Advanced Engineering Forum *Online: 2012-09-26*
ISSN: 2234-991X, Vols. 6-7, pp 309-314
doi:10.4028/www.scientific.net/AEF.6-7.309

A New Error Concealment Algorithm Based On Directional Texture Synthesis

Jiang-Xin Zhang[1,a], Jin Xie[2,b]

[1]Zhejiang Province Key Laboratory of Communications Network and Applications Hangzhou, 310023 China

[2]College of Engineering Information, Zhejiang University of Technology Hangzhou, 310023 China

[a]zjx@zjut.edu.cn, [b]xiejin305@163.com

Keywords: Error concealment; Directional texture synthesis; H.264; SSIM.

Abstract-In this paper, we propose a novel directional texture synthesis based error concealment algorithm to recover damaged video images. It uses the confidence level and structure information to calculate the priority of patch, which contributes to improve the ability to select the best matching block when the damaged area is very large. The JM86 model of H.264 standard is used to evaluate the algorithm. And experimental results show that our algorithm achieved a better image reconstruction results than the improved Multi-directional texture interpolation algorithm, with 1.2 to 1.4dB gain in PSNR and 0.5 percent to 1 percent gain in SSIM.

Introduction

Enacted in 2003, the H.264 standard is widely used in video conference, digital storage media, Internet and other fields[1]. Due to the application of inter/intra frame prediction technology in the video coding framework of H.264 standard, error will spread both in time and space domain. To get a better quality of image decoding without increasing redundancy and delay, error concealment should be used in the decoder.

Nowadays, error concealment methods can be classified into two common categories: spatial error concealment and temporal error concealment. Spatial approaches make use of the surrounding correctly received image data in the same frame to reconstruct the missing blocks. The simplest way is to use the adjacent pixel to interpolate, or find a suitable adjacent block and replace it. The best neighborhood matching (BNM) method[2] finds the best matching block in the same frame, making use of the similarity between blocks. The Multi-Texture Interpolation (MTI) method[3] calculates the texture directions of video image, and then a main direction and a secondary direction are selected. Admittedly, it has improved the accuracy of interpolation direction, but the reconstruction results of the damaged texture are unsatisfactory. In [4], by detecting the edge of surround blocks to get directions of edge of themselves, in order to get the interpolation direction for every pixel in the block, calculate the pertinent pixels and it's weighing determined by distance, and the method of interpolation decided by the status flag of surround blocks. All of those methods are better than MTI algorithm, but there is no good result in the area of texture just because of imprecise divide the direction of edge. In [5], an improved Multi-Texture Interpolation (IMTI) algorithm is proposed. It divides the multi-texture area of the lost macro-block adaptive into several sub-blocks, and uses different methods to conceal the error according to the type of sub-block. It performs well when the texture is plentiful. But the PSNR value would fluctuate around the threshold drastically when the picture texture is not plentiful.

In this paper, we proposed a new directional texture synthesis (DTS) error concealment algorithm in order to overcome the drawback of the algorithms which used in [3,5]. The newly proposed algorithm exploits the improved Sobel operator to calculate the main texture direction and makes full use of the structure information, which contributes to improve the calculate method of the priority of patch and the effect of error concealment. In addition, we use the JM86 model to simulate and compare the effect of DTS algorithm and IMTI algorithm.

The analysis of directional texture synthesis algorithm

Suffering from poor network performance, there might be lost or wrong transmitted packets. As a result, error macro-block will be detected during the decoding process. To reconstruct the damaged texture for video image, we need to calculate the texture's main direction prior to search the best matching block. In [6], Wei uses the Sobel edge detector to obtain the spectrum of edge images, and uses the spectrum characteristics to calculate texture's main direction. However, this direction is not precise enough, and the reconstruction results for video images are unsatisfactory when the texture structure is very complex. And Sun utilizes the gradient values of the pixel which located on the boundary of the target region to calculate of the main and the secondary direction[3]. However, when the damaged area is too large, the result is not satisfactory either.

