Reliable Camera Motion Estimation from Compressed MPEG Videos using Machine Learning Approach

Zheng Wang[1], Jinchang Ren[2, *],Yubin Wang[1], Meijun Sun[3] and Jianmin Jiang[3]

wzheng@tju.edu.cn jinchang.ren@strath.ac.uk

{wangyubin, sunmeijun, jmjiang}@tju.edu.cn

[1] School of Computer Software, Tianjin University, Tianjin, 300072, China

[2] Centre for excellence in Signal and Image Processing, University of Strathclyde, Glasgow, U.K.

[3] School of Computer Science and Technology, Tianjin University, Tianjin, 300072, China

Abstract: As an important feature in characterising video content, camera motion has been widely applied in various multimedia and computer vision applications. A novel method for fast and reliable estimation of camera motion from MPEG videos is proposed, using support vector machine (SVM) for estimation in a regression model trained on a synthesised sequence. Experiments conducted on real sequences show that the proposed method yields much improved results in estimating camera motions whilst the difficulty in selecting valid macroblocks and motion vectors is skipped.

Keywords: motion estimation, support vector machines, video signal processing, MPEG videos

1. Introduction

Camera motion estimation plays crucial roles in many multimedia and computer vision applications. In Tan et al [1], camera motion is estimated from compressed MPEG videos and used for video annotation. In Skulimowski and Strumillo [2], camera motion parameters are employed to refine extracted depth/disparity metrics. In Ren et al [3], camera motion is estimated from MPEG videos for event based video indexing and retrieval. In Jiang et al [4], the extracted camera motion are applied in detecting combined video events such as closing up of players in sports videos. In Ren et al [5], camera motion is estimated as global motion by using phase correlation and then applied to compensate frame difference for the detection of film dirt in archive restoration applications.

Since compressed-domain processing can avoid time-consuming fully decoding of the video, camera motion estimation from compressed videos is preferred [1, 3, 4, 6, 7]. Among these approaches, the classic work in [1] needs particular attention as it has successfully directly motivated several other approaches [3, 4, 6].

In Tan et al [1], a six-parameter affine transformation is simplified to three parameters, a zooming factor f and a 2-D shifts (p_x, p_y). These parameters are estimated using the motion vectors extracted from macroblocks of *p*-frames in compressed MPEG videos. The reason here is that unlike *b*-frames which contain bi-directional motion vectors thus need more complex processing in partially decoding of the videos, *p*-frames have only forward motion vectors and can be easily parsed.

Due to unavailable motion vectors in intra-coded marcobocks and inaccurate motion vectors in texture-free areas, Tan's approach suffers a fundamental problem in choosing reliable motion vectors for camera motion estimation where it suggests abandoning

macroblocks with zero motion vectors for improved accuracy. In practice, noisy motion vectors can be non-zero hence this problem can be generalised as how to remove outliers of motion vectors for robustness [3]. In Nikitidis et al [7], a stochastic model is established from noisy motion vector fields for camera motion estimation with the assistance of heuristic rules.

To overcome the problem in selecting macroblocks and motion vectors, a novel approach is proposed to apply machine learning for motion estimation, where support vector regression is employed. To the best of our knowledge, this is the first attempt to apply support vector machines in this field, and the approach and promising results are presented in the next two sections.

2. The Approach

In many video compression standards, such as MPEG and H.26x, block-based motion estimation and compensation is widely applied. Accordingly, the motion vector and the motion compensated resident can be used to restore the original image blocks. Since motion estimation is of very high computational cost, how to make use of these extracted motion vectors from compressed video sequences becomes a research trend [10].

Typically, the 6-parameter projective camera model is used as defined below [1, 3],

$$x_i = \frac{p_1 x_{i-1} + p_2 y_{i-1} + p_3}{p_5 x_{i-1} + p_6 y_{i-1} + 1} \tag{1}$$

$$y_i = \frac{-p_2 x_{i-1} + p_1 y_{i-1} + p_4}{p_5 x_{i-1} + p_6 y_{i-1} + 1} \tag{2}$$

where p_1 is the zoom factor ($p_1 > 1$ represents zoom in and $p_1 < 1$ represents zoom out) and (x_i, y_i) and (x_{i-1}, y_{i-1}) are the image coordinates of corresponding points in two consequent frames f_i and f_{i-1}, respectively. Parameters p_3 and p_4 denote camera shift,

and p_5 and p_6 refer to perspective distortion effects. Finally, p_2 represents rotation about the axis of the camera lens.

