Dynamic Texture Synthesis Using Motion Distribution Statistics

Ashfaqur Rahman

Computer Science Department, American International University Bangladesh
Email: ashfaqur.rahman.omi@gmail.com

Manzur Murshed

Gippsland School of IT, Monash University, Churchill VIC 3842, Australia
Email: Manzur.Murshed@infotech.monash.edu.au

In this paper we propose a motion based approach for synthesizing dynamic textures. Dynamic textures are natural phenomenon characterized by their distinctive motion patterns. Synthesis of these textures is thus considered as the regeneration of a motion pattern that has identical motion distribution of a source texture. In this paper we propose a synthesis technique where new textures are generated by computing their movement pattern from a known motion distribution followed by the generation of image frames. Experimental results demonstrate the ability of the proposed technique by producing visually promising dynamic textures.

ACM Classification: I.4

1. INTRODUCTION

A large class of objects commonly experienced in real world scenarios exhibit characteristic motion with an indeterminate spatial and temporal extent. The motion assembly adopted by a flock of flying birds, water streams, fluttering leaves, and waving flags are some of the most common examples that serve to illustrate such motion. Contemporary literature coined the term "dynamic texture" to collectively identify such motion patterns that exhibit spatiotemporal regularity but have an indeterminate spatial and temporal extent.

Dynamic textures are image sequences with an inherent time dimension along with two spatial dimensions. As a result, while pixel intensities play a direct role in image texture analysis, the temporal cue of pixel intensities, namely motion, plays a similar role for analysing dynamic textures. The dynamics of a dynamic texture is encoded by its motion distribution statistics and many researchers (Polana and Nelson, 1992; Bouthemy and Fablet, 1998; Fablet *et al*, 2002; Fablet and Bouthemy, 2003; Peh and Cheong, 1999; Péteri and Chetverikov, 2005; Péteri and Chetverikov, 2004; Fazekas and Chetverikov, 2007; Rahman and Murshed, 2007) have developed techniques to characterize dynamic textures using a diverse set of features based on motion distribution. Although motion based characterizations are commonly used for dynamic texture recognition, their application for synthesizing such textures, is unexplored. In this paper, we explore the potential of motion based dynamic texture characterization techniques to synthesize dynamic texture image sequences.

Manuscript received: 24 May 2007
Communicating Editor: Vlad Estivill-Castro

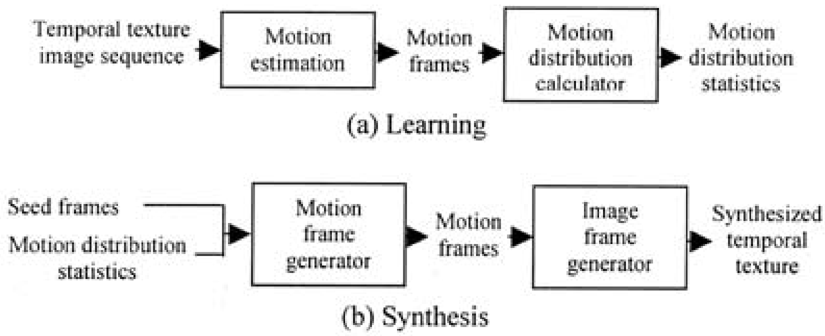


Figure 1: Framework of dynamic texture synthesis. (a) Learning; (b) Synthesis.

Before elaborating on our achievements in this paper, let us briefly explain what we mean by synthesis. Given the image sequence of an input dynamic texture T, we discover a generative model M and we would like to generate an image sequence T' from M, such that the texture of T' looks as realistic as T, but is not simply a copy of T. Often the width, height and number of frames in T' will differ from those in T. This definition of synthesis is commonly used in the current literature by a group of techniques known as image based synthesis techniques.

The basic philosophy of our proposed synthesis method that will be explored in this paper is presented in the framework in Figure 1. During learning, texture movement is identified by computing motion vectors and the movement pattern within a texture is then encoded by motion distribution statistics. Synthesis of new texture is performed in two steps. The motion frames of the synthesized sequence are generated first using the a priori motion distribution statistics. The sequence of image frames are then constructed from a seed image frame and synthesized motion frames using guidelines of the motion vectors and local properties of dynamic textures. We have explored each of the key steps of the abovementioned synthesis framework in this paper. We have synthesized a diverse set of dynamic textures and experimental results that demonstrate the ability of the proposed technique by producing visually promising dynamic textures.

The paper is organized as follows. Some works related to different steps of our proposed synthesis technique are presented in Section 2. Each of the key steps of our proposed synthesis technique is elaborated in Section 3. In Section 4 we present some experimental results and Section 5 concludes the paper.

2. RELATED WORKS

In this section we explore a set of existing works relevant to the key steps of our proposed framework in order to choose suitable mechanisms to carry out the synthesis steps. Motion distribution statistics are necessary for regenerating the motion frames effectively and we discuss some existing motion characterization techniques in Section 2.1 to find a suitable motion distribution. Regeneration of image frames requires guidelines from motion vectors, and a review of some existing motion estimation algorithms is presented in Section 2.2. Production of synthetic textures has long been a goal of computer graphics and there exists a number of works in the current literature for synthesizing dynamic textures. Although our main focus in this paper is to synthesize dynamic textures using a suitable motion based characterization technique, a brief review of existing synthesis techniques is presented in Section 2.3 for the sake of completeness.

2.1 Motion Characterization Techniques

A dynamic texture has one temporal dimension and two spatial dimensions and the effective characterization of dynamic textures lies in the optimal utilization of time–space motion correlation. Although a number of motion based dynamic texture characterization techniques exist (Polana and Nelson, 1992; Bouthemy and Fablet, 1998; Fablet *et al*, 2002; Fablet and Bouthemy, 2003; Peh and Cheong, 1999; Péteri and Chetverikov, 2005; Péteri and Chetverikov, 2004; Fazekas and Chetverikov, 2007; Rahman and Murshed, 2007), the most accurate characterization on a set of low quality dynamic textures is achieved by the Optimal Time Space Ratio (OTSR) technique (Rahman and Murshed, 2007) where textures are identified by a set of first order motion features, namely Motion Co-occurrence Matrix (MCM) that are computed along time and space dimensions and merged at an optimal time–space ratio. MCM is a tabulation of how often different combinations of motion vectors occur in a sequence of motion frames of a dynamic texture and thus encodes the motion distribution capturing the dynamics effectively. Thus, we use MCM as motion distribution statistics in our proposed synthesis framework.

2.2 Motion Estimation Algorithms

Image frames are regenerated in our proposed framework using the motion vectors in regenerated motion frames as guidelines. Motion vectors thus computed have to have two-fold properties in order to comply with our proposed synthesis framework – (i) In order for the motion distribution to capture the accurate dynamics of the texture the motion vectors need to reflect true motion content, and (ii) in order to regenerate image frames from motion frames, it is necessary that the source and destination of motion vectors have identical intensity patterns, as will be elaborated in Section 3.4.

There are two conventions for estimating 2D motion in the field of signal processing (Shi and Sun, 2000) – (i) Block motion estimation, and (ii) Pixel motion estimation. Block motion is mostly used in block-based video coding systems (Shi and Sun, 2000) (MPEG-1/2/4 and H.26X) where a video frame is partitioned into a set of non-overlapped, small rectangular blocks and a motion vector is associated with each block by finding its best match in the previous frame with maximum correlation. From a coding point of view the difference between a block and its best match is stored as error. As can be observed from our proposed synthesis framework (Figure 1), during synthesis only motion frames are regenerated not the difference errors. As a result, during the synthesis step the error information associated with the block is missing and we are unable to restore the image frames fully. Block motion is thus not a viable option to use for the purpose of synthesis although block motion computes true motion for dynamic textures (Rahman and Murshed, 2007).

