TEXTURE SYNTHESIS METHOD FOR GENERIC VIDEO SEQUENCES

Patrick Ndjiki-Nya, Christoph Stüber, and Thomas Wiegand

Fraunhofer Institute for Telecommunications - Heinrich-Hertz-Institut
Berlin, Germany
{ndjiki/wiegand}@hhi.de

ABSTRACT

An effective texture synthesis method is presented that is inspired by the work of Kwatra et al. [1]. Their algorithm is non-parametric and patch-based. Blending between overlapping patches is optimized using graph cut techniques. We generalize the initial approach [1] to achieve a new synthesis algorithm that yields improved results for a much larger class of natural video sequences. For that, two major extensions have been provided: 1) the ability to handle constrained texture synthesis applications and 2) robustness against global camera motion. Constrained synthesis thereby refers to integrating synthetic textures into natural video sequences, as opposed to unconstrained texture synthesis, where (infinite) spatio-temporal extensions of single textures are generated. Camera motion compensation enables applicability of the synthesis algorithm to video sequences with a moving camera. The results presented in this paper show that the proposed improvements yield significant subjective gains compared to the initial algorithm.

***Index Terms*—** Video, constrained, texture, synthesis, graph-cut, GMC

1. INTRODUCTION

Texture synthesis consists in generating a synthetic texture that is typically objectively different from a given texture example but it is perceptually similar. Typical applications of texture synthesis algorithms are special effects for cinema and television, computer-generated animation, computer games, and video restoration.

The major problems that have to be tackled in any texture synthesis process are roughly two-fold. The first challenge steers the accuracy of the synthetic textures and relates to the proper estimation of the underlying stochastic process of a given texture based on a finite sample of it. The second challenge is the efficient sampling of a high-dimensional probability density function (pdf) to generate new textures from a sample [2] determining the computational complexity of the texture generation procedure.

Texture synthesis approaches can be divided into two categories: parametric and non-parametric methods. Parametric synthesis approaches approximate the pdf by which the texture is assumed to be modeled using a compact model with a fixed parameter set [3]. Non-parametric synthesis approaches do not explicitly estimate the pdf by which the texture is assumed to be modelled. They rather measure the latter from the texture example, which can be a 2D image or a video signal. Non-parametric approaches typically formulate the texture synthesis problem based on Markov Random Field (MRF) theory [1],[2]. The generative stochastic process that is used as the texture model is assumed to be both local and stationary in the MRF context. Non-parametric approaches can be sample or patch-based. Sample-based algorithms update the synthetic texture sample-wise [2] while patch-based approaches operate a patch-wise update [1], i.e. a set of samples is updated simultaneously. Not only do non-parametric synthesis approaches typically yield better synthesis results than parametric algorithms, also can they be successfully applied to a much larger variety of textures [1].

In this work, a generalization of the approach by Kwatra et al. [1] is presented. Constrained texture synthesis under moving camera conditions is enabled by our approach. In the remainder of the paper, the initial synthesis framework by Kwatra et al. is introduced (Sec. 2). An in-depth description of the proposed improvements is given in Sec. 3, while experimental results are presented in Sec. 4.

2. VIDEO SYNTHESIS USING GRAPH CUTS

The synthesis algorithm developed by Kwatra et al. [1] can a priori be applied to plane (2D) and volumetric textures (2D+t). It is non-parametric and can thus render a large variety of video textures. The synthetic texture is updated patch-wise by disposing the patches in an overlapping manner. The originality of the approach by Kwatra et al. resides in the fact that it formulates the texture synthesis problem as a graph cut issue. Hence, the optimal seam between two overlapping patches, which is the seam yielding the best possible MRF likelihood among all possible seams for the given overlap, can be computed, thus minimizing subjectively annoying edges at patch transitions [1].

2.1. Sampling Procedure

Given a texture sample $\mathcal{T}$ and an empty lattice $\mathcal{S}$ to fill in, the sub-patch matching approach proposed by Kwatra et al. is used. The latter approach consists in placing patches in $\mathcal{S}$ in an overlapping manner, where the first patch is typically selected at random in the texture sample. The patches are

1-4244-1437-7/07/$20.00 ©2007 IEEE

thereby of a predefined size, typically much smaller than the texture example.

