, we briefly report how we compute the match in a bottom-up approach. This will be used as initial match to feed into the MCMC process above.

In computer vision, edges, ridges and valleys provide rich information for human beings to perceive geometric features of a scene. Firstly, we extract creases from a given image sequence $\mathbf{I}^{obs}[0,\tau]$ using the method in [9]. Then, based on the recent work of Guo *et. al.* [8], we obtain the primal sketch map. On top of that, we build up graphs following the way described in Section 2.

For computational ease, certain number of connected image patches on creases can be grouped into subgraphs, e.g. fire strokes sketches, river ridge curves, river ridge intersections, cloth folds, etc. A subgraph has the following properties.

1. Number of control points $\mathbf{c}$. (The center of each image patch is a control point.)
2. Shape $\mathbf{s}$. A set of these control points connectively define the shape of subgraph.
3. Appearance $\mathbf{a}$. (Pixel intensity of the image patches.)
4. Degree $\mathbf{d}$. (Number of curves in the subgraph.)

In the following, we also use $\mathbf{c}, \mathbf{s}, \mathbf{a}, \mathbf{d}$ as functions on subgraph index i, i.e., each returns the corresponding feature. For example, $\mathbf{c}(\mathbf{i})$ tells the number of image patches in the ith subgraph.

We adopt the graph matching algorithm in [23] and [11]. A matching is obtained by mapping subgraphs in a frame to their similar counterparts in the adjacent frames. Zhu and Yuille defined the similarity between subgraph g_i and g_j as the probability:

$$\begin{aligned} & P_{match}[g_i, g_j] \\ = \; & \frac{1}{Z}\exp\{-\frac{(\mathbf{c}(i)-\mathbf{c}(j))^2}{2\sigma_c^2} - \frac{(\mathbf{s}(i)-\mathbf{s}(j))^2}{2\sigma_s^2} \\ - \; & \frac{(\mathbf{a}(i)-\mathbf{a}(j))^2}{2\sigma_a^2} - \frac{(\mathbf{d}(i)-\mathbf{d}(j))^2}{2\sigma_d^2}\} \end{aligned}$$

where σ's are the variances of these features. This similarity measurement is also used in the inference part to compute the system energy.

It is worth mentioning that each vertex in the graph is a subgraph. They also possess the above properties. The matching procedure also apply to them.

When matching two given graphs $G(t) = (g_i(t), i = 1,\ldots,n)$, and $G(t+1) = (g_i(t+1), i = 1,\ldots,n)$, where n is the larger number of subgraphs in either of the two graphs, it is reasonable to allow some subgraph in $G(t)$ map to null, or multiple subgraphs in $G(t)$ map to the same subgraph in $G(t+1)$, and vice versa. Thus, the similarity between graph $G(t)$ and $G(t+1)$ is defined as the probability:

$$P[G(t), G(t+1)] = \prod_{i=1}^{n} P_{match}[g_i(t), g_i(t+1)]$$

After the graph matching, trajectories of graph elements can be extracted automatically. The graph matching results after MCMC sampling are shown in Fig.9, Fig.11, and Fig.14. A fire stroke trajectory and two river wave vertices are shown in Fig.7 and Fig.8, respectively.

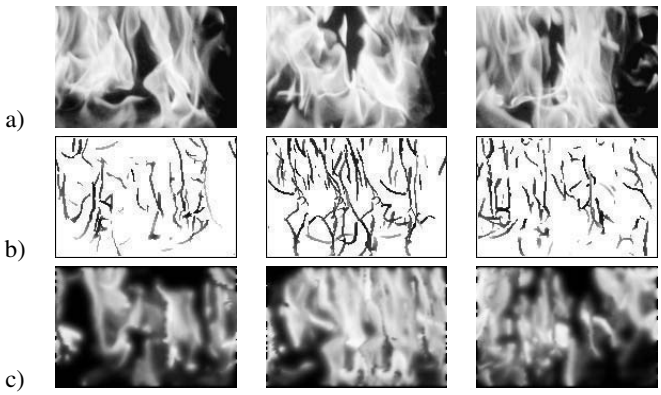


Figure 12: Fire sequence. a) Input sequence. b) Synthesized sketch sequence. c) Diffused synthesized sequence