In general, inter-frame camera motion is relatively small and contains minimal lens distortion effects [1, 18]. For simplicity, the distortion and the camera rotation are ignored, so that the model contains only shift and zooming. Accordingly, the model becomes by setting $p_2 = 0$, $p_5 = 0$ and $p_6 = 0$.

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} p_1 & 0 \\ 0 & p_1 \end{pmatrix} \begin{pmatrix} x_{i-1} \\ y_{i-1} \end{pmatrix} + \begin{pmatrix} p_3 \\ p_4 \end{pmatrix} \tag{3}$$

Actually, the simplified model has been successfully for content-based video annotation, indexing and retrieval [1, 3, 4, 6, 14, 18], where different videos such as sports, news, movies, surveillance and home generated videos are used. In most videos, rotation of the camera is rare, especially for surveillance, news and home generated videos. To this end, the simplification of the model is a reasonable practice in this context.

In Tan et al [1], corresponding pair of points are obtained automatically by checking the two macroblocks in f_i and f_{i-1}, where the two blocks are connected by the motion vectors extracted in P-frames of compressed MPEG video. Finally, p_1 is determined below, where N refers to the number of inter-coded macroblocks:

$$p_1 = \frac{\sum_{k=1}^{N} (w_{i(k)} - \overline{w}_i)^{\mathrm{T}} (w_{i-1(k)} - \overline{w}_{i-1})}{\sum_{k=1}^{N} \| w_{i-1(k)} - \overline{w}_{i-1} \|^2} \tag{4}$$

$$\begin{pmatrix} p_3 \\ p_4 \end{pmatrix} = \frac{\overline{w}_i}{p_1} - \overline{w}_{i-1} = \frac{1}{p_1} \begin{pmatrix} \bar{x}_i \\ \bar{y}_i \end{pmatrix} - \begin{pmatrix} \bar{x}_{i-1} \\ \bar{y}_{i-1} \end{pmatrix} \tag{5}$$

where $\overline{w}_j = N^{-1} \sum_{k=1}^{N} w_{j(k)} = (\bar{x}_j, \bar{y}_j)^{\mathrm{T}}$ and $w_{j(k)} = (x_{j(k)}, y_{j(k)})^{\mathrm{T}}$, with $j = i, i-1$.

Since the above solution suffers from false alarms caused by object motion and unreliable / non-exist motion vectors, selection of suitable motion vectors connected macroblocks to be used in (4) and (5) is required. For different videos, this seems quite arbitrary as different strategies need to be applied [11, 12].

The initial motivation to apply the SVM for camera motion estimation is to overcome the difficulty in selecting such suitable macroblock pairs. As a result, all macroblocks are used in training the SVM model. For those macroblocks without valid motion vectors, we simply assign the average motion vector over all available ones to them. To achieve this, firstly we extract motion vectors from all other macroblocks and calculate their mean as $(\bar{v}_x, \bar{v}_y)$. Then, $(\bar{v}_x, \bar{v}_y)$ is assigned as the motion vector for all the invalid motion vectors mentioned above.

For a given input vector $\mathbf{x}$, the output of the SVM is determined as follows [21],

$$f_{SVM}(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \phi(\mathbf{x}) + b \tag{6}$$

where the two parameters $\mathbf{w}$ and b respectively refer to a weight vector and a bias that can be determined in the training process through minimizing a given cost function, and $\phi(\cdot)$ is a linear or nonlinear mapping to map the input vector $\mathbf{x}$ into a higher dimensional space for easily separated by a linear hyperplane.

A training sample $(\mathbf{x}_i, y_i)$ is a support vector if it satisfies $y_i f_{SVM}(\mathbf{x}_i) \le 1$, where y_i is the designed output. If we denote $\mathbf{s}_k$ as extracted support vectors, $k \in [1, M]$, the SVM function can be re-written as

$$\begin{cases} f_{SVM}(\mathbf{x}) = \sum_{k=1}^{M} K(\mathbf{x}, \mathbf{s}_k) + b \\ K(\mathbf{x}, \mathbf{s}_k) = \phi^{\mathrm{T}}(\mathbf{x}) \phi(\mathbf{s}_k) \end{cases} \tag{7}$$

where $K(\cdot,\cdot)$ is a kernel function to represent the effect of the mapping $\phi(\cdot)$ in prediction, including both classification or regression.