The costs for a given translation s of a sub-patch of the output texture in $\mathcal{T}$ are given in (1). v corresponds to the portion of the input overlapping the sub-patch and p is a sample location in v [1]. $|v|$ is the size of v. The output sub-patch selected for this operation is the continuation part of the last patch placed in $\mathcal{S}$ (cp. Fig. 1). (1) basically corresponds to the normalized sum of squared errors. The offset selection is operated by means of a stochastic criterion that is dependent on E [1].

$$E = \frac{1}{|v|} \sum_{p \in v} \left| \mathcal{T}(p) - \mathcal{S}(p + s) \right|^2 \qquad (1)$$

The sub-patch matching approach can be applied to stochastic and volumetric textures. It captures the local coherency of spatially unstructured textures like water, smoke, etc. The size of the sub-patch, copied from the texture sample towards $\mathcal{S}$, is chosen in a way that it is slightly larger than the overlap region in the output texture. This is done to ensure that the output texture is grown with each update patch. For volumetric textures, the video sequence is seen as a volume composed of voxels (volume elements). The patches are spatio-temporal cuboids that can be placed anywhere in the synthesized texture volume $\mathcal{S}$.

2.2 Graph Cut Formulation of Texture Synthesis

Kwatra et al. [1] propose a graph cut formulation of the problem of finding an adequate seam between overlapping patches. Once the overlap region (synthetic texture) and the continuation patch (texture sample) have been found, the graph cut algorithm determines the path from one end to the other of the overlap region that minimizes the subjective annoyance of the blending. Fig. 1 delineates the approach based on a 2D texture synthesis example. The path specifies which irregular shaped portion of the continuation patch (patch 2), found in the texture sample, is transferred to the synthetic texture. Due to the irregular shape of the copied region, blocking effects can be avoided and seamless transitions be generated. Potentially subjectively annoying artifacts of the blending are captured by an adequate cost function that is applied to any sample transition in the overlap region.

The cost function M' used by Kwatra et al. [1] constrains the optimal path determined by the graph cut algorithm. Hence, its formulation is crucial with regard to the quality of the synthesis results. The graph cut formulation of the texture synthesis problem is depicted in Fig. 2. A 5x5 grid is shown, where each of the numbered square boxes corresponds to a sample in the overlap area. The samples marked with 'A' may be seen as the overlap region in the output texture, while the samples marked with 'B' would represent the corresponding portion of the continuation patch found in the example texture. The graph cut algorithm links adjacent sample pairs via the cost function. Let 'A' stand for the source and 'B' for the sink. Some samples are then linked to sink and source with an infinite weight. Hence, a cut at these transitions is made impossible as it would yield infinite costs. This is done in order to constrain samples adjacent to sink and source to come from B and A respectively, which reflects the fact that false boundaries at transitions between overlap region and sink (or source) should be avoided. The optimal cut (red line in Fig. 2), i.e. the cut yielding minimum costs, is determined by applying adapted optimization algorithms [1].

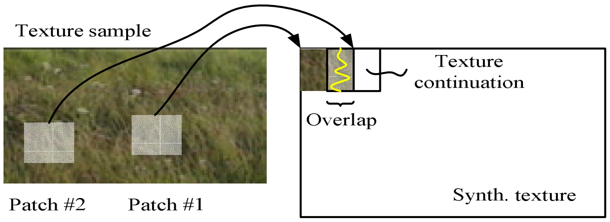

Fig. 1: Illustration of graph cut synthesis method in an unconstrained framework [1]

The cut specifies the contribution of each patch to the overlap region. For instance, in Fig. 2, the samples at the left hand side of the cut are provided by patch A, while the others come from patch B. For volumetric textures, the min-cut can be seen as a surface within the 2D+t space.

