Enhancing the Image Resolution of the Moving Object

Jen-Chi Huang[1], Wen-Shyong Hsieh[2]

[1] Dept. of Information Management, National Penghu University,
Penghu, 880, Twiawn

[2] Dept. of Computer Science and Information Engineering, Shu Te University,
Kaohsiung, 824, Taiwan

leohuang@cm1.hinet.net

論文摘要

在本論文中，我們提出一個新的增加物件影像解析度的方法。利用鄰近不同時間序列的物件影像資訊，來填補物件影像放大後缺少的影像資訊。實驗結果證明利用越多鄰近的物件影像來填補，則會有越高的 PSNR 值。

關鍵詞：物件分割、影像增強、移動偵測

Abstract

In this paper, we proposed a new method for enhancing the resolution of a moving object in the temporal domain, in which the pixels are patched by the neighboring frames. The more neighboring frames are patched, the better a PSNR of the moving object image is obtained.

Keywords: Video Object Segmentation, Image Enhancement, Change Detection.

Introduction

Over the past few years, the quality of video equipment has improved considerably. As a result the development of video-surveillance systems has attracted the attention of researchers. The film used in video-surveillance systems is valuable evidence in traffic offences, traffic accidents, a robbery or other criminal offences [1]. However, in order to reduce the capacity of the file, the video film is usually stored after the quad or the octal processing. It is this processing that causes an image of low quality and makes the moving object become indistinct. Hence, we propose a new method to enhance the image resolution of the moving object.

In order to improve the quality of the image, we have some methods available to us in the special domain, such as the dilation method or the mean method [2]. The dilation method is a simple method where one pixel expands into four pixels. The mean method involves averaging a pixel's 8 neighbors which are not null. In this paper, we propose a new method in the temporal domain where the pixels are patched by the neighboring frames. This is possible because an object can move to another location in video sequences, so that the pixels, which are not detected in one frame, may be detected in other frames. In other words, we can combine the pixels, which are from different frames but depict the same object, to become an image of high quality and resolution of the moving object.

Object segmentation is the most import issue for Mpeg-4 video coding, and the most popular scheme is the Change Detection Method [3] for inter-frame differences. It is simple to implement, and it enables automatic detection of new appearances. In the Change Detection Method, the Frame Different (FD), which indicates the change in pixels between frame n and frame n-1, is found by the following rule:

$$
\begin{aligned}
&\text{for all } (i, j) \in \text{the coordinates of the frame } n \\
&\quad FD_n(i, j) = | I_n(i, j) - I_{n-1}(i, j) | \\
&\quad \text{if } (FD_n(i, j) < V_{thr}) \text{ then } FD_n(i, j) = 0 \\
&\text{end for} \\
&DE_n = Edge\_Detection\,(FD_n)
\end{aligned}
$$

In which (i,j) is the coordinate of the frame n, I(i,j) is the pixel value of the frame n, Vthr is a threshold determined by the significance test, and the Edge_Detection ( ) is applied to the CDMn to extract the rough object shapes called the edge map of significant difference pixel (DEn). However, the Change Detection Method suffers from a 'fat boundary' and the problem of a 'double-edge'. The new wavelet-based object segmentation [4] was proposed to accurately obtain object shapes without the double-edge problem and with a lower time complexity.

However, most moving objects on the road, such as cars or motorcycles move much too fast for the Change Detection Method to work well. It is very easy to extract the newly appearing objects from the pure background without moving objects, and it is easy to find the pure background in a video sequence, because most moving objects are moving in and out in one frame. Hence we propose the pure background method to deal with the problem of the fast moving object.

