A Lattice-Based MRF Model for Dynamic Near-Regular Texture Tracking

Wen-Chieh Lin, *Member*, *IEEE*, and Yanxi Liu, *Senior Member*, *IEEE*

Abstract—A near-regular texture (NRT) is a geometric and photometric deformation from its regular origin—a congruent wallpaper pattern formed by 2D translations of a single tile. A dynamic NRT is an NRT under motion. Although NRTs are pervasive in man-made and natural environments, effective computational algorithms for NRTs are few. This paper addresses specific computational challenges in modeling and tracking dynamic NRTs, including ambiguous correspondences, occlusions, and drastic illumination and appearance variations. We propose a lattice-based Markov-Random-Field (MRF) model for dynamic NRTs in a 3D spatiotemporal space. Our model consists of a global lattice structure that characterizes the topological constraint among multiple textons and an image observation model that handles local geometry and appearance variations. Based on the proposed MRF model, we develop a tracking algorithm that utilizes belief propagation and particle filtering to effectively handle the special challenges of the dynamic NRT tracking without any assumption on the motion types or lighting conditions. We provide quantitative evaluations of the proposed method against existing tracking algorithms and demonstrate its applications in video editing.

Index Terms—Near-regular texture, visual tracking, dynamic near-regular texture tracking, model-based tracking, texture replacement, video editing.

1 Introduction

A near-regular texture (NRT) is a geometric and photometric deformation from its regular origin—a congruent wallpaper pattern formed by 2D translations of a single tile [26]. Dynamic near-regular textures are NRTs under motion. Correspondingly, we define the basic unit of a dynamic NRT *texton*, as a geometrically and photometrically deformed tile, moving through a 3D spatiotemporal space. Fig. 1 shows several sample snapshots of dynamic NRTs: a piece of moving fabric, a wallpaper pattern seen through disturbed water, or even a crowd in motion.

Dynamic NRTs can be viewed as an extension to the conventional dynamic textures, which refer to a sequence of image textures that exhibit certain statistical stationary properties in time [9], [33], [44], such as smoke, fire, or moving water. Different from the conventional dynamic textures, dynamic NRTs possess spatial topological invariance in time, but their motion along the time axis may not exhibit any statistical stationarity. The spatial regularity, including geometry, topology, and appearance brings both new challenges and useful cues for handling dynamic NRTs. While most existing work on dynamic textures addresses analysis, synthesis, or classification problems [2], [3], [5], [9], [22], [38], [44], [46], [49], dynamic NRT modeling, and tracking pose new problems that have not been addressed before.

- *W. Lin is with the Department of Computer Science, National Chiao-Tung University, 1001 Ta-Hsueh Rd., Hsinchu 300, Taiwan. E-mail: wclin@cs.nctu.edu.tw.*
- *Y. Liu is with the Computer Science and Engineering and Electrical Engineering Departments, Pennsylvania State University, University Park, PA 16802. E-mail: yanxi@cse.psu.edu.*

Manuscript received 8 Jan. 2006; revised 28 June 2006; accepted 10 Aug. 2006; published online 18 Jan. 2007.
Recommended for acceptance by S.-C. Zhu.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0011-0106.
Digital Object Identifier no. 10.1109/TPAMI.2007.1053.

The fundamental observation of dynamic NRTs is that, when an NRT is going through motion, its topological structure remains invariant. Therefore, a dynamic NRT can be modeled by a novel Markov-Random-Field (MRF) with a wallpaper-group-based lattice structure. Conventionally, dynamic texture analysis deals with stochastic textures [2], [3], [9], [22], [46], [49]. The problem of tracking the motion of individual textons of a general dynamic texture is ill-defined since there is no consensus on what a texton of a stochastic texture is [53]. In existing work, e.g., [5], [9], [47], a dynamic texture is usually treated as a statistical phenomenon and a statistical model is used to describe the texture's collective motion in the analysis and synthesis process. The texton of a dynamic NRT, on the other hand, is well-defined and characterized precisely based on their topological regularity. Thus, dynamic NRT analysis can be carried out through a computationally feasible process of tracking the motion of individual textons, leading to a complete understanding of the regularity and randomness of a dynamic NRT.

Tracking a dynamic NRT, however, poses new computational challenges: The similar appearance of the textons of an NRT introduces severe ambiguous correspondences (Fig. 1a). Furthermore, the tracking becomes very difficult when the textons of a dynamic NRT move rapidly or occlude each other on a folded surface. Due to these difficulties, tracking a dynamic NRT remained an unsolved problem.

The main contributions of this paper are: 1) proposing a novel and general lattice-based MRF model for dynamic NRTs, 2) developing a tracking algorithm that can effectively handle real-world dynamic NRTs with occlusions, and 3) demonstrating several video editing applications as a result of dynamic NRT tracking, e.g., real-world dynamic NRT replacement and video superimposition.

This paper is organized as follows: We first review related work on dynamic texture analysis and visual tracking (Section 2). We then define the scope of dynamic near-regular textures and discuss the challenges of dynamic NRT tracking

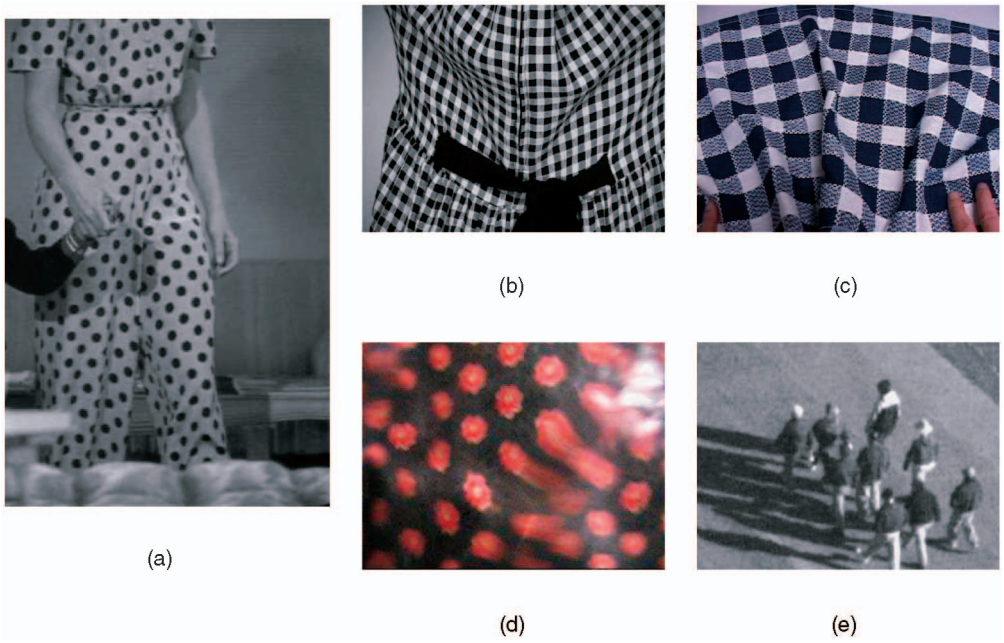


Fig. 1. Examples of dynamic near-regular textures. These images illustrate challenges of dynamic NRT tracking: ambiguous correspondences (a) and (b), occlusions (a), (b) and (c), and appearance and illumination variations (d). The texture in (d) is a pattern seen through disturbed water. The texture in (e) is a dynamic NRT formed by a crowd motion.

(Section 3). We propose a lattice-based MRF model for representing dynamic NRTs (Section 4) and develop a dynamic NRT tracking algorithm based on our MRF model (Section 5). Finally, we demonstrate the effectiveness of our tracking algorithm and its application in video editing (Section 6).

2 Related Work

Dynamic NRT tracking is related to dynamic texture analysis with similar goals: to model and analyze a time-varying texture. Methodology-wise, dynamic NRT tracking is closely related to visual tracking.

2.1 Dynamic Textures Analysis

A dynamic texture or temporal texture [32] is a sequence of texture images in a 3D spatiotemporal volume. Most work in dynamic texture analysis has mainly focused on dynamic stochastic textures [2], [3], [9], [22], [44], [49]. These algorithms assume that the motion is local and statistically stationary in time. Both Szummer and Picard [44] and Doretto et al. [9] used an autoregressive model to model the motion in videos. While the former directly operates on image pixels, the latter represents the input image sequence as a time series of filter response and constructs the model based on this time series. Recently, there have also been research efforts on modeling dynamic textures with structural patterns. Wang and Zhu [48] proposed a generative model to represent complex motion patterns. Their model consists of the geometric, photometric, dynamic, and topological components. MRF is used to model the interactions among image patches in their dynamic model. A difference between Wang and Zhu's model and ours is that they explicitly model the topology changes of a dynamic texture while we utilize the topology invariance property of dynamic NRTs in our MRF model. Doretto [8] extends Active Appearance Models (AAM) to the temporal domain to represent dynamic scenes, which are considered as deviations from a nominal state (a mean warp and a mean template). Doretto's view on modeling dynamic scenes is similar to ours while his approach, variational formulation, differs. A recent survey on dynamic texture analysis can be found in [5].

2.2 Visual Tracking

Our work is related to three types of tracking problems: deformable object tracking, cloth motion capture, and multitarget tracking. Image alignment is used in many deformable object tracking algorithms where different models are applied to confine the deformation space, such as PCA [7], [29], finite element mesh [41], or subdivision surface [16]. These models are not appropriate for tracking textons on a folded surface with occlusion because they assume the surface to be tracked is smooth and nonfolded. Recently, Pilet et al. [35] proposed a real-time nonrigid surface detection algorithm which tracks a nonrigid surface by repeatedly detecting and matching features in an image sequence. Features are detected using a classifier trained on a modal image. Feature matching between the input image and modal image are performed through an optimization process which minimizes the correspondence error and nonsmoothness. They do not handle NRT tracking where repeated patterns cause a serious feature correspondence problem (Section 3).

The goal of cloth motion capture is to capture the 3D motion of cloth. Special calibrated multicamera systems [17], [36], [39], [40], color-coded patterns [17], [40], or controlled lighting [40] are required. The special requirements on hardware and input patterns are used to reduce the tracking difficulties due to ambiguous feature correspondences or occlusion problems. Scholz and Magnor [39] combine optical flow with geometric constraints (distance, curvature, and contour) to track a synthetic motion of textured cloth under a calibrated multicamera setting. Guskov [15] developed an algorithm that can detect and track a black-white square pattern on cloth. His algorithm

does not work on general NRT since only the image features of black-white square pattern are used in the detection and image alignment process. Our tracking algorithm can serve as the front end of a cloth motion capture system where no special purpose color-coded cloth pattern or lighting and camera calibration are required.

Tracking textons of a dynamic NRT can also be considered as a special case of multitarget tracking with varying degrees of spatial constraints among different targets. The main difference is that the topology among targets remains invariant in dynamic NRT tracking but not so in general multitarget tracking. Modeling the spatial relation among tracked targets using an MRF has been applied to ant tracking [21], sports player tracking [52], and hand tracking [43]. In Section 6.5.4, we compare our algorithm with one of the multitarget tracking algorithms (Fig. 17).

Existing algorithms for deformable object tracking, cloth motion capture, or multitarget tracking succeed in their respective domains, but none of them deals with the general NRT tracking problem under various types of motion, viewed through different media (water, air, ...) and occlusion conditions as treated in this paper. By adopting MRF and image alignment into a specially designed, unified framework, our approach can effectively track various types of dynamic NRTs under different motion conditions.

3 Dynamic Near-Regular Textures

Despite various forms of dynamic NRTs (Fig. 1), they have the following common properties:

- **Statistical appearance regularity**. Even though the geometry and the appearance of individual textons in a dynamic NRT may vary, they bear strong similarity among themselves that can be considered as statistical deformations from the same texton.
- **Topological structural invariance**. The topological structure of a dynamic NRT remains invariant during motion, even though its geometry and appearance vary from frame to frame.

According to the spatial connectivity between textons, we can categorize dynamic NRTs into two types. We call a dynamic NRT *tightly coupled* if no gaps exist among neighboring textons of the dynamic NRT. On the other hand, we call a dynamic NRT *loosely coupled* if the neighboring textons of the dynamic NRT are allowed to move with a connected elastic constraint so that there may be a gap or overlap between two neighboring textons. Examples of dynamic NRT with loosely coupled textons include underwater pattern, or crowd motions, such as people in a crowd, a marching band or a parade (Fig. 1). Fig. 2 illustrates the lattice and textons of two types of the dynamic NRTs.