The large body of literature on pixel motion i.e. optical flow computation algorithms (Shi and Sun, 2000; Larsen *et al*, 1998; Berezait *et al*, 2000; Corpetti *et al*, 2002; Memin and Perez, 1999) can be roughly grouped into gradient based, correlation based, phase-based and spatiotemporal energy based approaches. These techniques however were developed considering properties of regular objects that are not usually observed in most dynamic textures. Optical flow estimation of dynamic textures in a more generalized way was first addressed, however in Rahman and Murshed (2007). In this technique, the choice of the flow vector for a pixel is guided by a probability density function obtained using the spatiotemporal autoregressive (STAR) model expressed as

$$I_t(x,y) = \sum_{i=1}^{m} A_i I_{t+1}(x + \Delta x_i, y + \Delta y_i) \tag{1}$$

where $I_t(x,y)$ is the intensity associated with pixel (x,y) at time t, $(\Delta x_i, \Delta y_i)$, is the displacement vector corresponding to the i-th possible destination pixel in search window of the next frame and A_i is the coefficient associated with the i-th destination pixel. A_i's constitute the probability density function and the pixel with the highest A_i is assumed to be the destination of pixel (x,y) at t-th image frame.

STAR model, however, fails to produce accurate motion content because of the following flaws – (i) The STAR model is appropriate for modelling energy and mass diffusion in thermodynamics and fluid dynamics respectively. As intensity does not represent energy or mass level, applying this model on intensity is inappropriate. (ii) In the STAR model a large neighbourhood structure is required for accurate approximation of coefficients in (1). However spatiotemporal motion uniformity may not exist for such a large neighbourhood. On the other hand choosing a small neighbourhood may not be sufficient for estimating the coefficients. All these observations on existing motion estimation algorithms leads us to develop a new motion estimation algorithm suitable for synthesizing dynamic textures as will be elaborated in the following sections.

2.3 Synthesis Techniques

The large number of works on dynamic texture synthesis can be divided into two groups – (i) Physics based approaches and (ii) Image based approaches. Synthesis of dynamic textures is performed in physics based approaches by building a physical model of the process that generates the scene. For example, steam, fog, smoke, and fire have been simulated in this manner (Ebert and Parent, 1990; Ebert *et al*, 1994; Stam and Fiume, 1993; Stam and Fiume, 1995). Explosions, fire, and waterfalls have been successfully simulated by animated particle systems (Reeves, 1983; Reeves and Blau, 1985; Sims, 1990). Simplified physically-based models have also been used to produce synthetic waves and surf (Fournier and Reeves, 1986; Peachey, 1986).

An alternative to the physics based techniques are image based ones. In this framework, new texture moves are generated using images without building a physical model of the process. Among these approaches one can distinguish between two subclasses, the so-called nonparametric approaches that forego the use of a model altogether and generate synthetic images by clever concatenation or repetition of image data, and parametric image based approaches that rely on a model, albeit not a physical one. The nonparametric methods generate dynamic textures by image quilting (Efros and Freeman, 2001) or by directly sampling original pixels (Wei and Levoy, 2000), frames (Schodl *et al*, 2000), wavelet-structures (Bar-Joseph *et al*, 2001), 2D patches (Kwatra *et al*, 2003), and 3D chunks (Edwards, 2002) from a training sequence and usually produce high quality visual effects. Compared with nonparametric approaches, the parametric methods provide better model generalization and understanding of the essence of dynamic texture. Our proposed synthesis technique also falls into the category. The typical parametric models include Szummer and Picards STAR model (Szummer and Picard, 1996), Campbell *et al*'s eigenspace representation (Campbell *et al*, 2002), Fitzgibbon's stochastic rigidity model (Fitzgibbon, 2001), Soatto *et al*'s linear dynamic system (LDS) (Soatto *et al*, 2001; Doretto *et al*, 2004), the Fourier descriptor based LDS by Bobby *et al* (Abraham *et al*, 2005), Yuan *et al*'s Closed-Loop LDS (Yuan *et al*, 2004), Wang and Zhu's moveton representation (Wang and Zhu, 2002; Zhu *et al*, 2005; Wang and Zhu, 2004a; Wang and Zhu, 2003), generative graph representation (Wang and Zhu, 2004b), and Che *et al*'s dynamic texture modelling with mixtures of locally linear subspaces (Li *et al*, 2005). They are very helpful for tasks such as dynamic texture editing, recognition, segmentation and image registration. However, most parametric models are less likely to generate dynamic textures with the same quality as the non-parametric models, in particular for videos of natural scenes.

3. PROPOSED SYNTHESIS TECHNIQUE

Each of the key steps of our proposed synthesis technique is elaborated in detail in the following sections.

3.1 Motion Estimation

As discussed in Section 2.2, not all pixel motion estimation algorithms are suitable for dynamic texture synthesis and we thus develop a new pixel motion estimation algorithm for dynamic textures. Dynamic textures are motion patterns with spatiotemporal motion uniformity and the majority of pixels of a small spatial group are supposed to undergo uniform displacement in a uniform direction. Like conventional flow estimation approaches, if we assume that the brightness of a point is constant over a short period of time, the flow estimation problem for a pixel can be stated as identification of flow vector that maximizes motion uniformity for a small spatial neighbouring group of pixels under this brightness constancy constraint.

Before elaborating the proposed motion estimation algorithm, we first formulate motion uniformity for a group of pixels centred at p_s. Let $\eta(p_s)$ represent the set of correlated neighbourhood pixels centred at p_s and $\delta(p_i)$ represent the set of possible destination pixels in the next frame under the brightness constancy constraint, i.e., pixels in the next frame whose intensity are similar to p_i within a search window. Uniformity for p_s under certain neighbourhoods and possible destination relationships can be formulated as –

$$\vartheta(\eta, \delta, p_s) = mode\left[\bigcup_{p_i \in \eta(p_s)} \bigcup_{p_j \in \delta(p_i)} \overrightarrow{p_i p_j}\right] \tag{2}$$

where ϑ is the uniformity vector and mode is the statistical mode function. Based on the abovementioned definition of uniformity our proposed algorithm is elaborated in Figure 2.

From an implementation point of view, the following adjustments are made to the proposed algorithm:

- Due to the quantization error in digital images, brightness constancy may not be observed in some cases. A small error window of $\pm\,\Delta$ is thus used when searching for similar pixels.
- The motion histogram is implemented as a two dimensional histogram indexed by magnitude and direction. However, due to the quantization error, motion uniformity may not be restricted along a salient magnitude and direction. In order to deal with this problem, bands of motion magnitude and direction are considered while estimating flow vectors and thus the uniformity

```
FUNCTION Motion_Estimation
ARGUMENT current_frame, next_frame
RETURN motion_vectors
FOR all pixel p_s in the current_frame
   FOR all correlated pixel p_i∈η(p_s)
      Populate a motion histogram h using all
      possible displacement vectors of p_i
      corresponding to similar intensity pixels in
      next_frame;
   ENDFOR
         motion_vectors(p_s) = The displacement vector of
         p_s that corresponds to maximum(h);
ENDFOR
```

Figure 2: Proposed motion estimation algorithm.

function returns one of the vectors in the maximum histogram, selected either randomly or by a spatial uniformity constraint.

- The set of destination pixels obtained under the brightness constancy assumption is reduced to those for which the source pixel has to travel the minimum distance before populating the motion histogram. This is to prohibit the assignment of erroneous flow vectors to static backgrounds.

Our proposed algorithm does not discriminate between types of textures by utilizing motion uniformity – universal to all types of textures – while computing optical flow thus proving a robust platform. It combines motion direction and magnitude using a histogram instead of considering them separately as is done in the STAR model (Rahman and Murshed, 2007). Our proposed algorithm is not constrained by size of the correlation structure. As noisy motion vectors are not uniform in nature, they are automatically eliminated by our algorithm. Our proposed algorithm has some added tuneable parameters like error window and band size to make it more adaptive compared to the STAR model.