Three commonly used kernel functions are summarized as follows, which include linear and two nonlinear functions. If the training samples are non-separable in linear cases, non-linear kernels like polynomial and Gaussian RBF functions are preferred. In addition, the associated parameters in the kernel functions, such as p and σ, can be determined in the training process.

$$K(\mathbf{x}_i,\mathbf{x}_j)=\begin{cases}\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j & linear\\ (\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j+1)^p & polynomial\\ e^{-\gamma\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2},\gamma=(2\sigma^2)^{-1} & RBF\end{cases} \tag{8}$$

With the finally extracted motion vectors, for each of the three parameters in (3), i.e. f, p_x, p_y, a regression model is learnt by using the SVM on a manually synthesised sequence generated by Maya with known camera parameters (as shown in Fig. 1). The generated video is converted to MPEG-1 format for training, where the macroblock size is 16*16 pixels. Motion vectors which were estimated using block-matching as defined in MPEG are then directly extracted from these macroblocks and used in learning the regression model.

Using 2-fold cross validation, the optimal parameters for SVM are determined via a grid search, as suggested by many other researchers [22, 23]. Usually, the SVM is trained with various combinations of parameter values, and the one which generates the best training results is selected as the best. For the generated ground truth, both polynomial kernel and Gaussian kernel are respectively employed for training. Not surprisingly, the Gaussian kernel produces more accurate training results as it is preferred in many applications [21-23]. However, the testing results using polynomial kernel for cross

validation seem much better, which indicates that Gaussian kernel causes severe over-fitting of the problem.

To avoid over-fitting, the polynomial kernel rather than the Gaussian kernel is adopted in our experiments, using the platform of libSVM [8] for implementation. The motion vectors are normalised into [-1,1] before they are inputted to SVM as features for prediction. For the training sequence, we then compare the mean squared error (MSE) of estimated camera motion vectors against known ground truth over all frames. The MSE for f, p_x, p_y is found as 0.032, 0.081 and 0.083, which shows a very high performance in training. Finally, the learnt regression models are applied on real sequences to valid the efficacy of the proposed approach.

3. Experimental Results

In our experiments, in total eight test sequences are used as summarised in Table 1. Four MPEG-1 sequences, Movie11, Movie3, River and Under the Sea of frame size 352*288, and four MPEG-2 video clips of frame size 720*576 and 720*480 are employed for performance validation. The MPEG-1 videos are from [9], which have all three camera motions, i.e. pan, tilt and zooming. Two MPEG-2 video clips are from youTube, which are for the football game between two teams, the Real Madrid and AC Milan, on Aug. 8, 2012, in which large object motions and fast camera movement are contained. In other two MPEG-2 sequences, one is from a surveillance camera in an airport and the other is cycling video. These test videos are selected to cover a wide range of video contents. This is not only reflected in the change of spatial resolutions, but also change of contents in various aspects. For example, we choose videos of single or multiple objects, indoor/outdoor scenes, home captured/professional videos, slow/fast movement as well as changes of illumination. Corresponding results are given as follows.

With the SVM models learnt from the synthesised sequence, we apply these models to estimate camera motions from the test sequences, where again motion vectors are extracted as input features of the SVMs. It is worth noting that the training is carried out on the simulated sequence, i.e. independent on any testing sequence. Consequently, the test results will inevitably prove the effectiveness of the proposed approach.

Since there is no ground truth information for these test sequences, motion compensated frame difference is calculated for performance evaluation. Let f, p_x, p_y be the estimated camera motion of two frames I_m and I_{m+1}, motion-compensated difference is given by

$$\begin{aligned} x' &= f \cdot x + p_x \\ y' &= f \cdot y + p_y \\ \Delta(I_m) &= \sum_{x,y} | I_{m+1}(x,y) - I_m(x',y') | \end{aligned} \tag{6}$$

Accordingly, the average frame difference over all frames, Ω, and the standard derivation Θ, are obtained below, where M is the number of frame pairs.

$$\begin{aligned} \Omega &= M^{-1} \sum_m \Delta(I_m) \\ \Theta &= (M-1)^{-1} \sqrt{\sum_m [\Delta(I_m) - \Omega]^2} \end{aligned} \tag{7}$$

For the eight test sequences, the average frame differences Ω and the standard derivations Θ are obtained and compared in Table 2, where the classic approach from Tan et al [1] and the more recent work in [6] are used for benchmarking. The running time of the two approaches are also shown in Table 2 for comparisons.

