Priority Belief Propagation-Based Inpainting Prediction with Tensor Voting Projected Structure in Video Compression

Hongkai Xiong, *Senior Member, IEEE,* Yang Xu,Yuan F. Zheng, *Fellow, IEEE,*
Chang Wen Chen, *Fellow, IEEE*

***Abstract*—This paper presents a new video compression framework over H.264/AVC scheme that integrates our proposed structured priority belief propagation (BP)-based inpainting prediction (IP) to exploit the intrinsic nonlocal and geometric regularity in video samples. Unlike the existing edge-based inpainting adopted in lossy image coding, the optimal predictor could maintain the pixel-wise fidelity and the robust error resilience without any assistant information. Beyond the local prediction limitation of traditional intra and inter-modes, the priority BP with regularized structure priors of a spatio-temporal Markov random field is imposed on the predictor in an adaptive and more convergent sense. Specifically, the structured sparsity of the predicted macroblock region is inferred by tensor voting projected from the co-located decoded regions. In turn, the priority and visiting order of nodes are assigned according to the sets of updated beliefs as the propagation of messages. Through relatively few iterations of forward and backward process, the sparse inference of priority BP would ensure a stable marginal belief distribution on the structure and texture through updating local messages and beliefs. Within the optimal mode selection on rate-distortion optimization (RDO), the IP-mode with structured priority BP outperforms the existing vision-based approaches, and specially achieves a better objective rate-distortion performance besides visual quality. The IP-mode with structured priority BP can be applied to both I and P frames to generate low entropy residue, e.g., homogeneous visual patterns, and the computation complexity is also competitive with one iteration of sparse inference. Moreover, it behaves more resilient with an intrinsic probabilistic inference than the intra and inter-modes.**

***Index Terms*—Belief propagation, H.264/AVC, inpainting, mode selection, tensor voting.**

Manuscript received September 14, 2010; revised January 1, 2011; accepted February 17, 2011. Date of publication March 28, 2011; date of current version August 3, 2011. This work was supported in part by the National Natural Science Foundation of China, under Grants 60772099, 60928003, 60736043, and 60632040, and the Program for New Century Excellent Talents in University, under Grant NCET-09-0554. This paper was recommended by Associate Editor R. Rinaldo.

H. Xiong and Y. Xu are with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: xionghongkai@sjtu.edu.cn; xuyang0815@gmail.com).

Y. F. Zheng is with the Department of Electrical and Computer Engineering, Ohio State University, Columbus, OH 43210 USA, and also with Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zheng@ece.osu.edu).

C. W. Chen is with the Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY 14260 USA (e-mail: chencw@buffalo.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2011.2133910

I. Introduction

THE state-of-the-art video compression schemes such as H.264/AVC have recently achieved a vital efficiency. These mainstream signal processing approaches focus on exploring statistical redundancy among pixels through intra and inter prediction [1]. To achieve a better performance, more intra prediction methods, e.g., template matching [2] and texture prediction [3], have been noticed. In 2005, high-performance video coding has been initialized for further investigation where various modes are advocated to suit regions of different properties. In the corresponding key technology area software [4], a bidirectional intra prediction and separable directional transforms are absorbed [5]. Besides improving traditional coding technique such as predictive coding and transform coding, there is another major research effort under way to improve coding efficiency through exploiting visual redundancy. In a perceptual rate-distortion (R-D) sense, a reconstruction (interpolation) from incomplete data has ever been proposed with a statistical correlation between a sparsely sampled low-resolution version and high frequency contents through learning-based approaches [6]. With the evolution into high efficiency video coding (HEVC) joint project in 2010, the requirements are expected to attain bit-rate reduction of 50% at the same subjective image quality comparing to H.264/AVC.

Parallel with the traditional prediction track, several computer vision and graphics technologies have been proposed to synthesize and hallucinate missing image contents with good perceptual quality. A remarkable image inpainting is first introduced by Bertalmio *et al.* [7] by using a third-order partial difference equation to smoothly propagate information to the damaged image area from the surrounding areas in isophote directions. Recently, this kind of texture synthesis methods [8], [9] has been playing a leading effort to exploit visual redundancy for restoring missing areas with inferable information. Those can be treated in a unified manner under the framework of Markov random field (MRF) [10], and optimization algorithms, e.g., belief propagation (BP) [11], [12], are concerned as an iterative solution. These approaches solve a wide class of problems in image processing and computer vision, but are space and time consuming till convergence [13]. In this situation, the assistant information, e.g., edges, is recognized significant to guide restoration process. The work of [14] combined "texture synthesis" and "inpainting"

to prioritize the completion order using the local image structure. Pixels in the hole maintain a confidence value which influences their filling priority together with image isophotes. The structure propagation has ever been advanced as a global optimization problem [15], which preserves important structure on condition that sharp curves are outlined by the user. Mathematically, this process can be interpreted as an iterated projection of the damaged image from prior knowledge including the undamaged content and constraints, e.g., regularity or sparsity.

Inspired by the insight, various image and video compression schemes have been revisited. An image-based compression framework has been designed where some smooth and flat blocks are removed from the source image [16]–[18]. At the decoder side, inpainting will be implemented to recover the lost content with the delivered assistant information. Applying similar idea to block-based video coding, some intuitive attempts involve in removing some macroblocks (MBs) in the prediction frame and restoring them based on various assistant side information, e.g., edge [16], sprite background image [19], and assistant parameters [18]. A related approach using texture analysis-synthesis scheme is proposed in [20], which reduces the entropy of source information by clustering the homogeneous area into a small patch that contains the epitome content of all associated regions. Furthermore, only the spatial consistency was inferred and enabled in intra frames. To maintain a temporal consistency, a space–time completion has recently been referred in a global optimization sense [21], [22]. Considering a semantic prior to some extent, a priority-based ordering in [23] was proposed to incrementally fill the specific missing texture by separating foreground and background layers with optical flow. Those schemes commonly need an explicit classification and segmentation criterion to extract a certain type of texture or object from a general natural video scene. Despite those schemes claim a bit-rate saving at similar visual quality levels compared with the traditional video codec, they fail to ensure a pixel-wise fidelity and sheer away from the quantitative validation. The coding burden from the assistant information is also a critical issue for generic video coding.

This paper proposes a structured priority BP-based inpainting prediction (IP) algorithm under the spatio-temporal MRF. The spatio-temporal consistency in video is translated into an optimization problem by minimizing the energy of the MRF model. Unlike the existing edge-based inpainting adopted in lossy image coding, the optimal predictor could maintain the pixel-wise fidelity and the robust error resilience without assistant information. Beyond the local prediction limitation and the ordered BP inference, the priority BP with regularized structure priors is imposed on the predictor in a nonlocal sense. It targets to achieve the reconstructed patch assignment of the largest expected probability and generate lower entropy residue. To be concrete, the structured sparsity of the predicted MB region is inferred by tensor voting projected from the co-located decoded regions, which is imposed on tuning the priority of message scheduling in BP with an adaptive and more convergent manner. The visiting order and the message scheduling policy of nodes in the spatio-temporal MRF are guided by the belief with structured sparsity. Through relatively few iterations of forward and backward process (typically, one iteration), the sparse inference of priority BP would ensure a stable marginal belief distribution on the structure and texture through updating local messages and beliefs.

We further propose a new video compression framework based on the H.264/AVC scheme, which adds the structured priority BP-based MB IP-mode for the optimal mode selection on rate-distortion optimization (RDO). The prediction mode can exploit the intrinsic nonlocal and geometric regularity in video samples. Through regularizing a global spatio-temporal consistency between the predicted region and the co-located known texture, the coded MBs with IP-mode is predicted to generate less residual, e.g., homogeneous visual patterns. The IP-mode with structured priority BP can be applied to both I and P frames, where only the MB header and residual data are included in the syntax element of the compressed bit-stream without any side information. Essentially, the IP-mode with structured priority BP at the encoder side of H.264/AVC outperforms the existing vision-based approaches at the decoder side, and specially achieves a better objective R-D performance besides visual quality. The computation complexity is also competitive with one iteration of sparse inference. Moreover, it behaves more resilient with an intrinsic probabilistic inference than the intra and inter-modes.

The rest of this paper is organized as follows. In Section II, we introduce the proposed IP-mode with structured priority BP (hereafter, IP-mode for short) in a generic video coding framework and discuss the motivated formulation on a regularized BP. In Section III, the structured sparsity on tensor voting is addressed. The priority BP is proposed in Section IV to clarify the prediction process. Extensive experimental results are validated in Section V on both objective and visual quality. In Section VI, we conclude this paper and discuss the future work.

II. Proposed Framework

A. Proposed Codec

The generic video coding framework with the proposed IP-mode is depicted in Fig. 1. Parallel with the existing intra and inter-modes, each MB would select the optimal mode in a R-D sense of Lagrange minimization [24]. The optimization formulation in the H.264/AVC standard is based on the assumption that the distortion D and incurred rate R of multiple MBs are independent of each other. If B_n is a MB in the current frame n, and $\hat{B}_{n-r}$ is the reconstructed blocks in the previously coded frame $n-r$ ($r=0$ denotes an intra-frame), then, the Lagrangian cost J_{MB} of the predicted MB B_n could be

$$\begin{aligned} &J_{MB}(B_n, \hat{B}_{n-r}, mode|Qp, \lambda_{mode}) = \\ &D(B_n, \hat{B}_{n-r}, mode|Qp) + \\ &\lambda_{mode} \cdot R(B_n, \hat{B}_{n-r}, mode|Qp) \end{aligned} \tag{1}$$

where Qp is the MB quantization value and λ_{mode} is the Lagrange parameter associated with Qp. The Lagrange coefficient λ_{mode} is empirically set $\lambda_{mode} = 0.85 \times 2^{(Qp-12)/3}$.