Our proposed motion estimation algorithm ensures that the source and destination pixels of the computed motion vectors have identical intensities thus conforming to the requirement of the image frame regeneration step of our proposed synthesis framework. Our proposed algorithm computes true motion for dynamic textures and we empirically establish the accuracy of the optical flow estimates obtained by our proposed method by classifying a diverse set of dynamic textures with high accuracy.

3.2 Motion Distribution Statistics

Motion distribution statistics encodes the movement pattern within a texture and we use MCM as the distribution statistics. The temporal MCM used in OTSR encodes motion correlation between successive motion frames and thus we can use temporal MCM to predict a future motion frame from the previous one. Spatial MCM however encodes motion correlation between spatial neighbours within a motion frame and thus lacks any ability to predict future motion frames. For accurate prediction of future motion frames we need to incorporate spatial information. We thus redefine MCM merging both spatial and temporal information into a unified spatiotemporal MCM considering one temporal neighbour and two spatial neighbours along three dominant axis directions while computing MCM as will be discussed next. We are motivated to consider three spatiotemporal neighbours by the optimality of natural 1:2 ratio of time-space motion correlation as established in OTSR.

Let a sequence of motion frames be represented by a function $M_t(x,y)$ such that (x,y) points to the spatial location at t-th motion frame. MCM Γ is a tabulation of how often different quadruples of motion vectors $(\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{v}_4)$ occur over the frame sequence such that $\vec{v}_1$ is associated with a pixel $M_t(x,y)$ and its neighbour $M_{t-1}(x,y)$, $M_{t-1}(x-1,y)$ and $M_{t-1}(x,y-1)$ are associated with $\vec{v}_2$, $\vec{v}_3$ and $\vec{v}_4$ respectively. During the motion regeneration of a point, we use the motion vectors of its spatio-temporal neighbourhood (Figure 3) in the previous motion frame to predict the motion vector of the point in the current motion frame using MCM.

3.3 Generation of Motion Frames

Given a seed motion frame M_0 and motion co-occurrence matrix Γ, we would like to generate a sequence of motion frames $M_1, M_2, \ldots, M_\tau$ where τ is the length of the synthesized texture. We considered a few alternate ways to compute future motion frames. Given $M_{t-1}(x,y)$, $M_{t-1}(x-1,y)$ and $M_{t-1}(x,y-1)$, $M_t(x,y)$ can be chosen –

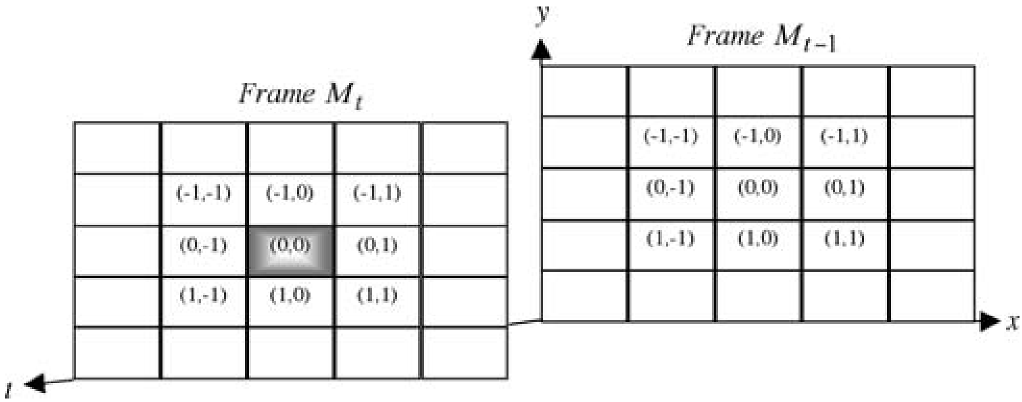


Figure 3: Spatiotemporal neighbourhood of point $M_t(0,0)$. The spatiotemporal neighbouring points considered for computing MCM are $M_{t-1}(0,0)$, $M_{t-1}(0,-1)$ and $M_{t-1}(-1,0)$.

- randomly i.e. $M_t(x,y)$ can be assigned any motion vector independent of its spatiotemporal neighbours. However such a selection will make the texture look too random.
- by using Maximum Likelihood (ML) criterion i.e.

$$M_t(x,y) = \underset{\vec{v}}{\arg\max}\ \Gamma(\vec{v}, M_{t-1}(x,y), M_{t-1}(x-1,y), M_{t-1}(x,y-1)). \tag{3}$$

However, this makes the selection process too much deterministic. The principal of synthesis deserves some degree of randomness that is lost while considering such a selection process.

- by choosing between too random and too deterministic i.e. a selection process that does not eliminate the possibility of selecting the most likely motion vector and on the other hand does not eliminate the possibility of selecting a random vector as well. We follow this criteria for selecting a motion vector for $M_t(x,y)$. Given the co-occurrence matrix Γ we compute a corresponding Probability Distribution Function (PDF) Γ_{pdf} and Cumulative Distribution Function (CDF) Γ_{cdf} as

$$\left.\begin{aligned} \Gamma_{pdf}(\vec{v}_1,\vec{v}_2,\vec{v}_3,\vec{v}_4) &= \frac{\Gamma(\vec{v}_1,\vec{v}_2,\vec{v}_3,\vec{v}_4)}{\sum\limits_{\forall\vec{v}}\Gamma(\vec{v},\vec{v}_2,\vec{v}_3,\vec{v}_4)} \\ \Gamma_{cdf}(\vec{v}_1,\vec{v}_2,\vec{v}_3,\vec{v}_4) &= \sum_{\vec{v}\le\vec{v}_1}\Gamma_{pdf}(\vec{v},\vec{v}_2,\vec{v}_3,\vec{v}_4) \end{aligned}\right\}, \tag{4}$$

where the symbol ≤ is used to denote the order of the motion vectors for indexing the co-occurrence matrix. We are now interested in predicting $M_t(x,y)$, given the neighbouring motion vectors $M_{t-1}(x,y)$, $M_{t-1}(x-1,y)$ and $M_{t-1}(x,y-1)$. $M_t(x,y)$ is computed according to the following steps – (i) A random number $0 < \Re \le 1$ is generated using a uniform distribution. (ii) $M_t(x,y)$ is set to $\vec{v}_j$ such that

$$\left.\begin{aligned} &\Gamma_{cdf}(\vec{v}_j, M_{t-1}(x,y), M_{t-1}(x-1,y), M_{t-1}(x,y-1)) \ge \Re \\ &\forall_{\vec{v}\le\vec{v}_j}\ \Re > \Gamma_{cdf}(\vec{v}, M_{t-1}(x,y), M_{t-1}(x-1,y), M_{t-1}(x,y-1)) \end{aligned}\right\}. \tag{5}$$

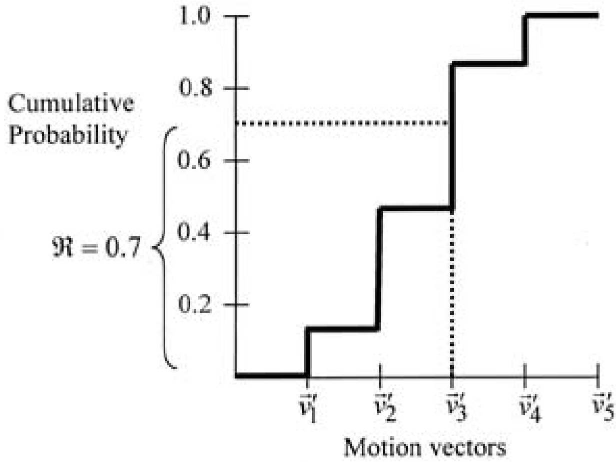


Figure 4: A graphical demonstration of motion selection process for $M_t(x,y)$. The graph shows the cumulative probability distribution of possible motion vectors given neighbouring motion vectors $M_{t-1}(x,y)$, $M_{t-1}(x-1,y)$, $M_{t-1}(x,y-1)$. For random number $\Re = 0.7$, $M_t(x,y)$ is set to $\vec{v}'_3$.

