MOTION VECTOR REFINEMENT FOR FRAME RATE UP CONVERSION ON 3D VIDEO

Yutao Liu, Xiaopeng Fan, Xinwei Gao, Yan Liu, Debin Zhao

Dept. of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
liuyutao2008@gmail.com, {fxp, xwgao.cs, liuyan, dbzhao}@hit.edu.cn

ABSTRACT

With the rapid development of digital video technology, frame rate up conversion is widely used. In this paper, a novel motion vector refinement method for frame rate up conversion on depth based 3D video is proposed. Our method involves two major stages in frame rate up conversion which are motion estimation and motion vector filtering. In the motion estimation process, the depth constraint to block matching algorithm is introduced in the bi-directional motion estimation method to obtain the motion vectors. In the motion vector filtering process, a depth-guided filter is designed to enhance the consistence of motions in the same depth plane. The refined motion vectors are used for frame interpolation. Experimental results show that the proposed method achieves 0.45 dB gain in terms of PSNR on average and improves the visual quality of the frame rate up-converted video.

Index Terms—frame rate up conversion, depth map, motion estimation, motion vector refinement

1. INTRODUCTION

With the development of display technology, the screen refresh rate is becoming higher and people can watch more vivid videos. However, this high refresh rate can't be fully used when the video source has a lower frame rate than the screen's refresh rate. In some other cases, for low bitrate video transmission, some frames in an original high frame rate video are transmitted and the remaining frames are interpolated at the receiver. To meet these application requirements, frame rate up conversion (FRUC) was proposed and widely used.

FRUC refers to the process to convert a low frame rate video to the corresponding high frame rate video by inserting some new frames into the low frame rate video. Early FRUC techniques include frame repetition and frame averaging. However, the motions in a video sequence are ignored in frame repetition and frame averaging techniques so that the motion blurring or ghost artifacts are produced in the moving regions. To deal with this, a more effective FRUC technique, named motion-compensated interpolation (MCI), was proposed by considering motion estimation (ME) in FRUC. ME determines the quality of interpolated frames so that it plays a key role in MCI. To improve the accuracy of ME, some relevant works have been done. In [1], de Hann et al. proposed a 3-D recursive search (3-DRS) approach in motion estimation to get the true motions. Choi et al. [2] proposed a bi-directional motion estimation method in FRUC. Tai et al. [3] proposed a multi-pass motion estimation scheme to get more accurate motion vectors (MVs) for FRUC. Since the pixels are absent in the to-be-interpolated frame, the MVs obtained by ME may be not accurate enough or the MV field is not consistent enough. Therefore, MV filtering is necessary for FRUC. A MV smoothing method based on the median filter [4] was designed to make MVs more consistent and the motion field smoother.

Frame interpolation is performed on the basis of the MVs obtained by ME. The to-be-interpolated frame is usually divided into regular blocks and each block is interpolated by averaging the reference blocks pointed by its MVs in the previous and following original frames. This traditional MCI method usually results in blocking artifacts at block boundaries. Overlapped block motion compensation (OBMC) [5] [6] was applied in FRUC to reduce the blocking artifacts. However, OBMC may generate blurring or over-smoothing artifacts, as the weights used in OBMC are fixed. An adaptive OBMC (AOBMC) [7] was presented to adjust the weights based on the reliability of neighboring motion vectors and it outperforms OBMC. In recent years, a spatio-temporal auto-regressive (STAR) model [8] that combines the spatial and temporal correlation of pixels was proposed for FRUC. Afterwards, A motion-aligned autoregressive model (MAAR) [9] that takes ME into consideration was proposed, in which the pixel to be interpolated is estimated as a weighted summation of the motion aligned pixels in the previous or following frame. In [10], Liu et al. proposed a multiple hypotheses Bayesian scheme which incorporated both the temporal motion model and the spatial image model for FRUC.

Nowadays, three-dimensional television (3DTV) technology develops rapidly. In order to represent 3D video, a new data format which includes multi-view video plus corresponding depth maps (MVD) was proposed. The depth map in MVD provides the distance information between the objects and the video camera. As well as FRUC on 2D video, FRUC on 3D video is also needed and some relevant techniques have been investigated. Choi et al. [11] proposed a framework for spatial and temporal up-conversion for depth video. In [12], Lu et al. proposed a depth-constrained motion vector post-processing FRUC scheme for 3D video.

