LETTER

Moving Object Completion on the Compressed Domain

Jiang YIWEI[†a)], ***Nonmember***, **Xu DE**[†], ***Member***, **Liu NA**[††], ***Nonmember***, ***and*** **Lang CONGYAN**[†], ***Member***

SUMMARY Moving object completion is a process of completing moving object's missing information based on local structures. Over the past few years, a number of computable algorithms of video completion have been developed, however most of these algorithms are based on the pixel domain. Little theoretical and computational work in video completion is based on the compressed domain. In this paper, a moving object completion method on the compressed domain is proposed. It is composed of three steps: motion field transferring, thin plate spline interpolation and combination. Missing space-time blocks will be completed by placing new motion vectors on them so that the resulting video sequence will have as much global visual coherence with the video portions outside the hole. The experimental results are presented to demonstrate the efficiency and accuracy of the proposed algorithm.

key words: *object completion, compressed domain, discrete cosine transform, video editing*

1. Introduction

Moving object completion is a problem of automatically filling missing parts of a moving object caused by occlusions in video sequences. Applications of object completion range from removal of occlusions to post-processing in the movie-making industry and video tracking.

Over the past few years, computer vision researchers have taken an increased interest in digital inpainting and a number of computable models have been developed [1]–[5]. Digital inpainting methods are mainly classified into two types: inpainting method and completion method. Inpainting method is to inpaint the small scale scratches in images. The inpainting technique based on partial differential equations is pioneered by Bertalmio [2]. The completion method is used for completing the large objects. Completion is mainly classified into two types: 2D completion and 3D completion. 2D completion approach restricts to the completion of spatial information. The temporal component of video has mostly been ignored. For the 3D completion, the missing video parts can be filled in by spatio-temporal patches. Most methods for 3D completion rely on periodic color transitions, layer extraction, or temporally local motion. Reference [3] solves 3D completion by sampling spatio-temporal patches from other video portions. Reference [4] introduces a method of moving object removal and background completion. Shiratori et al. proposes a video completion method using the motion field [5].

Although many completion models have been proposed, most of these models are developed based on the pixel domain. Compressed domain techniques deal with data directly in the compressed domain. The computational complexity can be significantly reduced owing to the smaller amount of processing information and the avoidance of the expensive inverse discrete cosine transform (DCT) computation required to convert values from the frequency domain to the pixel domain. In this paper, a moving object completion method in the compressed domain is presented.

The remainder of the paper is organized as follows. Section 2 describes the proposed moving object completion algorithm in the compressed domain. Experimental results are given in Sect. 3 to demonstrate the efficiency and accuracy of the proposed algorithm. Finally, the conclusion is given in Sect. 4.

2. Algorithm Description

A moving object completion algorithm in the compressed domain is described. The video stream format is IPPP. . . , that is, only the first frame is I frame and the others are P frames. The method is based on the assumption that object blocks have been extracted using the compressed-domain segmentation method [6]. The obstacle is static and based on the block-mode. The algorithm consists of three steps: motion field transferring, thin plate spline interpolation and combination.

2.1 The Pixel Trajectory

In the compressed domain, since the goal of motion compensation is to provide a good prediction, but not to find the correct optical flow, the original motion vectors are prone to errors due to the block-matching and quantization. We convolve the motion vectors of the inter-coded blocks with a Gaussian template to minimize the singularities. The trajectory of the pixel can be obtained using the motion vector. We define $Tr_t^m(x,y)$ as the position of the pixel (x,y) of the tth frame at the mth frame, and $m \leq t$.

$$\begin{aligned} Tr_t^t(x,y) &= (x,y) \\ Tr_t^{t-1}(x,y) &= (x,y) + mv_t^{(x,y)} \\ Tr_t^m(x,y) &= Tr_t^{m+1}(x,y) + mvb_{m+1}^{(i',j')} \end{aligned} \tag{1}$$

where $mv_t^{(x,y)}$ is the motion vector of (x,y) at the tth

Manuscript received January 19, 2009.
Manuscript revised March 22, 2009.