A graphical demonstration of this motion selection process for $M_t(x,y)$ is presented in Figure 4. This selection process does not eliminate the possibility of selecting most likely motion vectors as it is guided by Γ_{cdf} and on the other hand it does not eliminate the possibility of selecting a random vector. This point by point motion generation process is repeated to generate the motion frames in the order $M_1, M_2, \ldots, M_\tau$.

3.4 Generation of Image Frames

Given a seed image frame f_0 and the sequence of motion frames $M_1, M_2, \ldots, M_\tau$, we would like to generate a sequence of image frames $f_1, f_2, \ldots, f_\tau$, where τ is the length of the synthesized texture. Generation of f_t from f_{t-1} using M_t can be carried out pixel by pixel. The motion vector $M_t(x,y)$ associated with a pixel (x,y) in frame f_t, point to the source pixel in the frame f_{t-1} from where it moved. Thus the easiest way to regenerate the image frame is to assign the intensities of the source pixels in f_{t-1} to the destination pixel in f_t as guided by motion vectors of M_t. However, due to randomness in the motion regeneration step some erroneous motion vectors are generated. Thus a straightforward copy of intensity from the source pixel will create a noisy visual texture and also propagate this resulting error in future image frames.

In order to eliminate noisy motion vectors we perform group wise motion filtering. Our motivation to group wise filtering comes from the motion uniformity principle of dynamic textures which states that the majority of the pixels of a small spatial group are supposed to undergo uniform displacement in a uniform direction. To carry out group wise filtering, the image frame is divided into a set β of non overlapping, equal size small blocks. Motion uniformity is assumed to hold for each of these blocks. For each block $\sigma \in \beta$, group wise motion filtering is carried out using median filtering as –

$$\forall_{p \in \sigma} M_t(p) = median(\forall_{p' \in \sigma} M_t(p')) . \tag{6}$$

After completing motion filtering, regeneration of the image frame f_t is carried out by copying the intensities of the source pixels in f_{t-1} as guided by motion vectors of M_t.

Although block by block filtering eliminates noisy motion vectors, it leaves some block artefacts i.e. intensity discontinuity among neighbouring blocks in the synthesized texture. As motion filtering is performed on very small sized blocks (e.g. 2x2), we are able to use simple block artefact removal filtering techniques although a large number of filtering approaches to eliminate block artefacts exist (Meier *et al*, 1999) in the literature. We use mean filtering to remove block artefacts. Given f_t , the filtering is carried out as

$$I_t(p) = \frac{1}{|\theta(p)|} \sum_{\forall i \in \theta(p)} f_t(i) , \tag{7}$$

where I_t is the filtered image and $\theta(p)$ represents the spatial neighbourhood surrounding pixel p. Such a filtering makes the synthesized texture look bit blurred. However such blurriness is acceptable with non-rigid dynamic textures.

4. EXPERIMENTAL RESULTS AND DISCUSSION

In this section we present the outcomes of our proposed synthesis technique. The dynamic textures used in the experiments are a set of low quality Szummer dynamic textures (Szummer and Picard, 1996). Motion estimation plays the key role in the overall synthesis process. The meaning of motion distribution statistics and regeneration of image frames strictly depends on the genuineness of motion estimation and we present a set of experiments to establish the authenticity of estimated motion vectors in Section 4.1. The image sequences of the synthesized textures and some experimental results on how representative the synthesized textures are of their seeds are presented in Section 4.2.

4.1 Motion Estimation Results

Motion estimation is the key step of our proposed synthesis framework and the correctness of the estimated motion needs to be established before applying the synthesis. For this experiment we choose a set of dynamic texture sequences including smoke, fire, escalator, and spiral water where the image sequences themselves provide strong visual cues about flow directions so that we can visually evaluate the correctness of the estimated motion. The choice of textures is motivated by the diversity of their types including fluid, gaseous, rigid and fire type non-rigid sequences. Although we conducted experiments with different correlation and search windows, results are cited for a correlation window of size 9×9, search window size of 8×8 and histogram size of 2×4 as best results were obtained with these parameter values.

Figure 5 shows the motion vectors computed using our proposed algorithm. The image sequences themselves provide strong visual cues about flow directions. Thus we can visually evaluate the correctness of the results. In general, despite the diversity of texture types, the motion field estimates computed by our algorithm are consistent with human observations. The smoke sequence indicates that there is a wind from left to right as clearly portrayed by the motion field computed from smoke sequence. Another important observation is the presence of some background motion in the motion field of the smoke sequence. This is due to presence of fluid superimposed over the background. The escalator sequence indicates an upward movement and the motion field also agrees with this visual observation. Due to the movement of reflected edges, bands of flow vectors computed by our algorithm can be observed on both sides of the escalator. The quality of motion vectors computed from fire sequence can be evaluated visually observing their direction on the edges. Our algorithm did not perform as well for the spiral water sequence because the motion field is unable to portray visually the counter clockwise flow directions. This is partially due to the low quality of the images.

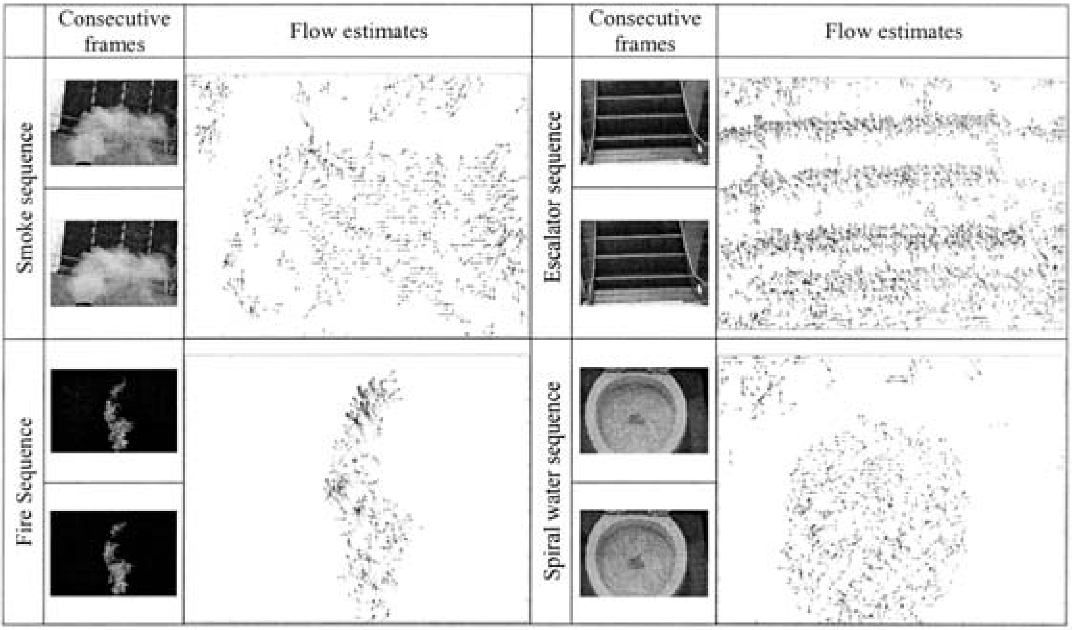


Figure 5: Motion estimates for different dynamic textures computed between two consecutive frames using proposed motion estimation algorithm.

We have also conducted a classification experiment to verify how well our proposed motion field estimates can identify dynamic textures and also to establish the superiority of our proposed technique over the STAR model. We used a set of ten different types of dynamic textures (Figure 6) for the classification experiment. For each texture type the image sequence of 100 frames was partitioned into a total of ten video clips each containing ten motion frames and MCMs were computed from each of these video clips. Classification was performed on the entire database of 100 video clips using the *k*-NN classifier. A destination class for a video is decided based on majority wins rule. In case of a tie, the video is classified into an undecided group. We computed distance between the image sequences in terms of their MCM using Kullback–Leibler (KL) divergence

Figure 6: Representative video clips for the ten different dynamic textures used in the experiments.