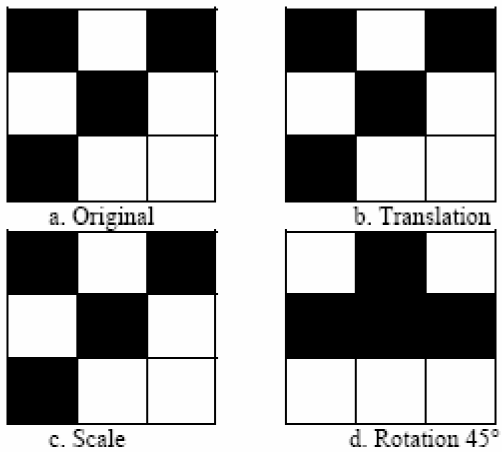

Fig. 1. Three Operations of Camera Motion

The object motion estimation is an import issue to estimate the object motion in two successive video frames. The Pixel-based or Point Mapping technique is currently the primary approach, and it usually uses some special feature point to be the matching point. In this paper, we use a robust feature point in a frame called ‘cross point’, which is located at the corner or the cross-section of the edge lines [5]. Fig. 1 shows the useful characters of the cross point. When the object is translated, rotated or scaled, the cross points in this frame remain the same, because the number of edge points in the 3x3 block are still the same after any motion operation. This allows us to use the relative cross points between two successive frames to find the object motion in the video sequences.

We use the Automatic Feature-based Global Motion Estimation [6] to find the Affine Model of object motion. The simple and efficient algorithm to detect the cross point is as follows:

Cross-Point-Detection Algorithm

$EM_n = CannyEdgeDetection(F_n)$

For all (x,y) in EM_n

$$if\ ((EM_n(x,y)=1) \text{ and } (\sum_{i=-1}^{1}\sum_{j=-1}^{1} EM_n(x+i,y+j) \geq 4))\ CPM_n(x,y)=1$$

$$\text{else } CPM_n(x,y)=0$$

End

Here Fn is the intensity of frame n, EMn is the edge map after applying the Canny Edge Detection method to frame n, and CPMn is the cross points map in frame n. We define a point as a cross point in the 3x3 block, if this point is on the edge line and more than 3 of its 8-neighbor-points are edge points also. If we detect too many cross points in , then we can define the detection algorithm strictly as the edge points of its 8-neighbor-points equaling 4 or 5 edge-points. The fewer number of cross points detected, the lower the time complexity. In addition, our detection algorithm uses a cluster of edge points to define the cross point. Using a cluster of edge points rather than the shape of the edge can deal with the half point problem and the rotation operation in any angle.

In the Cross-Point Match Algorithm [6] in this proposal, a search window of 7x7,is used to define the scope of the match algorithm. For every cross point in frame n, we count the number of the cross points within the search window in frame n-1. And, the cross point which holds only one cross point within the search window in frame n-1 is taken to form a Cross-Point Pair ($(x, y)_{n-1}, (x', y')_n$) and then this pair is recorded in the set of Cross Point Pair (SCPP), this is done because more than one cross point within the search window will create an ambiguous situation. The process of the Cross-Point Match Algorithm is as follows:

Cross-Point Match Algorithm

for all (x,y)

$$\text{if } (CPM_n(x,y)=1) \text{ and } (\sum_{i=-3}^{3}\sum_{j=-3}^{3} CPM_{n-1}(x+i,y+j)=1)$$

$$\text{Record } (x+i,y+i) \text{ and } (x,y) \text{ in SCPP, } where\ CPM_{n-1}(x+i,y+j)=1$$

Endif

Applying the Cross-Point Match Algorithm to CPMn and CPMn-1, we will find all Cross-Point Pairs, which will be used to find the Affine model of object motion.

In the Affine model, we must choice three Cross-Point Pairs to solve the parameters of the Affine Model, and verify the confirmation of these measured parameters for the rest of the Cross-Point Pairs in the SCPP. The ones with the largest confirmation will be assigned to present the Affine Model of the object motion.

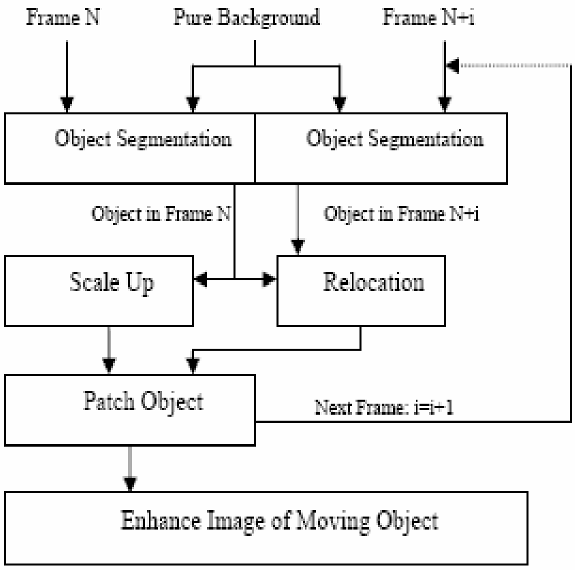

Fig. 2. Overall block diagram of the proposed moving object segmentation algorithm.