3.3 An example of graph operations

The graph inference is achieved by carefully designed MCMC sampling algorithm. The essence of the Markov chain design is to form an ergodic process in the space of all possible configurations of a graph. Also, 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 [15]. We design a pair of proposal probabilities for moving from A to B, with $q(A \rightarrow dB) = q(B|A)dB$, and back with $q(B \rightarrow dA) = q(A|B)dA$. The proposed move is accepted with probability

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

Due to the page limit, we only introduce one pair of Markov chain moves – split/merge. (For details of the other operations, please refer to our technical report.) The moves are illustrated in Fig.13 and they are jump processes between two states A and B,

$$\begin{aligned} A &= (N, G =< (V_-, v_j), (E_-, e_{i,j}) >) \\ &\rightleftharpoons (N-1, G' =< V_-, E_- >) = B, \end{aligned}$$

where N is the number of vertices in graph G. V_- and E_- denote the unchanged vertices set and edge set, respectively. $e_{i,j}$ is the edge between vertices v_i and v_j, and v_j is the vertex disappeared after merging. We define the proposal probabilities as follows.

$$\begin{aligned} q(A \rightarrow B) &= q_{s/m} \cdot q_m \cdot q(i) \cdot q(j) \\ q(B \rightarrow A) &= q_{s/m} \cdot q_s \cdot q'(i) \cdot q(pattern). \end{aligned}$$

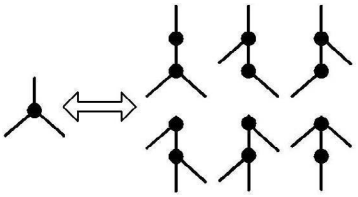


Figure 13: Split/merge graph operation diagram. A vertex can be split into two vertices with one of six edge configurations.

$q_{s/m}$ is the probability for selecting this split/merge move among all possible graph operations. q_m and q_s is the probability to choose either split or merge, respectively, where $q_m + q_s = 1$. $q(i)$ is the probability of selecting v_i as the anchor vertex for the other vertex to merge into, which is usually set to $1/N$. $q(j)$ is the probability to choose v_j from v_i's neighbors, which is set to be inversely proportional to the distance between v_i and v_j. Once v_j is merged into v_i, v_i becomes a symbolic vertex containing two real vertices. When proposing a split move, $q'(i)$ is the probability to choose v_i, which should contain more than one real vertices. It is assumed to be uniform among those qualified vertices. When a vertex with n edges is split, there are $1/(2^n - 2)$ ways for two vertices to share these n edges. Therefore, $q(pattern)$ is set to be $1/(2^n - 2)$.

3.4 Graph synthesis

The well acknowledged verification of the learned model being correct is through synthesis. When synthesizing a new sequence, the following steps are taken.

1. Initiate the first two frames. This is to ensure the AR model in Eq.3 is computable.
2. For the subsequent frames, we iterate the following steps.
   (a) Sample the subgraph $g_i(t+1)$ from $p(\omega_i(t))$ in Eq.4, according to the learned dynamics.
   (b) Sample a set of topological operator $\mathcal{T}_j(t)$ from $p(\mathcal{T}_j(t)|G(t))$ in Eq.4, based on the learned external force field.
   (c) When the synthesized sequence reaches its last frame, stop.

Some more experiment results of the cloth sequence and fire sequence are shown in Fig.10 - Fig.14.

4. Summary and Future Work

The current model and implementation still have some limitations. For example, if there is texture on the surface of

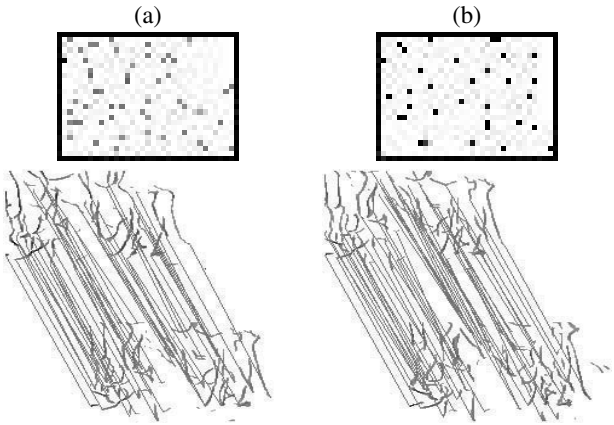


Figure 14: Fire sequence graph matching. (a) Birth map of fire strokes (projected from probability $p(\mathcal{T}_b)$). b) Death map of fire strokes ($p(\mathcal{T}_d)$). c) Graph matching between frames 1 & 2. d) Graph matching between frames 2 & 3.