In this paper, a novel motion vector refinement method for FRUC on 3D video is proposed. In the method, motion estimation and motion vector filtering in FRUC are considered. For motion estimation, the depth constraint to block matching algorithm is introduced in the bi-directional ME to obtain motion vectors. For motion vector filtering, a depth-guided filter is designed to refine the obtained motion vectors. The refined motion vectors are used for motion compensation based frame interpolation.

The rest of this paper is organized as follows. The proposed method is presented in Section 2. Experimental results and analysis performed on various video sequences are given in Section 3. At last, conclusions are drawn in Section 4.

2. PROPOSED METHOD

In this section, the proposed motion vector refinement method is presented in detail.

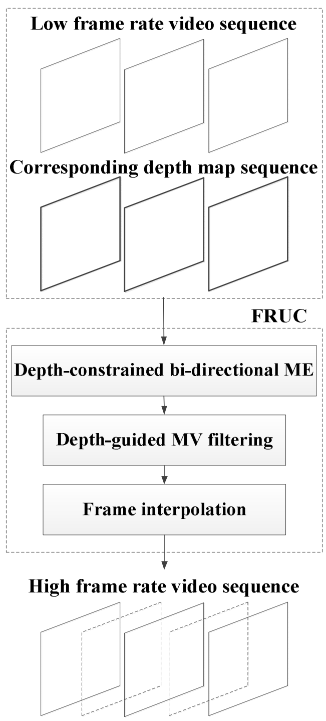


Fig. 1. Illustration of FRUC framework

2.1. FRUC framework

The FRUC framework is shown in Fig.1, the input includes low frame rate video sequence and the corresponding depth map sequence and the output is high frame rate video sequence. The proposed motion vector refinement method includes depth-constrained bi-directional ME and depth-guided MV filtering. Depth-constrained bi-directional ME is to get the MVs and depth-guided MV filtering is to smooth the MV field. At last, the refined MVs are used in frame interpolation. In the following, we will detail the proposed motion vector refinement method.

2.2. Depth-constrained bi-directional ME model

Depth-constrained bi-directional ME is based on bi-directional ME [2] which is an effective ME method to obtain MVs. Bi-directional motion estimation method centers on the to-be-interpolated block B_i and searches for the best linear motion trajectory that passes through B_i using block matching algorithm. As depicted in Fig.2, f_{2k} and f_{2k+2} represent the previous and following original frames, the intermediate frame f_{2k+1} represents the frame to be interpolated and the temporal distance between f_{2k+1} and f_{2k} is equal to that between f_{2k+1} and f_{2k+2}. The MV of block B_i is determined by:

$$MV(B_i) = \underset{MV}{\operatorname{argmin}} \sum_{p \in B_i} \left| f_{2k}(p - MV) - f_{2k+2}(p + MV) \right|, \tag{1}$$

where p represents the coordinate (x,y) in B_i .

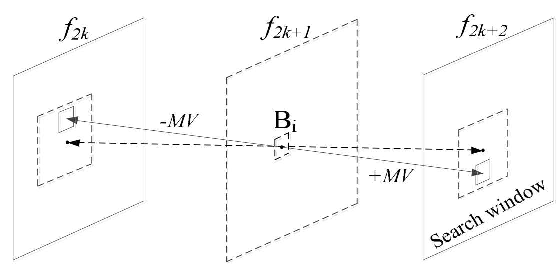


Fig. 2. Illustration of bi-directional ME model

As can be seen in Fig.2, after bi-directional ME, the two most similar blocks corresponding to block B_i in f_{2k} and f_{2k+2} are obtained. However, they are likely to be different in depth so that they may belong to different objects with similar textures. As we know, motions of different objects are usually irrelevant. Therefore, the MV obtained by equation (1) is not accurate enough. Now we make a more rigorous assumption that the best linear motion trajectory that passes throng B_i is in the same depth plane within a short time interval. Based on this assumption, we propose a depth-constrained bi-directional ME model by taking depth constraint into consideration. As shown in Fig.3, d_{2k} and d_{2k+2} are the depth maps corresponding to f_{2k} and f_{2k+2}. D_i is the depth block corresponding to block B_i, the two blocks in f_{2k} and f_{2k+2} need to be constrained by the two corresponding depth blocks in d_{2k} and d_{2k+2}. The MV of block B_i is obtained by:

$$MV(B_i) = \underset{MV}{\operatorname{argmin}} \sum_{p \in B_i} \left(\Delta f + \lambda \Delta d \right), \tag{2}$$

where λ is a positive parameter and

$$\Delta f = \left| f_{2k}(p - MV) - f_{2k+2}(p + MV) \right|, \tag{3}$$

$$\Delta d = \left| d_{2k}(p - MV) - d_{2k+2}(p + MV) \right|. \tag{4}$$

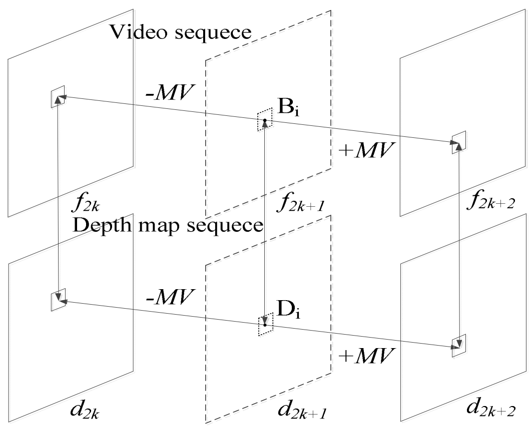


Fig. 3. Illustration of depth-constrained bi-directional ME model

2.3. Depth-guided motion vector filtering

After bi-directional ME, each block in the to-be-interpolated frame is assigned a pair of motion vectors: -MV points to the previous block and +MV points to the following block, as shown in Fig.2. However, the MV which is directly used for frame interpolation may cause blocking artifacts due to the local inconsistence of MV field. Therefore, motion vector filtering is introduced after ME to maintain the local consistence of MV field. The vector median filtering method [13] was utilized for MV filtering in [14] as follows:

$$MV(E) = \underset{K}{\operatorname{argmin}} \left[\sum_{J=1}^{8} \left| MV(K) - MV(J) \right| \right], \tag{5}$$

where *MV(E)* represents the to-be-filtered MV of the current block E and *MV*(*K*) is one of the eight neighbor motion vectors, depicted in Fig.4.

1	2	3
4	E	5
6	7	8

Fig. 4. Illustration of local MV filtering window

The median filter for MV filtering is able to maintain the local consistence of MV field by considering the eight neighbor MVs surrounding the to-be-filtered MV. However, the depth-affected correlation among the MVs of adjacent blocks is not taken into account. Generally speaking, the adjacent blocks in the same depth plane likely to belong to the same object and their motions tend to be consistent. So the correlation among the MVs of adjacent blocks in the same depth plane tends to be stronger than the correlation among the MVs of adjacent blocks in different depth planes. To measure the depth-affected correlation between *MV(E)* of the to-be-filtered block E and *MV(i)* of its neighbor block i, we define ω_i as:

$$\omega_i = \frac{1}{\sum_x \sum_y \left| D_i(x,y) - D_E(x,y) \right| + \alpha}, \tag{6}$$

where D_E and D_i represent the corresponding depth blocks of the current block E and its neighbor block i, (x,y) represents the coordinate in D_i and D_E, α is a positive parameter. However, the corresponding depth map of the to-be-interpolated frame is not available, D_i and D_E can't be directly used. Instead, the depth blocks in the previous and following original depth maps aligned by the motion vectors are utilized to estimate D_i as follows:

$$D_i(x,y) = \frac{d_{2k}[(x,y) - MV(B_i)] + d_{2k+2}[(x,y) + MV(B_i)]}{2}, \tag{7}$$

where d_{2k} and d_{2k+2} are the previous and following depth maps . B_i is the corresponding block to depth block D_i as Fig.3 shows. D_E is calculated in the same way. Finally the MV of block E is filtered as a weighted summation of the MVs in the local MV filtering window (shown in Fig. 4):

$$MV(E) = \sum_j \eta_j \cdot MV(j), \tag{8}$$

where $j \in \{1,2,3,4,E,5,6,7,8\}$ and *MV(j)* is filtered by equation (5) at first, η_j is a normalized parameter calculated by:

$$\eta_j = \omega_j \Big/ \sum_i \omega_i, \tag{9}$$

where $i \in \{1,2,3,4,E,5,6,7,8\}$, ω_j and ω_i are calculated by equation (6).