Proposed Method

We propose this method in order to enhance the image resolution of the moving object. Fig. 2 shows the over-all block diagram of the proposed algorithm. First, we use the Change Detection Method with frame N and a pure background to segment the moving objects in frame N. We also segment the moving objects in the successive frames using the same method. We choose the frame N as the base frame, and the neighboring frames as patching frames. We use the Cross Point feature and the Affine Model to describe the relative location between the same object in the base frame and in the patching frames. Then we up-scale the QIF image of the moving objects in frame N to the base frame image (I_{BF}) in CIF format, and the algorithm is as follows:

$$
\begin{array}{l}
\text{Scaling - up Algorithm} \\
\text{for all } (x, y) \\
\quad I_{BF}(2x-1, 2y-1) = I_N(x, y); \\
\text{End for}
\end{array}
$$

We use the Affine Model to patch the pixels that are not defined in frame N, by means of the moving object image in the successive frames. As a result we obtain an enhanced image of the moving object. The algorithm is as follows:

$$
\begin{array}{l}
\text{Patch Algorithm} \\
\text{for all } (x, y) \\
\quad [\, x' \; y' \,] = \text{Afine\_Parameter} * [\, 1 \; x \; y \,]; \\
\quad I_{BF}(x', y') = I_N(x, y), \text{ where } I_{BF}(x', y') \text{ is null;} \\
\text{End for}
\end{array}
$$

In the Patch Object step, we use the threshold to avoid the luminance effect and the non-rigid object deformation. If the pixel value of the neighboring frames is less than the maximum value of its 8-neighboring pixels and more than the minimum value of its 8-neighboring pixels, then we will patch the pixel value of the neighboring frames in the null value pixel of the base frame.

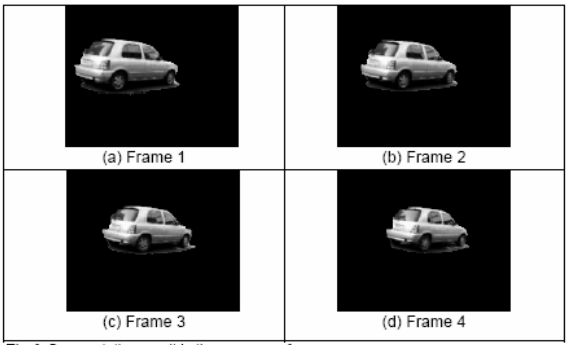

Fig. 3. Segmentation result in the sequence frames

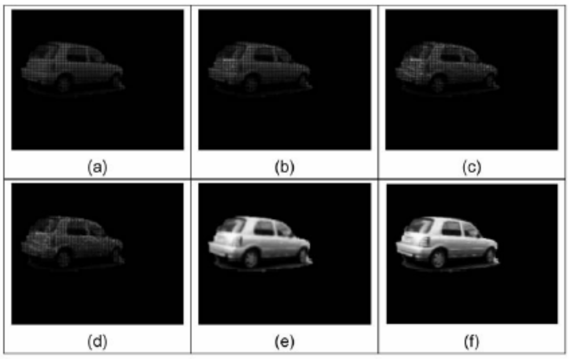

Fig. 4. The sequences of the patch algorithm with frames 1 to 4

Results

We applied the proposed algorithm to "Car", which is a surveillance type video in 352x288 CIF format. First we simulated the Quad Processing System to reduce the resolution to 176x144 QIF format by keeping the odd numbered pixels. Then we determined the pure background and applied the Change Detection Method. After that we obtained the object segmentation results in the sequence frames. Fig. 3 shows the segmentation results of the sequence frames. We defined frame 1, see fig. 4 (a), as the base frame, and scaled it up to a 352x288 CIF format from the 176x144 QIF format in the “Small Car” video sequence. We observed that the base frame contained many pixels of null value. Hence, we used the neighboring frames as the patching frames to patch the null value pixels of the base frame. In figs. 4 (b) to (d), we patched the null value pixels from frames 2 to 4. Finally, we used the mean method, which is the average value of its 8-neighboring pixels, to patch the remaining null-value pixels, and the result is shown in fig. 4 (e). Fig. 4 (f) shows the original CIF image in the “small car” video sequence.