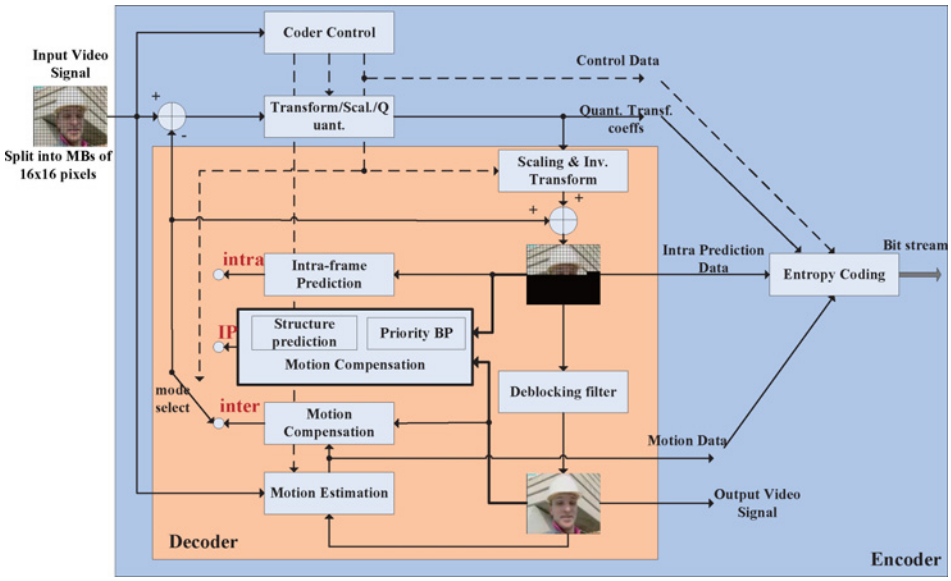


Fig. 1. Proposed codec with the structured priority BP-based IP.

By adding the IP-mode with structured priority BP to the existing intra and inter-modes in H.264/AVC, the respective possibilities of modes are

$$mode_{intra} \in \{\text{I4MB, I16MB, } \mathbf{IP}\}$$
$$mode_{inter} \in \{\text{SKIP, I4MB, I16MB, INTER, } \mathbf{IP}\}. \quad (2)$$

From the definition of (2), the IP-mode can be turned on in I, P, and B frames. Similar to intra and inter-modes, the predictor of IP-mode is subtracted from the current MB to generate a residue which is transformed, quantized, and entropy encoded to form the compressed bit-stream in a network abstraction layer. There is no assistant side information to be included in the bit-stream, so that the IP-mode's R only contains the MB header (IP-mode flag) and the corresponding discrete cosine transform residual blocks. The D in (1) is a measure to quantify the difference between B_n and $\hat{B}_n$ which is the predictor of the IP-mode from the reconstructed blocks $\hat{B}_{n-r}$ as follows:

$$D_{IP}(B_n, \hat{B}_n|Qp) = \sum_x \sum_y |B_n(x, y) - \hat{B}_n(x, y|Qp)|^2. \quad (3)$$

By minimizing the Lagrangian cost J_{MB} in (1) among all the candidate modes, the optimal mode can be selected for each MB.

In traditional inpainting scenarios, the restoration of the missing region is an ill-posed problem. However, the missing region might keep similar statistical, geometric, and surface reflectivity regularities as those in the surroundings, which makes the ill-posed problem possible to be solved. For the MB inpainting-based prediction problem, it could be formulated as follows by a weighted sum of both color distortion and gradient deformation terms as follows:

$$\min \sum_x (|f(x) - \hat{f}(x)|^2 + \beta|\nabla f(x) - \nabla \hat{f}(x)|^2) \quad (4)$$

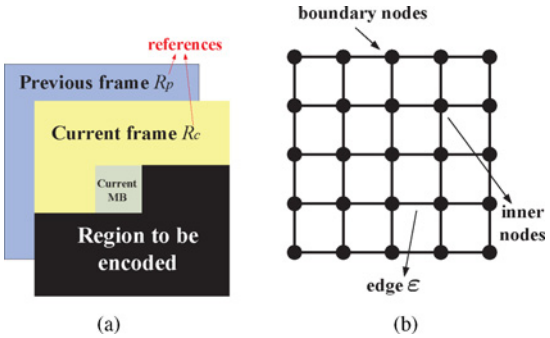


Fig. 2. (a) Obtained candidate patches from the reconstructed region in both the current and previous frames. (b) Nodes and edges of MRF.

where x is the position in the region to be predicted, $f(x)$ is the original pixel value, and $\hat{f}(x)$ is the inpainting result of $f(x)$. ∇ represents the gradients, and β is the weighted parameter. Generally, the gradients of the pixels can be represented by the structure in the region which is of prominent importance in inpainting process to preserve geometric and photometric regularities.

B. Basic Model

In the IP-mode, the prediction result is obtained by selecting and copying suitable candidate patches from the reconstructed regions. The target region to be inpainted is the current MB M, and the candidate patches are obtained from the reconstructed region in both the current and previous frames $S = R_c \cup R_p$, as depicted in Fig. 2(a). The candidates dictionary C consists of all $w \times w$ patches from the source region S.

We model the inpainting problem by the spatio-temporal MRF. As shown in Fig. 2(b), the nodes of the MRF are de-sampled from the pixels in M, and the edges ε of the MRF make up a four-neighborhood system. The nodes consist of boundary nodes, whose neighborhood intersect the source region S, and the inner nodes. Under this model, the unknown

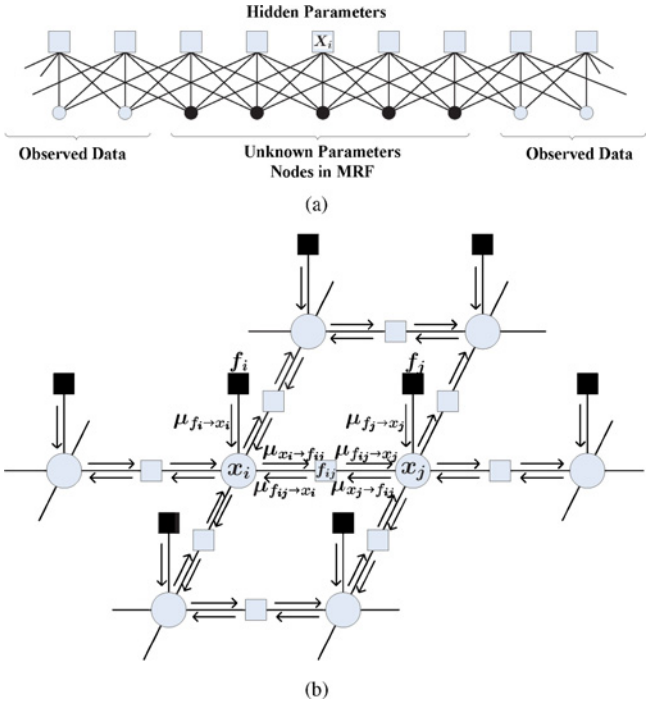


Fig. 3. (a) MRF model in one dimension for the video inpainting problem. Black circles are nodes in MRF, white circles are observed points in the reconstructed region, and squares are hidden variables referring to the patch overlapping the current MB region. (b) Factor graph represents the distribution of MRF, and local messages pass in the network. White circles are variables in MRF, white squares are factors between the neighboring nodes, and black squares are factors from the observed regions.

parameter, denoted by Ω, is the pixel values in the current MB to be predicted, and $\Omega = \{i_p | p \in M\}$. The known pixels around the MB are the observed variables $Y = \{i_p \in R_p \cup R_c \backslash M\}$. The windows $\{W_n\}_1^N$ are defined as small $w \times w$ spatial patches overlapping with the region M, and N is the total number. The patches $\mathbb{X} = \{X_n\}_1^N$ are hidden variables corresponding to W_n, as depicted in Fig. 3(a) in one dimension. Each patch W_n is associated with one hidden variable X_n, which is connected to all the pixel locations it covers, $\{i_p | p \in W_n\}$. The possible assignments for each X_n are chosen from the candidate patches in dictionary C. We assume that each X_n can have any assignment with a probability. As a consequence, the joint probability of the observed data and the hidden variables given the parameters Ω is as follows:

$$\begin{aligned} P(Y, X; \Omega) &= \frac{1}{Z} \prod_{n=1}^{N} \prod_{p \in W_n} \phi(i_p, X_n) \\ &= \frac{1}{Z} \prod_{n=1}^{N} \exp[-F(X_n, W_n)] \end{aligned} \tag{5}$$

where p denotes a pixel, either unknown or reconstructed, and Z is a normalizing constant. $\phi(i_p, X_n)$ is the joint potential function and $F(X_n, W_n)$ is the matching cost function of patches X_n and W_n.

Within the spatio-temporal MRF, we assume that the probability distribution of the values for a pixel is only related to the values of its neighboring pixels, and independent of the rest of the frame. As depicted in Fig. 2(b), we construct a four-neighborhood Markov network, and convert it into a factor graph, as Fig. 3(b). For a node $x_i \in M$, the factor $f_i(x_i)$ represents the probability of X_n on condition of the observed data Y, and the pairwise factor $f_{ij}(x_i, x_j)$ denotes the potential of neighboring nodes x_i and x_j as follows:

$$f_i(x_i) = \frac{1}{Z} \exp[-F(X_i, W_i)] \tag{6}$$

$$f_{ij}(x_i, x_j) = \frac{1}{Z} \exp[-F(X_i, X_j)]. \tag{7}$$

Based on the joint probability definition in (5), to find the maximum *a posteriori* (MAP) solution, we carry out a message passing process by propagating messages between the variable nodes and the factor nodes. By using the logarithm of the joint distribution, the max-product term is converted to max-sum. The messages transferred between the nodes are represented as follows:

$$\begin{aligned} \mu_{f_i \to x_i}(X) &= \max_{X_1, X_2, \ldots, X_N} \ln f(X, X_1, X_2, \ldots, X_N) \\ &\quad + \sum_{m \in \epsilon(f_i) \backslash x_i} \mu_{x_m \to f}(X_m) \end{aligned} \tag{8}$$

$$\mu_{x_i \to f_i}(X) = \sum_{l \in \epsilon(x_i) \backslash f_i} \mu_{f_l \to x_i}(X). \tag{9}$$

Thus, we have two distinct kinds of messages: those go from variable nodes to factor nodes, denoted $\mu_{x \to f}(X)$, and those go from factor nodes to variable nodes, denoted $\mu_{f \to x}(X)$. In this case, the messages passed along a link are always a function of the variable associated with the variable node which the link is connected to. As shown in (8), the messages sent by a factor node to a variable node along the link is composed by the sum of incoming messages along all other links coming into the factor node, and then marginalizes over all of the variables associated with the incoming messages. To evaluate the messages sent by a variable node to an adjacent factor node along the connecting link, we simply take the sum of the incoming messages along all of the other links, as displayed in (9).