3.1 Challenges of Dynamic NRT Tracking

Tracking dynamic NRTs present new computational challenges, including highly ambiguous texton correspondences (Fig. 1a), drastic temporal variations (Fig. 1d), and occlusions (Figs. 1a, 1b, and 1c).

The ambiguous correspondence problem in NRT tracking is caused by the strong appearance and geometry resemblance of NRT textons. Although textons of an NRT may have different appearance or geometry across a textured region due to variation of the surface geometry and lighting, the appearance of textons within a local region are similar (Fig. 1). This causes a tracking algorithm to mistake one texton for another easily and lose the temporal correspondence of individual textons during tracking process. Spatial aliasing can also introduce ambiguous temporal correspondences. This occurs on a rapidly moving dynamic NRT where the movement of a texton between two frames is larger than the size of a texton (Fig. 1d). The ambiguous correspondence problem of dynamic NRT tracking is challenging because neither motion continuity nor appearance difference can be used as cues to distinguish neighboring textons.

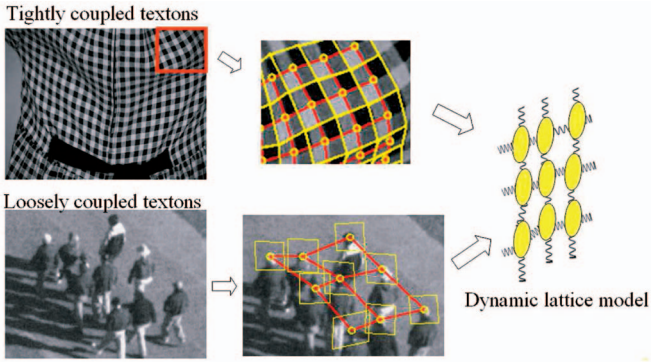


Fig. 2. This figure shows the lattices (red lines) and textons (yellow quadrilaterals) of two types of NRTs: tightly and loosely coupled textons. We model the lattice of an NRT as a 2D MRF where each node represents a texton.

The appearance and geometry of a dynamic NRT may vary dramatically during the course of a motion. For example, if a texton is on a 3D surface, its image intensities and shape change because the lighting condition varies as the surface geometry deforms, such as shading variations or shadowing effects. An extreme case of geometry and appearance variations happens in the dynamic underwater texture, where textons are seen through disturbed water (Fig. 1d). Surface refraction/reflection and motion blur cause the shape and appearance of a texton vary drastically.

There are two types of occlusions in dynamic NRT tracking: self and external occlusion. Fig. 1c illustrates an example of these two problems. External occlusion happens when a texture is occluded by another object. It is easier to overcome the external occlusion problem if the appearance of an external object is substantially different from that of the NRT textons. The most difficult case occurs when an NRT has a self-occlusion. Simply relying on the appearance difference cannot resolve the foreground/background separation because the occluding and occluded textons have similar appearance. We need additional information, such as global structure, local geometric relation, and tracking history to resolve the confusion.

4 Mathematical Model of Dynamic NRTs

The unique properties of NRT create new computational challenges for dynamic NRT tracking. Meanwhile, they provide important and helpful tracking cues. An effective computational model that respects and exploits the properties of the NRT is the key to solve the dynamic NRT tracking problem. We propose a lattice-based MRF model that integrates a high-level topological structure model and a low-level registration-based texton geometry

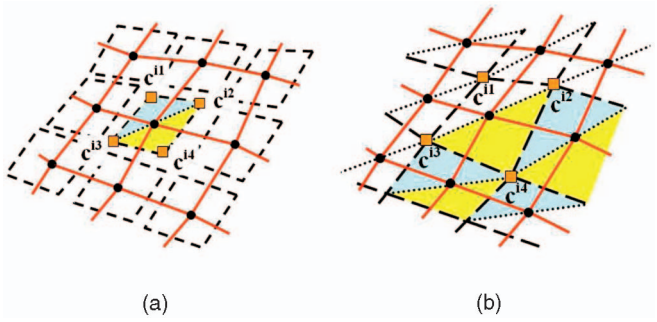


Fig. 3. Texton geometry model. Red lines form the quadrilateral lattice structure of a dynamic NRT. $\mathbf{c}^{i1}, \mathbf{c}^{i2}, \mathbf{c}^{i3}, \mathbf{c}^{i4}$ denote image coordinates of the four vertices of texton i. A texton is divided into two triangles (shaded blue and yellow, respectively) and the vertex coordinates of each triangle parameterize an affine transformation. (a) Loosely coupled textons: The vertex positions of a texton are determined independently. (b) Tightly coupled textons: The vertex positions of a texton are jointly determined by its neighboring textons (e.g., $\mathbf{c}^{i4}$ is involved in four textons and six affine transformations).

and appearance model. Topology invariance and geometry regularity are exploited in the lattice structure model to resolve ambiguous correspondences and occlusion problems. Geometry and appearance regularity are used to detect textons and deal with temporal variations. The proposed model characterizes the lattice topology, geometry, and appearance of individual textons of a dynamic NRT.

Based on wallpaper group and lattice theory, the topology and geometry of an NRT can be described by a quadrilateral lattice L [14], [25]. L is a graph of degree 4 connecting all textons of an NRT. The degree-4 graph lattice topology comes from crystallographic group theory that all wallpaper patterns have a quadrilateral lattice characterizing its fundamental generating region. Therefore, a pair of linearly independent vectors $\mathbf{t}_1$ and $\mathbf{t}_2$ is sufficient to represent the quadrilateral lattice of a regular texture.

4.1 Texton Geometry Model

The texton geometry model defines the local appearance of an NRT. Specifically, the geometry of a quadrilateral texton is represented by two nonindependent affine transformations[1] (Fig. 3). These two affine transformations map image pixels from a rectangular domain $[1, w] \times [1, h]$, an *aligned texton*, to the quadrilateral region of a texton in an image. Each of the two transformations is parameterized by the positions of three vertices of half of a quadrilateral. These affine parameters (i.e., vertex positions) are determined through image alignment processes ((6) and (7)). For loosely coupled textons, affine transformations of each texton are computed independently. For tightly coupled textons, a connected constraint between neighboring textons is enforced via a piecewise affine model, where a shared texton vertex is used to parameterize multiple texton affine transformations (Section 4.2.2). That is, all affine transformations are computed together in a single image alignment process. Fig. 3 illustrates the geometric model of loosely and tightly coupled textons.

1. The reason for choosing two affine transformation over one homograph to represent the geometry of a quadrilateral texton is that an affine transformation needs fewer parameters to specify. This is important since, when a texton is partially occluded, we can still use the top or bottom triangular region to compute the parameters of the affine transformation.

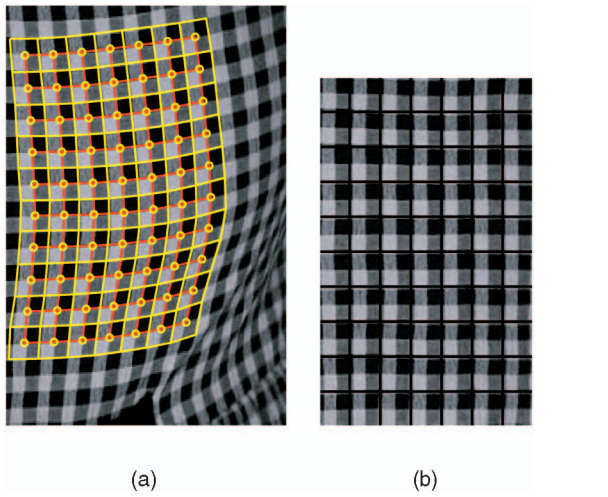


Fig. 4. (a) The first frame of an dynamic NRT where each yellow quadrilateral represents a texton. A lattice is formed by connecting the centers of textons (red lines). (b) Aligned textons from textons in (a). These aligned textons serve as initial texton templates and are updated in successive frames.

In the image alignment process, each texton is registered to a *texton template* which is composed of aligned textons obtained from the tracking initialization process (Section 5.1). Fig. 4b shows an example of aligned textons in the first frame of a video. Each aligned texton in Fig. 4b serves as a texton template for the corresponding texton in the video.

4.2 Lattice-Based MRF Model

The lattice L of a dynamic NRT functions as the topological skeleton of a texture. When an NRT moves, its lattice deforms accordingly, but its topology remains invariant. This resembles the behavior of a network of springs in which each spring controls the mutual distance between two textons locally. The network combines the forces from individual springs to maintain a global spatiotemporal structure. Our observation of the similarity between a lattice structure and a spring network is inspired by the physics-based cloth motion simulation [37], where a spring-damper network is used to model the dynamics of cloth motion. The specific topology of our network, however, is soundly and conveniently based on the mathematical theory of wallpaper groups [11], [14], [25].

We can also view the lattice structure and the relation (or *interaction*) between textons from a statistical viewpoint; the probabilities of the states of textons are locally dependent. The lattice structure provides a well-defined neighborhood structure. The probability of the position of a texton is influenced more by neighboring textons than by distant textons. The shape of a texton has similar properties to its neighbors. These Markov properties make the MRF model a natural candidate to embed the global lattice structure of a dynamic NRT under a statistical framework.

An MRF is an undirected graph $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the set of vertexes and edges in the graph, respectively. Each vertex in the graph corresponds to a random variable. The joint probability of all random variables is factored as a product of local potential functions at each node and the interactions between nodes are defined on neighborhood cliques represented by the connected edges in the graph. The most common form of MRF is a pairwise MRF in which each clique is a pair of connected nodes in the

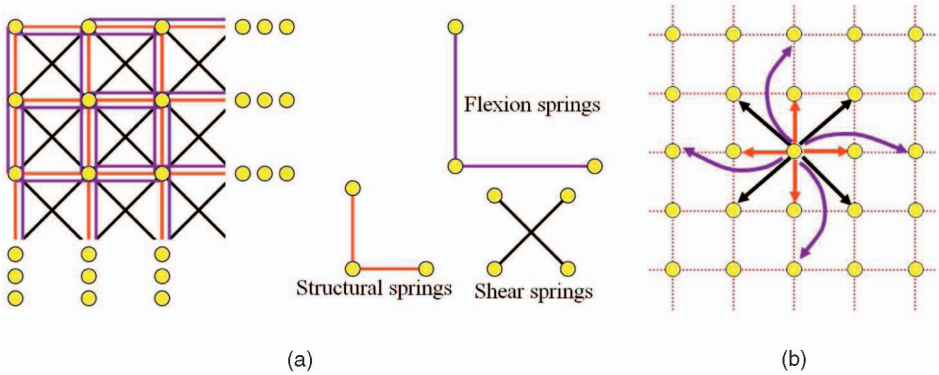


Fig. 5. (a) The cloth spring configurations in [37]. The circles and lines represent mass particles and springs, respectively. There are 1) structural springs (red lines) connecting nearest-neighbor particle along vertical and horizontal lines, 2) shear springs (black lines) connecting a particle's nearest-neighbor particles along diagonals, and 3) flexion springs (purple lines) connecting a particle with its second neighbor along vertical and horizontal lines. (b) Neighborhood configuration of our MRF model: A node is connected to 12 neighbors.

undirected graph. Representing each texton as a node and the probabilistic dependency among textons as edges in the graph, we can model the probability of a lattice configuration of an NRT as follows:

$$p(x^1, x^2, \ldots, x^N, z^1, z^2, \ldots, z^N) \propto \prod_{(i,j)\in\mathcal{E}} \varphi(x^i, x^j) \prod_{i=1}^{N} \phi(x^i, z^i), \tag{1}$$

where x^i and z^i are random variables representing the state and image observation of texton i, respectively. N is the total number of the textons in the NRT. The state of a texton is defined as $x^i = (c^{i1x}, c^{i1y}, c^{i2x}, c^{i2y}, c^{i3x}, c^{i3y}, c^{i4x}, c^{i4y}, v^i)$. The pair (c^{ikx}, c^{iky}) denotes the image coordinates of the kth $(k = 1, 2, 3, 4)$ vertex of the texton and $v^i \in [0, 1]$ represents the visibility score of texton i. The potential function $\varphi(x^i, x^j)$ is defined as a measurement of the spring energy between two connected textons i and j. If the function value is large, the state probabilities of textons x^i and x^j are highly dependent. The image likelihood function $\phi(x^i, z^i)$ associates the probabilistic relation between the state of a texton x^i and its image observation z^i.