In this paper, we propose to use the improved Sobel operator to obtain the spectrum of edge images, and determine the direction of texture according to the gradient values of pixel. During decoding, the quality of the output image reconstruction on the edge of the damaged area is highly influenced by the order in which the filling process proceeds, so we must calculate the priority of each patch in order to determine the order of reconstruction. When looking for the best matching block, we need to use different size of patch for different region if the frequency of the texture is not the same. Otherwise, it will lead to unsatisfactory results for the large frequency region, because of the patch is too large, or the patch is too small, or the covered process of small frequency region is too cumbersome that leads to rapidly increase in computing amount. Thereby, we change the patch selection method in this paper. Because of the self-similar of texture, we should start from the center of the patch to find the best matching block for each block and replace it along the texture's main direction of the image. But, when comparing the block obtained by error concealment with the block with no error, the effect of the former will be relatively worse as a matching block. Thus, we set the confidence level as one of weight to search for the matching block. The flowchart of the proposed algorithm is shown in Fig. 1.

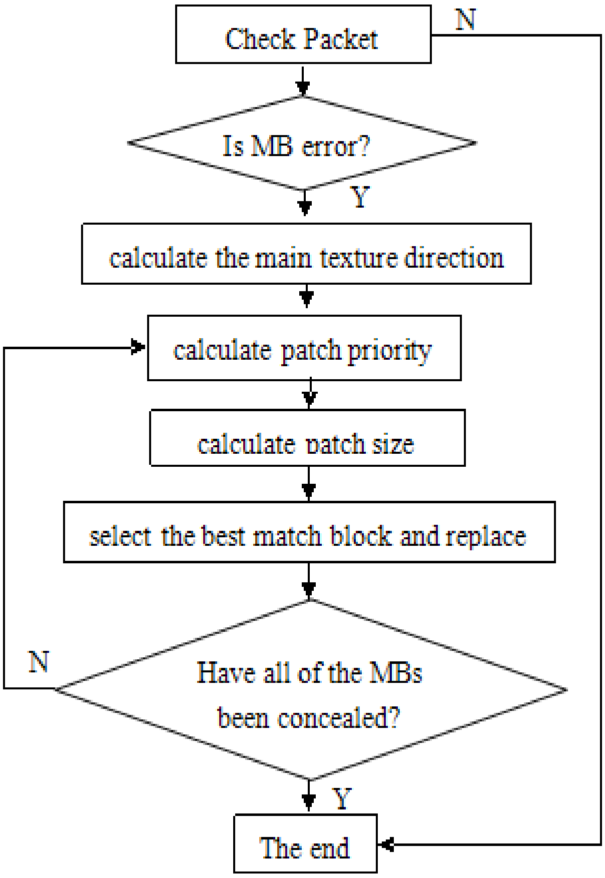

Check Packet
N
Is MB error?
Y
calculate the main texture direction
calculate patch priority
calculate patch size
select the best match block and replace
N
Have all of the MBs been concealed?
Y
The end

Fig .1 The processes of DTS error concealment algorithm

The design of algorithm modules

The calculation of video image's main texture direction. In order to calculate the main texture direction more accurately, this paper adopts the Sobel operator to extract the image's edge. However, it has an opposite direction compared with the commonly coordinate system, and it will cause trouble to the calculation of angle and slope. Therefore, the operator of y-direction has been amended, as shown in eq. 1, and the operator of x-direction as shown in eq. 2. The gradient vector of pixel in an adjacent space with a size of 3×3 is composed by G_x and G_y which belong to the x-direction and y-direction respectively. The gradient modulus and gradient direction to current pixel can be derived by eq. 3 and eq. 4.

$$\nabla_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{1}$$

$$\nabla_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{2}$$

$$\left|Grad(x,y)\right| = \sqrt{G_x^{\ 2} + G_y^{\ 2}} \tag{3}$$

$$\theta(x,y) = \arctan\left(\frac{G_x}{G_y}\right) \tag{4}$$

In order to characterize the texture direction precisely, *θ(x,y)* is evenly divided into 32 parts. And then the dexter 16 direction is selected. The selection process is as follows:

A) Set a counter for each direction;

B) Calculate gradient direction and gradient modulus for each pixel in adjacent space, the counter will summate the gradient modulus on the corresponding direction;

C) The direction associated with counter which has the maximum value is considered as video image's main texture direction.