As can be seen, in general our approach yields the best performance among all three methods in terms of the sum of average frame differences Ω. In fact, our results are consistently better than Tan's approach, also it slightly outperforms Weng's approach. Regarding the standard derivation Θ, the results from the three approaches are quite

comparable, though our approach generates slightly less sum of standard derivation as summarized in Table 2. Detailed comparisons over four sequences are also illustrated in Fig. 2 and Fig. 3, respectively.

Regarding running time, the performance of the three approaches are quite comparable and all satisfy real-time requirements, though Weng's approach seems more efficient, followed by ours and Tan's approach. This is because the majority efforts in common are used to extract motion vectors from compressed videos, yet estimation of camera motion itself requires much less computational power. However, in at least two sequences when there are complex (object) motions, Movie3 and Airport, Weng's approach generates the worst results. This has demonstrated that Weng's approach fails to deal with such complex cases, although the running time is reduced.

It is worth noting that in all the three approaches, ours and those in [1] and [6], as a standard procedure, no object detection is required. As a result, object motion may affect the estimated results as the extracted motion vectors become unreliable or inaccurate, especially when the object is too large [19, 24]. In this case, the basic assumption that the camera motion will be the dominant one in the frame will become invalid. One possible solution is to remove the outliers of motion vectors that do not fit well with the global motion model and re-estimate the camera motion model in an iterative process [24]. However, if the object is too large and persists for a certain period, the accuracy of estimation will be still questionable.

To illustrate how object motion affects the accuracy of camera motion estimation, Fig. 4 gives the motion compensated residual images for the Movie11 sequence, using the results from our approach. If we compare the original frame image in Fig. 2 with the residual image in Fig. 4, we can clearly find that these residuals are mainly caused by object motion. Therefore, detection of objects for improved estimation of camera motion

can be a possible solution, although it needs fully decoding the video thus the overall efficiency may be degraded. On the other hand, compressed domain processing has provided a reasonable comprise in this context.

4. Conclusions

A novel machine learning based approach is presented for camera motion estimation from compressed MPEG videos, using support vector regression with extracted motion vectors as feature of input. When the support machine models is learnt from synthesised sequence, the test results on real sequences produce much improved performance in terms of motion compensated frame difference. Since selection of valid macroblocks and motion vectors is skipped, our proposed approach provides a more feasible solution in this context.

Acknowledgement

This work is partially supported by National Science Foundation of China under grants numbered 61202165 and 61003201.

References

1 Tan, Y., Saur, D. D., Kulkarni, S. R., and Ramadgeet, P. J.: 'Rapid estimation of camera motion from compressed video with application to video annotation'. *IEEE Trans. Circuits Syst. Video Technol. (T-CSVT)*, 2000, **10**, (1), pp. 133-146

2 Skulimowski, P. and Strumillo, P.: 'Refinement of depth from stereo camera ego-motion parameters'. *Electronics Letters*, 2008, **44**, (12), pp. 729-730.

3 Ren, J., Jiang, J., Chen, J., and Ipson, S. S.: 'Extracting objects and events from MPEG videos for highlight-based indexing and retrieval'. *Journal of Multimedia*, 2010, **5**, (2), pp. 95-103

4 Jiang, J., Kohler, J., Williams, C., Zaletelj, J., Guntner, G., Horstmann, H., Ren, J., Loffler, J., and Weng, Y.: 'LIVE: an integrated production and feedback system for intelligent and interactive TV broadcasting'. *IEEE Trans. Broadcasting*, 2011, **57**, (3), pp. 646-661

5 Ren, J. and Vlachos, T.: 'Detection of dirt impairments from archived film sequences: survey and evaluations'. *Optical Engineering*, 2010, **49**, 067005

6 Weng, Y. and Jiang, J: 'Fast camera motion estimation in MPEG compressed domain'. *IEEE Trans. Consumer Electronics*, vol. 57, no. 3, pp. 1329-1335, 2011

7 Nikitidis, S., Zafeiriou, S. and Pitas, I.: 'Camera motion estimation using a novel online vector field model in particle filters'. *IEEE T-CSVT*, 2008, **18**, (8), pp. 1028-1039

8 libSVM: *www.csie.ntu.edu.tw/~cjlin/libsvm/*

9 MUSCLE-VCD-2007: https://www.rocq.inria. fr/imedia/civr-bench/data.html

10 Zhang, H. J., Low, S. Y., and Smoliar, S. W.: 'Video parsing and browsing using compressed data,' *Multimedia Tools and Apps.*, 1(1), pp. 89-111, March 1995.