We expect to find the marginal for every variable node in the graph, and the marginal is given by the product of the incoming messages along all of the links arriving at that node, which is converted to sum by logarithm. Each of the messages can be computed recursively in terms of other messages. For a specific node x_i, we can view it as the root of the factor graph and the messages passing process begins at the leaf nodes. Each node sends messages toward the root once it has received messages from all of its other neighbors. By propagating messages recursively till messages have been sent along every link, and the root node has received messages from all its neighbors, the marginal of the root is computed as follows:

$$p(x_i) = \sum_{l \in \epsilon(x_i)} \mu_{f_l \to x_i}(X). \tag{10}$$

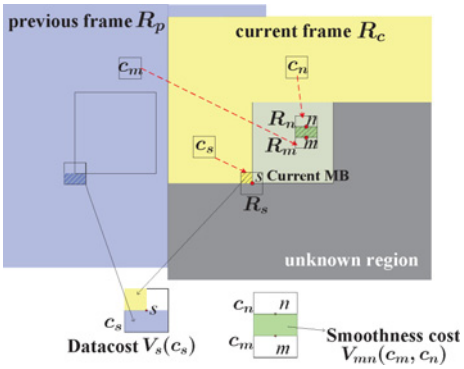


Fig. 4. Computed data cost and smoothness cost.

Thus, the most probable value of the root can be searched as follows:

$$X_i^{max} = \arg\max_X \left[\sum_{l \in \epsilon(x_i)} \mu_{f_l \to x_i}(X) \right]. \tag{11}$$

As we aim to find the marginal distribution for all the variable nodes in the spatio-temporal MRF, the problem could be solved by viewing each node as the root of the graph and repeating the algorithm, which is a huge waste of time and resource. As a result, we consider an efficient procedure by decomposing the overall message propagation course into forward and backward processes. Arbitrarily pick a node as the root, and propagate messages from leaves to the root, which is formulated as a forward process. When the root has received messages from all of its neighbors, it then sends out messages to the neighbors, i.e., the messages pass outwards from the root to the leaves, which is called a backward process. In this way, messages are passed in both directions across every link in the graph, and all the nodes can receive messages from all their neighbors. The set of values that jointly have the largest probability, or the vector $\mathbb{X}^{max}$ maximizes the joint distribution $p(\mathbb{X})$ is as follows:

$$\mathbb{X}^{max} = \arg\max_{\mathbb{X}} \ p(\mathbb{X}) \tag{12}$$

$$s.t. \quad p(\mathbb{X}^{max}) = \max_{\mathbb{X}} \ p(\mathbb{X}). \tag{13}$$

By bringing factor nodes into the network and propagating messages around the graph, the maximum of the joint distribution is achieved.

C. Video Inpainting as a Discrete Optimization Problem

The prediction of the current MB is to copy suitable candidate patches over the position of nodes. It is critical that the most suitable candidate should match both the known region and the neighborhood. As a result, it is necessary to define the matching cost or the distance of two patches A and B. With regard to the minimized distortion objective in (4), it is formulated as follows:

$$D(A, B) = \|A(x_A) - B(x_B)\|^2 + \beta \|\nabla A(x_A) - \nabla B(x_B)\|^2. \tag{14}$$

According to the formulation in Section II-B, we define a discrete optimization problem of minimizing the energy of a spatio-temporal MRF. By the matching cost in (14), we define the potential $V_s(c_s)$ and the pairwise potential $V_{mn}(c_m, c_n)$ of single node as in Fig. 4. The potential $V_s(c_s)$ or data cost for putting the patch c_s over the node s, presents how well the patch agrees with the source region around the node s as follows:

$$V_s(c_s) = D(c_s, R_c) + \gamma D(c_s, R_p) \tag{15}$$

where $D(c_s, R_c)$ is the matching cost between the candidate patch c_s and the known region around the node s in the current frame, which is depicted as R_c, and $D(c_s, R_p)$ corresponds to the matching cost between the candidate patch and the previous reconstructed reference frame, which is represented as R_p. γ is the coefficient for motion change between the current and the preceding frames. If the current frame is intra frame and there is no a previous reference frame, the second term is zero. In a similar context, the smoothness potential $V_{mn}(c_m, c_n)$, measures how well patches c_m and c_n over the neighbor nodes m and n agree at the overlapping region as follows:

$$\begin{aligned} V_{mn}(c_m, c_n) = D(c_m, c_n) \quad &= \sum_{x \in R_m \cap R_n} \{|c_m(x) - c_n(x)|^2 \\ &+ \beta |\nabla c_m(x) - \nabla c_n(x)|^2\}. \end{aligned} \tag{16}$$

Generally, the overlapping region of the patches assigned on two neighboring nodes is half the area of the patch to obtain a smooth result and make the best use of dependency between the neighbors.

Based on the data cost and the smoothness cost, our goal is to minimize the energy of spatio-temporal MRF by assigning a suitable patch $\hat{c}_m \in C$ to each node m as follows:

$$\min E(\hat{c}) = \sum_{m \in M} V_m(\hat{c}_m) + \sum_{\substack{m,n \in M \\ (m,n) \in \varepsilon}} V_{mn}(\hat{c}_m, \hat{c}_n). \tag{17}$$

Intuitively, an algorithm to optimize the energy attempts to assemble a huge jigsaw puzzle, where the source patches correspond to the puzzle pieces and the region M represents the puzzle itself.

III. Structured Sparsity

As a typical inpainting process is usually an ill-posed problem, a variety of regularization algorithms have been developed to solve it. Hereafter, the completion problem in (17) is formulated into the energy minimization in spatio-temporal MRF, and the energy is composed of pixel and gradient distortion which can be interpreted into what the pixel and its direction are. It is well known that the gradient and direction of the pixels can be represented by the structured sparsity. For an efficient compression, the structured sparsity is inferred through the clues from the reconstructed regions.

For a non-iterative solution, a tensor voting algorithm is adopted for the edge prior of the decoded regions to cast votes on nodes in the current MB. Tensor is used for token representation, while voting for communication among tokens [25]. By postulating smooth connections among tokens, the voting field is a dense tensor field combining tensor and voting.

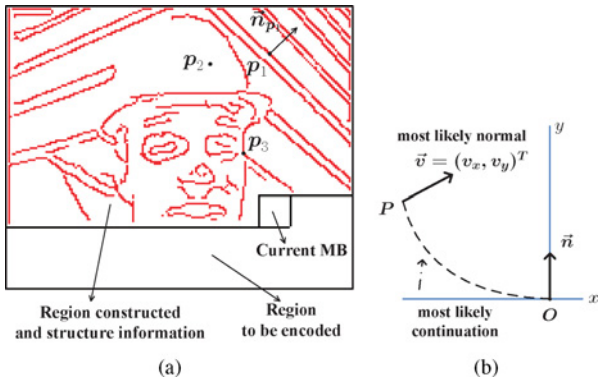


Fig. 5. (a) Structure information and the token representation. (b) Illustrative design of a fundamental 2-D voting field.

Fig. 6. Structured sparsity of nodes in MRF, and the whiter nodes have higher structured sparsity. (a) *Foreman*. (b) *Highway*.

In this section, we address the stick tensor field with structured sparsity.

A. Token Representation

To infer the structured sparsity in the unknown regions, we extract the edge prior in the neighboring known region. The Canny edge detector [26] is used because it could define a series of detection and localization criteria in a mathematical form, e.g., low error rate ensuring that edges should not be missed and there be no responses to non-edges, well localized edge points at a minimum distance between the edge pixels from the detector and the actual edge, and single response criterion.

Once the edge information of the decoded region is detected and extracted, we translate the pixels in the decoded region into the inputs of token. As depicted in Fig. 5(a), if the point is on a curve, the tensor can be described by its associated tangent or normal, e.g., node p_1 has normal $(n_x, n_y)^T$, and it can be expressed by a second-order tensor as follows: $\begin{pmatrix} n_x^2 & n_x n_y \\ n_y n_x & n_y^2 \end{pmatrix}$. If the node is not on a curve (e.g., node p_2), or a intersection (e.g., node p_3), we can translate it into $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

B. Token Communication

The tensor voting algorithm identifies local feature descriptions by spreading the information associated with shape-related input within a field while enforcing a smoothness constraint. Each data point communicates its information in a region of field through a voting process. The more information that is received at each data point, the stronger likelihood of a geometric feature being present at a certain location. This likelihood could be expressed through a confidence measure, saliency, which is often used in feature extraction.

There are two elements in tensor voting algorithm: data representation which is described in Section III-A, and token communication through linear voting which is a process similar to linear convolution. In the underlying problem, the points in the reconstructed region play as inputs, and each node in MRF receives votes from inputs within their voting field which is decided by voting size. For example, the vote casted by an input O to node P is illustrated in Fig. 5(b). We claim that the osculating circle connecting O and P is the most likely connection since it keeps the curvature constant along the hypothesized circular arc. The most likely normal $\vec{v}$ at P is given by the normal to the circular arc at P, and its inner product with normal $\vec{n}$ at O is nonnegative. The length of this normal represents the strength of the vote, which is inversely proportional to the arc length and the curvature of the underlying circular arc as follows:

$$|\vec{v}(r, \varphi, \sigma)| = e^{-\left(\frac{r^2 + c\varphi^2}{\sigma^2}\right)} \tag{18}$$

where r is the length of arc OP, φ is the curvature, and c is a constant controlling the decay with high curvature. σ controls the smoothness and determines the effective voting size.