The calculation of patch priority. In order to determine the order to replace each patch, we define its priority $P(p)$ as the products of the following two terms:

$$P(p) = C(p) + \gamma D(p) \tag{5}$$

where $C(p)$ and $D(p)$ are the confidence term and the data term respectively, and γ is a weight factor for $D(p)$. The definition of $C(p)$ and $D(p)$ are as follows:

$$C(p) = \frac{\sum_{q \in \Psi_p} C(q)}{\left|\Psi_p\right|} \tag{6}$$

$$D(p) = \frac{\left|\nabla I_p^{\perp} \cdot n_p\right|}{\alpha} \tag{7}$$

where $|\Psi_p|$ is the area of Ψ_p with default value is 9×9, α is a normalization factor with typical value 255 for gray-level image, n_p is a unit vector orthogonal to the front $\delta\Omega$ in the point *p*, and ⊥ denotes the orthogonal operator. The priority *P(p)* is computed for every border patch, with distinct patches for each pixel on the boundary of the target region. The function *C(q)* is initialed as follows:

$$C(q)=0\ \ \forall q\in\Omega\text{, and }\ C(q)=1\ \ \forall q\in\varphi \tag{8}$$

as is shown in Fig. 2.

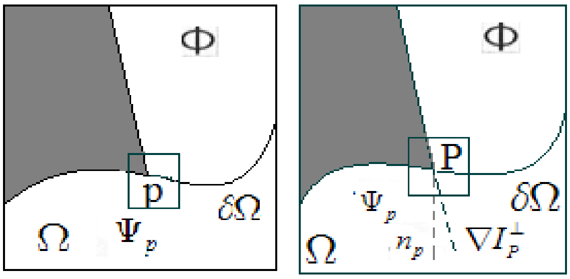


Fig.2 Symbol Map

Having found the proper source patch Ψ_p', the value of each missing pixel, $q\in\Psi_p\cap\Omega$, is replaced by the corresponding pixel in the obtained source patch, $q'\in\Psi_p'$, and the confidence level should be updated to give a lower priority to the pixels which already filled a part of the target region. Therefore the confidence level is updated as:

$$C(q)=\beta C(q') \qquad q\in\Omega\cap\Psi_p \tag{9}$$

where β is a weight factor lower than 1 for the previous confidence level $C(q)$.

Because the change in gradient $\nabla I(p)$ of the pixel on the boundary of the target region reflects the change of image frequency, the higher the value, the faster change the frequency, and more structure information and texture information the image contained. Under this situation, $D(p)$ is even more important, so we determinate the value of γ according to $D(p)$ in this paper. In [6], it will obtain better results for complex texture image when $\gamma = 2$. And letting $\gamma = 1$ to deal with the image which has simple structure[7]. After several trials, we decide to calculate the value of γ by Table 1.

Table 1 The formula to calculate the value of γ

$\lvert\nabla I(p)\rvert$	γ
$\lvert\nabla I(p)\rvert\geq 1.0$	2.25
$0.75\leq\lvert\nabla I(p)\rvert<1.0$	2.00
$0.25\leq\lvert\nabla I(p)\rvert<0.75$	1.75
$\lvert\nabla I(p)\rvert<0.25$	1.50

The calculation of patch size. As is shown in the previous section, the change in the gradient of pixel will reflect the change of image frequency, while the spatial frequency reflects the amount of information contained by image. The smaller the frequency, the smoother the image, therefore the template should be larger; rather than vice versa, the template should be smaller. The size of template has four options: 3×3, 5×5, 7×7, 9×9.