[†] The authors are with the Institute of Computer Science and Engineering, Beijing Jiaotong University, Beijing China.
[††] The author is with Lenovo Corporate R & D, Beijing China.
a) E-mail: javayw@163.com

DOI: 10.1587/transinf.E92.D.1496

frame. $mvb_t^{(i,j)} = (mvbx_t^{(i,j)}, mvby_t^{(i,j)})$. $mvb_t^{(i,j)}$ and $(mvbx_t^{(i,j)}, mvby_t^{(i,j)})$ represent the motion vector of the block (i, j) at the tth frame. If $pixel(x, y) \in block(i, j)$, then $mv_t^{(x,y)} = mvb_t^{(i,j)}$. The position of the pixel $Tr_t^{m+1}(x, y)$ is in the block (i', j') at the frame $m + 1$. From Eq. (1), the backward trajectory of the pixel (x, y) can be represented using $Tr_t^m(x, y)$.

2.2 Motion Field Transferring

Most methods for dynamic completion rely on periodic color or intensity transitions. Instead of transferring color/intensity information directly, Ref. [5] transfers motion field into missing areas, and the author explains the advantage of using motion field. In this step, we follow the idea of [5] using motion vectors to seek the most similar source patch. The distance between two motion vectors can be defined as [5]:

$$d_v(mv_0, mv_1) = 1 - \frac{mv_0 \cdot mv_1}{|mv_0|\,|mv_1|} \tag{2}$$

where mv_0 and mv_1 represent motion vectors.

The similarity between the source patch P_s and the target patch P_t can be calculated as:

$$d_p(P_s, P_t) = \frac{1}{|D|} \sum_{(x,y)\in D} d_v(mv_s^{Tr_t^s(x,y)}, mv_t^{(x,y)}) \tag{3}$$

where D represents collection of pixels in the P_t.

Then the most similar source patch which is outside the hole or the completion result of the previous frame can be obtained as minimizing the Eq. (4).

$$\hat{P}_{\hat{s}} = \arg\min_{P_s} d_p(P_s, P_t) \tag{4}$$

When the object transforms from the previous frame to the current frame, some parts may be disappeared for the reason of being occluded. We record the pixel which is in the range of the object blocks in the previous frame, but losts in the current frame. For those pixels, we calculate the possible position as

$$mv_{\hat{s}}^{Tr_{t-1}^{\hat{s}}(x,y)} + (x, y) \tag{5}$$

If the result is in the range of the obstacle blocks, we can record the motion vector of the missing pixel as

$$mv_t^{F_t^{\hat{s}} \cdot Tr_{t-1}^{\hat{s}}(x,y)+(x,y)} = F_t^{\hat{s}} \cdot mv_{\hat{s}}^{Tr_{t-1}^{\hat{s}}(x,y)} \tag{6}$$

where $F_t^{\hat{s}}$ represents motion vector ratio between the frame t and the frame $\hat{s}$. Because of the block-mode of the DCT, the motion vector of the block can be calculated as the average of the motion vectors of the pixels in the block.

From the above analysis, the motion field transferring model can be written as:

$$X_t = F_t^{\hat{s}} X_{\hat{s}} \tag{7}$$

where X_t is the block motion vector of the frame t.

2.3 Thin Plate Spline (TPS) Interpolation

The previous step completes the missing patch by transferring motion field to the missing blocks in the whole video sequence. In this step, local neighbor information of the missing part will be considered. For the missing object blocks, the new motion vector can be estimated from the local remaining block using the TPS interpolation methods [7].