11 Dante, A. and Brookes, M.: 'Precise real-time outlier removal from motion vector fields for 3D reconstruction,' In: *Proc. ICIP*, I, pp. 393-396, 2003

12 Ewerth, R., Schwalb, M., Tessmann, P., and Freisleben, B.: 'Estimation of arbitrary camera motion in MPEG videos,' In: *Proc. ICIP*, I, pp. 512-515, 2004

13 Papadopoulos, G. Th., Briassouli, A., Mezaris, V., Kompatsiaris, I., and Strintzis, M. G.: 'Statistical motion information extraction and representation for semantic video

analysis,' *IEEE Trans. Circuits Syst. Video Techn*., vol. 19, no. 10, pp. 1513-1528, Oct. 2009

14 Abdollahian , G., Taskiran, C. M., Pizlo, Z., and Delp, E. J.: 'Camera Motion-Based Analysis of User Generated Video,' *IEEE Trans. Multimedia*, vol. 12, no. 1, pp. 28-41, Jan. 2010

15 Hu, W., Xie, N., Li, L., Zeng, X., and Maybank,, S.: 'A Survey on Visual Content-Based Video. Indexing and Retrieval', *IEEE Trans. Systems, Man and Cybernetics*, Part C, vol. 41, no. 6, Nov. 2011, pp. 797-819

16 Chen, Y., and Bajic, I. V., 'A joint approach to global motion estimation and motion segmentation from a coarsely sampled motion vector field,' *IEEE Trans. Circuits Syst. Video Techn*., vol. 21, no. 9, pp. 1316-, 1328, Sept. 2011

17 Khatoonabadi, S. H., and Bajic, I. V.: 'Video object tracking in the compressed domain using spatio-temporal Markov random fields,' *IEEE Trans. Image Processing*, vol. 22, no. 1, pp. 300-313, Jan. 2013

18 Kas, C., and Nicolas, H.: 'Compressed domain indexing of scalable H.264/SVC streams,' Signal Processing: Image Communication, vol. 24, no. 6, pp. 484-498, 2009 MPEG-2

19 Wang, J., Patel, N. V., Grosky, W. I., and Fotouhi, F.: 'Moving camera moving object segmentation in compressed video sequences,' *Int. J. Image Grap*. **09**, 609, 2009

20 Badu, R. V., and Ramakrishnan, K. R., 'Compressed domain video retrieval using object and global motion descriptors,' Multimedia Tools and Applications, vol. 32,no. 1, pp. 93-113, 2007

21 Ren, J.: 'ANN vs. SVM: Which one performs better in classification of MCCs in mammogram imaging,' *Knowledge Based System*, vol. 26, pp. 144-153, 2012

22 Hsu, C.-W., and Lin, C.-J., ‘A comparison of methods for multiclass support vector machines,’ *IEEE Trans. Neural Networks*, vol. 13, no. 2, pp. 415-425, 2002.

23 Zhuang, L., Dai,, H., ‘Parameter optimization of kernel-based one-class classfier on imbalance learning,’ *Journal of Computers*, vol. 1, no. 7, 2006

24 Wang, H., Divakaran, A., Vetro, A., Chang, S.-F.,and Sun, H.: ‘Survey of compressed-domain features used in audio-visual indexing and analysis,’ *J. Vis. Commun. Image R.*, vol. 14, pp. 150-183, 2003.

Biographies

Dr Zheng Wang graduated from School of Computer Science and Technology in 2002 and also obtained his Master and PhD degrees from the same school in Tianjin University, China in 2005 and 2008, respectively. He was a joint PhD student in INRIA, France from Sept. 2007 to Aug. 2008. From Sept. 2009, he is a Lecturer in the School of Computer Software, Tianjin University. His research interests focus on image and graphics, and he has published over 20 papers and held several patents in this field.