The selection of the best matching block for each patch and replacing. Generally, we assume that the probability distribution of color values are independent for each other[6] to pixel in adjacent space, so we can calculate the difference between target block and candidate block, and select the best matching block. After repeated trials, we found that the performance of the patch obtained via error concealment operation might be relatively worse than the patch with no error concealment operation. Thus, we use the confidence level of each pixel as a weight factor when it comes to evaluate $d(\Psi_p, \Psi_s)$.

$$\Psi_{\hat{p}} = \arg\min_{\Psi_s \subset \hat{\varphi}} d(\Psi_p, \Psi_s) \tag{10}$$

$$d(\Psi_p,\Psi_s)=\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{m}(4\text{-}\beta_1\text{-}\beta_2)\left[\left(p_{ij}^{r}-q_{ij}^{r}\right)^2+\left(p_{ij}^{g}-q_{ij}^{g}\right)^2+\left(p_{ij}^{b}-q_{ij}^{b}\right)^2\right]} \tag{11}$$

where Ψ_p is the target block, $\Psi_{\hat{p}}$ is the best match block, $\hat{\varphi}$ is the search area in the φ along the texture direction, β_1 and β_2 are the confidence level of the pixel p and q respectively, and $m\times m$ is the size of patch. After the best matching block is obtained, we use it to replace the corresponding patch.

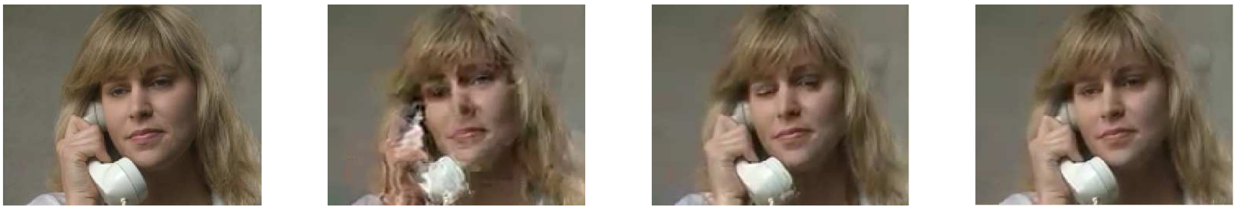
Simulation results

We use JM86 as the test platform. The algorithms are tested on three QCIF video sequences: *Suzie* and *Pamphlet*. Because this paper only focus on the error concealment algorithm which used in intra, all the test sequences are set as III Frame structure and encoded in QCIF format at 30Hz frame rate. Each frame is divided into four slices. The quantization parameter is set at 28. After coding, RTP format packet stream would be generated. In order to show the effectiveness of DTS algorithm, a comparison between the performance of the proposed algorithm and the IMTI algorithm is made. The number of packets is randomly dropped according to the error pattern. In this simulation, the packet loss rate is set as 15%, 28% and 35% respectively. SSIM and PSNR are used as measures of the objective criteria of image quality.

Fig. 3 and Fig. 4 show the result of visual quality of reconstructed video images. Fig.3 (a) and (b) show the original and damaged frame respectively, Fig.3 (c) and (d) show the result generated by IMTI algorithm and DTS algorithm respectively. We can find that the DTS algorithm obtained better visual quality, as well as better complete boundary and smoothness.

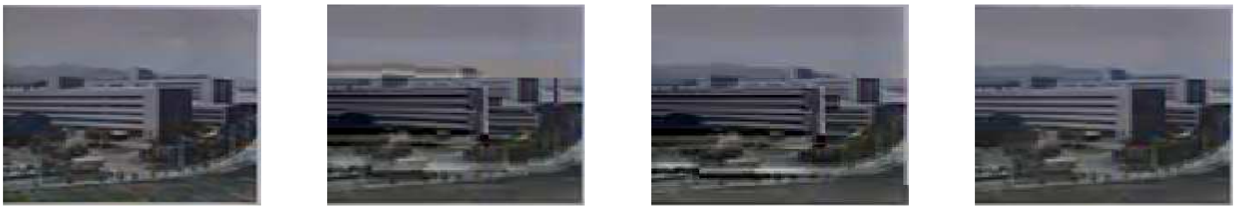
Table 2 shows the PSNR and SSIM performance of the reconstructed sequences with packet loss rate is at 28%. We can see that DTS algorithm is always superior to the IMTI algorithm, especially in the test case that β*=0.95*. Thus, in the subsequent test, we set β*=0.95*.