The MRF defined in (1) is also called the Markov network [13]. Equation (1) can be represented as a posterior probability:

$$p(X|Z) \propto p(X, Z) \propto \prod_{(i,j)\in\mathcal{E}} \varphi(x^i, x^j) \prod_{i=1}^{N} \phi(x^i, z^i), \tag{2}$$

where $X = (x^1, x^2, \ldots, x^N)$ and $Z = (z^1, z^2, \ldots, z^N)$ are the state of lattice configuration and the image observations of all textons, respectively. $\prod \varphi(x^i, x^j)$ is a lattice structure model and $\prod \phi(x^i, z^i)$ is an image observation model. The lattice structure model captures the global structure of a lattice and resolves ambiguous correspondences using the topological relation between neighboring textons. The image observation model integrates a texton geometry model (Section 4.1) which aligns textons by minimizing image differences to handle local deformation of individual textons.

Although the MRF model has been applied to texture analysis and texture synthesis in the past, they are usually used for low-level processing, such as modeling the probabilities of pixel intensities [23]. The effective combination of a global lattice structure with MRF in this work enables us to capture the innate property of a dynamic NRT: globally and topologically regular while locally and appearance-wise (geometric and photometric) random.

4.2.1 Lattice Structure Model

The potential function in our MRF is defined as follows:

$$\varphi(x^i, x^j) = e^{-\beta \cdot d_g(x^i, x^j)}, \tag{3}$$

$$d_g(x^i, x^j) = (\|\mathbf{c}_m^i - \mathbf{c}_m^j\| - l^{ij})^2 \cdot v^i v^j, \tag{4}$$

where β is a global weighting scalar that is applied to all springs. β weights the influence of the lattice model versus the observation model in the Markov network ($\beta = 2$ is used in our experiments). d_g is a function that measures the geometric deformation (spring energy function). $\mathbf{c}_m^i \in \mathbb{R}^{2\times 1}$ is the mean position of four vertices of the texton i. This potential function acts like a spring that adjusts the position of textons based on their mutual distance. The rest length l^{ij} of the spring is spatially dependent. To handle occlusion, v^i and v^j in (4) are used to weigh the influence of a node by their visibility status.

The topology of the graph $(\mathcal{V}, \mathcal{E})$ defines how textons are related to each other in the lattice structure model. To model the global constraints and local variations of the probabilities of the states of textons properly, we use an analogy between an MRF and a network of springs. If graph $(\mathcal{V}, \mathcal{E})$ is a complete graph, each texton is connected with all other textons. Thus, the motion of a texton would be directly affected by all textons leading to an overly constrained lattice model. On the other hand, if there are no edges in the graph, the global structure among textons reduces to isolated unconstraint textons.

We adopt a lattice structure topology similar to the spring connection configuration used in [6], [37] (Fig. 5a). It has been shown in cloth simulation that the 12-neighbor configuration provides a good balance between structural constraint and local deformations. We can convert this spring configuration into the topology of a graph. According to this graph, we can therefore define the neighborhood configuration of the MRF where the state of a node depends on the states of its 12 neighbors. Fig. 5b shows the neighborhood configuration of our MRF. Additional experiments validate that the 12-neighbor configuration is appropriate for our application (Section 6.5.1).

4.2.2 Image Observation Model

We define the image likelihood as follows:

$$\phi(x^i, z^i) \propto e^{-\frac{1}{v^i} d_a(x^i, z^i, T^i)}, \tag{5}$$

where the appearance difference function d_a is weighted by the visibility score v^i of a texton so that visible textons contribute more in the likelihood function. $d_a = \sum_{r=1}^{2} \sum_{\mathbf{p}} \|z^{ir}(\mathbf{p}) - T^i(\mathbf{p})\|^2$ is the sum of squared differences (SSD) between the observed texton and a texton template T^i. r denotes top or bottom triangle of a quadrilateral texton. $z^{ir} = I(\mathbf{W}(\mathbf{p}; \tilde{\mathbf{a}}^{ir}))$ is an aligned texton obtained from the affine warp $\mathbf{W}$. $\mathbf{p}$ denotes a pixel location in the coordinate frame of the template. The parameters of the affine warp $\tilde{\mathbf{a}}^{ir}$ are computed using the Lucas-Kanade algorithm. In this image alignment process, each quadrilateral texton is represented as the combination of two triangles and the vertex coordinates of each triangle are used to parameterize $\mathbf{a}^{i1}$ and $\mathbf{a}^{i2}$, respectively. For example, in Fig. 3, $\mathbf{a}^{i1} = (c^{i1x}, c^{i1y}, c^{i2x}, c^{i2y}, c^{i3x}, c^{i3y})$ and $\mathbf{a}^{i2} = (c^{i2x}, c^{i2y}, c^{i3x}, c^{i3y}, c^{i4x} c^{i4y})$.

For loosely coupled textons, the affine parameters for each texton are computed independently. This allows the observation model to handle more flexible motion, such as underwater texture or people in a crowd. The optimized vertex coordinate $\tilde{\mathbf{a}}^{ir}$ is obtained from:

$$\begin{aligned} \widetilde{\mathbf{a}}^i = argmin & \sum_{\mathbf{p}} v^i \cdot [T^i(\mathbf{p}) - I(\mathbf{W}(\mathbf{p}; \mathbf{a^{i1}}))]^2 \\ & + \sum_{\mathbf{p}} v^i \cdot [T^i(\mathbf{p}) - I(\mathbf{W}(\mathbf{p}; \mathbf{a^{i2}}))]^2, \end{aligned} \tag{6}$$

where $\mathbf{a}^i = \mathbf{a}^{i1} \cup \mathbf{a}^{i2}$.

For tightly coupled textons, the textured region is modeled as a piecewise affine warp and the position of each texton vertex is affected by at most four neighboring textons. $\tilde{\mathbf{a}}^{ir}$ is computed as follows:

$$\begin{aligned} \widetilde{\mathbf{A}} = argmin & \sum_{i} \sum_{\mathbf{p}} v^i \cdot [T^i(\mathbf{p}) - I(\mathbf{W}(\mathbf{p}; \mathbf{a^{i1}}))]^2 \\ & + \sum_{i} \sum_{\mathbf{p}} v^i \cdot [T^i(\mathbf{p}) - I(\mathbf{W}(\mathbf{p}; \mathbf{a^{i2}}))]^2, \end{aligned} \tag{7}$$

where $\mathbf{A} = (c^{i1x}, c^{i1y}, \cdots, c^{ikx}, c^{iky}, \cdots, c^{iNx}, c^{iNy})$ contains all texton vertex coordinates and $\widetilde{\mathbf{A}}$ denotes optimized vertex coordinates. $\mathbf{a}^{i1}$ and $\mathbf{a}^{i2}$ follow the same definition in (6). Note that the coordinates of a vertex (c^{ikx}, c^{iky}) are usually determined by the image alignment of neighboring connected textons (at most four). This enforces the hard connected constraints among textons when computing $\tilde{\mathbf{a}}^{ir}$. For details about this image alignment process, please see the appendix in [24].

4.2.3 Visibility Computation

The visibility of a texton is determined by constraints and measurements related to the geometry and appearance of a texton. The constraints, which include topology, side length, and area difference with its neighboring textons, are used to decide whether a texton is visible and can be included in the tracking process. The visibility score v^i of a valid texton i is defined as

$$v^i = \frac{1}{1+\rho} \left(\frac{s^i}{s^*} + \frac{\rho}{4} \sum_{k=1}^{4} \left| 1 - \frac{|b_k^i - b_k^*|}{b_k^*} \right| \right). \tag{8}$$

Note that $0 \leq v^i \leq 1$ and ρ is a constant to weigh the influence of area and side length variations. s^i and s^* are the area of texton i and the initial texton (see Fig. 8a and Section 5.1). b_k^i and b_k^* are the kth side length of texton i and the initial texton. A visibility map V is constructed based on the visibility scores of all textons:

$$V = \{round(v^i), i = 1, \ldots, N\}. \tag{9}$$

4.2.4 Temporal Lattice-Based MRF Model

So far, the lattice-based MRF model provides the probabilities of lattice configurations at a single time instance. To incorporate the temporal variations of the MRF model for dynamic NRTs, we formulate the temporal lattice-based MRF model of a dynamic NRT at frame t as:

$$p(X_t|Z_t) \propto \prod_{(i,j)\in\mathcal{E}} \varphi(x_t^i, x_t^j) \prod_{i=1}^{N} \phi(x_t^i, z_t^i). \tag{10}$$

The potential function $\varphi(x_t^i, x_t^j)$ and image likelihood function $\phi(x_t^i, z_t^i)$ in (10) are defined, respectively, as

$$\varphi(x_t^i, x_t^j) = e^{-\beta \cdot d_g(x_t^i, x_t^j)}, \tag{11}$$

$$d_g(x_t^i, x_t^j) = (\|\mathbf{c}_m^i - \mathbf{c}_m^j\| - l_t^{ij})^2 \cdot v_t^i v_t^j, \tag{12}$$

$$\phi(x_t^i, z_t^i) \propto e^{-\frac{1}{v_t^i} d_a(x_t^i, z_t^i, T_t^i)}, \tag{13}$$

where all notations follow the same definitions given in (2)-(5) but with an additional time index. Fig. 6 illustrates our temporal lattice-based MRF model.

To handle the temporal variation of the lattice structure, the rest length l_t^{ij} of the spring becomes not only spatially dependent (on the vertex indexes) but also temporally adaptive (time-variant spring rest length). We use an exponentially decaying function to model the temporal variation of the rest length of springs

$$l_t^{ij} = \frac{\sum_{f=1}^{\infty} l_{t-f}^{ij} e^{-\gamma f}}{\sum_{f=1}^{\infty} e^{-\gamma f}}, \tag{14}$$

where f is a frame index for previous frames involved in the weighted average. $\gamma > 0$ is a parameter that controls how fast the exponential function decays, i.e., γ determines the weights of the spring length of previous frames in the weighted average. $\gamma = 0.2$ is used in our experiments.

5 Dynamic NRT Tracking Algorithm

Our dynamic NRT tracking algorithm consists of four components:

1. **texton detection**,
2. **spatial inference**,
3. **temporal tracking**, and
4. **template update**.

Fig. 7 shows an overview of our algorithm. In the **initialization stage**, the texton detection algorithm finds all textons in the first frame based on a single texton. All detected textons

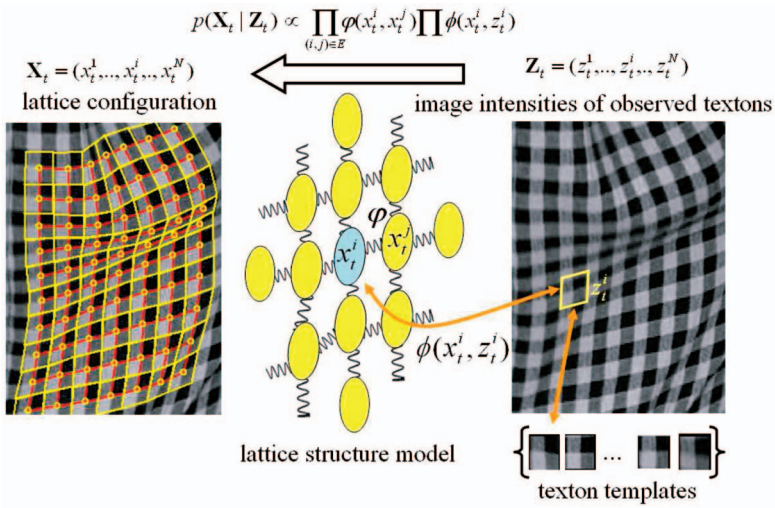


Fig. 6. Illustration of our temporal lattice-based MRF model. Our model consists of 1) a global lattice structure that characterizes the topological constraint among multiple textons and 2) an image observation model that handles local geometry and appearance variations.

are then geometrically aligned. We call these aligned textons from the first frame *texton templates*. A quadrilateral lattice is constructed by connecting the centers of detected textons. In the **tracking stage**, texton detection is performed at each frame to include any additional texton entering the scene and removing the ones leaving the scene. We handle the texton tracking problem through a statistical inference process consisting of spatial inference and temporal tracking. The states of a texton (position, shape, and visibility) are sampled and its distribution is modeled by a particle filter in the tracking process. In each frame, a set of sampled states is drawn and a dynamic model generates the predicted states for the next frame. Belief propagation (BP) [13], [34], [51] is then applied to these predicted states to find the most likely lattice configuration based on the lattice structure model and image data. BP also provides the probability of each texton state, which is used to refine the approximation of the distribution of texton states through particle filtering. The above process iterates until the whole image sequence is tracked. In addition, the texton template set is updated to handle the variation of texton image intensities during tracking.