Table 1. Comparison with the original CIF frame

PSNR comparison		Small Car 		Large Car 	
		PSNR	Patch Number	PSNR	Patch Number
Quad Processing	(a) Mean with Frame 1	30.83 dB	0	29.07 dB	0
	(b) Only Patch Frame 2	30.96 dB	329	29.16 dB	752
	(c) Only Patch Frame 3	30.88 dB	454	29.15 dB	666
	(d) Only Patch Frame 4	30.91 dB	331	29.09 dB	555
	(e) Patch Frame 2 and 3	30.97 dB	809	29.21 dB	1442
	(f) Patch Frame 2 ,3,4	31.01 dB	1181	29.18 dB	2073
Octal Processing	(g) Mean with Frame 1	22.29 dB	0	21.49 dB	0
	(h) Only Patch Frame 2	22.52 dB	79	21.56dB	70
	(i) Only Patch Frame 3	23.05 dB	195	21.61 dB	105
	(j) Only Patch Frame 4	22.44 dB	62	21.56 dB	96
	(k) Patch Frame 2 and 3	23.23 dB	264	21.68 dB	180
	(l) Patch Frame 2 ,3,4	23.37 dB	341	21.74 dB	287

Table 1 shows the experiment results in two different video sequences, the “small car” and the “large car”. Table 1 (a)-(f) used the Quad Processing. In (a), we only used the mean method, which is the average of its 8-neighboring pixels. In (b)-(d), we used frame 1 as the base frame and patched it by using only one frame, and then we used the mean method to patch the remainder null-value pixels. In (e), we used frame 1 as the base frame and patched it using frames 2 and 3, and then we used the mean method to patch the remainder null-value pixels. In (f), we used frame 1 as the base frame and patched it using frames 2, 3 and 4, and then we used the mean method to patch the remainder null-value pixels. It was evident that the more neighboring frames we patched, the better the PSNR of the moving object image we obtained. However, we found the PSNR of the “Large Car” in Table 1 (d) not satisfactory, because the object of “Large Car” in frame 4 was deformed too much. Hence, the “Large Car” in frame 4 caused a decreased PSNR in Table 1 (f). Table 1 (g)-(l) used the same general procedure as in (a)-(f) but used the Octal Processing.. The result shows that a better performance of enhancing the PSNR is obtained in Octal Processing compared to Quad Processing. Table 1 (L) shows the best performance in our experiment; we enhance the 1.08 dB of PSNR in the “small car”, after patching it with frames 1 to 4. This also indicates that we obtained a better PSNR by patching more frames.

Conclusion

We proposed a new enhancing resolution method for a moving object. The experiment results have shown that our method has good performance and obtained an enhanced image of the moving object. The more neighboring frames we patched, the better the PSNR of the moving object image we obtained. The performance of our proposed method depends on the accurate Affine parameter estimation of the object motion and the object deformation.

References

1. Rita Cucchiara, Costantino Grana, Massimo Piccardi, and Andrea Prati: Detecting moving objects, ghosts, and shadows in video streams. Lecture Notes in IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 25 , Issue 10 (2003) 1337-1342
2. Gonzalez, Rafael C., and Richard E. Woods.: Digital Image Processing. Addison-Wesley (1992). 518-519
3. C. Kim and J.-N. Hwang: Object-Based Video Abstraction for Video Surveillance Systems. Lecture Notes in IEEE Transactions on Circuits and Systems for Video Technology, VOL. 12, NO. 12 (2002).1128-1138
4. Jen-Chi Huang and Wen-Shyong Hsieh: Wavelet-based moving object Segmentation. Lecture Notes in IEE Electronics Letters, Vol. 39, NO. 19 (2003) 1380-1382
5. L. G. Brown: A survey of image registration techniques. Lecture Notes in ACM Computing Surveys, 24 (1992) 325-376
6. Jen-Chi Huang and Wen-Shyong Hsieh: Automatic Feature-based Global Motion Estimation in Video Sequencess. Lecture Notes in IEEE Transactions on Consumer Electronics,Vol. 50, Issue 3 (2004). 911- 915