In the compressed domain, P frames are coded more efficiently using motion compensated prediction from past intra or predictive coded frames. In the P frame, inter mode blocks undergo motion transformation from the previous frame to the current frame. With known corresponding object parts between two consecutive frames, we compute TPS parameters. Then for the object block which is hidden from the obstacle, the motion vector can be obtained from the TPS model as:

$$O_t = P_t Z + B_t \tag{8}$$

where O_t represents the motion vector of the occluded object block at the frame t;

$$P_t = \begin{bmatrix} a_1^x & a_2^x & a_3^x \\ a_1^y & a_2^y & a_3^y \end{bmatrix} \tag{9}$$

$$B_t = \begin{bmatrix} \sum_{i=1}^{n} w_i^x U(|(x_{tar}^i, y_{tar}^i) - (x'_{tar}, y'_{tar})|) - x'_{tar} \\ \sum_{i=1}^{n} w_i^y U(|(x_{tar}^i, y_{tar}^i) - (x'_{tar}, y'_{tar})|) - y'_{tar} \end{bmatrix} \tag{10}$$

$$U(r) = r^2 \log r^2 \tag{11}$$

$$Z = [x'_{tar}\ \ y'_{tar}\ \ 1]^T \tag{12}$$

n is the number of the corresponding object blocks between the frame t and $t - 1$. (x_{tar}, y_{tar}) represents the object block position in the frame t which have corresponding object part in $t - 1$. (x'_{tar}, y'_{tar}) represents the obstacle block position. P_t and w_i are the TPS parameters computed from corresponding object parts. U is the TPS basis functions. It defines a spatial mapping which maps any location in space to a new location. r is the distance from the Cartesian origin.

2.4 Video Completion on the Compressed Domain

The missing space-time blocks will be completed by placing new motion vectors on them. From previous steps, we obtain two completion results. The motion field transferring method depends on the similar motion outside the hole, when the motion of the object is irregular, the result will be less precise. The problem can be solved by combining with the TPS interpolation method, which obtains missed motion vectors using the known motion vectors between two neighbour frames. Then we combine the two methods to get the more precise result. The two results can be integrated following the idea of Kalman filter.

$X_{t|s}$ can be regarded as the complete result using the motion field transferring method evolved from the previous frame s which is the frame outside the hole or has been finished completing. $\hat{X}_{t|t}$ represents the estimate of motion vectors at time t given observations up to and including time t. $P_{t|t}$ represents the measure of the estimated accuracy of the completing result.

$$P_{t|t} = \text{cov}(X_t - \hat{X}_{t|t}) \tag{13}$$

At time t, an observation O_t defined as the completing result of the TPS interpolation method can be considered as $O_t = X_t + v_t$, where v_t is the observation noise which is assumed to be zero mean Gaussian white noise with covariance R_t.

X_t is the true motion vectors at time t. To obtain the more accurate result, we seek to minimize the expected value of the square of the magnitude of the vector, $E\left[\left|X_t - \hat{X}_{t|t}\right|^2\right]$. This is equivalent to minimizing the trace of the posterior estimate covariance matrix $P_{t|t}$. The step has two phases: Predict and Update. The predict phase uses the result of the previous frame to produce an estimate of motion vectors at the current frame. In the update phase, measurement information of the current frame is used to refine the prediction to arrive at a more accurate motion vector estimate.

Predict Step:

For the motion field transferring step, motion vectors of the space-time hole are transferred from the most similar patch outside the hole in the video sequence. The formula can be written as

$$\hat{X}_{t|s} = F_t^s \hat{X}_{s|s} + g_t \tag{14}$$

where g_t is the process noise which is assumed to be drawn from a zero mean multivariate normal distribution with covariance Q_t.