5.1 Tracking Initialization and Texton Detection

The appearance and geometry regularity of an NRT can be used for automatic texton detection. We consider an NRT as being formed by translating a texton on a plane where the shape and image intensities of the texton may vary. The process of texton detection can thus be viewed as a tracking problem on this plane, more precisely, a *spatial tracking* problem. That is, each texton can be treated as a target to be tracked and the trajectories of all texton centers form a lattice. We "grow" the lattice from regions where textons are more regular and reliable by propagating the lattice spatially outward to regions where textons are distorted or occluded.

In the initialization stage, the user identifies a single texton in the first image frame by specifying two vectors $\mathbf{t}_1$ and $\mathbf{t}_2$ (three points) that form a parallelogram (Fig. 8a). A texton template T is constructed by transforming the parallelogram region in the image to a rectangular region $[1, w] \times [1, h]$, where $w = length(\mathbf{t}_1)$, $h = length(\mathbf{t}_2)$, and the affine transformation matrix $\mathbf{A}_1$ is parameterized by the image coordinates of texton vertices (c^{1x}, c^{1y}), (c^{2x}, c^{2y}), (c^{3x}, c^{3y}), (c^{4x}, c^{4y}),

$$\mathbf{A}_1 = \begin{bmatrix} c^{1x} & c^{3x} & \frac{c^{2x}+c^{4x}}{2} \\ c^{1y} & c^{3y} & \frac{c^{2y}+c^{4y}}{2} \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & w \\ 1 & h & \frac{h+1}{2} \\ 1 & 1 & 1 \end{bmatrix}^{-1}. \quad (15)$$

Once the first texton is identified, the second, third, and fourth texton, are obtained by translating the first texton by $\mathbf{t}_1$, $-\mathbf{t}_1$, and $\mathbf{t}_2$. Using the first four textons as the basis for the initial lattice, the lattice grows by repeating the *spatial prediction* and the *validation* steps below.

In the spatial prediction step, the vertices of a texton are estimated from existing textons. The varying geometry regularity of an NRT is utilized to predict the shape and

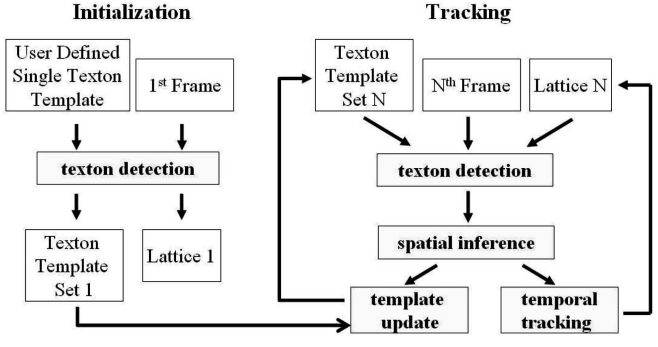


Fig. 7. An overview of our dynamic NRT tracking algorithm.

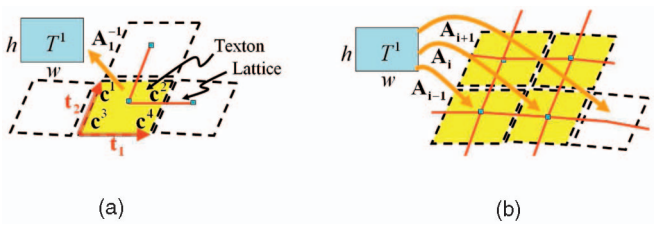


Fig. 8. (a) Initial texton (yellow parallelogram formed by $\mathbf{t}_1$ and $\mathbf{t}_2$) and lattice (red lines). The neighboring textons are estimated by translating the first texton by $\mathbf{t}_1$, $-\mathbf{t}_1$, and $\mathbf{t}_2$. (b) Spatial prediction of the position of a new texton.

position of a new texton. Assuming the geometric variations of textons are smooth locally, the shape and position of a new texton can be estimated by linearly extrapolating from the affine parameters of the neighboring textons. Let $\mathbf{A}_i$ be the affine transformation matrix that maps pixels of the texton template T^i to the texton i in the image. If textons $i-1$, i, and $i+1$ are on the same lattice row or column and $\mathbf{A}_{i-1}$ and $\mathbf{A}_i$ are known, $\mathbf{A}_{i+1}$ is predicted by $\mathbf{A}_{i+1} = (\mathbf{A}_i \cdot \mathbf{A}_{i-1}^{-1}) \cdot \mathbf{A}_i$ [24].

In the validation step, we verify if the predicted texton is valid by checking its associative topology constraints and its area and side length difference with the neighboring textons. Additionally, the vertex positions of all valid textons are refined through an image alignment process where a global optimization that involves the whole lattice is performed ((6) and (7)). The texton geometry model (Section 4.1) is used to confine the transformation space in the image alignment process by computing the affine parameters of all detected textons simultaneously. This makes the image alignment process more robust since it is a global adjustment.

The spatial prediction and validation steps are repeated until no new texton is detected. A texton template set $\mathcal{T}_1 = \{T_1^i\}_{i=1}^N$ is constructed by collecting all valid texton template T_t^i, where $\mathcal{T}_t$ denotes the template set at frame t. The initial configuration of lattice is obtained by connecting all the centers of textons.

5.2 Spatial Inference

The temporal lattice-based MRF model (10) describes the posterior probabilities of the lattice configuration of a dynamic NRT given an image observation of the NRT at frame t. Solving (10) can be considered as a spatial inference problem where the most likely configuration of the lattice is inferred from the image observation of an NRT and the lattice structure model. We can apply the belief propagation algorithm (BP) [13], [34], [51] to solve this inference problem.

BP is an iterative algorithm for computing marginal distributions of random variables on a graphical model, such as MRF, Bayesian network, and factor graph. The BP algorithm introduces variables such as $m_{ij}(x_t^j)$ to propagate the marginal distribution among nodes. $m_{ij}(x_t^j)$ is a vector of the same dimensionality as x_t^j. Intuitively, $m_{ij}(x_t^j)$ can be interpreted as a *message* passing from hidden node i to hidden node j about what state node j should be. Messages are computed iteratively using the following update rule:

$$m_{ij}(x_t^j) \leftarrow \sum_{x_t^i} \phi(x_t^i, z_t^i)\varphi(x_t^i, x_t^j) \prod_{k \in \mathcal{N}(i)\backslash j} m_{ki}(x_t^i), \qquad (16)$$

where $\mathcal{N}(i)$ denotes the neighbors of nodes i.

The marginal distribution of x_t^i, which is called the *belief* at node i, is proportional to the product of the local evidence at that node ($\phi(x_t^i, z_t^i)$), and all messages coming into node i. By iteratively computing (16), the marginal distribution at each node can be obtained using

$$p(x_t^i) = \frac{1}{Q}\phi(x_t^i, z_t^i) \prod_{j \in \mathcal{N}(i)\backslash j} m_{ji}(x_t^i), \qquad (17)$$

where Q is a constant for normalization. It has been shown that BP converges to an exact inference solution if the graph is a tree structure [34] and an approximated inference solution if the graph contains loops [50]. For more details about belief propagation, please see [51].

Since the conventional BP algorithm works on discrete variables while the configuration of a lattice is described by continuous variables, we need to either discretize the state variables or apply continuous BP algorithms [18], [42], i.e., the message function in (16) becomes

$$m_{ij}(x_t^j) \leftarrow \int \phi(x_t^i, z_t^i)\varphi(x_t^i, x_t^j) \prod_{k \in \mathcal{N}(i)\backslash j} m_{ki}(x_t^i) dx_t^i. \qquad (18)$$

The integral in (18) is computationally expensive. For computational efficiency, we choose to use the discrete BP and adopt the sample-based statistics to represent the continuous state variables for each texton. Particle filtering [10], [19] is applied to update the particle set for each texton in the temporal tracking process.

5.3 Temporal Tracking

The spatial inference results provide the probabilities of lattice configurations at a single time instance. To track a lattice, these probabilities need to be propagated temporally. This poses the temporal tracking problem as a sequential inference problem. A general approach to handling non-Gaussian and nonlinear probability distributions in sequential inference is particle filters [10]. Particle filtering is flexible in that it does not require any assumptions about the probability distributions of the data. It approximates the posteriori distribution by a set of particles where each particle is weighted by an observation likelihood and is propagated according to a dynamic model. We therefore apply particle filtering to solve the temporal tracking problem.

The distribution of the lattice configurations in 3D spatiotemporal space are represented and maintained by particle filtering in temporal tracking. The belief distribution computed by the BP is used in importance sampling to draw new samples. The dynamic model is then applied to predict a set of states for each texton and the discrete BP is applied to infer the most likely configuration based on these predicted states.

We use a second-order dynamic model, i.e., the current state of the lattice depends on the previous two states:

$$p(X_t|X_{t-1}, X_{t-2}) \propto \prod p(x_t^i|x_{t-1}^i, x_{t-2}^i), \qquad (19)$$

where a constant velocity model with Gaussian noise is used for the dynamic model for each texton:

$$p(x_t^i|x_{t-1}^i, x_{t-2}^i) = \mathcal{N}(x_t^i - 2x_{t-1}^i + x_{t-2}^i; 0, \Lambda_i). \qquad (20)$$

Λ_i is a diagonal matrix whose diagonal terms correspond to the variance of the state at different dimensions. Although we assume that the movement of each texton is independent in our dynamic model, it does not imply that there is no constraint among textons. Instead, the constraint for maintaining the topological structure of a dynamic NRT is already enforced through the piecewise affine alignment process (7). For more details about particle filtering, please see [10], [19] and our technical report [24].

Our approach of combining BP and particle filter is similar to PAMPAS [18] in spirit; however, PAMPAS incorporates particle filter in the message propagation process while we use particle filter to carry the texton states between image frames. Specifically, PAMPAS adopts particle filter to compute (18) for message propagation but we use (16) for message propagation in a discrete BP. Guskov et al. [17] also

used the Markov network to associate color-coded quadrilaterals in an image with the quadrilaterals of the surface model. They did not use the Markov network to infer the position and the shape of the textons.

5.4 Template Update

Since the appearance of textons vary during tracking process, it is necessary to update the texton template set. We adopt the template updating algorithm in [30], where the basic idea is to correct the drift in each frame by additionally aligning the current image with the template at the first frame using the Lucas-Kanade algorithm. Let T_t and I_t be the texton template and the image at the current frame t, respectively.[2] The warping parameter $\mathbf{a}_t$ computed from the first alignment process is

$$\mathbf{a}_t = argmin_{\mathbf{a}=\mathbf{a}_{t-1}} \sum_{\mathbf{p}} [T_t(\mathbf{p}) - I_t(\mathbf{W}(\mathbf{p};\mathbf{a}))]^2. \quad (21)$$

After aligning the current image with the previous frame, the computed warping parameters are used as the initial values in the additional alignment process to correct any drift:

$$\widetilde{\mathbf{a}}_t = argmin_{\mathbf{a}=\mathbf{a}_t} \sum_{\mathbf{p}} [T_1(\mathbf{p}) - I_t(\mathbf{W}(\mathbf{p};\mathbf{a}))]^2. \quad (22)$$

If the difference of the warping parameters $|\widetilde{\mathbf{a}}_t - \mathbf{a}_t|$ is less than a threshold, the template is updated, $T_{t+1} = I_t(\mathbf{W}(\mathbf{p};\widetilde{\mathbf{a}}_t))$; otherwise, the template remains unchanged. This updating strategy prevents our tracking algorithm from being distracted by outliers while maintaining the flexibility to handle the appearance variations of textons during tracking.

6 Experimental Results

6.1 Texton Detection

Our texton detection algorithm (Section 5.1) can be used to initialize tracking as well as to extract the lattice of a static NRT [26] with minimal user input. Fig. 9 shows four texton detection results. In the dress example, there are large deformations on the left and right boundaries. These results show that our detection algorithm can detect textons on flat or deformed surfaces with perspective distortions.

