Asian Journal of Applied Science and Technology (AJAST)
(Open Access Quarterly International Journal) Volume 2, Issue 2, Pages 526-530, April-June 2018

An Efficient Algorithm for Video Restoration

Anagha Chandran N[1] and Mr.Shehin A.U[2]

[1]S4-M.Tech (ECE-CE), Royal College of Engineering and Technology, Thrissur, Kerala, India.
[2]Department of ECE, Royal College of Engineering and Technology, Thrissur, Kerala, India.

Article Received: 29 January 2018 *Article Accepted: 26 February 2018* *Article Published: 30 April 2018*

ABSTRACT

An approach for filling-in blocks of missing data in wireless video transmission is describing in this paper. For transmission of videos first it divided into image frames. Images are first filled into 8*8 pixels. During the transmission of these videos over fading channels, due to noise entire block of image frame may be lost. Considering the variance difference between patches can reflect the image edge structure, we redefine the priority calculation. The fruit fly optimization algorithm is used for selecting the best matching patch, which avoids the heavy workload and inaccuracy caused by the global search, and it improves the efficiency and accuracy of the algorithm. By restoring damaged images, we can get a better restoration result.

1. INTRODUCTION

During the wireless transmission of, the image/video is transmitted over the wireless channel block by block. Due to severe fading, we may lose an entire block even several consecutive blocks of an image. In the worst case, a whole line of image blocks might be lost. The previous image restoration algorithms have some disadvantages: during the priority calculation, the credibility is not high. The accuracy is also not high when searching the best matching patch. All of these will affect the final results of the image restoration. This paper improves the above shortcomings and the restoration results of the algorithm are more accurate and reliable.

A. FLOWCHART OF PROPOSED SCHEME

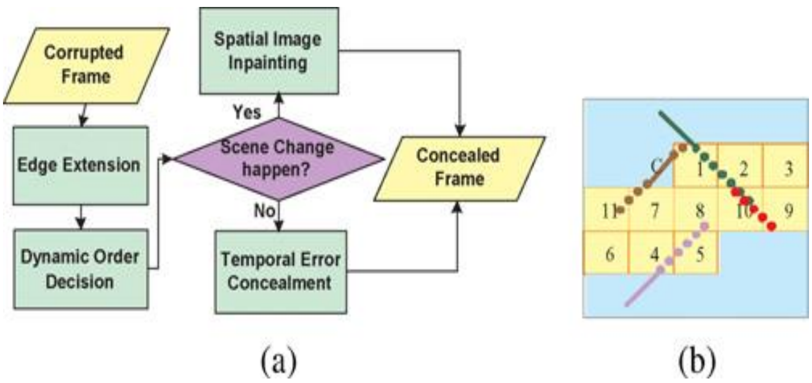


Fig. 1. (a) Flow chart of the current algorithm. The edge extension engine extends the existing edges from uncorrupted MBs into corrupted MBs according to edge direction and strength. In order to retain the edge continuity, a dynamic order decision engine is applied to decide the processing order for the corrupted MBs. Scene change detection analyzes the correlation of the current frame and previous frame. If scene change is determined to have occurred, the spatial image in painting algorithm is applied or temporal error concealment is applied to repair the corrupted image. (b) Edge extension step. The extended points are extended from the detected edge following its direction.

The flowchart of the proposed algorithm is shown in Fig. 1(a). The structure of the corrupted region will be calculated using fruitfly algorithm which extends the existing edges in uncorrupted MBs to corrupted MBs according to isophote direction and variance. As the points shown in Fig. 1(b), the edge points are extended from the detected edge in the same direction. The inpainting process is executed patch by patch. That is, the corrupted patch is retrieved patch by patch.

Moreover, in order to retain the edge structure in each stage structure of missing block is calculated. As shown in Fig. 1(b), the inpainting order for points on extended edges will be similar to that of the original inpainting algorithm. After that, the system can start to perform spatial image inpainting or temporal error concealment as the final stage, as shown in Fig.1(a).

B.FRUITFLY OPTIMISATION ALGORITHM

We introduce the Fruit Fly Optimization algorithm to search the best matching patch. The Fruit Fly Optimization algorithm is a global optimization method based on the fruit fly ransacking behavior. According to the ransacking process, we get the basic principles of the algorithm.