2.4. Frame interpolation

Frame interpolation is implemented at last in the FRUC framework. The motion compensation based frame interpolation method can be utilized in the FRUC framework. In this paper, MCI, AOBMC [7] and MAAR [9] are chosen for experiments.

3. EXPERIMENTAL RESULTS

In this section, various experiments are performed to evaluate the performance of the proposed motion vector refinement method. In the first subsection, we will give the experiment settings. In the second subsection, the objective and subjective interpolation performance comparisons will be presented.

3.1. Experiment settings

We integrate the proposed motion vector refinement method into MCI, AOBMC [7] and MAAR [9] respectively. 3-D video sequences: *Kendo* (1024×768, 30fps), *Dancer* (1920×1088, 30fps), *Pantomime* (1280×960, 30fps), *Lovebird1* (1024×768, 30fps) and *Poznan_Street* (1920×1088, 30fps) are used as test video sequences. In the experiments, we remove the first 30 even frames and interpolate them by FRUC. The interpolated 30 even frames are compared with the original 30 even frames to evaluate the interpolation performance. The block size in bi-directional ME is 8×8. In equation (2), λ=1 and in equation (6), α=1.

3.2. Performance comparison

To verify that the proposed depth-constrained bi-directional ME model in equation (2) is more accurate than bi-directional ME model in equation (1), we do some experiment without adding the proposed depth-guided MV filtering, then we add the depth-guided MV filtering method to confirm the advantage of the whole proposed motion vector refinement method. We compute the average PSNR and SSIM [15] between the original 30 even frames and the interpolated 30 even frames, shown in Table 1, Table 2 and Table 3. In Table 1, Table 2 and Table 3, DCME refers to the method that we replace the original bi-directional ME with the proposed depth-constrained bi-directional ME in MCI, AOBMC and MAAR (initialized by AOBMC) respectively. DCME+DMVF refers to the method that we add the depth-guided MV filtering based on DCME, namely the whole proposed method. From Table 1, it can be observed that for all test sequences the PSNRs of DCME are consistently higher than the PSNRs of MCI and the average PSNR gain is 0.32 dB which means that the proposed depth-constrained bi-directional ME can obtain more accurate MVs than the original bi-directional ME. Meanwhile, for all the test sequences the PSNRs of DCME+DMVF are consistently higher than the PSNRs of DCME and the average PSNR gain is 0.15 dB compared to DCME and 0.47 dB compared to MCI which indicates that DCME+DMVF refines the MVs obtained by DCME. It can also see that the average SSIM of DCME+DMVF is the highest among the competing methods. In general, the proposed method enhances the accuracy of the motion vectors and improves the quality of frame interpolation in MCI, AOBMC and MAAR. The same conclusion can be drawn from analysis of Table 2 and 3. The visual quality comparisons are given in Fig.5, 6 and 7. From Fig.5, 6 and 7, the blocking artifacts or incorrect interpolated blocks can be seen in (b). AOBMC can reduce the blocking artifacts to some extent shown in (d). As we can see in (f), MAAR reduces the blocking artifacts obviously and gets better visual quality than AOBMC and MCI. However, the blocking artifacts aren't eliminated completely by MCI, AOBMC and MAAR due to the inaccurate MVs used for interpolation. Conversely in (c), (e) and (g), the blocking artifacts can't be seen which proves that the proposed method integrated into MCI, AOBMC and MAAR respectively enhances the accuracy of MVs and improves the visual quality.