Dr Jinchang Ren received his undergraduate degree in Computer Software, MEng in Image Processing, DEng in Computer Vision, all from Northwestern Polytechnical University, Xi'an, China. He was also awarded a PhD in Electronic Media from Bradford University, U.K. Currently he is a Lecturer and Deputy Director of Hyperspectral Imaging Centre in the University of Strathclyde, Glasgow, U.K. His research interests focus mainly on intelligent visual computing and multimedia signal processing, especially on semantic content extraction for video analysis and understanding and hyperspectral imaging. He has published over 100 peer reviewed journal/conferences papers, and acts as an Associate Editor for two international journals including Multidimensional Systems and Signal Processing (Springer) and International Journal of Pattern Recognition and Artificial Intelligence.

Mr Yubin Wang is a Master student in the School of Computer Science and Technology, and his research interests are within image and video analysis.

Dr Meijun Sun obtained her Bachelor, Master and PhD degrees in 2002, 2005 and 2009, respectively all from School of Computer Science and Technology, Tianjin University, China. She is now an Associate Professor in the same School. Her research interests are mainly in using image and graphics techniques for Chinese paint simulation and analysis. She has published over 20 papers with several Chinese patents in this field.

Prof Jianmin Jiang received the B.Sc. degree from Shandong Mining Institute, Shandong, China, in 1982, the M.Sc. degree from China University of Mining and Technology, Xuzhou, China, in 1984, and the Ph.D. degree from the University of Nottingham, Nottingham, U.K., in 1994. He is currently a professor of digital media with the School of Computer Science and Technology, Tianjin University, China. He is also an adjunct professor with the University of Surrey, Surrey, U.K. He is an Evaluator for the Sixth and Seventh Framework Programmes (FP6/FP7) of the European Union. He has published around 400 refereed research papers. His research interests include image/video processing in the compressed domain, digital video coding, stereo image coding, medical imaging, computer graphics, machine learning, and artificial intelligence applications in digital media processing, retrieval, and analysis. Dr. Jiang is a Chartered Engineer, a Fellow of the Institution of Electrical Engineers and the Royal Society of Arts, Manufacture and Commerce, and a Member of the Engineering and Physical Sciences Research Council College.

Table 1. Sample frames of the 8 test sequences

			
Movie11	River	Movie3	Under the sea
Single object slow motion	*Fast camera motion, outdoor*	*Multi-object, home-video, indoor*	*Natural outdoor scene multi-object*
			
Airport	Cycling	Football-1	Football-2
Multi-object indoor	*Multi-object, fast outdoor motion*	*Multi-object, slow motion, sports*	*Multi-object, slow motion, sports*

Table 2: Performance comparison of our method and those in [1] and [6].

Sequences	Frame pairs and (size)	Ω (Θ)			Running time (s)		
		Tan [1]	Weng [6]	Ours	Tan [1]	Weng [6]	Ours
Movie11	88 (352x288)	14.00(2.163)	13.22(2.160)	12.15(2.161)	0.55	0.51	1.14
Movie3	176 (352x288)	25.93 (0.705)	26.04 (0.688)	20.38 (0.643)	1.41	1.38	1.42
River	205 (352x288)	15.30 (0.552)	15.09 (0.551)	14.87 (0.527)	1.52	1.29	1.54
Under the sea	383 (352x288)	5.64(0.971)	5.41(0.971)	5.36(0.975)	2.98	2.17	2.34
Airport	212 (720x576)	10.70(0.536)	11.66(0.540)	11.23(0.541)	8.95	4.26	4.33
Cycling	41 (720x480)	37.64 (1.307)	35.25 (1.329)	34.53 (1.287)	1.14	0.98	1.65
Football clip 1	201 (720x576)	10.77 (0.160)	10.23 (0.139)	9.39 (0.149)	1.69	1.50	1.57
Football clip 2	315 (720x576)	13.49 (0.165)	12.79 (0.141)	12.16 (0.167)	2.20	1.96	2.10
Sum	**1621**	**133.47 (6.559)**	**129.69 (6.519)**	**120.07 (6.450)**	**20.44**	**14.05**	**16.09**

List of Figure Captions:

Fig.1. Virtual Scene constructed by Maya.

Fig. 2. Comparisons of motion compensated frame difference of the Movie11 sequence using Tan's approach [1], Weng's approach [6] and ours.

Fig. 3. Comparisons of motion compensated frame difference of the Movie3 sequences using Tan's approach [1], Weng's approach [6] and ours.

Fig. 4. Comparisons on two football sequences of higher resolution and large object motions.