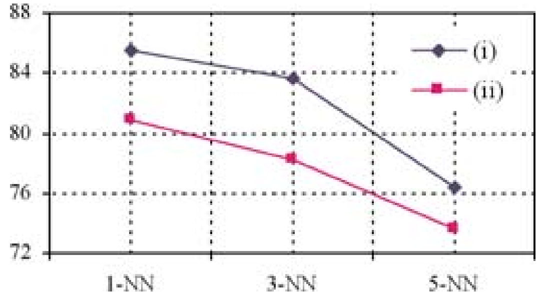


Figure 7: Classification accuracy using motion vectors computed by (i) our proposed algorithm and (ii) STAR model for different -NN classifier.

(Rahman and Murshed, 2007) that measures the amount of information lost when two motion probability distributions replace each other.

The classification results are presented in Figure 7 using motion fields obtained by the proposed method and the STAR model. For the sake of fairness the same set of parameters (correlation window, destination window etc.) were used for both of the algorithms. Results were obtained only

(a) Seed smoke sequence

(b) Synthesized image frames computed from filtered motion frames

(c) Synthesized image frames after block artifact removal.

Figure 8: Synthesis steps of the smoke sequence

(a) Seed fire sequence

(b) Synthesized image frames computed from filtered motion frames

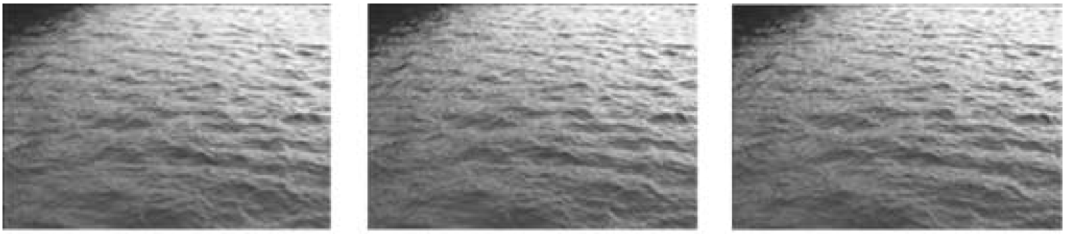
(c) Synthesized image frames after block artifact removal.

Figure 9: Synthesis steps of the fire sequence

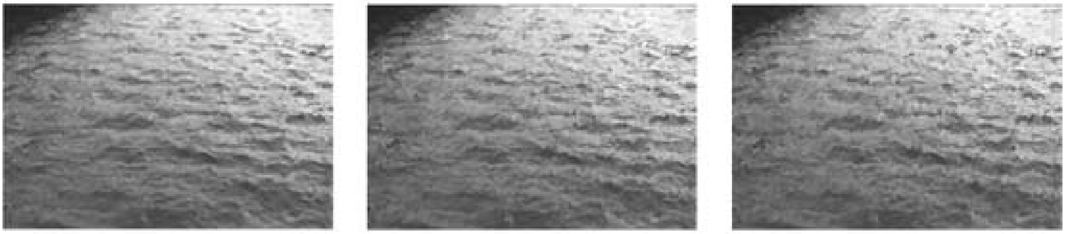
for odd values of k where the majority is unarguably decidable. Although it is natural for any majority win k-NN classifier to degrade accuracy rate slowly with k, the degradation rate in Figure 7 is significantly high due to the limited number of representative video clips per class. Classification accuracy level obtained using optical flows of the proposed technique is relatively 5.42% better (across all $k = 1, 3$, and 5) than that of the STAR model. This better classification result is due to the fact that motion vectors obtained using our algorithm are more representative of true motion than that obtained using the STAR model. Thus we conclude that our proposed motion estimation algorithm estimates true motion for dynamic textures.

4.2 Synthesis Results

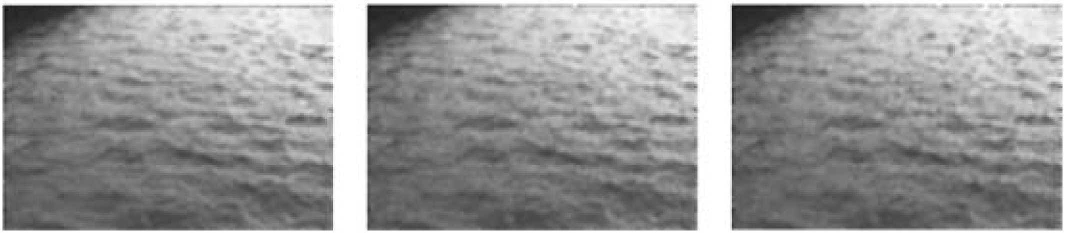
We have synthesized a diverse set of dynamic textures including fire, smoke, boiling water, plastic and river sequence using our proposed synthesis technique. Each of the grey scale seed sequences of these textures have an image resolution 170 × 116 and we considered 20 image frames from each of them for computing motion frames and their spatiotemporal MCM. Motion estimation was conducted using the parameters mentioned in Section 4.1. For computing spatiotemporal MCM we considered a 4-dimentional array with the last three dimensions indexed by motion vectors of neighbouring pixels of the pixel whose motion vector indexes the first dimension. While regenerating the



(a) Seed river sequence



(b) Synthesized image frames computed from filtered motion frames



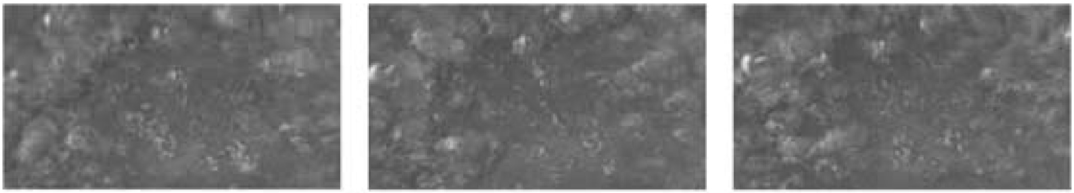
(c) Synthesized image frames after block artifact removal.

Figure 10: Synthesis steps of the river sequence

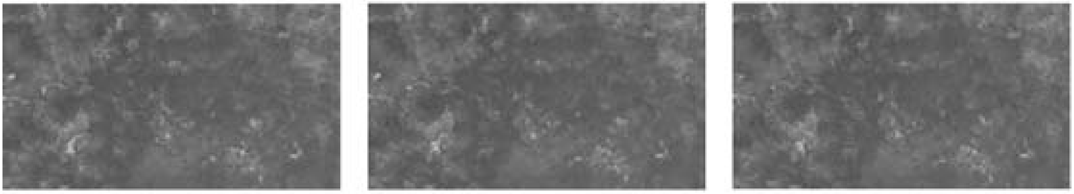
motion frames we considered uniform distribution for generating a random number $\Re$. For motion filtering we used a window size of 2×2. We have synthesized 30 image frames of spatial resolution 170×116 for each texture. All the experiments were conducted on MATLAB 6.5.1.

Figures 8 to 12 show the behaviour of our proposed technique while synthesizing the above-mentioned dynamic texture sequences. In each case, on the first row we show a few images from the original dataset, on the second row we show the synthesized image sequences computed from filtered motion frames and on the third row we show the image sequences after removing block artefacts. For each case block artefacts are clearly visible in all image frames after synthesizing them from filtered motion frames. In order to eliminate these artefacts we apply mean filtering. Mean filtering eliminates block artefacts as is visible from the final row in each figure. Such filtering makes the image sequences look a bit blurred although it is unnoticeable for dynamic textures. Our proposed method produces visually promising textures for non-rigid sequences like smoke, fire, and boiling water. For sequences like plastics and rivers that are not strictly non-rigid our method did not perform as well.