Each node in spatio-temporal MRF receives votes from inputs within its corresponding voting field, and the second-order tensor sums all the votes and collects into a covariance matrix as follows:

$$V_{sum} = \begin{bmatrix} \sum v_x^2 & \sum v_x v_y \\ \sum v_y v_x & \sum v_y^2 \end{bmatrix}. \tag{19}$$

The corresponding eigensystem consists of two eigenvalues $\lambda_1 \geq \lambda_2 \geq 0$ and two corresponding eigenvectors e_1 and e_2. Therefore, V_{sum} can be rewritten as follows:

$$V_{sum} = (\lambda_1 - \lambda_2) e_1 e_1^T + \lambda_2 (e_1 e_1^T + e_2 e_2^T) \tag{20}$$

where $e_1 e_1^T$ is a 2-D stick tensor with e_1 indicating curve normal direction, and $e_1 e_1^T + e_2 e_2^T$ is a 2-D ball tensor. As a result, the curve saliency S_m of node m is represented by $\lambda_1 - \lambda_2$ as follows:

$$S_m = |\lambda_1 - \lambda_2|. \tag{21}$$

The structured sparsity of nodes in the unknown region could denote the possibility that the node is on a curve, and we are implied to pay more attention to nodes with higher structured sparsity. Intuitively, Fig. 6 shows two examples of structured sparsity. In the block rounded by red lines, the gray levels represent the structured saliency of nodes, and whiter nodes own higher structured sparsity.

IV. Belief Propagation

A. Message Propagation

BP is an iterative algorithm to find a MAP estimator by iteratively solving a finite set of equations till convergence. Through continuously propagating local messages within the nodes of the spatio-temporal MRF, beliefs of every node are updated and an optimal output would be achieved when

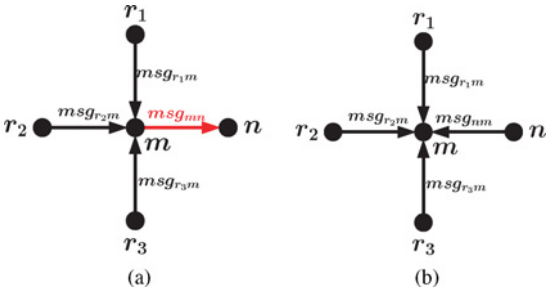


Fig. 7. (a) If a node m needs to send messages to its neighbors n, it must make use of the messages msg_{r_1m}, msg_{r_2m}, and msg_{r_3m} from all the other neighbors. (b) If a node m needs to calculate belief $b_m(c_m)$, it should collect the message coming from all the neighbors.

all the messages have been stabilized. Within the graphical model in Section II-B, since each factor node has no more than two neighboring variable nodes, the messages can be converted into the opinions between the neighboring variable nodes. That is, the messages reflect the node's opinion to its neighbors, and each message is a vector whose dimensions is given by the number of candidates. Generally, the set of messages sent from node m to its neighbor n is denoted as $\{msg_{mn}(c_n)_{c_n \in C}\}$, and implies the opinion of m about assigning label $c_n \in C$ to node n as follows:

$$msg_{mn}(c_n) = \min_{c_m \in C} \{V_{mn}(c_m, c_n) + V_m(c_m) + \sum_{\substack{r \neq n \\ (r,m) \in \varepsilon}} msg_{rm}(c_m)\} \quad (22)$$

where c_m and c_n are the candidate patches in the patch dictionary C, and $V_{mn}(c_m, c_n)$, $V_m(c_m)$ are the smoothness cost and the data cost of the neighboring nodes m and n.

From the above equation, we can find that if one node m expects to send messages to its neighboring node n, it must first traverse each one of its own candidates c_m and decide which one of them could provide the greatest support for assigning the candidate c_n to the node n. The support of the candidate c_n to the node n is determined by the compatibility between candidates c_m and c_n [smoothness cost $V_{mn}(c_m, c_n)$], and the likelihood of assigning the candidate c_m to the node m [data cost $V_m(c_m)$], as well as the opinion of its other neighbors about the candidate c_m [sum of messages $\sum_{r \neq n,(r,m) \in \varepsilon} msg_{rm}(c_m)$]. As a result, the node m must first collect messages from all its other neighbors, and then add its own opinion into the messages sent to the node n [see also Fig. 7(a)].

As we discussed in Section II-B, BP in spatio-temporal MRF can be decomposed into forward and backward processes, and the arrangement of the root and leaves of the network is the prominent factor to affect the speed of convergence. In other words, the organization of the message flow is of paramount importance. In the proposed inpainting-based MB prediction, we adopt the message scheduling principle that the nodes who are more confident about their candidates should transmit outgoing messages to their neighbors earlier, i.e., the nodes are located at the bottom leaf nodes. On the contrary, the nodes who are less confident and do not own enough idea about the candidates to select should send messages later, which are positioned near the root. It can be reasoned in two aspects. On the one hand, the more confident a node is, the more informative its messages are going to be, meaning that these messages can help the neighbors to increase their own confidence. On the other hand, by first propagating the most informative messages around the graphical model, it can help BP converge much faster and reduce the computation complexity to prune a large amount of useless messages.

Based on the message scheduling principle, BP is completed by a few iterations and each iteration includes forward and backward processes. Nodes are arranged in a special sequence and visited from head to tail in the forward process, and inversely in the backward process. Once a node is accessed, it sends messages to the neighbors and updates their beliefs. A set of beliefs $\{b_m(c_m)\}_{c_m \in C}$ for each node expresses the probability of assigning patch c_m to node m as follows:

$$b_m(c_m) = -V_m(c_m) - \sum_{r:(r,m) \in \varepsilon} msg_{rm}(c_m). \quad (23)$$

The set of beliefs is related to the node's data cost and the messages from all its neighbors [see Fig. 7(b)]. Actually, as denoted in (10), each belief $b_m(c_m)$ approximates the maximum conditional probability given the fact that node m has been assigned the patch c_m, and it is the evidence to select candidates.

After a number of iterations, when the messages are asymptotically stable, the candidate of maximum belief is assigned for each node as follows:

$$\hat{c}_m = \arg\max_{c_m \in C} b_m(c_m). \quad (24)$$

B. Ordered BP

As discussed in Section IV-A, the message scheduling principle would ensure the nodes with higher confidence to propagate messages earlier. Thus, we can arrange all the nodes in a order according to their confidence and correspondingly visit each node in sequence. The following problem becomes how to evaluate the confidence of nodes and in what order the nodes should be arranged to effectively reduce the computation complexity and speed up the convergence. In the context, it is obvious that boundary nodes with more priors from the surrounding known regions are most confident to determine which kind of patches are suitable, while the inner nodes only rely on the opinions from their neighbors to select the assigned patches. Moreover, it is also recognized that the nodes which have a larger intersection area with known regions are more confident. For example, the node in the top left corner is the most confident one in the MB region. In this way, nodes would be arranged into a fixed list of decreasing confidence. As depicted in Fig. 8, in the forward process, we scan the nodes from the top left to the bottom right along the boundary, and from the outside layer to the inside, like peeling an onion. The ordered list is stored in an array. In the backward process, the nodes are visited in the reverse order from tail to head through the order list, which is from inner to outside in MRF.

Fig. 9 gives a set of examples applying a fixed order BP to predict MBs, where both structure and texture regions are evaluated. Through a number of iterations along the ordered

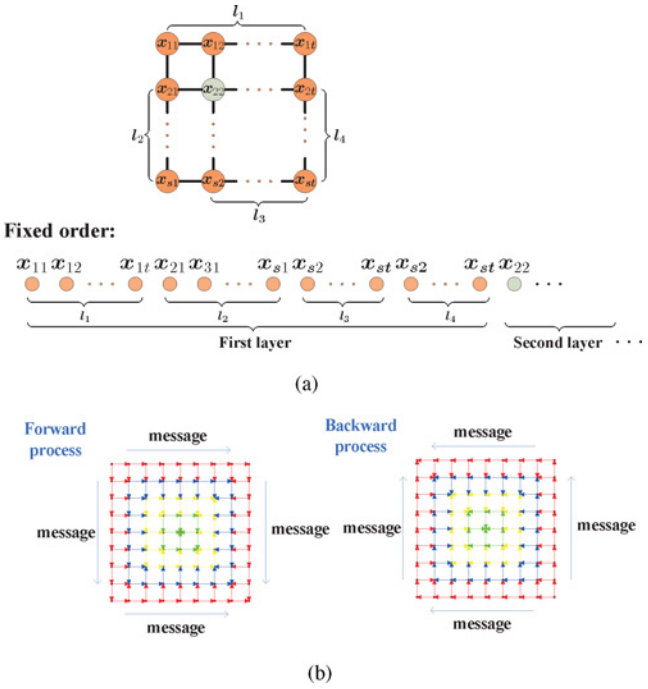


Fig. 8. In the ordered BP with forward process and backward process, nodes are visited in order from boundary layer to inner side. (a) Ordered BP. (b) Forward and backward process.