Fig. 5. Examples of residual images after motion compensation from Movie11 sequences

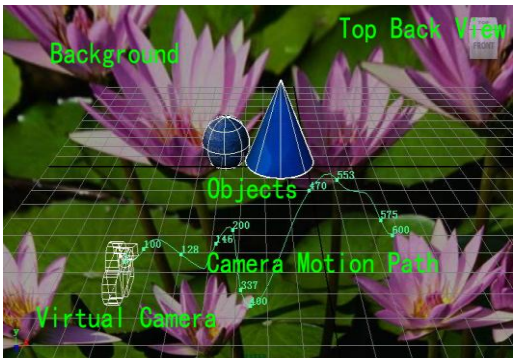


Fig.1. Virtual Scene constructed by Maya

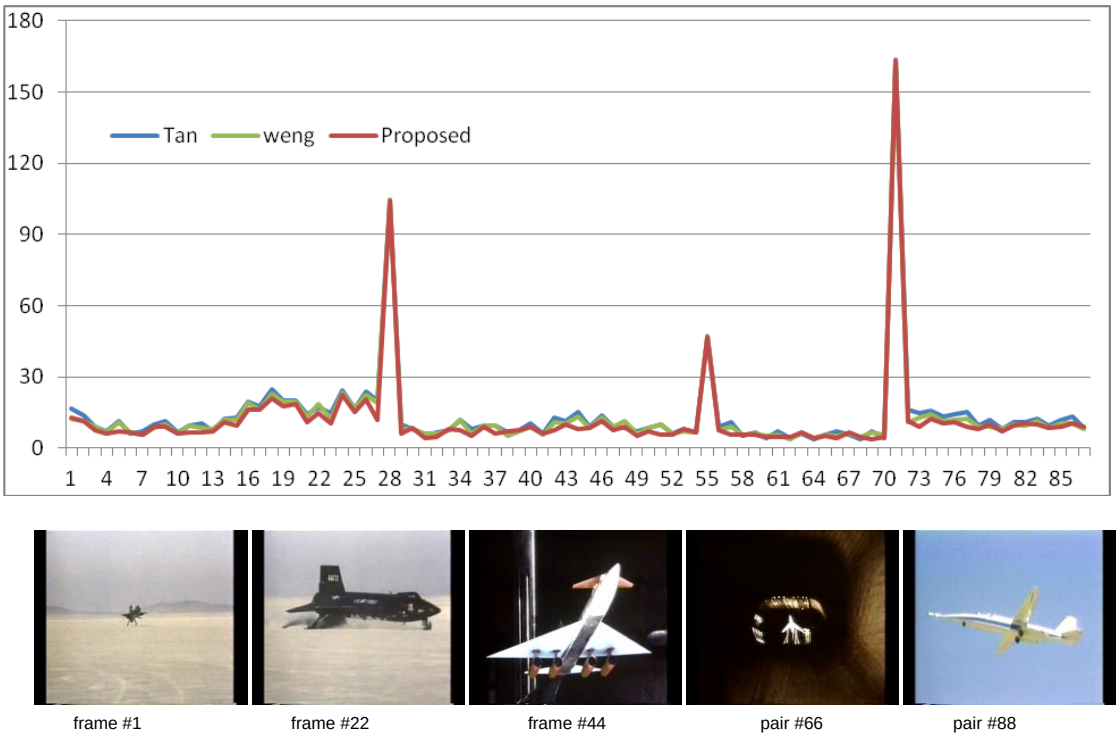


Fig. 2. Comparisons of motion compensated frame difference of the Movie11 sequence using Tan's approach [1], Weng's approach [6] and ours.