3. PROPOSED IMPROVEMENTS

The approach by Kwatra et al. [1] yields impressive synthesis results for various spatio-temporal textures with and without local motion activity. Their algorithm is, however, designed to synthesize single, autarkic textures. A further limitation is that they assume camera motion to be absent. Hence, enhancements are required to generalize the underlying algorithm by Kwatra et al. to achieve successful synthesis of natural video sequences. Synthesis of the latter is particularly critical, as synthetic textures are inserted into video scenes featuring natural textures. Therefore, even small inconsistencies might become visible if the appearance of the synthetic textures does not match the natural textures in terms of motion, sharpness, intensity, etc.

3.1 Constrained Texture Synthesis

Some application scenarios like content-based video coding require the extension of the algorithm by Kwatra et al. to a texture synthesis module with boundary constraint handling. In the context of constrained texture synthesis, missing textures can be considered as large spatio-temporal “holes” in a given video sequence that must be “filled” (cp. Fig. 3). The boundary constraint relates to the texture(s) surrounding the area to be synthesized. This constraint is taken into account in order to avoid subjectively annoying artifacts at the transition between synthetic and natural textures. Constraint texture synthesis is a somewhat complicated task as both spatial and temporal inferences are required. Inappropriate synthesizer decisions may yield annoying artifacts as flickering or spurious spatio-temporal edges.

In the constrained synthesis scenario, the input video sequence is temporally segmented as depicted in Fig. 4. The first group of pictures consists of a reference burst (R1) that temporally precedes the synthetic burst (S1). The synthetic burst is itself followed by another reference burst (R2) in temporal order. The two reference bursts and the synthetic burst give a group of bursts (GOB) R1S1R2. The reference bursts are (manually) chosen such that they contain the sample texture $\mathcal{T}$ required to synthesize the empty lattice $\mathcal{S}$ in the synthetic burst. The second GOB consists of the last reference burst of the first GOB, R2, and the next synthetic (S2) and reference (R3) bursts to give R2S2R3. Hence, an overlapping GOB structure is used. The succeeding GOBs are composed accordingly until the end of the video sequence is reached.

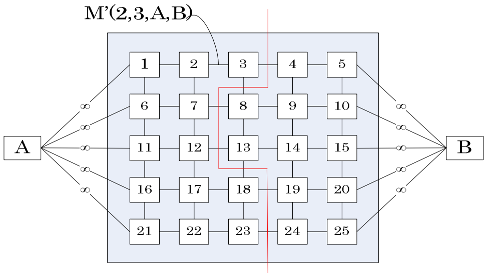

Fig. 2: Illustration of graph cut for texture synthesis

The patch placement procedure is affected in the given scenario. In fact, due to the boundary constraints, the first patch can not be selected at random from $\mathcal{T}$. The patches at the boundary of the synthesis area must be placed such that they overlap the boundary texture (cp. Fig. 3). This in turn implies that the constraint texture must be of the same class as the texture to be synthesized, which must be determined by the texture analysis method. The latter is assumed to be given as the current work focuses on texture synthesis. The graph cut algorithm is now applied to aforesaid overlap region, which yields an irregular boundary between the spatio-temporal constraint region and the synthetic video texture. This ideally decreases the perceptibility of the boundary given an adequate cost function. Irregular boundaries are also obtained in temporal direction. That is, the cut generated by the graph cut algorithm typically leads through some reference pictures. Hence the latter usually feature a small proportion of synthetic samples if they are close to an S burst, which allows a smooth transition between R and S bursts.

3.2 Temporal Alignment

Kwatra et al. implicitly assume a static camera scenario [1]. This is a very restrictive framework, as many natural video sequences feature some degree of camera motion. This constraint has to be relaxed for achieving a generic texture synthesis tool. Camera motion is typically not known a priori and requires a motion estimation process. The proposed global (camera) motion compensation (GMC) algorithm, also called temporal alignment in the following, is based on dense motion fields. Robust statistics are operated on the estimated motion vectors to derive the apparent camera motion and compensate it.