Predicted estimate covariance is

$$P_{t|s} = F_t^s P_{s|s} (F_t^s)^T + Q_s \tag{15}$$

Update Step:

Updated state estimate can be obtained as:

$$\hat{X}_{t|t} = \hat{X}_{t|s} + K_t(O_t - \hat{X}_{t|s}) \tag{16}$$

Updated estimate covariance can be obtained as:

$$P_{t|t} = (I - K_t)P_{t|s} \tag{17}$$

K_t is the Kalman gain and can be defined as:

$$K_t = P_{t|s}(P_{t|s} + R_t)^{-1} \tag{18}$$

Then by minimizing the covariance matrix, we integrate the two completion results and the more accurate result can be obtained. Obstacle blocks which are not contain object can be inpainted by the image inpating method in the compressed domain.

3. Experimental Results

The following experimental results are presented to demonstrate the efficiency and accuracy of the proposed algorithm. Obstacle blocks are pointed out from the DC image to indicate the completion region. Frames in Fig. 1 are from a test sequence of [3]. The black blocks are obstacle blocks. Some parts of the person are missed due to obstacle blocks. Figure 1 (a) are the original frames. Figure 1 (b) and Fig. 1 (c) are the completion results of the moving person on the lawn corresponding to left images using our proposed method and the method in the pixel domain from Ref. [3] respectively. For the image on the left of the first row, the arm of the person is hidden from obstacle blocks. For the image of the second row, the arm and the leg are missed. Using the proposed completion method, missing parts can be completed on the compressed domain. Frames in Fig. 2 are from another test sequence of [3]. In the Fig. 2, completion results of the moving person hidden from the umbrella on the beach are presented on the second column and the third column. Our completion method shows no less precise than the method in the pixel domain. Figure 3 and Fig. 4 are other completion results in the compressed domain. The frame is from the test sequence ship in Fig. 3 and from the test sequence hall in Fig. 4. Figure 3 completes the part of the moving ship

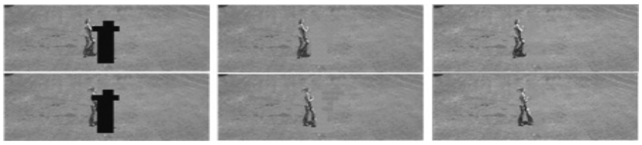

(a) Original frames. (b) Compressed domain. (c) Pixel domain.

Fig. 1 Experimental results of completing the moving person.

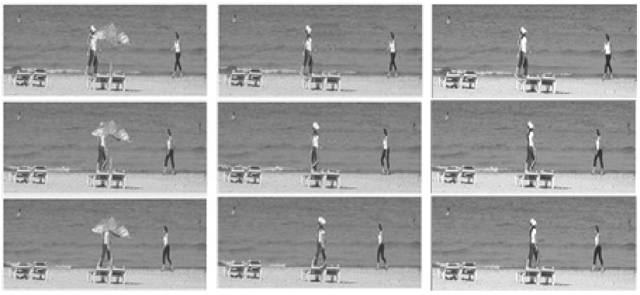

(a) Original frames. (b) Compressed domain. (c) Pixel domain.

Fig. 2 Experimental results of completing the moving person.

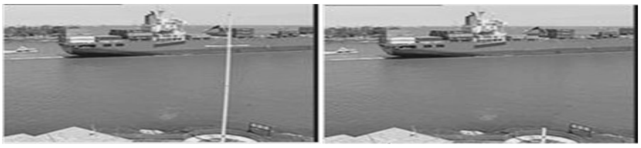
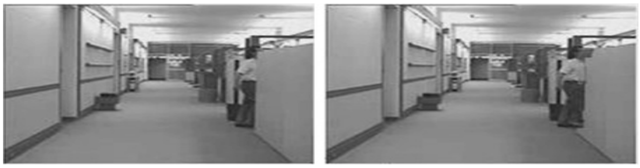
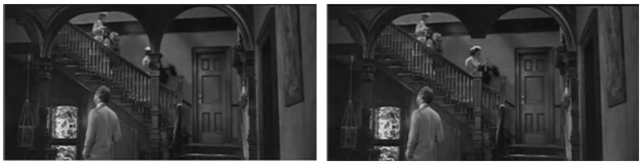

(a) Original frame. (b) Completion result.