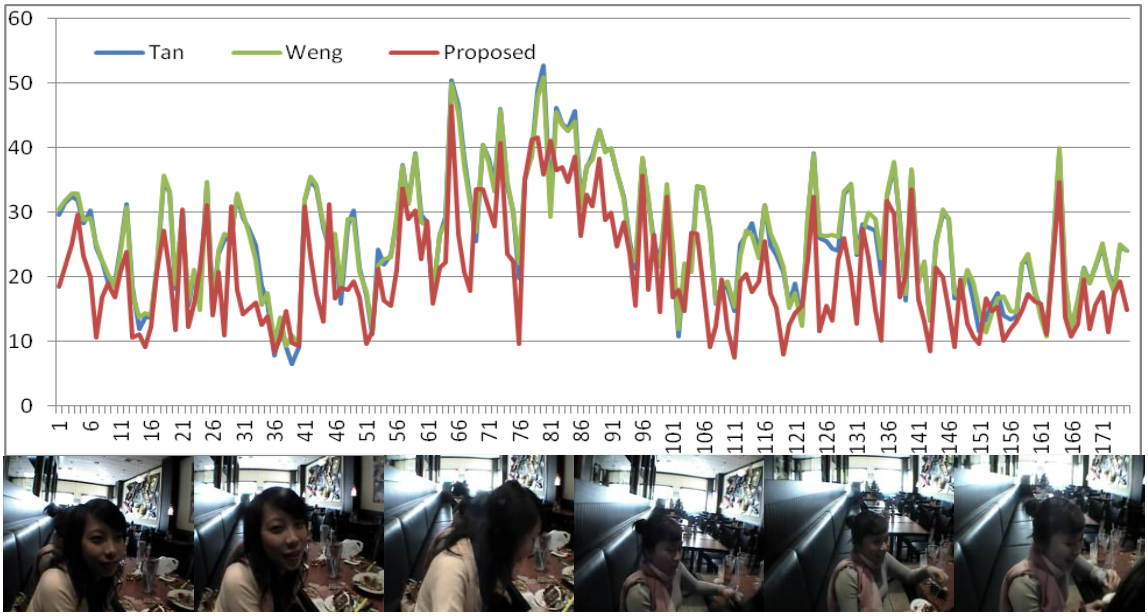


Fig. 3. Comparisons of motion compensated frame difference of the Movie3 sequence using Tan's approach [1], Weng's approach [6] and ours.

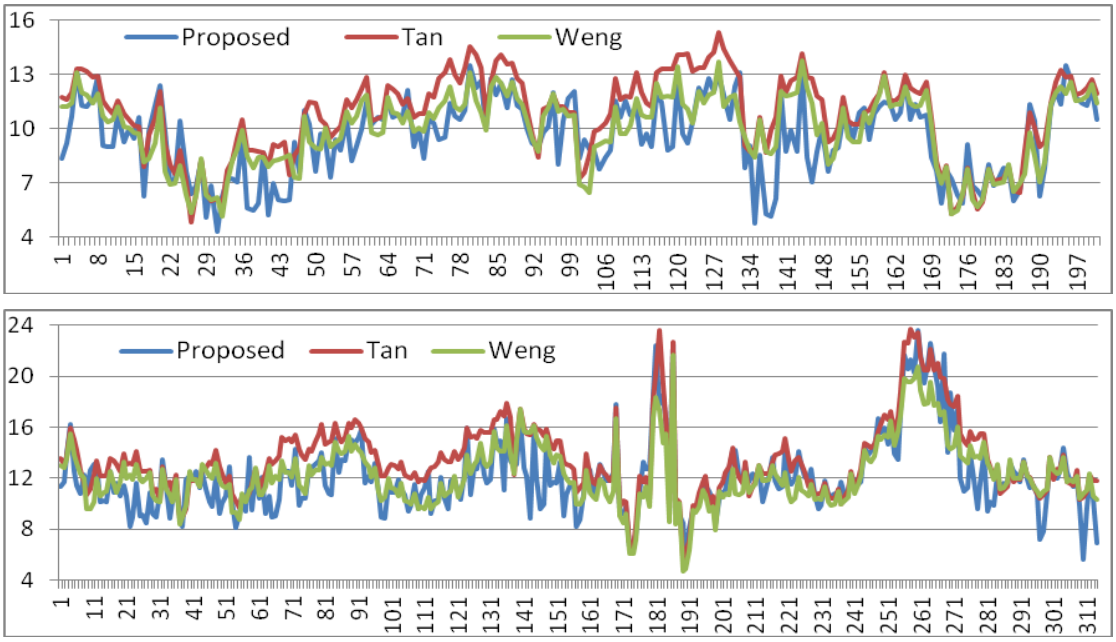


Fig. 4. Comparisons on two football sequences of higher resolution and large object motions.

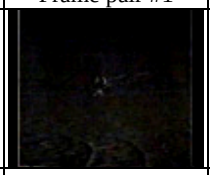
	Frame pair #1	Frame pair #22	Frame pair #44	Frame pair #66	Frame pair #88
Residual image					
e_{Ours}	12.7756	22.5303	11.2387	3.95588	8.81223

Fig. 5. Examples of residual image after motion compensation using our approach from the Movie11 sequence