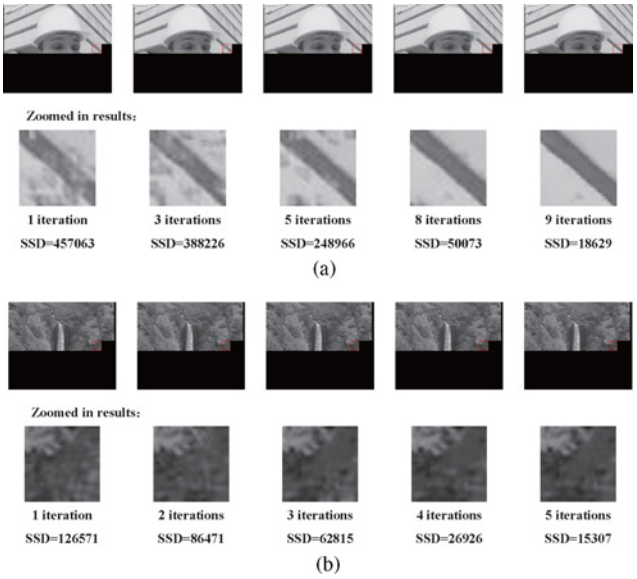


Fig. 9. IP results with an ordered BP for both structure and texture regions. (a) Structure region. (b) Texture region.

list, the sum of squared differences between the predicted result and the original signal is decreasing. It is obvious that structure regions will require a larger number of iterations than texture region to achieve acceptable results because the orientation of the structure may not be so consistent with the direction of the messages propagation.

C. Priority BP

The ordered BP has achieved a decent reconstruction in inpainting, while there is much space to promote the performance and reduce the computation complexity, i.e., the

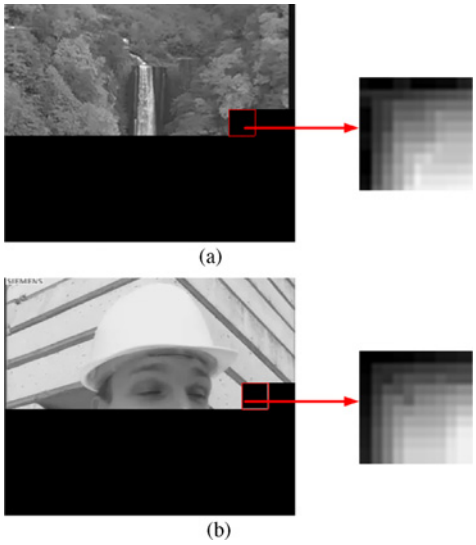


Fig. 10. Visiting order in the first forward process according to the belief and priority of nodes. The darker a node is, the earlier it will be visited in the forward process. (a) Texture region, according to the belief and priority. (b) Structure region, according to the tuned belief and priority in (26).

convergent number of iterations. As the message propagates, beliefs of nodes are updated to reflect the dynamic confidence of nodes as the process goes along. In view of the basic message scheduling principle in Section IV-A, we tune the priority of nodes to quantify their confidence which is tightly related to the set of beliefs. With the change of beliefs, the priorities of nodes are dynamically updated throughout the process.

Based on the definition of beliefs in (23), the confidence of node m depends only on the current set of beliefs $\{b_m(c_m)\}_{c_m \in C}$ that has been estimated by BP. One simple way to measure the confidence is to count the number of suitable candidate patches, i.e., the candidates whose belief exceed a control threshold b_{thred}: $P_m = |\{c_m \in C : b_m(c_m) \geq b_{thred}\}|$.

By computing the priority P_m of each node m, it is possible to select the node of a maximum priority as the next node to visit

$$\hat{m} = \max_m P_m. \tag{25}$$

For texture regions, the visiting order of the nodes in the first forward process is shown in Fig. 10(a). The darker a patch is, the earlier the corresponding node is visited. It implies that the nodes near the known region would be visited earlier. Fig. 11(a) displays the predictor after one iteration, which is better than that after a few iterations from a ordered BP in Fig. 9(a) in both visual and objective measurement.

For structure regions, it might be more intractable. Fig. 11(b) displays the predictor by the priority BP where message flow is assigned along the structure direction. We take tensor voting to infer the structure prior of the unknown regions, and include the structured sparsity into the definition of beliefs in (23) to adjust the priority and visiting order of nodes as follows:

$$b_m(c_m) = -V_m(c_m) - \sum_{r:(r,m)\in\varepsilon} msg_{rm}(c_m) + \alpha S_m. \tag{26}$$

Algorithm 1 Priority BP algorithm

begin

/* ***Initiation*** */

$order = \varnothing$; initiate the max priority to 0: $p_{max} = 0$;

for $i = 1$ to N

$vflag(x_i) = 0$

if node x_i is inner node

$bflag(x_i) = 0$

else

compute datacost and belief of x_i

$bflag(x_i) = 1$

end for

/* ***Forward Process*** */

while there are nodes unvisited

/* *select the node with max priority* */

for $i = 1$ to N

if $vfalg(x_i) == 0$ && $bflag(x_i) == 1$

compute priority of node x_i: $p(x_i)$

if $p(x_i) > p_{max}$

$target = x_i$

end for

/* *send messages and update beliefs* */

push $target$ into $order$; $vflag(target) = 1$;

for neighbors of $target$ && $vflag(neighbor) == 0$

$target$ sends messages to $neighbor$

update beliefs of $neighbor$

$bflag(neighbor) = 1$

end for

/* ***Backward Process*** */

for $i = 1$ to N

$target = order(i)$; $vflag(target) = 0$

for neighbors of $target$ && $vflag(neighbor) == 1$

$target$ sends messages to $neighbor$

update beliefs of $neighbor$

end for

end for

/* ***Patch assignment*** */

for $i = 1$ to N

$\hat{c}_i = \arg\max b_i(c_j)$, $c_j \in C$

end for

end

From the equation, the beliefs not only are related to the node's data cost and messages from neighbors, but also depend on the node's structured sparsity S_m which is weighted by a parameter α. As a consequence, the nodes with higher structured sparsity get larger beliefs than others on average. It would efficiently approximate the stable marginal belief distribution by message propagation. Fig. 10(b) displays the visiting order of the priority BP after belief adjustment of (26). The darker nodes indicate an earlier visit, and the nodes with high structured sparsity are arranged forth in the visiting sequence. In Fig. 11(b), the predictor after only one iteration is displayed, which is close to the result after three iterations without the structure prior.

For either texture or structure regions, the message arrangement and visiting sequence of nodes are dynamic. The next node to send messages is always the node with the highest confidence currently. A whole message propagation course consists of a forward process and backward process, and behaves as one iteration through an appropriate definition of belief and priority. We also need an array to store the index list of nodes to be visited, which is defined as *order*. Noted that the list is not fixed in each iteration or settled before the propagation. At the beginning, the list *order* is empty, all the messages are initialized to zero, and the data cost and

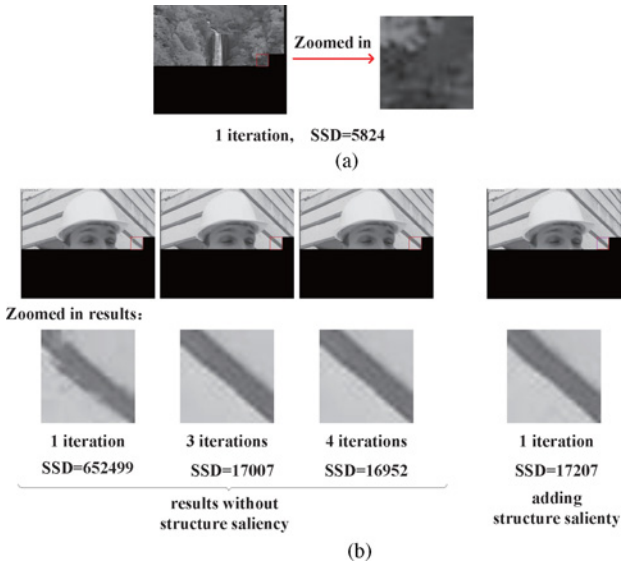


Fig. 11. Prediction results by the priority BP in the texture and structure regions. (a) For the texture regions, the predictor after one iteration. (b) For the structure regions, the predicting comparison of beliefs with and without structure consideration.

the structured sparsity of nodes are calculated. In the forward process, as the nodes are visited and messages are propagated, beliefs of all nodes are updated. It is worth mentioning that for nodes which do not have data cost (e.g., inner nodes in MRF) and receive any messages from their neighbors currently, their beliefs are not counted and considered in the current selection of nodes to visit. When we select the next node to send messages, in other words, only the nodes who have either data cost (information from the neighboring known region) or the received messages from neighbors are considered. Once the node of the largest priority is decided, it sends messages to all the neighbors who have not been visited, and the index of the node is pushed into list *order* as well as the node is tagged as "visited." The beliefs of nodes would be updated and the selection of the next target begins, till all the nodes are accessed. In the backward process, nodes are visited from tail to head in sequence through the list *order*, so that all nodes received feedbacks from their less confident neighbors. The detailed procedure could be referred to as follows.

Input: nodes in MRF, $X = \{x_i, i = 1, 2, \ldots, N\}$
candidates dictionary, $C = \{c_j, j = 1, 2, \ldots, L\}$

Output: patches assignment $X(\hat{c}) = \{x_i(\hat{c}_i), \hat{c}_i \in C, i = 1, 2, \ldots, N\}$

Important intermediate variables:

order: an array to store the nodes

vflag: an array to label whether the node has been visited

bflag: an array to label whether the node has beliefs currently.

D. Composition of Final Patches

After the final patches have been selected for each MRF node, we need to compose them to generate the final result. As there are some overlapping regions between the patches of the neighboring nodes, every pixel in the MB region is covered

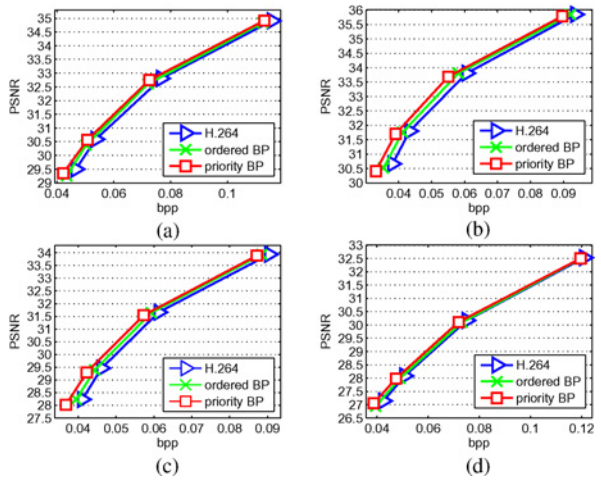


Fig. 12. R-D performance comparison of the test sequence. (a) *Foreman_cif*. (b) *Highway_cif*. (c) *Container_cif*. (d) *Waterfall_cif*.