4. CONCLUSIONS

In this paper, we proposed a novel motion vector refinement method for frame rate up conversion on depth based 3D video. In the method, a depth-constrained bi-directional motion estimation model and a depth-guided motion vector filtering method were proposed. In the depth-constrained bi-directional motion estimation model, depth constraint to block matching algorithm was added to the original bi-directional motion estimation model to get more accurate motion vectors. In the depth-guided motion vector filtering method, the depth-affected correlation among motion vectors of adjacent blocks was analyzed and a depth-guided motion vector filter was designed. The refined motion vectors were used for frame interpolation. Experimental results show that the proposed method enhances the accuracy of motion vectors and improves the overall quality of the up-converted video in subjective and objective aspects.

Table 1. PSNRs and SSIMs of MCI based interpolated frames

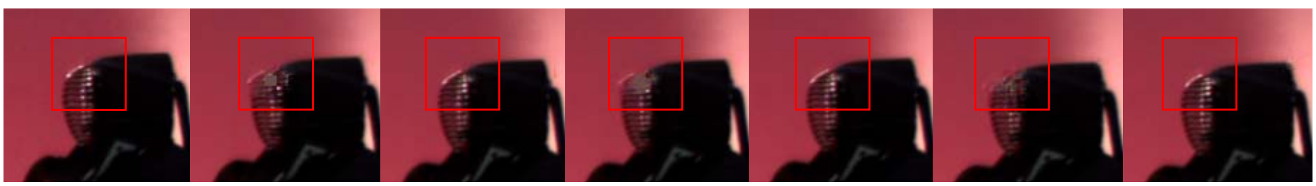
Sequence	MCI		DCME		DCME+DMVF	
	PSNR(dB)	SSIM	PSNR(dB)	SSIM	PSNR(dB)	SSIM
Kendo	32.01	0.9610	32.73	0.9612	32.86	0.9616
Dancer	31.25	0.9601	31.49	0.9593	31.56	0.9593
Pantomime	29.62	0.9529	29.80	0.9527	29.94	0.9530
Lovebird1	44.15	0.9898	44.25	0.9899	44.62	0.9903
Poznan_Street	37.33	0.9447	37.68	0.9451	37.70	0.9450
Average	**34.87**	**0.9617**	**35.19**	**0.9616**	**35.34**	**0.9618**

Table 2. PSNRs and SSIMs of AOBMC based interpolated frames

Sequence	AOBMC		DCME		DCME+DMVF	
	PSNR(dB)	SSIM	PSNR(dB)	SSIM	PSNR(dB)	SSIM
Kendo	32.18	0.9630	32.97	0.9637	33.10	0.9637
Dancer	31.44	0.9620	31.68	0.9615	31.73	0.9609
Pantomime	29.69	0.9542	29.90	0.9543	30.05	0.9546
Lovebird1	44.31	0.9901	44.44	0.9902	44.78	0.9906
Poznan_Street	37.58	0.9456	37.92	0.9459	37.93	0.9459
Average	**35.04**	**0.9630**	**35.38**	**0.9631**	**35.52**	**0.9631**

Table 3. PSNRs and SSIMs of MAAR based interpolated frames

Sequence	MAAR		DCME		DCME+DMVF	
	PSNR(dB)	SSIM	PSNR(dB)	SSIM	PSNR(dB)	SSIM
Kendo	32.41	0.9620	33.12	0.9627	33.25	0.9629
Dancer	31.58	0.9616	31.81	0.9613	31.87	0.9620
Pantomime	30.16	0.9566	30.30	0.9563	30.37	0.9566
Lovebird1	44.13	0.9895	44.23	0.9895	44.42	0.9898
Poznan_Street	37.54	0.9476	37.94	0.9481	37.93	0.9480
Average	**35.16**	**0.9635**	**35.48**	**0.9636**	**35.57**	**0.9639**



(a) (b) (c) (d) (e) (f) (g)

Fig. 5. FRUC results for *Kendo* (2nd frame). (a) Original, (b) MCI, (c) Proposed method integrated into MCI, (d) AOBMC, (e) Proposed method integrated into AOBMC, (f) MAAR, (g) Proposed method integrated into MAAR

(a) (b) (c) (d) (e) (f) (g)

Fig. 6. FRUC results for *Lovebird1* (16th frame). (a) Original, (b) MCI, (c) Proposed method integrated into MCI, (d) AOBMC, (e) Proposed method integrated into AOBMC, (f) MAAR, (g) Proposed method integrated into MAAR