6.2 Tracking Dynamic NRTs without Occlusion

We test our tracking algorithm on several dynamic NRTs (Fig. 10) under different types of motions and compare it against the robust optical flow algorithm [4] and the Lucas-Kanade algorithm [1], [28]. These two algorithms are chosen as baseline comparisons because they are popular general purpose tracking algorithms. We try different values of the regularization term in the robust optical flow algorithm and find that the value of 2.5 achieves the best tracking performance for the examples in Fig. 10. For the Lucas-Kanade algorithm, 2D affine transformation is used in the tracking process. Due to the ambiguous correspondence challenge of NRT textons (discussed in Section 3), both the robust optical flow algorithm and the Lucas-Kanade algorithm are distracted by neighboring textons. The patterns viewed through disturbed water (Figs. 10d, 10e, 10f, 10g, 10h, and 10i) vary rapidly because of the water surface refraction and motion blur. Despite these difficulties, our algorithm is able to track these highly dynamic and varied textons successfully. These

2. The superscript i is omitted for illustration simplicity.

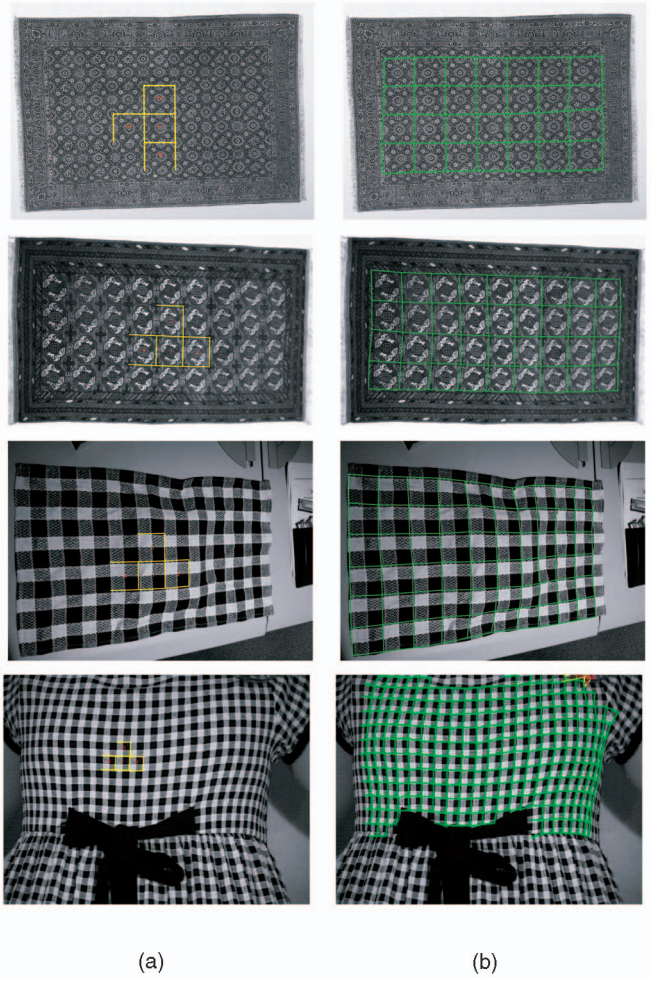


Fig. 9. Texton detection results by spatial tracking. (a) Initial textons. (b) Final result. For videos, please see http://www.cs.cmu.edu/~wclin/dnrtPAMI/dnrt.html.

experiments demonstrate that, even without occlusion, dynamic NRT tracking can be challenging to the robust optical flow algorithm and the Lucas-Kanade algorithm.

The textons of the underwater texture are modeled as a loosely coupled MRF allowing flexible motion of textons. Figs. 10j, 10k, and 10l shows another example of tracking loosely coupled textons. In this example, a texton is defined as a local patch around the head region of a person. The marching motion presents a relatively large global motion with small local deformation of individual textons compared to the motion of tightly coupled textons. Also, the appearance of textons varies drastically due to shadows. The underwater texture and crowd marching examples show that our algorithm is able to handle large illumination changes, rapid geometric deformation, and intensity variations in the tracking process.

6.3 Tracking Dynamic NRTs with Occlusion

Occlusion is one of the major challenges in dynamic texture tracking. Textons may leave/enter the scene or be occluded by other objects or other textons on a folded surface. We test our tracking algorithm on different cloth motion under different degrees of occlusions.

For a checkerboard pattern on a T-shirt, our tracking algorithm achieves similar performance as by Guskov's algorithm [15]; however, Guskov's algorithm is specially

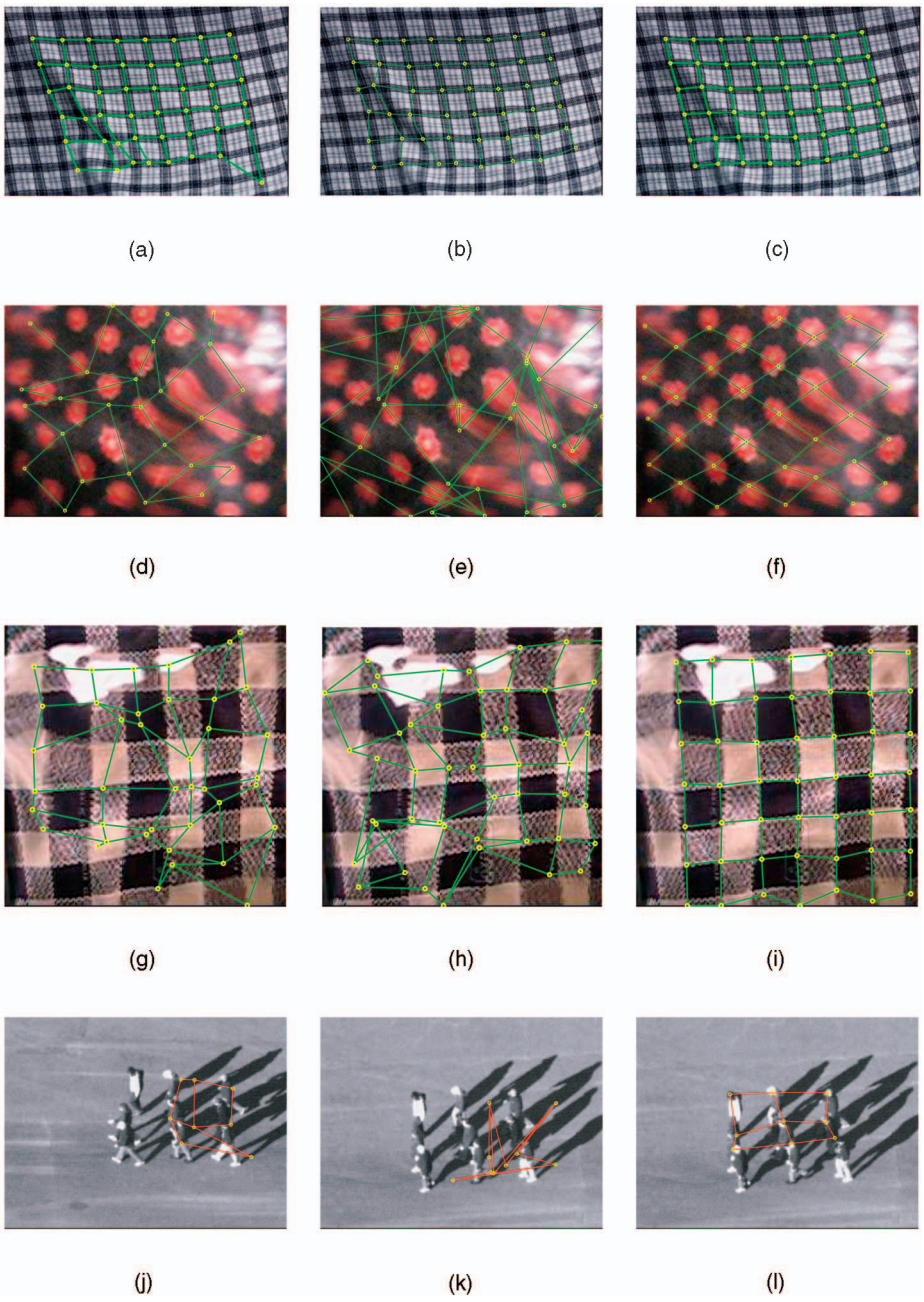


Fig. 10. Tracking results of different algorithms on dynamic NRTs without occlusion. Left column: robust optical flow. Middle column: Lucas-Kanade. Right column: Our results. For videos, please see http://www.cs.cmu.edu/~wclin/dnrtPAMI/dnrt.html. (a), (b), and (c) Slowly waving cloth (frame 86). (d), (e), and (f) An underwater texture seen through disturbed water (frame 91). (g), (h), and (i) Another underwater texture (frame 70). (j), (k), and (l) Crowd motion (frame 210). One can observe that the tracking error accumulates quickly in the optical flow and Lucas-Kanade results. Note that there are serious motion blurs and large lighting variations due to reflection highlights in underwater textures ((d), (e), (f), (g), (h), and (i)).

designed for real-time tracking of black-white checkerboard patterns, while our algorithm can handle general dynamic NRTs. The root mean square error of Guskov's result and ours against hand-labeled ground truth are 2.94 and 2.57 pixels, respectively. The videos of Guskov's and our tracking results can be seen in http://www.cs.cmu.edu/~wclin/dnrtPAMI/dnrt.html.

Fig. 11a shows our tracking result on a fabric pattern under self-occlusion and textons leaving/entering the scene during tracking. The lattice, visible textons, and occluded textons are shown in red, yellow, and cyan colors, respectively. Fig. 11b shows the straightened-out visibility map of textons where blackened regions correspond to detected occluded textons and all visible textons.

Fig. 12 shows another tracking result where a fabric pattern is being folded. There are a few textons totally occluded in the middle and two occluded by a finger in the bottom-right region (Fig. 12b). Despite self and external occlusions in the video, our algorithm can successfully track this folding fabric pattern. When the texton is at the boundary of a lattice, the BP inference result for the texton is less reliable since it receives messages from fewer neighboring nodes. This is the reason why there are some tracking errors in the cyan lattice at the boundary, e.g, top-middle in Fig. 12c.

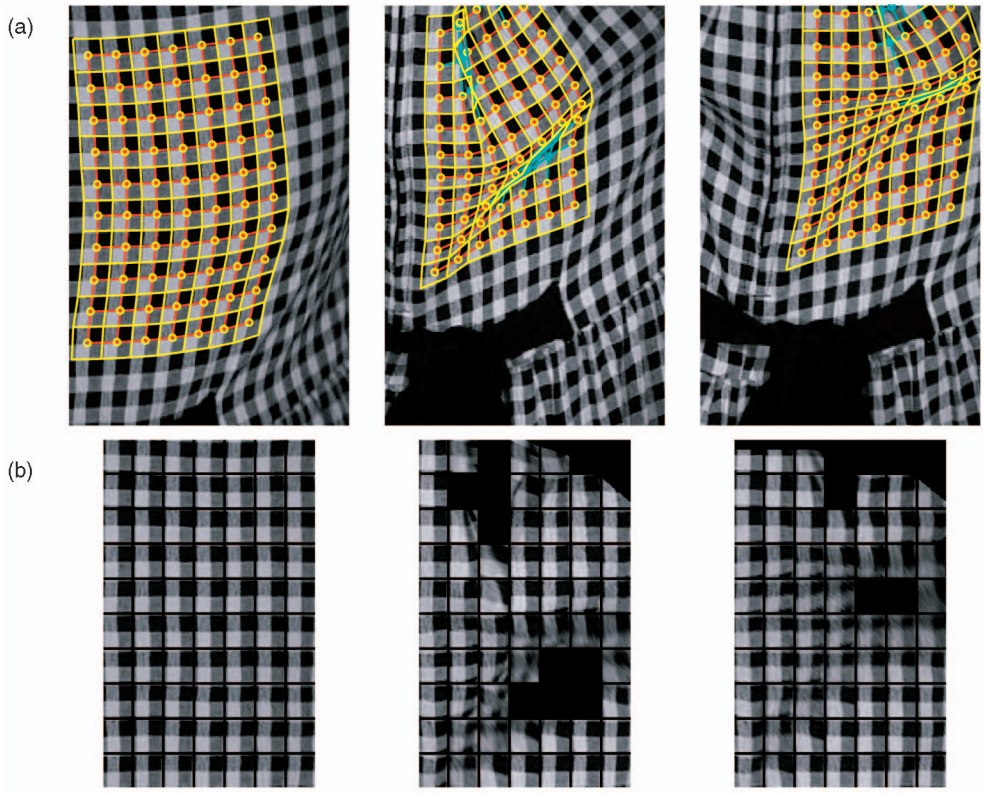


Fig. 11. Tracking results of a fabric pattern under occlusion. (a) Tracked lattice. (b) Visibility map. The visible lattice, occluded lattice, visible textons, and occluded textons are shown in red, cyan, yellow, and cyan color. The visibility map shows visible aligned textons. For videos, please see http://www.cs.cmu.edu/~wclin/dnrtPAMI/dnrt.html.