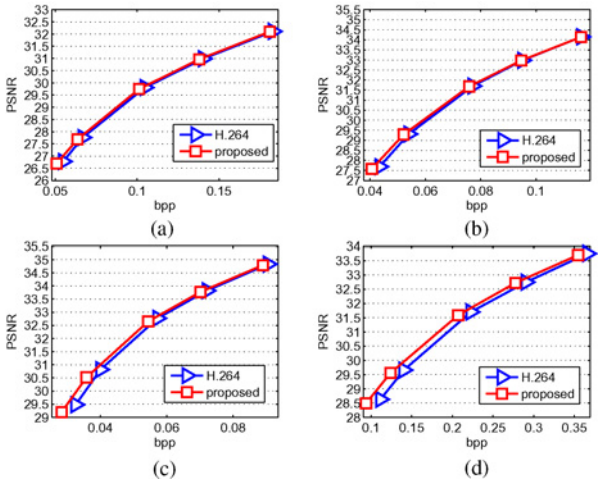


Fig. 13. R-D performance comparison of the test sequence. (a) *BlowingBubbles*_416 × 240. (b) *BQMall*_832 × 480. (c) *ParkScene*_1920 × 1080. (d) *ParkJoy*_2560 × 1600.

by several patches centered in its surrounding nodes. Since the corresponding gray level of candidates for each pixel is quite similar and the usual graph-cut algorithm is complex, the most likely color of the pixel should satisfy an overlapped block motion compensation in a posterior probability of similarity-based confidence [22]. Here, the final result of the pixel is composed by blending the patches with weights related to the node confidence as follows:

$$I(x) = \sum_{i:x\in R_i} w_i \hat{c}_i(x)$$
$$s.t. \quad w_i \propto \frac{1}{i}, \quad \sum_i w_i = 1. \tag{27}$$

For each pixel x in the current MB, the weights of the candidate patch for each node whose corresponding patch region covers the pixel are in inverse proportion to the position i of the node in *order*. That is, the candidate patch with the node positioned in the forth of *order* would get a larger weight to generate the final result. In this way, the predicted result is obtained smoothly and the block effect is avoided.

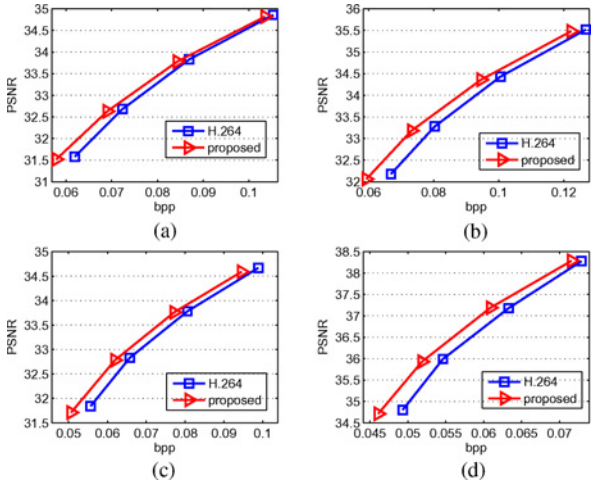


Fig. 14. R-D performance comparison with the frame structure of "IBBPBBP." (a) *Foreman_cif*. (b) *Highway_cif*. (c) *Cactus* 1280×720. (d) *BlueSky* 1920×1080.

V. Experimental Results

A. Implementation

The proposed IP-mode with structured priority BP has been implemented in the Joint Model 15.1 of H.264/AVC [27]. In the experiments, all test sequences are with the YUV 4:2:0 format, 30 Hz frame rate. The IP-mode is enabled in both I, P, and B slices, and is compared with a hierarchical tree-structured intra and inter-prediction modes: I4×4, I16×16, P16×16, P16×8, P8×16, P8×8, P8×4, P4×8, and P4×4 modes. Various types of test sequences including the common intermediate format (CIF) resolution videos (352×288), *Foreman*, *Highway*, *Container*, *Waterfall*, and a WQVGA resolution *BlowingBubbles* (416×240), a standard definition *BQMall* (832×480), a 720p definition *Cactus* (1280×720), the high-definition *ParkScene* and *BlueSky* (1920×1080), and a ultrahigh-definition *ParkJoy* (2560×1600), are evaluated.

Both an ordered BP and a priority BP are imposed on CIF sequences, and structured sparsity is enabled on *Foreman* and *Highway* with significant geometric regions. The proposed prediction algorithm would involve with an ordered BP and a priority BP in the inpainting process, and the iteration number is set to 8 in the ordered BP and 1 in the priority BP.

B. Validated Results

Fig. 12 shows the coding performance of the proposed scheme with the ordered BP and the priority BP in comparison to the traditional H.264/AVC with the "IPPP···" structure and the group of pictures (GoP) size 10 for the CIF sequences. For the medium and high spatial resolution sequences, Fig. 13 gives the comparison of the proposed scheme with the priority BP. When applying the IP-prediction mode in B frames with the frame structure of "IBBPBBPBBP···" and the GoP size 15 over both the CIF sequences and higher resolution sequences, Fig. 14 shows the R-D performance comparison. It can be seen that the proposed inpainting-based prediction with structured priority BP can increase up to 0.8 dB peak signal-to-noise ratio (PSNR) at the same bit-rate (bit per pixel, b/p) *versus*

TABLE I

COMPUTING COMPLEXITY COMPARISON BETWEEN ORDERED BP AND PRIORITY BP

Sequences	Bit-Rate (b/p)	Ordered BP (Eight Iterations)		Priority BP (One Iteration)	
		Selection Ratio (%)	Complexity (pixels/s)	Selection Ratio (%)	Complexity (pixels/s)
Foreman	0.04	47.0	314.32	46.5	1156.53
BQMall	0.05	44.9	45.48	43.1	140.59
Stockholm	0.05	37.2	265.29	35.7	1119.91
Cactus	0.06	56.9	132.41	56.4	668.12
Bluesky	0.05	47.8	72.32	45.1	591.09
ChristmasTree	0.05	32.0	385.36	32.9	1206.64

TABLE II

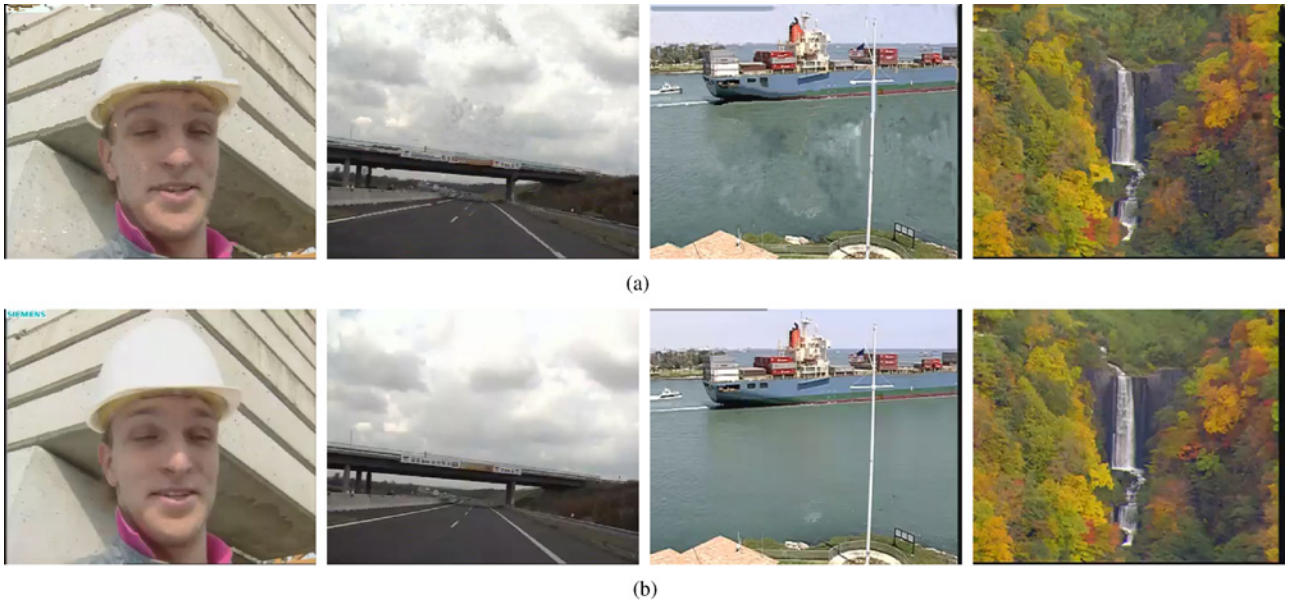
ERROR RESILIENT CAPABILITY OF THE PROPOSED SCHEME WITH THE *IP*-MODE

Package Losing Rate	10%		15%	
Sequences	H.264/AVC (dB)	Proposed Scheme (dB)	H.264/AVC (dB)	Proposed Scheme (dB)
Foreman	32.45	32.64	32.13	32.61
Highway	33.65	33.80	33.64	33.80
Container	31.59	31.62	31.57	31.61
Waterfall	30.08	30.10	30.05	30.10

TABLE III

R-D PERFORMANCE COMPARISON BETWEEN THE BLOCK-SKIP AND EDGE-BASED INPAINTING ALGORITHM [16] AND THE PROPOSED ALGORITHM

Sequences	Edge-Based Inpainting in [16]			Proposed Scheme		
	b/p	PSNR (dB)	Skip Rate (%)	b/p	PSNR (dB)	IP-Mode Rate (%)
Foreman	0.47	26.83	44.7	0.45	36.19	3.78
Container	0.65	24.92	48.6	0.64	35.21	3.03
Highway	0.39	16.85	41.8	0.36	37.42	2.53
Waterfall	0.95	26.48	37.6	0.91	33.55	0.51



(a)

(b)

Fig. 15. Reconstructed performance comparison of sequence *Foreman_cif*, *Highway_cif*, *Container_cif*, and *Waterfall_cif*. (a) Block-skip and edge-based inpainting algorithm in [16]. (b) Proposed inpainting-based prediction scheme with IP-mode.