cloth, the graph structure will be more complicated. Consequently, the computational expense is going to be higher accordingly. Furthermore, although we allow six types of graph editing operators, we assume no combination of these operators. Thus, larger motions with complex topological changes at the same site cannot be computed. We will extend our model to graph morphing in the future. We plan to learn a set of graph operators with probability for perceptually appealing morphing and thus we can define meaningful geodesic distance and metrics between two images based on the "natural" motion.

Acknowledgments

We'd like to thank for the support from research grant NSF-IIS-0244763, NSF-0240148, the MIT motion texture database for the fire sequence, and Jinhui Li for Fig.2.

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 Visualization and Computer Graphics*, (to appear).

[2] M. Brand and A. Hertzmann, "Style Machines", *Siggraph2000*, 183-192, 2000.

[3] C. Bregler and J. Malik, "Learning Appearance Based Models: Mixtures of Second Moment Experts", *Advances in Neural Information Processing Systems*, 9, 845, 1997.

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

[5] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling.",*ICCV, 2:10338*, Sep 1999.

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

[7] M.G. Gu, "A stochastic approximation algorithm with MCMC method for incomplete data estimation problems", *Preprint*, Dept. of Math. and Stat., McGill Univ. 1998.

[8] C.E. Guo, S.C. Zhu and Y.N. Wu, "A Mathematical Theory of Primal Sketch and Sketchability", *ICCV*, Nice, France, 2003.

[9] R. Haralick, "Ridges and Valleys on Digital Images", *CVGIP*, vol 22, no. 10, 28-38, Apr. 1983.

[10] B. Julesz, "Textons, the elements of texture perception and theri interactions", *Nature*, 290:91-97, 1981.

[11] P. Klein, T. Sebastian, and B. Kimia, "Shape matching using edit-distance: an implementation", *SODA*, 781-790, 2001.

[12] Y. Li, T. Wang, H.Y. Shum, "Motion texture: a two-level statistical model for character motion synthesis", *Siggraph2002*, 465-472, 2002.

[13] A.M. Lopez, F. Lumbreras, J. Serrat and J.J. Villanueva, "Evaluation of methods for ridge and valley detection." *PAMI*, 21(4):327–335, April 1999.

[14] D. Marr, *Vision*, W. H. Freeman and Company, 1982.

[15] 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.*

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

[17] A. Schodl, R. Szeliski, D. Salesin, and I. Essa, "Video texture", *Proc. of SIGGRAPH*, 2000.

[18] S. Soatto, G. Doretto, and Y.N. Wu, "Dynamic texture", *ICCV*, Vancouver, Canada, July, 2001.

[19] M. O. Szummer and R. W. Picard, "Temporal texture modeling", *ICIP*, Lausanne, Switzerland, 1996.

[20] Y.Z. Wang and S.C. Zhu, "Modeling Textured Motion: Particle, Wave and Sketch", *ICCV*, Nice, Oct. 2003.

[21] L.Y. Wei and M. Levoy, "Fast texture synthesis using tree structured vector quantization", *Proc. of SIGGRAPH*, 2000.

[22] S.C. Zhu, C. E. Guo, Y. N. Wu, and Y.Z. Wang, "What are Textons?", *ECCV*, Copenhagen, Denmark, 2002.

[23] S. C. Zhu and A. L.Yuille, "FORMS: A Flexible Object Recognition and Modeling System", *IJCV*, Vol.20, No.3, 187-212, 1996.

[24] S. C. Zhu, Y.N. Wu, and D. B. Mumford, "Minimax entropy principle and Its Applications to Texture Modeling", *Neural Computation*, Vol. 9, 1627-1660, Nov. 1997.