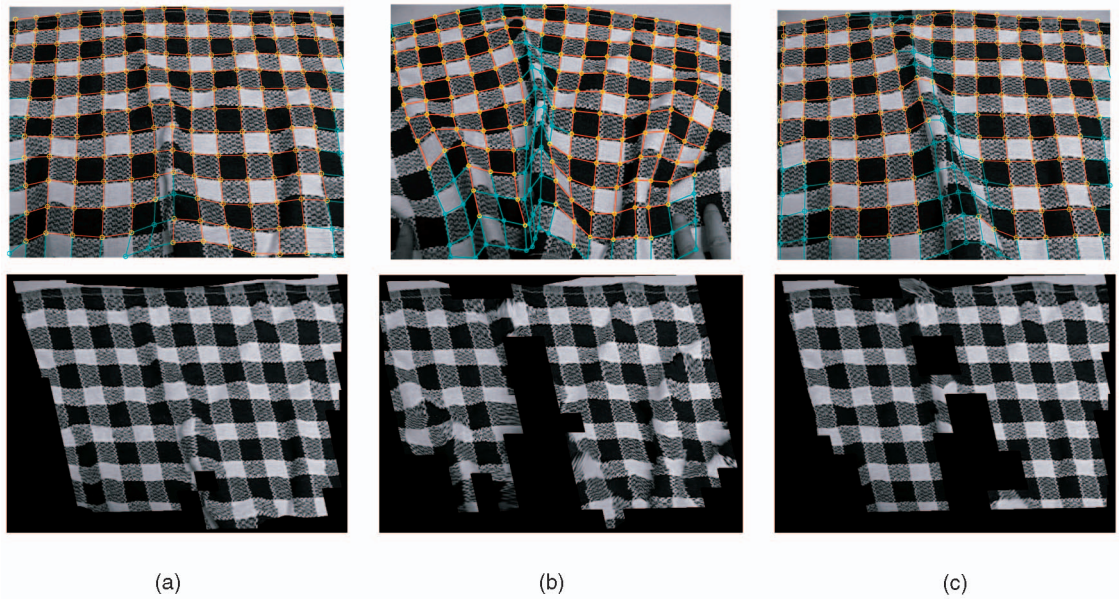


Fig. 12. (a) Frame 25. (b) Frame 50. (c) Frame 100. Top row: tracking results of a folding fabric pattern. There are a few textons totally occluded in the middle and two textons are occluded by a finger in the lower-right region. Bottom row: visibility map. For videos, please see http://www.cs.cmu.edu/~wclin/dnrtPAMI/dnrt.html.

6.4 Computational Speed

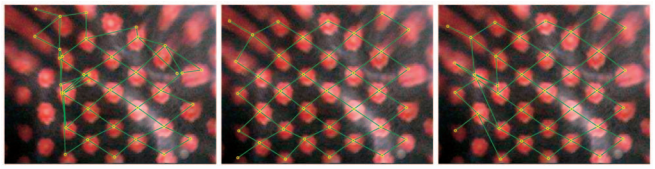
Table 1 summarizes the processing time per frame for all tracking experiments. Depending on the number of textons to be tracked in the scene, the number of particles used in particle filtering, and the type of a dynamic NRT, the computational time for processing a frame ranges from 10 seconds to 639 seconds on a 2.2 GHz PC with nonoptimized MATLAB code.

6.5 Validation and Comparison

We conduct several experiments to validate our MRF model and compare our tracking algorithm with mesh-based tracking and multitarget tracking algorithms. We first verify if the 12-neighbor configuration is the best setting for dynamic NRTs by testing our tracking algorithm with 8 and 16-neighbor configuration on underwater texture motion. We also test our tracking algorithm with multiple

TABLE 1
Processing Time in Different Experiments

Figure	10(c)	10(f)	10(i)	10(l)	11	12
Dynamic NRT Type	tightly	loosely	loosely	loosely	tightly	tightly
Number of textons	48	47	49	9	70	208
Number of particles	5	441	441	40	5	5
Time per frame (second)	40	90	90	10	200	639



(a) (b) (c)

Fig. 13. Tracking results of an underwater texture using different number of neighbors at frame 100. (a) Eight neighbors. (b) Twelve neighbors. (c) Sixteen neighbors. For videos, please see http://www.cs.cmu.edu/~wclin/dnrtPAMI/dnrt.html.

texton templates and single texton template settings and texton detection algorithm initialized at different positions. Finally, we compare the performance of our tracking algorithm with deformable object tracking and multitarget tracking algorithms.

6.5.1 Validation of 12-Neighbor Configuration

The 12-neighbor configuration in our MRF model is adopted from the spring configuration in physics-based cloth simulation. Our tracking results show that this 12-neighbor configuration works well for different types of dynamic NRTs, from highly dynamic underwater textures to slowly varying fabric textures. To further validate that the 12-neighbor MRF model is appropriate for dynamic NRTs, we test our tracking algorithm with different neighborhood configurations. Fig. 13 shows several static frames of tracking results with different number of neighbors used in the MRF model. From this experiment, we find that an 8-neighbor configuration cannot provide sufficient constraints to maintain the lattice structure, while a 16-neighbor configuration introduces too strong structural constraints. This makes the tracking algorithm less adaptive to highly dynamic motion. Once the algorithm loses track of a texton, the algorithm cannot catch the texton again (Fig. 13).

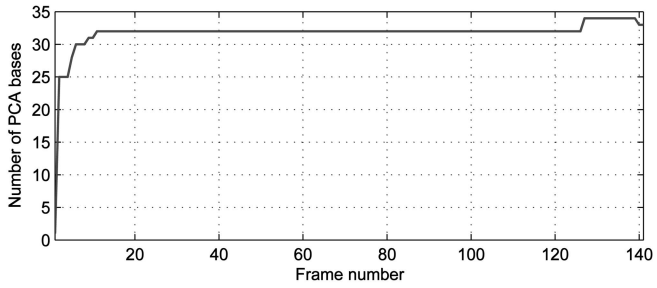


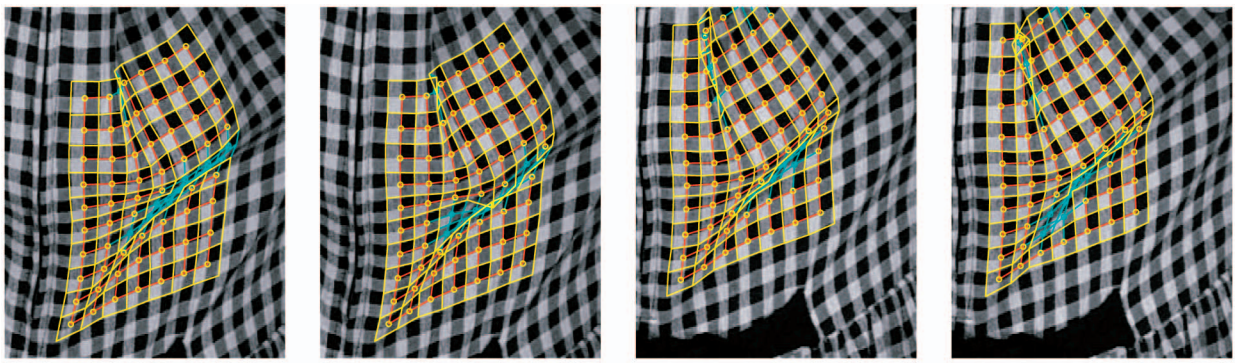
Fig. 14. Plot of the number of PCA bases (95 percent energy) used to represent texton templates in the tracking process (see Fig. 15 for tracking results). At the first frame, only one basis is used since a single texton template is used. As tracking proceeds, the number of bases increases.

6.5.2 Multiple Texton Templates versus Single Texton Template

We explore if PCA can be used to represent multiple texton templates, allowing a more compact representation of texton templates. Fig. 14 plots the number of PCA bases (95 percent energy) used during tracking (see Fig. 15). At the first frame, only one basis is needed since a single texton is used. As the tracking proceeds, the number of PCA bases increases and reaches its maximum of 34 at frame 127. If no PCA is applied, 70 texton templates are used during tracking process. Fig. 15 compares the tracking results of multiple and single texton templates at several frames. The tracking results show that using a single texton template, although providing more compact representation, has a slightly larger tracking error than using multiple texton templates.

6.5.3 Comparison of Texton Detection with Different Initial Positions

We investigate how the position of the initial texton (the first texton specified by the user) affects texton detection result. We start the texton detection algorithm (Section 5.1) at different initial positions. The texton detection results of



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

Fig. 15. Comparison of tracking results using multiple texton templates and a single texton template at frames 16 (a) and (b) and 68 (c) and (d). (a) and (c) Multiple templates. (b) and (d) Single template. One can observe that the tracking results for multiple texton templates are more accurate. For instance, there are tracking errors in the middle of (b) where several lattice nodes are not located at the centers of textons. In (d), there are visible textons misjudged as occluded, and a texton on the top row is not tracked correctly.

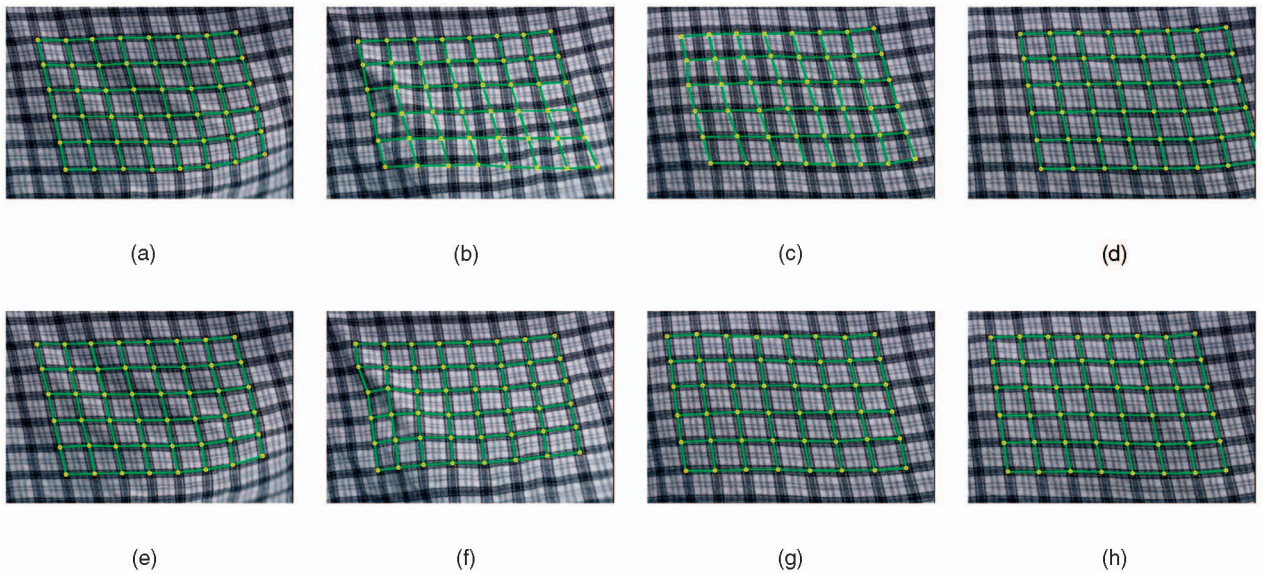


Fig. 16. Top: A short image sequence showing the drifting of lattice in the AAM tracking result. (a) frame 41, (b) frame 47, (c) frame 53, and (d) frame 56. Bottom: Our tracking result. (e) frame 41, (f) frame 47, (g) frame 53, and (h) frame 56. The lattice drifts because the AAM is not able to represent local nonlinear deformation and tracking cannot be recovered once the lattice drifts to other texton locations. For videos, please see http://www.cs.cmu.edu/~wclin/dnrtPAMI/dnrt.html.