TABLE IV
PROPOSED **IP**-MODE RATIO UNDER DIFFERENT QP LEVELS

IP-Mode Ratio (%)	Size	$QP = 30$	$QP = 32$	$QP = 36$	$QP = 40$	$QP = 42$
Foreman	352 × 288	10.8	12.9	18.8	26.8	32.2
Highway	352 × 288	25.0	29.9	40.3	45.1	45.7
Container	352 × 288	35.7	36.0	36.9	34.8	36.2
Waterfall	352 × 288	15.4	19.1	31.7	50.2	57.3
BlowingBubbles	416 × 240	12.5	18.4	25.9	32.3	40.4
BQMall	832 × 480	20.4	26.8	35.2	50.8	61.6
ParkScene	1920 × 1080	14.2	19.3	24.1	28.9	32.4
ParkJoy	2560 × 1600	20.2	23.6	27.6	35.6	40.8

H.264/AVC, and the coding gain is more obvious in low bit-rate region. With structured sparsity regularization and priority arrangement, the computation complexity of the IP-mode is hugely reduced to an acceptable level from the ordered BP inference. That is, only one iteration of forward and backward process could obtain little residue and would be promising in the future HEVC project. In fact, pruning-based scheduling with multidimensional feature tensors could be further explored to speed up the convergence. Table I shows the complexity comparison between the ordered BP and the priority BP, while the complexity is evaluated by the decoding pixels per second. To be more reliable, the ratios of the selected IP-mode and the bit-rate in both the ordered BP and the priority BP are set to be very close. By extracting the structure information and adjusting the message propagation direction, the prediction result can be fine enough for the priority BP with only one iteration. Obviously, the priority BP with only one iteration achieves much lighter complexity compared to the ordered BP with eight iterations, so that it largely reduces the computing complexity and speeds up convergence.

The error resilient capability of the proposed scheme with IP-mode is shown in Table II. Using data partition in the H.264/AVC standard [28], the bits in IP-mode which only contain the MB header and residual coefficients, are classified into data partitions A and C, respectively. To say it simply, the residual data of the IP-mode occupies the least importance and the most tendency to loss. It is compared with H.264/AVC at the same packet loss ratio, e.g., 10% and 15%. With an intrinsic probabilistic coherence, it can be seen that the proposed scheme is of more error resilient because the IP-mode is less dependent of the determinate reconstruction than the existing modes in H.264/AVC.

Also, the proposed compression scheme is compared with the block-skip and edge-based inpainting algorithm in [16]. This anchor algorithm extracts edge information, skips some blocks at the encoder side, and restores the regions through edge-based inpainting at the decoder side. Considering this method is originally used in image compression and H.264/AVC intra-picture coding, we correspondingly enable the I slices to cater for the spatial context of the block-skip framework. The QP levels of the skip block framework and the proposed inpainting-based prediction scheme are set to 28 and 31, respectively, in order to obtain a similar bit-rate. Fig. 15 shows the perceptual (visual) quality of reconstructed frames from both the proposed IP-mode prediction and the

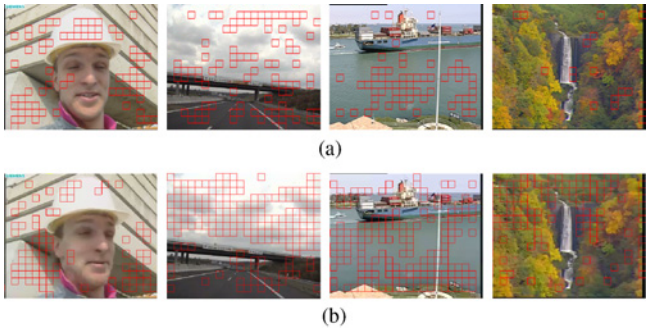


Fig. 16. Distributions of the IP-mode MBs in test sequences. From left to right, the sequences are *Foreman*, *Highway*, *Container*, and *Waterfall*. (a) I frames. (b) P frames.

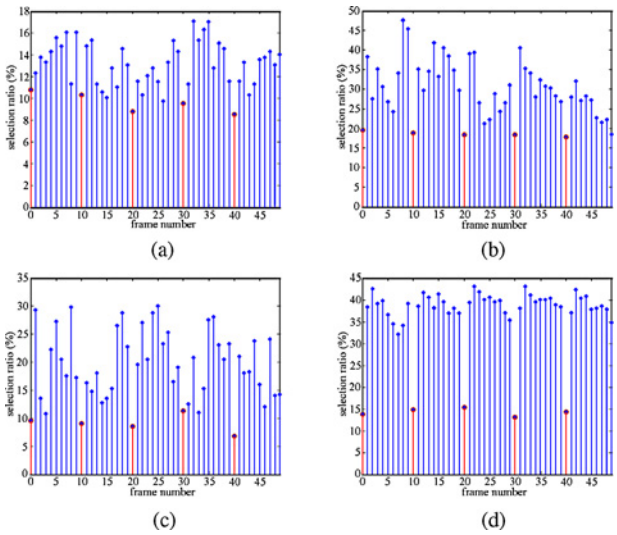


Fig. 17. Distribution ratio of the proposed IP-mode in each frame. (a) *Foreman*. (b) *Highway*. (c) *Waterfall*. (d) *Container*.

edge-based inpainting algorithm in [16]. To meet the image compression criterion, the bit-rate metric has been changed from kbit/s to b/p. Corresponding to the visual quality, Table III provides the average R-D result (b/p and PSNR) of the two approaches. Although the block-skip method claims up to 33% bit-savings compared with H.264/AVC intra coding, the so-called "similar visual quality levels" are not clearly given, and the objective results are poor. The advantage of the proposed inpainting-based prediction scheme is obvious because it is not required to contain edge information in the bit-stream. It is achieved through inferring structured sparsity

TABLE V
RATIO BETWEEN THE ORIGINAL MODES IN H.264/AVC AND THE *IP*-MODE

Replacing Ratio (%)	P16 × 16	P16 × 8	P8 × 16	P8 × 8	I4 × 4	I16 × 16
Container	89.8	3.14	2.88	0.59	0.78	2.81
Waterfall	97.1	1.12	1.27	0.00	0.32	0.16
Foreman	88.0	3.23	2.83	0.40	0.67	4.85
Highway	94.1	1.24	1.05	0.07	0.07	3.53

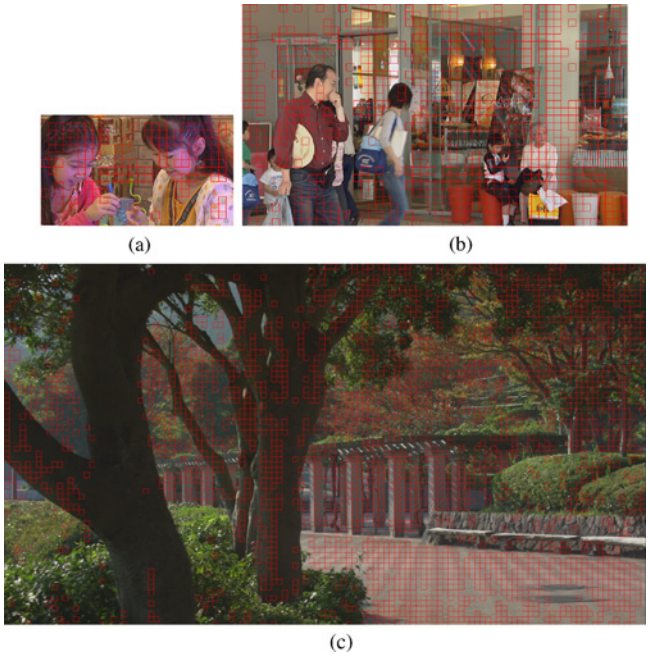


Fig. 18. IP-mode MBs distribution of the WQVGA resolution, standard definition, and high definition sequences. (a) *BlowingBubbles*_416 × 240. (b) *BQMall*_832 × 480. (c) *ParkScene*_1920 × 1080.

by tensor voting from the reconstructed regions, and the bits are more effectively assigned to the residue.

C. Discussion

Table IV gives the average IP-mode ratio under different quantization parameter (QP) levels for the testing sequences. It can be observed that the ratio of IP-mode is increasing with higher QP levels, and the IP-mode through inpainting can save bits for homogeneous visual patterns (texture) than original inter-modes with a more accurate predictor. To be more concrete, Fig. 16 displays a set of distributions of the IP-mode MBs in both I frames and P frames which are outlined by red rims. It can be observed that the IP-mode can be selected not only in homogeneous textural regions, e.g., the cloud in *Highway*, the water in *Container*, and the leaves in *Waterfall*, but also the salient structure regions, e.g., the leaves in *Foreman* and the road in *Highway*. It infers that the predictor by the regularized inpainting is more optimal than intra prediction and motion-compensated prediction in H.264/AVC. In Fig. 17, we also analyze the percentage of IP-MBs for each frame along the time, where the QP level is set to 32, the GoP size is ten frames, and the I frames are labeled in red circles. It can be seen that the IP-mode ratio in P frames is larger than I frames.

Fig. 19. IP-mode MBs distribution of the ultrahigh-definition sequence *ParkJoy*_2560 × 1600.