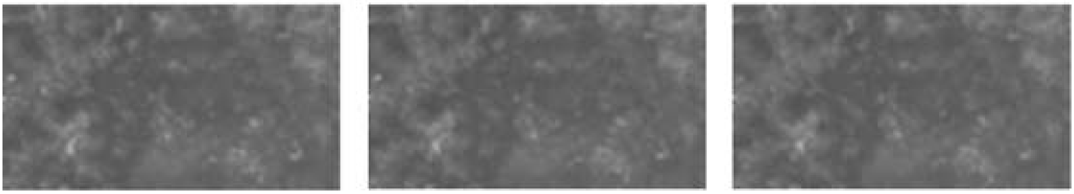
Evaluation of quality of a synthesized dynamic texture needs to address using two aspects – (i) Visual quality and (ii) Dynamic quality. In order to evaluate the visual quality of the synthesized texture we computed grey-level histograms of both seed and synthesized dynamic textures as



(a) Seed boil sequence



(b) Synthesized image frames computed from filtered motion frames



(c) Synthesized image frames after block artifact removal.

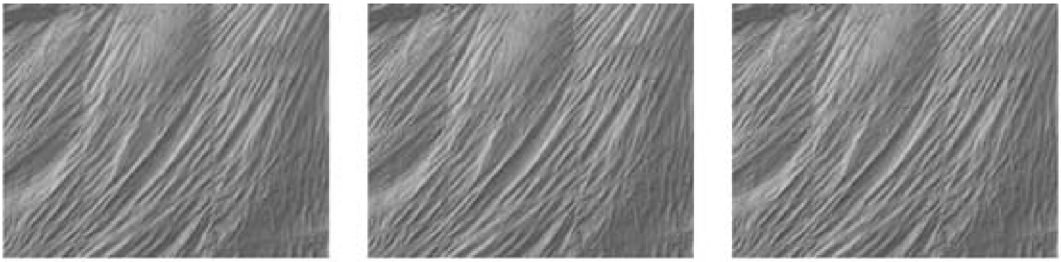
Figure 11: Synthesis steps of the boil sequence

presented in Figures 13 to 17. It can be observed that histograms of the synthesized dynamic textures closely matches that of their seed textures which clearly establishes the fact that, visually the textures are well reproduced.

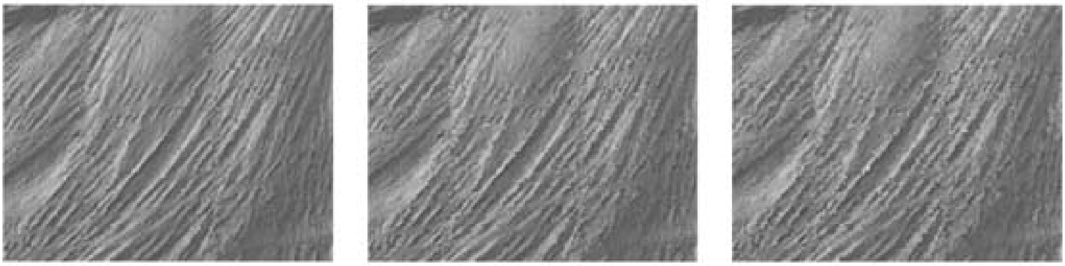
In order to evaluate the quality of dynamics in the synthesized dynamic textures, we need to assess how well the computed motion frames follow motion distribution of the seed sequence. We thus conducted some empirical analysis on how representative the motion distribution of the synthesized textures is of their seeds. We computed dissimilarity between each synthesized texture and actual texture in terms of their MCM using KL divergence and results are presented in Figure 18. It can be observed that the dissimilarity between synthesized texture and its seed is very small compared to that between the synthesized texture and other textures. It establishes the fact that synthesized textures reproduce the dynamics of the original texture.

5. CONCLUSION

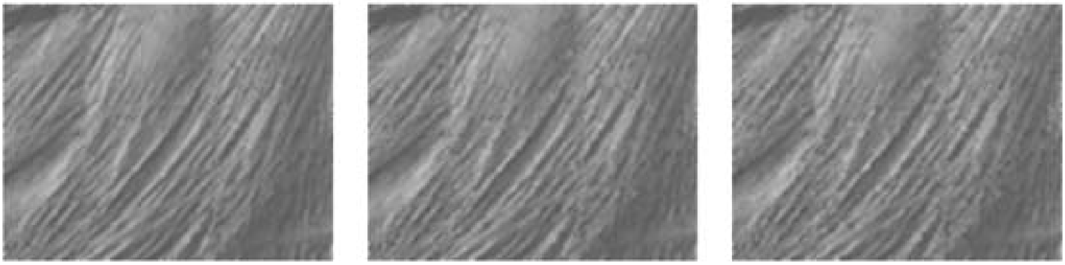
In this paper we explored the potential of synthesizing dynamic textures using underlying motion distribution statistics. More precisely we used Motion Co-occurrence Matrix (MCM) to encode the dynamics of a texture and reproduced texture having identical motion distribution. We developed a motion estimation algorithm suitable for synthesis applications and established the correctness of motion estimates empirically. Our proposed synthesis technique produces visually promising textures for short length strictly non-rigid sequences. Experimental results establish the fact that the synthesized textures both visually and dynamically reproduce the original sequences.



(a) Seed boil sequence

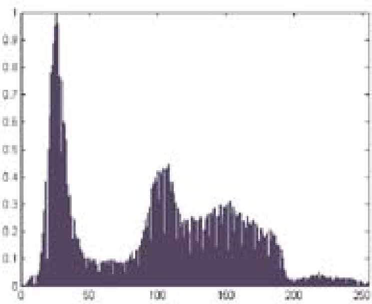


(b) Synthesized image frames computed from filtered motion frames

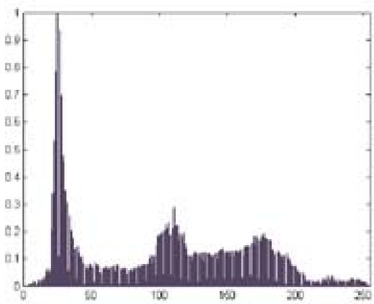


(c) Synthesized image frames after block artifact removal.

Figure 12: Synthesis steps of the plastic sequence

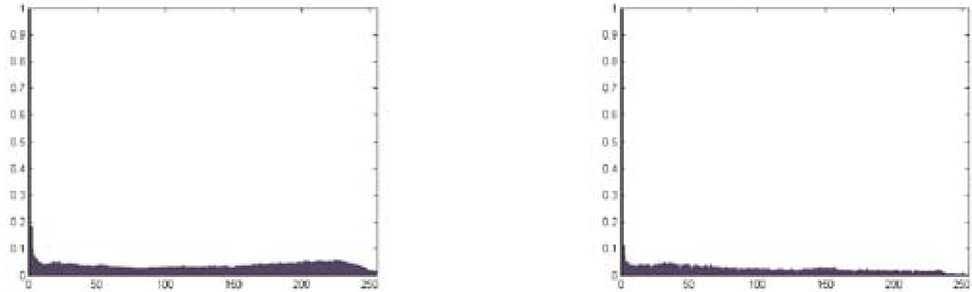


(a)



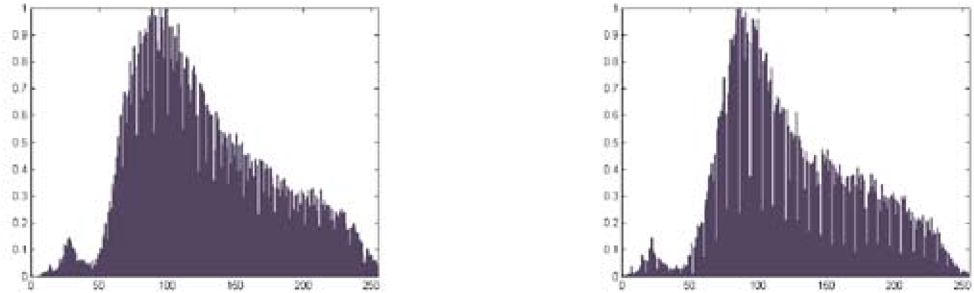
(b)

Figure 13: Gray-level histogram of (a) original smoke sequence and (b) synthesized smoke sequence.