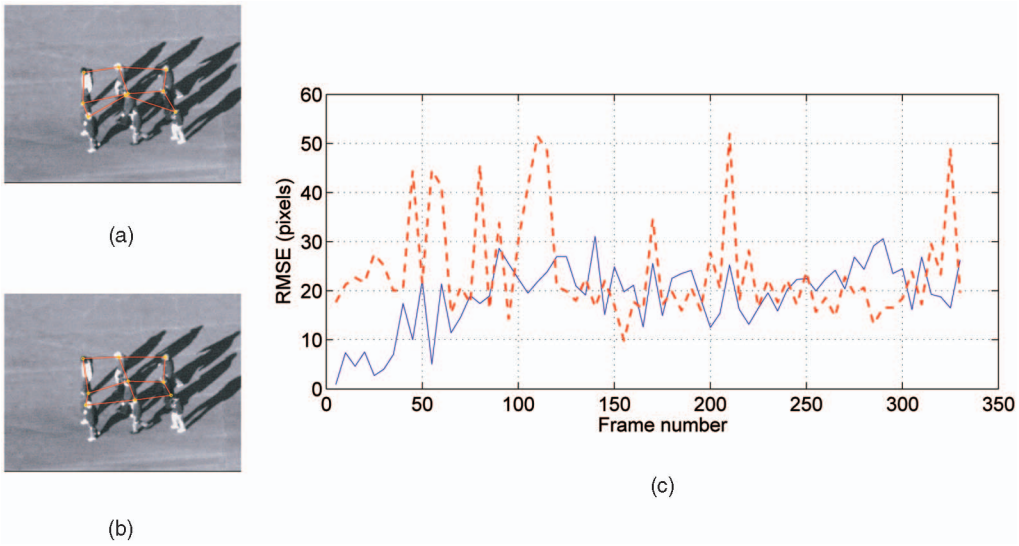


Fig. 17. (a) Comparison of our tracking result and (b) Yu and Wu's result [52] at frame 210. (c) Comparison of root mean square error against the hand-labeled ground truth. The red dash line shows the RMSE curve of Yu and Wu's tracking result and the blue solid line is the RMSE curve of our tracking result. The total RMSE of Yu and Wu's results and ours are 25.9 and 20.2 pixels, respectively. For videos, please see http://www.cs.cmu.edu/~wclin/dnrtPAMI/dnrt.html.

fabric textures on a towel and a dress can be seen in http://www.cs.cmu.edu/~wclin/dnrtPAMI/dnrt.html. In one of the detection results of the towel example (initialized at the bottom-right corner), two textons at the top-left corner are not detected successfully because the image intensities of these textons are much darker than those of the initial texton. The towel and dress examples show that the spatial detection algorithm is not sensitive to different initial positions.

6.5.4 *Comparison with Deformable Object Tracking and Multitarget Tracking*

Dynamic NRTs exhibit wide range of motion characteristics, from slowly periodic motion to rapidly moving motion, and from surface deformation to loosely coupled crowd motion. If we arrange different dynamic NRTs based on structural constraints among individual textons, these dynamic NRTs form a spectrum along the structural regularity axis. It appears that, on one end of the spectrum, dynamic NRTs may be considered as a deformable object tracking problem, while, on the other end, dynamic NRTs can be treated as a multitarget tracking problem.

Among deformable object tracking algorithms, we choose the Active Appearance Model (AAM) [7] for comparison since most deformable object tracking algorithms are developed based on AAM or a similar concept [29]. These algorithms use a mesh to represent the shape of a deformable object and apply an image alignment algorithm to fit the appearance of an input image sequence with a modal image or an appearance model constructed by a set of training images. Fig. 16 shows the AAM tracking result of a slowly varying dynamic NRT using the AAM code implemented by Matthews and Baker [29]. In this

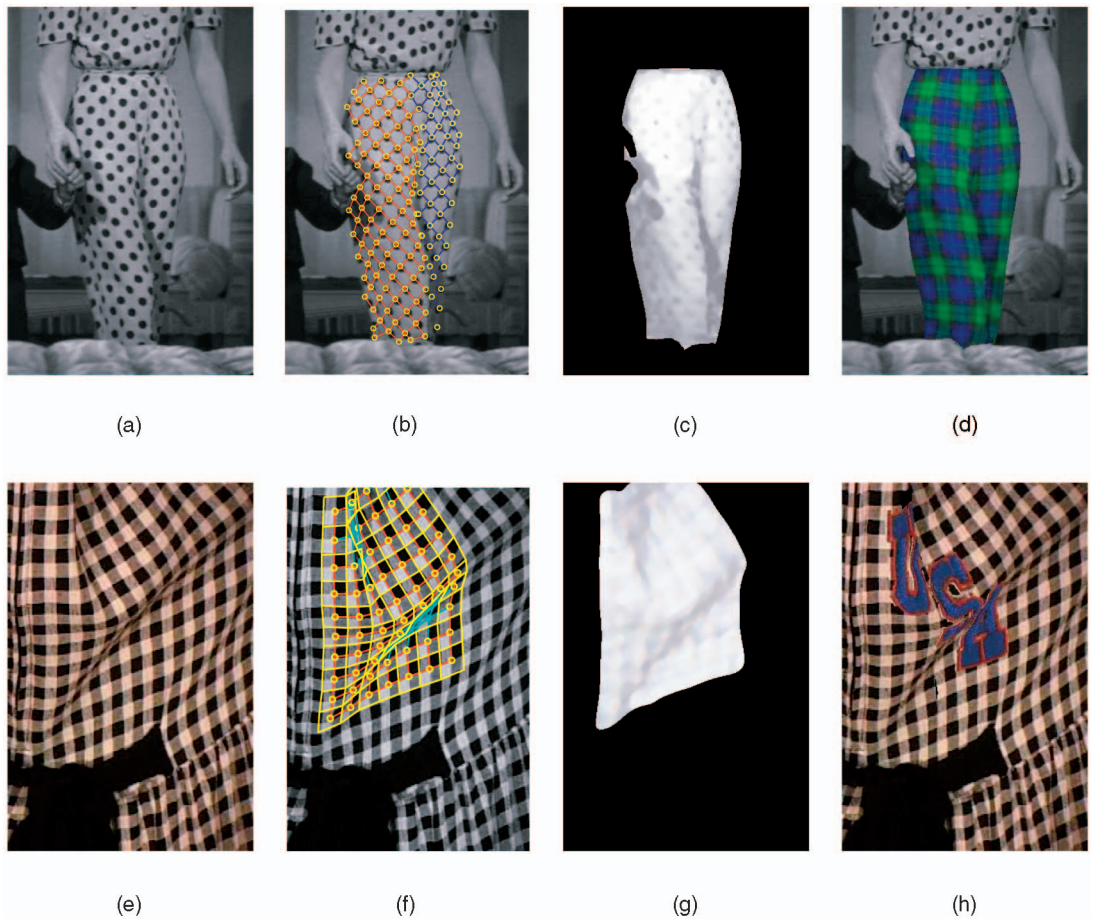


Fig. 18. Top: Dynamic NRT replacement of a fabric texture. Bottom: Video superimposition of letters "USA." (a) An input frame. (b) Tracked lattice. (c) Extracted lighting. (d) Replacement result. (e) An input frame. (f) Tracked lattice. (g) Extracted lighting. (h) Superimposed text. For videos, please see http://www.cs.cmu.edu/~wclin/dnrtPAMI/dnrt.html.

experiment, we select 10 frames (uniformly sampled) from the input image sequence to construct an AAM model. The 2D meshes used for constructing the AAM model are manually labeled. The lattice drifts due to ambiguous correspondences, although the lattice structure is maintained in the AAM tracking result. Fig. 16 shows image frames where drifting occurs in the AAM tracking result.

There are two important differences between AAM-based approaches and our tracking algorithm. First, our MRF model integrates a spring-network-like statistical lattice structure model and a registration-based image observation model. The role of the image observation model is similar to the image alignment algorithm in AAM-based approaches; however, our lattice structure model makes a crucial difference. It allows local nonlinear deformation while AAM only models global linear deformation. If a surface deformation cannot be represented by an AAM, AAM may lose tracking of textons and the whole lattice is attracted by neighboring textons with similar appearance (Fig. 16). Second, we use a linear dynamic model (20) to predict the position of lattice in the next frame. The predicted lattice position provides a better initial condition for the image alignment algorithm such that the image alignment process is more likely to converge to the correct solution.

We also compare the performance of our tracking algorithm against multitarget tracking algorithms. In particular, we choose Yu and Wu's tracking algorithm [52] as they also use an MRF to represent the spatial constraints between targets. Each target is tracked by a tracker and these trackers collaborate with each other under an MRF model to resolve ambiguous correspondences among multiple identical targets.

Fig. 17 shows the comparison of our tracking results and Yu and Wu's on the crowd motion. This comparison shows that our tracking algorithm can keep tracking all targets steadily through the entire sequence, while Yu and Wu's algorithm may lose tracking of targets from time to time. To quantitatively compare their results and ours, we manually track the lattice every five frames in the video. The total root mean square errors (RMSE) of Yu and Wu's results and ours are 25.9 and 20.2 pixels, respectively. Fig. 17c is an RMSE plot of tracking results. The reason that our algorithm is more robust than Yu and Wu's is because we explicitly model the 4-degree topological structure of the textons, which remains invariant despite all kinds of motions a dynamic NRT may undergo.

The comparison with deformable object tracking and multitarget tracking algorithms demonstrate the effectiveness of our lattice-based MRF model on dynamic NRT tracking. The comparison also shows that our tracking algorithm not only provides a unified framework for tracking dynamic NRT under a wide range of motion but also outperforms algorithms that are specialized at certain type of motion-deformable objects and multitargets. We should mention that our method is not real-time, but the

AAM approach and Yu and Wu's approach that we are comparing with are.

6.6 Video Editing Applications

Our tracking algorithm can benefit many applications besides texture tracking, e.g., video editing, cloth motion capture, and fashion design preview. Fig. 18 demonstrate two applications in dynamic texture replacement and video superimposition. Texture replacement in photos changes an NRT in an image without knowing the scene geometry while preserving the geometric deformations, and photometric realism, such as shading and shadows. Although existing algorithms [12], [26], [27], [45] can replace a texture in still images, including an attempt in the early 1970s [20], texture replacement in videos under occlusion and rapid movements has not been done. The major challenge is to achieve realistic replacement and maintain temporal coherence simultaneously.

The temporal coherence problem can be solved by combining an effective texture tracking algorithm and a spatiotemporal smoothing algorithm. We first apply our tracking algorithm to track lattices of a dynamic texture. With the tracked lattices that capture the temporal information of the texture, we use an algorithm proposed by Liu et al. [26] to compute geometric and lighting deformation fields (DFs) separately, which are essentially pixel-wise mappings that define the geometric and photometric variations of the NRT. Temporal coherence is further addressed by smoothing the geometric and lighting DFs spatiotemporally. The smoothed geometric and lighting DFs can then be applied to any texture to achieve realistic and coherent video texture replacement. Note that spatiotemporal stitching is applied when there are more than one NRT patches in an image frame, e.g., the left and right trousers in Fig. 18b are represented by two patches. For more video editing examples, please see http://www.cs.cmu.edu/~wclin/dnrtPAMI/dnrt.html.

7 Conclusion

We propose a lattice-based MRF model for dynamic NRTs. Textons of a dynamic NRT are treated as separate moving objects connected by a topological constraint, while allowing individual textons to vary flexibly in geometry and appearance. Our lattice-based MRF model consists of a lattice structure model that characterizes the topological constraint of a dynamic NRT and a registration-based image observation model that handles the geometry and appearance variations of individual textons. We treat dynamic NRT tracking as a spatiotemporal inference problem using the belief propagation and the particle filtering algorithms. We demonstrate the effectiveness of our algorithm on tracking dynamic NRTs under rapid movements, motion blurring, folding, occlusion, and illumination changes through different mediums. In future study, we will allow the lattice topology to adapt during a dynamic NRT tracking process and evaluate its pros and cons. A possible direction is to adopt a dynamic Bayesian network [31] to model a varying topology. We would also like to extend our MRF to model the folding topology and to combine a shape-from-texture algorithm to capture 3D surface geometry, which will further expand the applications of various types of dynamic NRTs.