Table V contrasts the IP-mode MBs to their original modes in the traditional H.264/AVC scheme, where we can observe that the P16×16 mode of P slices are mainly replaced with less burden of side information and more accurate predictor. More examples of IP-mode examples in P frames can be referred to Figs. 18 and 19.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a video coding framework with a structured priority BP-based MB IP-mode. Compared to the local prediction of traditional intra and inter-modes, the optimal predictor can generate lower entropy residue and behave more resilient by exploiting the intrinsic nonlocal and geometric regularity in video samples. Moreover, it can

maintain a better pixel-wise fidelity and the robust error resilience without any assistant information than the existing edge-based inpainting in lossy image coding. Under a global spatio-temporal MRF, the structured sparsity of the coded MBs with IP-mode is inferred by tensor voting projected from the co-located decoded regions, which is imposed on tuning the priority and the visiting order of nodes with an adaptive and more convergent manner. Through relatively few iterations of forward and backward process, the sparse inference of priority BP would ensure a stable marginal belief distribution on the structure and texture through updating local messages and beliefs. Within mode selection on RDO, the IP-mode with structured priority BP could get the best patch arrangement by a spatio-temporal correlation. The computation complexity is shown competitive with one iteration of sparse inference.

In the future, more vision-based technologies are envisaged to get into the video coding framework. We will continue to investigate multidimensional feature tensors, e.g., structure, texture, color, and others, which can be extracted and analyzed for matching and completion, to achieve higher compression ratio and better performance.

REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[2] T. K. Tan, C. S. Boon, and Y. Suzuki, "Intra prediction by template matching," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2006, pp. 1693–1696.

[3] J. Balle and M. Wien, "Extended texture prediction for H.264/AVC intra coding," in *Proc. IEEE Int. Conf. Image Process.*, vol. 6. Sep. 2007, pp. 93–96.

[4] *Key Technology Area Reference Software* [Online]. Available: http://iphome.hhi.de/suehring/tml/download/kta

[5] Y. Ye and M. Karczewicz, "Improved H.264 intra coding based on bi-directional intra prediction, directional transform, and adaptive coefficient scanning," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2008, pp. 2116–2119.

[6] H. Xiong, Z. Yuan, and Y. Xu, "A learning-based framework for low bit-rate image and video coding," in *Proc. IEEE Pacific-Rim Conf. Multimedia*, Dec. 2009, pp. 232–244.

[7] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. IEEE Int. Conf. Comput. Graphics Interact. Tech.*, Jul. 2000, pp. 417–424.

[8] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proc. IEEE Int. Conf. Comput. Vision*, Sep. 1999, pp. 1033–1038.

[9] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 882–889, Aug. 2003.

[10] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 6, no. 6, pp. 721–741, Nov. 1984.

[11] N. Komodakis and G. Tziritas, "Image completion using global optimization," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Patt. Recog.*, vol. 1. Jun. 2006, pp. 442–452.

[12] J. Sun, N. Zheng, and H. Shum, "Stereo matching using belief propagation," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 25, no. 7, pp. 787–800, Jul. 2003.

[13] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Patt. Recog.*, vol. 1. Jul. 2004, pp. 261–268.

[14] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.

[15] J. Sun, L. Yuan, J. Jia, and H. Y. Shum, "Image completion with structure propagation," in *Proc. IEEE Int. Conf. Comput. Graphics Interact. Tech.*, vol. 24. Jul. 2005, pp. 861–868.

[16] D. Liu, X. Sun, F. Wu, S. Li, and Y. Zhang, "Image compression with edge-based inpainting," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 10, pp. 1273–1287, Oct. 2007.

[17] C. Wang, X. Sun, F. Wu, and H. K. Xiong, "Image compression with structure-aware inpainting," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, pp. 21–24.

[18] Z. Xiong, X. Sun, and F. Wu, "Block-based image compression with parameter-assistant inpainting," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1651–1657, Jun. 2010.

[19] C. Zhu, X. Sun, F. Wu, and H. Li, "Video coding with spatio-temporal texture synthesis and edge-based inpainting," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jun. 2007, pp. 813–816.

[20] A. Dumitras and B. G. Haskel, "An encoder-decoder texture replacement method with application to content-based movie coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 6, pp. 825–840, Jun. 2004.

[21] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 29, no. 3, pp. 463–476, Mar. 2007.

[22] Z. Yuan, H. Xiong, and Y. F. Zheng, "A generic video coding framework based on anisotropic diffusion and spatio-temporal completion," in *Proc. IEEE Int. Conf. Acou., Speech, Signal Process.*, Mar. 2010, pp. 926–929.

[23] K. A. Patwardhan, G. Sapiro, and M. Bertalmio, "Video inpainting of occluding and occluded objects," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2005, pp. 69–72.

[24] A. C. Yu, N. K. Ngi, and G. R. Martin, "Efficient intra and inter-mode selection algorithms for H.264/AVC," *J. Vis. Commun. Image Representat.*, vol. 17, no. 2, pp. 322–344, Apr. 2006.

[25] J. Jia and C. Tang, "Inference of segmented color and texture description by tensor voting," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 26, no. 6, pp. 771–786, Jun. 2004.

[26] J. Canny, "A computational approach to edge detection," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–714, Nov. 1986.

[27] Joint Video Team. *JVT Reference Software, Version JM 15.1* [Online]. Available: http://iphome.hhi.de/suehring/tml/download

[28] S. Kumar, L. Xu, M. K. Mandal, and S. Panchanathan, "Error resiliency schemes in H.264/AVC standard," *J. Vis. Commun. Image Representat.*, vol. 17, no. 2, pp. 425–450, Apr. 2006.

Hongkai Xiong (M'01–SM'10) received the Ph.D. degree in communication and information systems from Shanghai Jiao Tong University (SJTU), Shanghai, China, in 2003.

Since then, he has been an Associate Professor with the Department of Electronic Engineering, SJTU. From December 2007 to December 2008, he was a Research Scholar with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA. His current research interests include source coding/network information theory, signal processing, computer vision and graphics, and statistical machine learning. He has published over 90 international journal/conference papers.

Dr. Xiong was a recipient of the New Century Excellent Talents in University Award in 2009. In 2008, he obtained the Young Scholar Award of SJTU. In SJTU, he directs the Intelligent Video Modeling Laboratory and multimedia communication area in the Key Laboratory of Ministry of Education of China—Intelligent Computing and Intelligent Systems, which is also co-granted by Microsoft Research. He has served on various IEEE conferences as a technical program committee member. Also, he acts as a member of the Technical Committee on Signal Processing of Shanghai Institute of Electronics.

Yang Xu received the B.S. degree in electronic engineering from Shanghai Jiao Tong University (SJTU), Shanghai, China, in 2008. She is currently pursuing the M.S. degree in electronic engineering at SJTU.

Her current research interests include video compression, signal processing, and computer vision and graphics.

Yuan F. Zheng (F'97) received the M.S. and Ph.D. degrees in electrical engineering from Ohio State University, Columbus, in 1980 and 1984, respectively. His undergraduate education was received at Tsinghua University, Beijing, China, in 1970.

From 1984 to 1989, he was with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC. Since August 1989, he has been with Ohio State University where he is currently a Professor, and was the Chairman of the Department of Electrical and Computer Engineering from 1993 to 2004. From 2004 to 2005, he spent a sabbatical year with Shanghai Jiao Tong University, Shanghai, China, and continued to be involved as the Dean of the School of Electronic, Information and Electrical Engineering until 2008. His current research interests include two aspects. One is wavelet transform for image and video, and object classification and tracking, and the other is robotics, which includes robotics for life science applications, multiple-robot coordination, legged walking robots, and service robots.

Dr. Zheng was and is on the editorial boards of five international journals. He received the Presidential Young Investigator Award from Ronald Reagan in 1986, and research awards from the College of Engineering of Ohio State University in 1993, 1997, and 2007. He and his students received the Best Conference and Best Student Paper Award a few times in 2000, 2002, and 2006, and received the Fred Diamond for Best Technical Paper Award from the Air Force Research Laboratory, Rome, NY, in 2006. He was appointed to the International Robotics Assessment Panel by the NSF, NASA, and NIH to assess robotics technologies worldwide in 2004 and 2005.

Chang Wen Chen (F'04) received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 1983, the M.S.E.E. degree from the University of Southern California, Los Angeles, in 1986, and the Ph.D. degree from the University of Illinois at Urbana-Champaign, Urbana, in 1992.

Since 2008, he has been a Professor of computer science and engineering at the State University of New York at Buffalo, Buffalo. From 2003 to 2007, he was Allen S. Henry Distinguished Professor with the Department of Electrical and Computer Engineering, Florida Institute of Technology, Melbourne. He was on the Faculty of Electrical and Computer Engineering at the University of Missouri-Columbia, Columbia, from 1996 to 2003, and at the University of Rochester, Rochester, NY, from 1992 to 1996. From 2000 to 2002, he served as the Head of the Interactive Media Group at the David Sarnoff Research Laboratories, Princeton, NJ. He has also consulted with Kodak Research Laboratories, Microsoft Research, Beijing, China, Mitsubishi Electric Research Laboratories, NASA Goddard Space Flight Center, and the U.S. Air Force Rome Laboratories.

Dr. Chen was the Editor-in-Chief for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from January 2006 to December 2009. He has served as an Editor for the PROCEEDINGS OF THE IEEE, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, the IEEE MULTIMEDIA, the *Journal of Wireless Communication and Mobile Computing*, the *EUROSIP Journal of Signal Processing: Image Communications*, and the *Journal of Visual Communication and Image Representation*. He has also chaired and served on numerous technical program committees for the IEEE and other international conferences. He was elected a fellow of the IEEE for his contributions in digital image and video processing, analysis, and communications, and a fellow of the SPIE for his contributions in electronic imaging and visual communications.