The first step of the temporal alignment algorithm consists in determining the perspective motion parameters [4],[5] describing the camera motion between adjacent pictures starting from the outmost pictures.

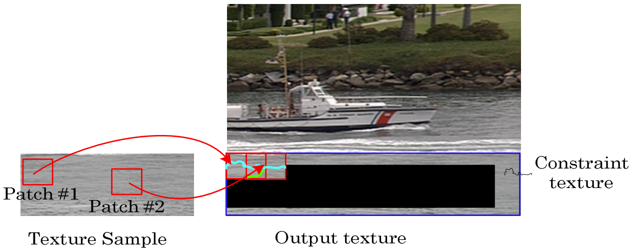

Fig. 3: Constrained 2D texture synthesis principle

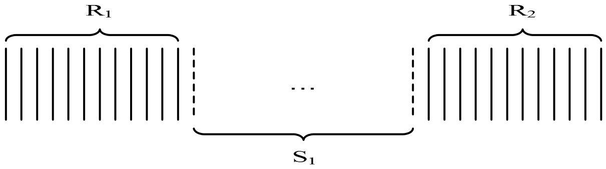

Fig. 4: Video sequence structure for texture synthesis

Once the frame to frame global motion is known, the reference time instance t_0 is shifted towards the designated frame, e.g. the mid-picture of the GOB, by accumulation of the motion parameter sets, which can be obtained by chaining single, i.e. frame-to-frame, perspective transformations.

Let F_t and F_{t+1} be two successive frames of a video sequence. Then the dense motion field between the two is first estimated using the approach by Black and Anandan [6]. The samples belonging to the background, i.e. regions without local motion activity underlying only global camera motion, are determined based on robust statistics, namely M-estimation [4]. The latter is an iterative model-fitting approach that detects outliers within a dataset a posteriori and without any prior knowledge of outlier characteristics. The observations are defined as a set of motion vectors in our specific framework, while outliers (non-background samples) can be seen as motion vectors that reveal different motion properties than the inliers (background samples). The observed motion field [6] is approximated using the perspective motion model as already explained above. This motion model is selected due to its ability to describe translation, rotation, and scaling of a planar patch in 3D as we assume this geometry for background textures. The M-estimator minimizes the influence of outliers on the model optimization by penalizing motion vectors yielding high modelling costs [4],[5]. The cost function is thereby given as the deviation between the observed [6] and the modelled dense motion field.

After alignment w.r.t. the reference time instant t_0 has been done, texture synthesis can be operated as described in the previous sections, which results in a synthetic texture w.r.t. t_0. Back-warping the synthetic pictures towards the genuine time instant yields a total of two interpolation steps per sample (warp and back-warp), which may give blurry results. Hence, the number of interpolations is minimized for improved visual quality by operating "virtual synthesis" in the warped domain. That is, each sample (texture samples, constraint and synthetic regions) is assigned a unique index within a GOB in the genuine coordinate system. Warping is applied both to the video signal and to the index maps yielding a first set of spatio-temporal index maps, $\mathcal{M}$, in the warped domain. The synthetic samples are inserted into the warped video, while their indexes are inserted into a second set of index maps, $\mathcal{M}'$, during synthesis. Finally, the synthetic samples, of which the indexes are held by $\mathcal{M}'$, are assigned to the unwarped region to be synthesized by looking up their destination in $\mathcal{M}$ at the same spatio-temporal location (x,y,t).

4. EXPERIMENTAL RESULTS

Experimental evaluations are conducted to demonstrate that the proposed generalizations of the approach by Kwatra et al. [1] entail significant perceptual gains. For that, two video sequences are used that show water with significant local motion activity within a natural video sequence. The video clips have CIF resolution (352x288), a frame rate of 15 Hz, and both feature non-zero camera motion. The reference burst length is set to 20 pictures, while the synthetic burst length is set to 40 pictures. The spatial boundary conditions are sized 16 samples each (x and y direction), while the temporal boundary condition is sized eight samples. Note that the temporal boundary condition is a subset of the reference bursts, while the spatial boundary condition is a subset of the synthetic burst. The patch size is set to 32x32x16 (height x width x temporal depth). Up to half of a patch overlaps either the spatio-temporal boundary condition and/or neighboring patches.