The first is smell inquiry process: using the smell to perceive the various gases in air and determine the food position to close to it. The second is visual orientation process: in the visible range, determining the food position accurate and flying to it.

The specific procedure for the algorithm is as follows:

Step 1 Determine the size of the fruit fly, the maximum number of iteration represented by max *gen* and other parameters. Also, initialize the fruit fly position represented by (X_{axis} ,Y_{axis});

Step 2 Give fruit fly the direction and distance to search the food by smell:

$$\begin{cases} X_i = X_{axis} + RandomValue \\ Y_i = Y_{axis} + RandomValue \end{cases}$$

Step 3 Due to the specific position of the food is not known. So, determine the value of the taste concentration S_i by calculating the distance represented by $Dist_i$ during the individual and origin. The calculation formula is as follows:

$$\begin{cases} Dist_i = \sqrt{\left(X_i^2 + Y_i^2\right)} \\ S_i = 1 / Dist_i \end{cases}$$

Step 4 Take the taste concentration value S; into the function and calculate the taste concentration value of each fruitfly represented by $smell_i$:

$smell_i$ = function(S_i)

Step 5 According to the smell concentration value, and find the individual with the lowest concentration:

[$bestSmell$,$bestindex$] = min($smell_i$)

Step 6 Retain the best smell concentration value and the Position (X,Y) , At this point, all fruit fly population fly to there:

$$\begin{cases} Smellbest = bestSmell \\ X_{axis} = X(bestindex) \\ Y_{axis} = Y(bestindex) \end{cases}$$

Step 7 Conduct the iterative optimization process. First, determine whether the best smell concentration is better than the last best smell concentration, if it is better, carry out the Step 6, and if it is not better, repeat the steps from 2 to 5. Until the terminate condition is reached, stop the process.

We apply the fruit fly optimization algorithm to the best matching patch search process and the smell concentration judgment function is as follows: d(Ψ_p,Ψ_q) When finding the patch with minimal smell concentration, it is the best matching patch.

C.PROCESS OF FRUIT FLY OPTIMIZATION ALGORITHM

Step 1 Mark the areas to be restored in images with a special color;

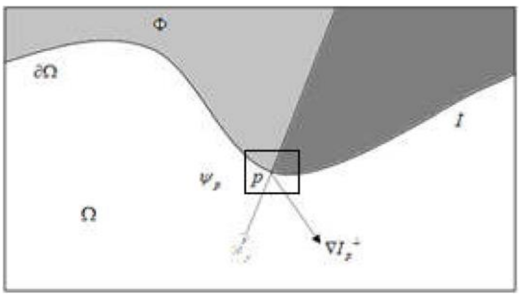
Step 2 Initialize the boundary of the restored areas and record it as a $\partial\Omega$

Step 3 Calculate priority of the restored patch which centers on the contour point, and by comparing, we select the restored patch with the high priority;

Step 4 Use the Fruit Fly Optimization to Search the best matching patch in the known area, then we copy corresponding pixel information in the best matching patch to the unknown area in the restored patch;

Step 5 Update the confidence value and continuing to search the next restored patch with the maximum priority;

Step 6 Repeat steps 3-5 until the damaged area is fully restored. At this point, the restoration process completed.



a. The redefinition of the priority

First, we analysis the damaged patches, as the figure 2 shows, when restoring the damaged patches, we segment the patch along the direction of the n_p. The patch is segmented evenly to be two patches I and 2. We calculate their variances of the known pixels, which are F (1) and F (2). Next, we calculate the variance difference during patches land 2, which is represented by F (p) . The variance difference can reflect the edge characteristics of the damaged areas and it will influence the selection of the damaged patches. So we add it to the priority calculation.

However, as the filling and the selection of the template patch is inaccurate, *C (p)* will quickly drop to zero. At this point, *D(p)* almost have no effect on priority calculation, which leads to wrong filling order and the bad restoration results, so we add them so that they will have no impact on each other. So, we redefine the priority as follows

$$P(p) = C(p) + D(p) + \delta F(p)$$

Where, F(p) = |F(1)- F(2)| and, *C(p)* represents confidence value, which is used to measure the known information proportion in the target patch. *D(p)* represents data item, which decides that the restorations of the image are conducted along the direction of the isophote. n_p represents the external normal line direction of an, a represents a normalized factor, which values 255.