Table 3 shows the PSNR and SSIM performances of the reconstructed sequences with different packet loss rates. One can see that DTS algorithm outperforms IMTI algorithm for all sequences, with 1.2~1.4dB gain in PSNR and about 0.7% gain in SSIM.



(a)The original image (b)Damaged Image (c)The effect of the IMTI (d)The effect of the DTS

Fig.3 The 129th frame of *Suzie*



(a)The original image (b)Damaged Image (c)The effect of the IMTI (d)The effect of the DTS

Fig.4 The 197th frame of *Pamphlet*

Table 2 PSNR(dB) and SSIM performance of the reconstructed sequences with different β

loss rate		15%		28%		35%	
Algorithm		IMTI	DTS	IMTI	DTS	IMTI	DTS
Suzie	PSNR	32.56	34.04	31.81	33.13	30.57	31.84
	SSIM	0.9395	0.9421	0.9346	0.9414	0.9312	0.9370
Pamphlet	PSNR	33.42	34.55	31.67	33.21	30.71	31.96
	SSIM	0.9410	0.9453	0.9348	0.9419	0.9301	0.9355

Table 3 The PSNR(dB) and SSIM performance of the reconstructed sequences with different packet loss rates

Packet loss ratio 28%		IMTI	DTS (β=0.98)	DTS (β=0.95)	DTS (β=0.9)
Suize	PSNR	32.62	33.87	34.04	33.59
	SSIM	0.9397	0.9413	0.9421	0.9396
Pamphlet	PSNR	33.53	34.21	34.55	33.86
	SSIM	0.9417	0.9431	0.9453	0.9367

Conclusion

In this paper, we propose a new error concealment algorithm based on improved directional texture synthesis, and use the improved Sobel operator to calculate the main texture direction for video image. In addition, a new method is used to calculate the priority of patches. We treat the confidence level of the pixel obtained from concealment with the confidence level of correctly received pixels differently, in order to make concealment for the damaged block to find the best matching block. The results of experiments show that the DTS algorithm has effectively solved the problem of selecting the best matching block from the reference blocks and it has a fine reconstruction quality for both different packet loss rates and video sequences.

Acknowledgements:

This paper is sponsored by the project of Zhejiang Provincial Key Science and Technology Innovative Team (Project Number.2012 R 10011-01).

References

[1] W.-F. Qin, Research of Error Controls And Concealment Technology Based on H.264/AVC, Ph.D dissertation, Sichuan University, Chendu, 2005.

[2] Y. Wang, Q.-F. Zhu, and Leonard SHAW. Maximally Smooth Image Recovery in Transform Coding, IEEE Transactions on Communications. 41 (1993) 1544-1551.

[3] X.-H. Sun, H.-N. Wang, New Spatial Error Concealment Method Based on Multi-texture Direction, Journal of Chinese Computer Systems. 30 (2009) 2264-2267.

[4] Q. Li, J.-M. He, and Y. Ming, Error Concealment Algorithm Based on Edge Detection And Directional Weight for H.264 Intra Frame, Application Research of Computers. 27 (2010) 4798-4800.

[5] X.-H. Wang, F.-H. Li, and Y.-Z. Huang, Improved Spatial Error Concealment Method Based on Multi-texture Direction, Journal of Chinese Computer Systems. 32 (2011) 1452-1455.

[6] L. Wei, X.-H. Chen, Algorithm of Image Inpainting Based on Texture Orientation, Computer Applications. 28 (2008) 2315-2317.

[7] Y.-L. Lin, J.-H. Zhao, and X.-F. Zhu, Y.-J. Hu, Image Inpainting Technology Based on Image Decomposition, Computer Engineering. 36 (2010) 187-192.