Fig. 3 Experimental result of completing the part of the moving ship hidden from the backstay.



(a) Original frame. (b) Completion result.

Fig. 4 Experimental result of completing the person when he goes into the room.



(a) Original frame. (b) Completion result.

Fig. 5 Experimental result of completing the person who is hidden from the pillar.

Table 1 Comparison of the processing time (s).

	Compressed Domain	Pixel Domain
Lawn (Fig. 1)	19	25
Beach (Fig. 2)	23	32
Ship (Fig. 3)	8	10
Hall (Fig. 4)	32	46
Phoenixty (Fig. 5)	37	54

hidden from the backstay. Figure 4 shows the result of completing the person when he goes into the room. When the person goes into the room, some parts of the person will be disappeared. We complete the person and the missed parts of the person occluded by the wall are completed. The frame in Fig. 5 is from the famous movie “Phoenixty”. In this experiment, we complete the person who is hidden from the pillar.

In the original work [5], the author computes the motion field first. Then seeks the most similar source patch for the given target patch in order to assign the motion vectors to the missing pixels in the target patch. Our method is based on the compressed domain. The biggest advantage of processing in the compressed domain is the decrease of the processing time. The avoidance of the expensive inverse DCT computation and the less amount of the processing information in the compressed domain are the two reasons. In addition, the motion field can be obtained directly in the compressed domain, which also can reduce the processing time. It is difficult to get more precise result from the compressed domain than that from the pixel domain. What we can do is improve its accuracy as far as possible. So we combine the two computation methods, which can improve the accuracy and well fit human’s visual perception. Table 1 shows comparison of the processing time between the proposed method and the method in the pixel domain [5].

From the experimental results, we can see that our completion method on the compressed domain can obtain the results that can well fit humans visual perception and largely reduce the processing time.

4. Conclusion

We present a moving object completion method in the compressed domain. The algorithm integrates the motion transferring method and the interpolation method. Since the proposed method uses features extracted from the compressed video directly, computational complexity can be reduced. However, the precision of the method is based on the accurate motion vectors drawn from the compressed domain. Thus it will obtain a more precise result when the moving object is not in a high speed. Because of the block-mode of DCT, we have to take the obstacle as block-mode, which is a limit of the method. In addition, our method depends on the result of object segmentation. Object segmentation in the compressed domain is a difficult problem, which limits the applicability of the proposed method. Future research can be launched to investigate an object segmentation method in the compressed domain.

Acknowledgements

This work is supported by The National Science Fundation project (No.60803072).

References

[1] A. Efros and T. Leung, “Texture synthesis by nonparametric sampling,” Computer Vision, Greece, vol.2, pp.1033–1038, Sept. 1999.

[2] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “Image inpainting,” Proc. SIGGRAPH 2000, New Orleans, USA, July 2000.

[3] Y. Wexler, E. Shechtman, and M. Irani, “Space-time completion of video,” IEEE Trans. Pattern Anal. Mach. Intell., vol.29, no.3, pp.463–476, March 2007.

[4] S.-Y. Park, C.-J. Park, and I. Lee, “Moving object removal and background completion in a video sequence,” Image Vis. Comput., Nov. 2005.

[5] T. Shiratori, Y. Matsushita, S.B. Kang, and X. Tang, “Video completion by motion field transfer,” CVPR 2006.

[6] Z. Liu, Z.Y. Zhang, and L. Shen, “An efficient compressed domain moving object segmentation algorithm based on motion vector field,” J. Shanghai University (English Edition), vol.12, no.3, pp.221–227, June 2008.

[7] F.L. Bookstein, “Principal warps: Thin-plate splines and the decomposition of deformations,” IEEE Trans. Pattern Anal. Mach. Intell., vol.11, no.6, pp.567–585, 1989.