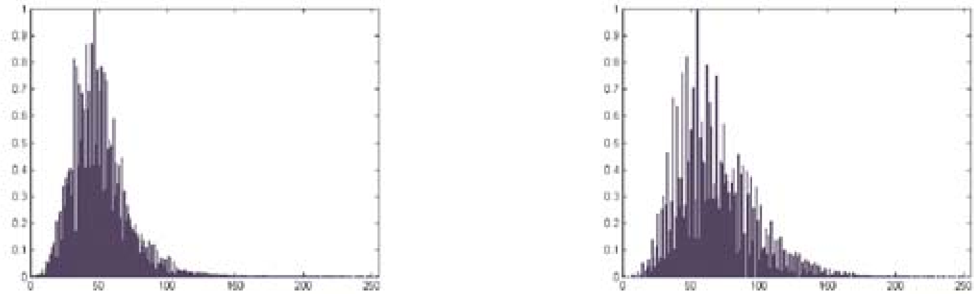
(a) (b)

Figure 14: Gray-level histogram of (a) original fire sequence and (b) synthesized fire sequence.



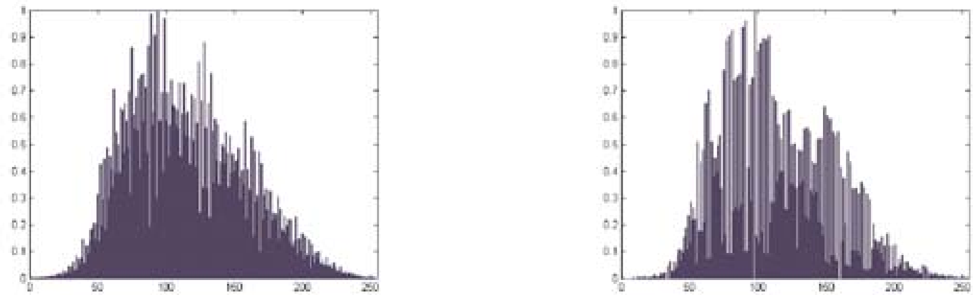
(a) (b)

Figure 15: Gray-level histogram of (a) original river sequence and (b) synthesized river sequence.



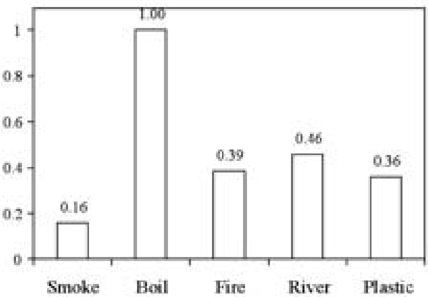
(a) (b)

Figure 16: Gray-level histogram of (a) original boil sequence and (b) synthesized boil sequence.

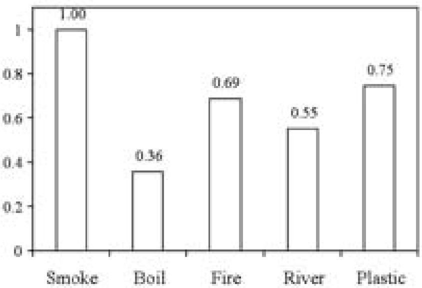


(a) (b)

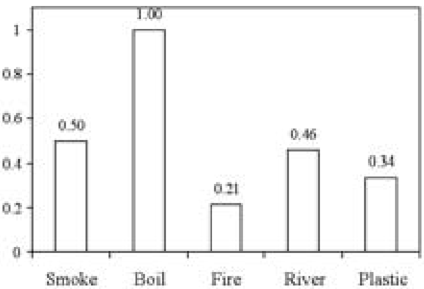
Figure 17: Gray-level histogram of (a) original plastic sequence and (b) synthesized plastic sequence.



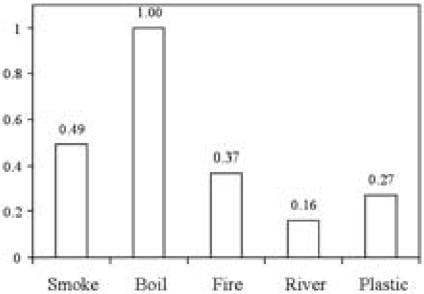
(a) Normalized distance between synthesized smoke and actual textures



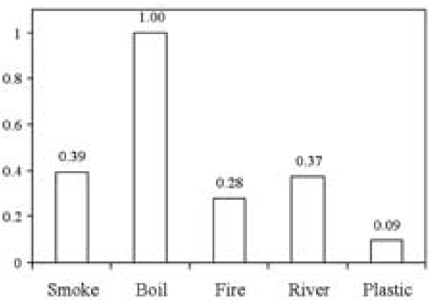
(b) Normalized distance between synthesized boil and actual textures



(c) Normalized distance between synthesized fire and actual textures



(d) Normalized distance between synthesized river and actual textures



(e) Normalized distance between synthesized plastic and actual textures

Figure 18: Distance between synthesized textures and actual textures in terms of motion co-occurrence statistics.

Our proposed technique, however, has some inherent limitations. It is not suitable to produce long sequences as the spatial constraints to keep the appearance as well as dynamics is limited. Our proposed technique also works under the constraint that the synthetic image sequences have the same resolution of the seed image sequence. Finally our proposed technique models only the dynamics and a mapping is done from dynamics to appearance which could be handled well if the model incorporates appearance as well. Altogether there is still room for improvement and we look forward to working on these issues in future.

REFERENCES

ABRAHAM, B., CAMPS, O. and SZNAIER, M. (2005): Dynamic texture with Fourier descriptors, In Texture 2005: *Proceedings of the 4th International Workshop on Texture Analysis and Synthesis*, 53–58.

BAR-JOSEPH, Z., EL-YANIV, R., LISCHINSKI, D. and WERMAN, M. (2001): Texture mixing and texture movie synthesis using statistical learning, *IEEE Transactions on Visualization and Computer Graphics*, 7(2): 120–135.

BEREZAIT, D., HERLIN, I. and YOUNES, L. (2000): A generalized optical flow constraint and its physical interpretation, *Proceedings IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

BOUTHEMY, P. and FABLET, R. (1998): Motion characterization from temporal cooccurrences of local motion-based measures for video indexing, *Proc. ICPR*, 1: 905–908, Brisbane.

CAMPBELL, N.W., DALTON, C., GIBSON, D. and THOMAS, B. (2002): Practical generation of video textures using the auto-regressive process, *British Machine Vision Conference*, 434-443.

CORPETTI, T., MEMIN, E. and PEREZ, P. (2002): Dense motion analysis in fluid imagery, *Proceedings European Conference on Computer Vision (ECCV)*, 676-691.

DORETTO, G., JONES, E. and SOATTO, S. (2004): Spatially homogeneous dynamic textures, *Proc. of the Eur. Conf. on Comp. Vision (ECCV)*.

EBERT, D., CARLSON, W. and PARENT, R. (1994): Solid spaces and inverse particle systems for controlling the animation of gases and fluids, *The Visual Computer*, 10(4):179–190

EBERT, D.S. and PARENT, R.E. (1990): Rendering and animation of gaseous phenomena by combining fast volume and scanline A-buffer techniques, In Forest Baskett, editor, Computer Graphics *SIGGRAPH '90 Proceedings*, 24: 357–366, August.

EDWARDS, D. (2002): Coherence-based temporal texture synthesis using optical flow information, Master's thesis, University of Illinois at Urbana-Champaign.

EDWARDS, D., CHANG, J., SHI, L. and YU, Y. (2003): Motion field estimation for temporal textures, *Digital Image Computing - Techniques and Applications*, December.