The synthetic video sequences are subjectively evaluated using the Double Stimulus Continuous Quality Scale (DSCQS) method [4]. That is, 10 test subjects are asked to comparatively rate the quality of a synthetic clip and the corresponding original video sequence on a scale from 0 to 100. Subjective opinion scores are obtained as a result. Perceptual degradations due to texture synthesis can thus be measured. Fig. 5 depicts the subjective evaluations obtained for the synthetic video clips with camera motion. Note that the so-called whiskers are drawn from the lower (upper) quartile to the smallest (largest) subjective score and thus cover the full span of the given data. The horizontal line within a box represents the median of the corresponding data samples. It can be seen that the synthetic clip (BP 1) generated without adequate motion compensation received significantly lower rates from the test subjects compared to the original video sequence (BP 2). Subjective ratings are significantly improved (median opinion score moves from 45 to 55), when temporal alignment is conducted (BP 3). The overlapping notches of BP 3 and BP 4 show that no statistically relevant difference can be observed by the test subjects between the reference video clip (BP 4) and the synthetic clip (BP 3). Note that the rates assigned by the test subjects to the same reference sequence (cf. BP 2 and BP 4) vary depending on the video it is compared to.

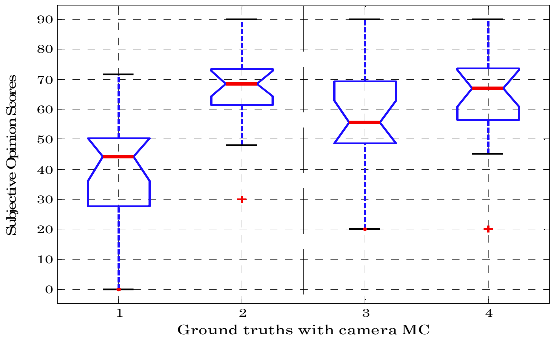

Fig. 5: Boxplots (BP) of opinion scores for constrained texture synthesis with and without camera MC. Synthetic videos, no MC (BP 1), references (BP 2), synthetic, MC (BP 3), references (BP 4)

The experiments conducted in this section show that temporal alignment and constrained texture synthesis optimizations proposed in the present work are important for the perceived quality of a synthetic video sequence. The test sequences can be viewed at our web-page http://ip.hhi.de/imagecom_G1/ImprovedTS.htm.

5. CONCLUSIONS

A graph cuts video synthesis approach for generic video sequences has been presented in this paper. The relevance of the proposed algorithm is given by the fact that it is non-parametric and patch-based, which makes it applicable to a large class of 2D and 3D textures. The basic approach proposed by Kwatra et al. [1] is generalized by extending it to realistic video sequences with regard to constrained texture synthesis and camera motion compensation.

6. REFERENCES

[1] V. Kwatra et al., "Graphcut Textures: Image and Video Synthesis using Graph Cuts", *Proc. SIGGRAPH*, pp. 277-286, 2003.

[2] L.-Y. Wei and M. Levoy, "Fast Texture Synthesis using Tree-structured Vector Quantization", *SIGGRAPH 00*, pp. 479-488, New Orleans, USA, 2000.

[3] J. Portilla and E. P. Simoncelli, "A Parametric Texture Model based on Joint Statistics of Complex Wavelet Coefficients", *International Journal of Computer Vision*, vol. 40, No. 1, pp. 49-71, 2000.

[4] J.-R. Ohm, "Multimedia Communication Technology", ISBN 3-540-01249-4, Springer, Berlin Heidelberg New York, 2004.

[5] A. Smolić, "Globale Bewegungsbeschreibung und Video Mosaiking unter Verwendung parametrischer 2-D Modelle, Schätzverfahren und Anwendungen", *PhD Thesis*, Aachen University of Technology, Germany, 2001.

[6] 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.