(a) (b) (c) (d) (e) (f) (g)

Fig. 7. FRUC results for *Poznan_Street* (52th frame). (a) Original, (b) MCI, (c) Proposed method integrated into MCI, (d) AOBMC, (e) Proposed method integrated into AOBMC, (f) MAAR, (g) Proposed method integrated into MAAR

5. ACKNOWLEDGEMENT

This work was supported in part by the Major State Basic Research Development Program of China's 973 Program under Grant 2009CB320905, the National Science Foundation of China (NSFC) under grants 61272386 and 61100096, and the Program for New Century Excellent Talents in University (NCET) of China (NCET-11-0797).

6. REFERENCES

[1] G. de Haan, P. W. Biezen, H. Huijgen, O. A. Ojo, "True-motion estimation with 3-D recursive search block matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 5, pp. 368–379, Oct. 1993.

[2] B. T. Choi, S. H. Lee, S. J. Ko, "New frame rate up-conversion using bi-directional motion estimation," *IEEE Trans. Consum. Electron.*, vol. 46, no. 3, pp. 603–609, Aug. 2000.

[3] S. C. Tai, Y. R. Chen, Z. B. Huang, C. C. Wang, "A multi-pass true motion estimation scheme with motion vector propagation for frame rate up-conversion applications," *J. Display Tech.*, vol. 4, no. 2, pp. 188–197, Jun. 2008.

[4] Suk-Ju Kang, Dong-Gon Yoo, Sung-Kyu Lee, Young-Hwan Kim, "Design and implementation of median filter based adaptive motion vector smoothing for motion compensated frame rate up-conversion," *Consumer Electronics, 2009 IEEE 13th International Symposium on*, pp.745-748, 25-28 May 2009.

[5] M. T. Orchard, G. J. Sullivan, "Overlapped block motion compensation: An estimation-theoretic approach," *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 693–699, Sep. 1994.

[6] J. Zhai, K. Yu, J. Li, S. Li, "A low complexity motion compensated frame interpolation method," *in Proc. ISCAS*, vol. 5, pp.4927–4930, May 2005.

[7] B. D. Choi, J. W. Han, C. S. Kim, S. J. Ko, "Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 4, pp. 407–416, Apr. 2007.

[8] Yongbing Zhang, Debin Zhao, Xiangyang Ji, Ronggang Wang, Wen Gao, "A Spatio-Temporal Auto Regressive Model for Frame Rate Upconversion," *IEEE Trans. Circuits Syst. Video Technol.*, vol.19, no.9, pp.1289-1301, Sept. 2009.

[9] Y. B. Zhang, D. B. Zhao, S. W. Ma, R. G. Wang, W. Gao, "A motion-aligned auto-regressive model for frame rate up conversion," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1248–1258, May 2010.

[10] Hongbin Liu, Ruiqin Xiong, Debin Zhao, Siwei Ma, Wen Gao, "Multiple Hypotheses Bayesian Frame Rate Up-Conversion by Adaptive Fusion of Motion-Compensated Interpolations," *IEEE Trans. Circuits Syst. Video Technol.*, vol.22, no.8, pp.1188-1198, Aug. 2012.

[11] Jinwook Choi, Dongbo Min, Bumsub Ham, Kwang¬hoon Sohn, "Spatial and temporal up-conversion technique for depth video," *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pp.3525-3528, 7-10 Nov. 2009.

[12] Qingchun Lu, Xiangzhong Fang, Chong Xu, Yongzhe Wang, "Frame Rate Up-Conversion for Depth-Based 3D Video," *Multimedia and Expo (ICME), 2012 IEEE International Conference on*, pp.598-603, 9-13 July 2012.

[13] J. Astola, P. Haavisto, Y. Neuvo, "Vector median filters," *Proceedings of the IEEE*, vol.78, no.4, pp.678-689, Apr. 1990.

[14] Z. Gan, L. Qi, X. Zhu, "Motion compensated frame interpolation based on H.264 decoder," *Electronics Letters*, vol. 43, no. 2, pp. 96-98, January 18 2007.

[15] Zhou Wang, Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol.13, no.4, pp.600-612, April 2004.