EFROS, A.A. and FREEMAN, W.T. (2001): Image quilting for texture synthesis and transfer, *Proceedings of SIGGRAPH*, 341– 346.

FABLET, R. and BOUTHEMY, P. (2003): Motion recognition using nonparametric image motion models estimated from temporal and multiscale co-occurrence statistics, *IEEE transaction on pattern analysis and machine intelligence*, 25(12): 1619–1624, December.

FABLET, R., BOUTHEMY, P. and PEREZ, P. (2002): Nonparametric motion characterization using casual probabilistic models for video indexing and retrieval, *IEEE Trans. on Image Processing*, 11: 393–407.

FAZEKAS, S. and CHETVERIKOV, D. (2007): Analysis and performance evaluation of optical flow features for dynamic texture recognition, *Special Issue of Signal Processing: Image Communication Journal on Content-Based Multimedia Indexing*, 22(7-8): 680-691.

FITZGIBBON, A.W. (2001): Stochastic rigidity: image registration for nowhere-static scenes, *Proceedings ICCV*, 662-669.

FOURNIER, A. and REEVES, W.T. (1986): A simple model of ocean waves, In David C. Evans and Russell J. Athay, editors, Computer Graphics *SIGGRAPH '86 Proceedings*, 20(4): 75–84, August.

KWATRA, V., SCHODL, A., ESSA, I., TRUK, G. and BOBICK, A. (2003): Graphcut textures: Image and video synthesis using graph cuts, *Proceedings SIGGRAPH*, 277-286.

LARSEN, R., CONRADSEN, K. and ERSBOLL, B.K. (1998): Estimation of dense image flow fields in fluids, *IEEE Transaction on Geoscience and Remote Sensing*, 256-264.

LIU, C., LIN, R. and AHUJA, N. (2005): Modeling dynamic textures using subspace mixtures, *IEEE International Conference on Multimedia & Expo (ICME)*.

MEIER T., NGAN, K.N. and CREBBIN, G. (1999): Reduction of blocking artifacts in image and video coding, *IEEE Transactions on Circuits and Systems for Video Technology*, 9: 490-500, April.

MEMIN, E. and PEREZ, P. (1999): Fluid motion recovery by coupling dense and parametric vector fields, *Proceedings IEEE International Conference on Computer Vision (ICCV)*.

PEACHEY, D.R. (1986): Modeling waves and surf, In EVANS, D.C. and ATHAY, R.J. editors, Computer Graphics *SIGGRAPH '86 Proceedings*, 20(4): 65–74, August.

PEH, C.H. and CHEONG, L.-F. (1999): Exploring video content in extended spatiotemporal textures, In *Proc. 1st European workshop on Content-Based Multimedia Indexing*, 147–153, Toulouse, France.

PÉTERI, R. and CHETVERIKOV, D. (2004): Qualitative characterization of dynamic textures for video retrieval, In *Proc. International Conference on Computer Vision and Graphics (ICCVG 2004)*, Warsaw, Poland.

PÉTERI, R. and CHETVERIKOV, D. (2005): Dynamic texture recognition using normal flow and texture regularity, *Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA 2005), LNCS*.

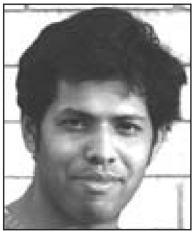
POLANA, R. and NELSON, R. (1992): Recognition of motion using temporal texture, *Proc. CVPR '92*, 129–134.

RAHMAN, A. and MURSHED, M. (2007): A temporal texture characterization technique using block-based approximated motion measure, *IEEE Transaction on Circuits and Systems for Video Technology*, 17(10): 1370–1382, October

REEVES, W.T. (1983): Particle systems – a technique for modeling a class of fuzzy objects, *ACM Trans. Graphics*, 2:91–108, April.

REEVES, W.T. and BLAU, R. (1985): Approximate and probabilistic algorithms for shading and rendering structured particle systems, In BARSKY, B. A. editor, Computer Graphics *SIGGRAPH '85 Proceedings*, 19: 313–322, July.
SCHODL, A., SZELISKI, R., SALESIN, D. and ESSA, I. (2000): Video textures, *Proceedings ACM SIGGRAPH*.
SHI, Y.Q. and SUN, H. (2000): Image and video compression for multimedia engineering, CRC Press.
SIMS, K. (1990): Particle animation and rendering using data parallel computation, In Forest Baskett, editor, Computer Graphics *SIGGRAPH '90 Proceedings*, 24: 405–413, August.
SOATTO, S., DORETTO, G. and WU, Y.N. (2001): Dynamic textures, *Proc. ICCV*.
STAM, J. and FIUME, E. (1993): Turbulent wind fields for gaseous phenomena, In KAJIYA, J.T. editor, Computer Graphics *SIGGRAPH '93 Proceedings*, 27: 369–376, August.
STAM, J. and FIUME, E. (1995): Depicting fire and other gaseous phenomena using diffusion processes, In COOK, R. editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, 129–136. ACM SIGGRAPH, AddisonWesley, August.
SZUMMER, M. and PICARD, R.W. (1996): Temporal texture modeling, *Proc. IEEE International Conference on Image Processing* (ICIP).
WANG, Y. and ZHU, S. (2002): A generative method for textured motion: Analysis and synthesis, *Proc. ECCV*, 583-598.
WANG, Y. and ZHU, S.C. (2003): Modeling textured motion: particle, wave and sketch, *Proceedings IEEE International Conference on Computer Vision (ICCV)*, 1: 213–220.
WANG, Y. and ZHU, S.C. (2004a): Analysis and synthesis of textured motion: particles and waves, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10): 1348–1363, October.
WANG, Y.Z. and ZHU, S.C. (2004b): Modeling Complex Motion by Tracking and Editing Hidden Markov Graphs, *Proc. CVPR*.
WEI, L.Y. and LEVOY, M. (2000): Fast texture synthesis using tree structured vector quantization, *Proc. ACM SIGGRAPH*.
YUAN, L., WEN, F., LIU, C. and SHUM, H.–Y. (2004): Synthesizing dynamic texture with closed-loop linear dynamic system, *Proc. ECCV*.
ZHU, S.C., GUO, C., WANG, Y. and XU, Z. (2005): What are textons? *International Journal of Computer Vision, special issue on texture,* 62(1/2): 121–143, April/May.

BIOGRAPHICAL NOTES



Ashfaqur Rahman

Ashfaqur Rahman received his B.Sc.(Hons) degree in Computer Science and Engineering in 2001 from Bangladesh University of Engineering and Technology (BUET), Bangladesh and a Ph.D. degree in Computing from Monash University, Australia in 2007.

He is currently serving as an Assistant Professor in the Computer Science Department at American International University Bangladesh, where his major research interests are in the fields of multimedia signal processing and communication, data mining, ad-hoc networks and artificial intelligence. He has published more than 20 peer-reviewed journal articles and conference papers.

Dr Rahman is the recipient of numerous academic awards including the International Postgraduate Research Scholarship (IPRS), Monash Graduate Scholarship (MGS) and FIT Dean Scholarship by Monash University Australia.

Manzur Murshed

Manzur Murshed received his B.Sc.Eng.(Hons.) degree in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET), Bangladesh, in 1994 and Ph.D. degree in Computer Science from the Australian National University, Australia, in 1999.

He is an Associate Professor and the Head of the Gippsland School of Information Technology at Monash University, Australia, where his major research interests are in the fields of multimedia signal processing and communications, parallel and distributed computing, simulations, and multilingual systems development. He received many research grants *including a prestigious Australian Research Council (ARC) Discovery Projects grant. He has published more than 100 peer-reviewed journal articles, book chapters, and conference papers.*

Dr Murshed is the recipient of numerous academic awards including the University Gold Medal by BUET. He recently received the inaugural Faculty of Information Technology Award for Excellence in Research for Early Career Researchers and was one of the nominees for the Vice Chancellor's Award for Excellence in Research for Early Career Researchers.