Acknowledgments

The authors would like to thank the reviewers for their valuable comments and Robert T. Collins, Alexei A. Efros, Greg Turk, Jing Xiao, Chieh-Chih Wang, Jiayong Zhang, Sanjiv Kumar, Srinivasa Narasimhan, and Tim Cootes for their insightful suggestions on this work. They also thank Ting Yu, Igor Guskov, and Changbo Hu for providing the results of their algorithms in dynamic NRT tracking comparison. This work was supported in part by US National Science Foundation grant IIS-0099597, Taiwan National Science Council grant 95-2218-E-009-207, and Taiwan MOE ATU Program.

References

[1] S. Baker and I. Matthews, "Lucas-Kanade 20 Years On: A Unifying Framework," *Int'l J. Computer Vision,* vol. 56, no. 3, pp. 221-255, 2004.
[2] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman, "Texture Mixing and Texture Movie Synthesis Using Statistical Learning," *IEEE Trans. Visualization and Computer Graphics,* vol. 7, no. 2, pp. 120-135, Apr.-June 2001.
[3] K. Bhat, S. Seitz, J. Hodgins, and P. Khosla, "Flow-Based Video Synthesis and Editing," *ACM Trans. Graphics,* vol. 23, no. 3, pp. 360-363, 2004.
[4] M.J. Black and P. Anandan, "The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields," *Computer Vision and Image Understanding,* vol. 63, no. 1, pp. 75-104, 1996.
[5] D. Chetverikov and R. Péteri, "A Brief Survey of Dynamic Texture Description and Recognition," *Proc. Int'l Conf. Computer Recognition Systems,* pp. 17-26, 2005.
[6] K.-J. Choi and H.-S. Ko, "Stable but Responsive Cloth," *Proc. ACM SIGGRAPH Conf.,* pp. 604-611, 2002.
[7] T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active Appearance Models," *Proc. European Conf. Computer Vision,* pp. 484-498, 1998.
[8] G. Doretto, "Modeling Dynamic Scenes with Active Appearance," *Proc. Conf. Computer Vision and Pattern Recognition,* pp. 66-73, 2005.
[9] G. Doretto, A. Chiuso, Y.N. Wu, and S. Soatto, "Dynamic Textures," *Int'l J. Computer Vision,* vol. 51, no. 2, pp. 91-109, 2003.
[10] A. Doucet, N.D. Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice.* Springer-Verlag, 2001.
[11] E.S. Fedorov, "The Elements of the Study of Figures," *Proc. St. Petersburg Mineralogical Soc.,* vol. 21, pp. 1-289, 1885.
[12] D.A. Forsyth, "Shape from Texture without Boundaries," *Proc. European Conf. Computer Vision,* pp. 225-239, 2002.
[13] W.T. Freeman, E.C. Pasztor, and O.T. Carmichael, "Learning Low-Level Vision," *Int'l J. Computer Vision,* vol. 40, no. 1, pp. 25-47, 2000.
[14] B. Grünbaum and G.C. Shephard, *Tilings and Patterns.* W.H. Freeman and Company, 1987.
[15] I. Guskov, "Efficient Tracking of Regular Patterns on Non-Rigid Geometry," *Proc. Int'l Conf. Pattern Rocognition,* 2002.
[16] I. Guskov, "Multiscale Inverse Compositional Alignment for Subdivision Surface Maps," *Proc. European Conf. Computer Vision,* pp. 133-145, 2004.
[17] I. Guskov, S. Klibanov, and B. Bryant, "Trackable Surfaces," *Proc. ACM Symp. Computer Animation,* pp. 251-257, 2003.
[18] M. Isard, "Pampas: Real-Valued Graphical Models for Computer Vision," *Proc. Conf. Computer Vision and Pattern Recognition,* pp. 613-620, 2003.
[19] M. Isard and A. Blake, "Condensation—Conditional Density Propagation for Visual Tracking," *Int'l J. Computer Vision,* vol. 29, no. 1, pp. 5-28, 1998.
[20] T. Kanade and T. Sakai, "Color TV Display and TV Camera Input for Computer Image Processing," *Interface,* vol. 5, pp. 21-36, 1976 (in Japanese).
[21] Z. Khan, T. Balch, and F. Dellaert, "An Mcmc-Based Particle Filter for Tracking Multiple Interacting Targets," *Proc. European Conf. Computer Vision,* pp. 279-290, 2004.
[22] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut Textures: Image and Video Synthesis Using Graph Cuts," *Proc. ACM SIGGRAPH Conf.,* pp. 277-286, 2003.
[23] S.Z. Li, *Markov Random Field Modeling in Image Analysis,* second ed. Springer, 2001.

[24] W.-C. Lin, "A Lattice-Based MRF Model for Dynamic Near-Regular Texture Tracking and Manipulation," Technical Report CMU-RI-TR-05-58, PhD thesis, Robotics Inst., Carnegie Mellon Univ., Dec. 2005.
[25] Y. Liu, R.T. Collins, and Y. Tsin, "A Computational Model for Periodic Pattern Perception Based on Frieze and Wallpaper Groups," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, pp. 354-371, 2004.
[26] Y. Liu, W.-C. Lin, and J. Hays, "Near-Regular Texture Analysis and Manipulation," *Proc. ACM SIGGRAPH Conf.,* pp. 368-376, 2004.
[27] A. Lobay and D.A. Forsyth, "Recovering Shape and Irridiance Maps from Rich Dense Texton Fields," *Proc. Conf. Computer Vision and Pattern Recognition,* pp. 400-406, 2004.
[28] B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Proc. Int'l Joint Conf. Artificial Intelligence,* pp. 674-679, 1981.
[29] I. Matthews and S. Baker, "Active Appearance Models Revisited," *Int'l J. Computer Vision,* vol. 60, no. 2, pp. 135-164, Nov. 2004.
[30] I. Matthews, T. Ishikawa, and S. Baker, "The Template Update Problem," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 6, pp. 810-815, June 2004.
[31] K. Murphy, "Dynamic Bayesian Networks: Representation, Inference and Learning," PhD thesis, Univ. of California, Berkeley, 2002.
[32] R.C. Nelson and R. Polana, "Qualitative Recognition of Motion Using Temporal Texture," *CVGIP Image Understanding,* vol. 56, no. 1, pp. 78-89, July 1992.
[33] A. Papoulis and S.U. Pillai, *Probability, Random Variables, and Stochastic Processes,* fourth ed. McGraw-Hill, 2002.
[34] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,* second ed. Morgan Kaufmann, 1998.
[35] J. Pilet, V. Lepetit, and P. Fua, "Real-Time Non-Rigid Surface Detection," *Proc. Conf. Computer Vision and Pattern Recognition,* pp. 822-828, 2005.
[36] D. Pritchard and W. Heidrich, "Cloth Motion Capture," *Proc. Eurographics,* 2003.
[37] X. Provot, "Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior," *Proc. Conf. Graphics Interface,* pp. 147-154, 1995.
[38] P. Saisan, G. Doretto, Y.N. Wu, and S. Soatto, "Dynamic Texture Recognition," *Proc. Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 58-63, Dec. 2001.
[39] V. Scholz and M. Magnor, "Cloth Motion from Optical Flow," *Proc. Ninth Int'l Fall Workshop Vision, Modeling, and Visualization,* 2004.
[40] V. Scholz, T. Stich, M. Keckeisen, M. Wacker, and M. Magnor, "Garment Motion Capture Using Color-Coded Patterns," *Proc. Eurographics,* pp. 439-448, 2005.
[41] S. Sclaroff and J. Isidoro, "Active Blobs," *Proc. Int'l Conf. Computer Vision,* pp. 1146-1153, 1998.
[42] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky, "Nonparametric Belief Propagation," *Proc. Conf. Computer Vision and Pattern Recognition,* pp. 605-612, 2003.
[43] E.B. Sudderth, M.I. Mandel, W.T. Freeman, and A.S. Willsky, "Distributed Occlusion Reasoning for Tracking with Nonparametric Belief Propagation," *Neural Information Processing Systems,* pp. 1369-1376, 2004.
[44] M. Szummer and R.W. Picard, "Temporal Texture Modeling," *Proc. IEEE Int'l Conf. Image Processing,* vol. 3, pp. 823-826, 1996.
[45] Y. Tsin, Y. Liu, and V. Ramesh, "Texture Replacement in Real Images," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 539-544, Dec. 2001.
[46] R. Vidal and A. Ravichandran, "Optical Flow Estimation and Segmentation of Multiple Moving Dynamic Textures," *Proc. Conf. Computer Vision and Pattern Recognition,* pp. 516-521, 2005.
[47] Y. Wang and S.C. Zhu, "Analysis and Synthesis of Textured Motion: Particles and Waves," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 10, pp. 1348-1363, Oct. 2004.
[48] Y. Wang and S.C. Zhu, "Modeling Complex Motion by Tracking and Editing Hidden Markov Graphs," *Proc. Conf. Computer Vision and Pattern Recognition,* pp. 856-863, 2004.
[49] L.-Y. Wei and M. Levoy, "Fast Texture Synthesis Using Tree-Structured Vector Quantization," *Proc. ACM SIGGRAPH Conf.,* pp. 479-488, 2000.
[50] Y. Weiss and W. Freeman, "Correctness of Belief Propagation in Gaussian Graphical Models of Arbitrary Topology," *Neural Computation,* vol. 13, no. 10, pp. 2173-2200, 2001.
[51] J. Yedidia, W. Freeman, and Y. Weiss, "Understanding Belief Propagation and Its Generalizations," *Proc. Int'l Joint Conf. Artificial Intelligence,* 2001.
[52] T. Yu and Y. Wu, "Decentralized Multiple Target Tracking Using Netted Collaborative Autonomous Trackers," *Proc. Conf. Computer Vision and Pattern Recognition,* pp. 939-946, 2005.
[53] S.C. Zhu, C. Guo, Y. Wang, and Z. Xu, "What Are Textons?" *Int'l J. Computer Vision,* vol. 62, nos. 1-2, pp. 121-143, 2005.

Wen-Chieh Lin received the BS and MS degrees in control engineering from the National Chiao-Tung University, Hsinchu, Taiwan, in 1994 and 1996, respectively, and the PhD degree in robotics for dynamic near-regular texture tracking and manipulation from Carnegie Mellon University, Pittsburgh, Pennsylvania, in 2005. He joined the Department of Computer Science and the Institute of Multimedia Engineering at National Chiao-Tung University as an assistant professor in 2006. Dr. Lin's current research interests include computer vision, computer graphics, and computer animation. He is a member of the IEEE and the IEEE Computer Society.

Yanxi Liu received the BS degree in physics/electrical engineering in Beijing and the PhD degree in computer science for group theory applications in robotics from the University of Massachusetts. Her postdoctoral training was performed at LIFIA/IMAG, Grenoble, France. She also spent one year at DIMACS (US National Science Foundation center for Discrete Mathematics and Theoretical Computer Science) with an NSF research-education fellowship award. Dr. Liu has been a faculty member in the Robotics Institute (RI) of Carnegie Mellon University (CMU) and affiliated with the Machine Learning Department at CMU. In the fall of 2006, she joined the Computer Science and Engineering and Electrical Engineering Departments at Pennsylvania State University. She is an adjunct associate professor in the Radiology Department of the University of Pittsburgh and a guest professor at the Computer Science Department, Huazhong University of Science and Technology in China. Her research interests span a wide range of applications in computer vision, computer graphics, robotics, and computer aided diagnosis in medicine, with two central themes: computational symmetry and discriminative subspace learning. With her colleagues, Dr. Liu won the first place in the clinical science category and the best paper overall at the Annual Conference of Plastic and Reconstructive Surgeons for the paper "Measurement of Asymmetry in Persons with Facial Paralysis." Dr. Liu chaired the First International Workshop on Computer Vision for Biomedical Image Applications (CVBIA) in conjunction with ICCV 2005 and coedited the book: *CVBIA: Current Techniques and Future Trends* (Springer-Verlag LNCS). She serves as a reviewer/committee member/panelist for all major journals and conferences as well as NIH/NSF panels on computer vision, pattern recognition, biomedical image analysis, and machine learning. She was a chartered study section member for Biomedical Computing and Health Informatics at the US National Institutes of Health. She is a senior member of the IEEE and the IEEE Computer Society.