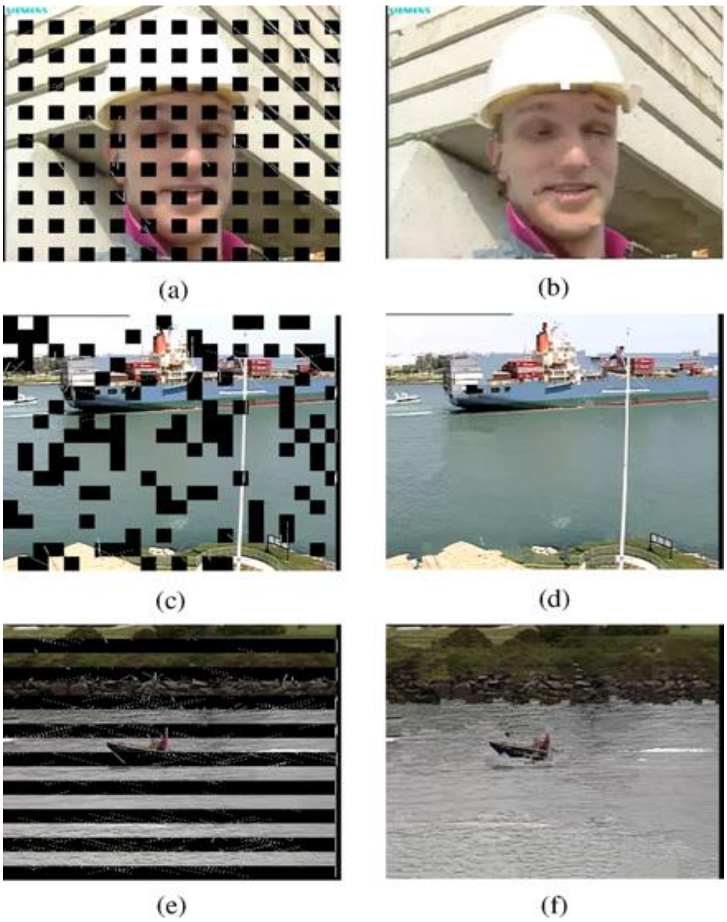


Fig. 4. Experimental results of the proposed inpainting algorithm with edge extension map. (a) Edge extension map for *Foreman*. (b) Inpainting result for *Foreman*. (c) Edge extension map for *Container*. (d) Inpainting result for *Container*. (e) Edge extension map for *Coastguard*. (f) Inpainting result for *Coastguard*.

Fig. 4(a), (c), (e) shows the results based on the proposed algorithm. Fig. 4(b), (d), (f) shows the respective inpainting results.

2. CONCLUSION

In this paper, we improve the original algorithm. When calculating the priority of damaged patch, with the conduction of the restoration, the confidence will decline rapidly. At this point, the data items have no effects on the priority, so we change their geometric relationship to be addition, which avoid the effects they impact on each other. Also, we consider that variance difference between patches can reflect the image edge structure. In order to get a more accurate result, we introduce it into the priority calculation. When searching for the best matching patch, because the fruit fly optimization algorithm has the benefits of easy realization, small calculation and strong global optimization ability and so on, we use it to search for the best matching patch. By restoring the different damaged images, we can see that not only the PSNR values are improved, but also we can get a better restoration result..

REFERENCES

[1] Harrison P.A nonhierarchical procedure for re synthesis of complex texture!! Proc. Int. Conf. Central Europe Computer Graphics, Visualization, and Computer Vision, Plzen, 2001.

[2] Iddo D, Daniel C, Hezy Y. Fragment based image completion. ACM Transaction on Graphics, 22(3):303-312.

[3] Borikar S, Biswas K K, Pattanaik S. Fast Algorithm for Completion of Image with Natural Scenes. http://graphics.cs.ucf.edulborikar!BorikarPaper.pdf, 2005

[4] Criminisi A. Perez p. Toyama K. Region filling and object removal by exemplar-based image inpainting. IEEE Transactions on Image Processing, 2004, 13(9):1200-1212.

[5] Zhang Song, Zhou Xuya. An improved scheme for Criminisi's inpainting algorithm[C]. International Congress on Image and Signal Processing, 2011:2048-2051.