

SYMMETRIC DISTRIBUTED CODING OF STEREO VIDEO SEQUENCES

*M. Tagliasacchi, G. Prandi, S. Tubaro **

Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy

ABSTRACT

In this paper we present a novel video coding scheme to compress stereo video sequences. We consider a wireless sensor network scenario, where the sensing nodes cannot communicate with each other and are characterized by limited computational complexity. The joint decoder exploits both the temporal and inter-view correlation to generate the side information. To this end, we propose a fusion algorithm that adaptively selects either the temporal or the inter-view side information on a pixel-by-pixel basis. In addition, the coding algorithm is symmetric with respect to the two cameras. We also propose a practical stopping criterion for turbo decoding that determines when decoding is successful. Experimental results on stereo video sequences show that a coding efficiency gain up to 4dB can be obtained by the proposed scheme at high bit-rates.

Index Terms— Video coding, motion analysis

1. INTRODUCTION

Traditional video coding is usually concerned with the compression of one source at a time, representing the scene taken from a single viewpoint. With the proliferation of cheap acquisition devices, it is easier to take simultaneously several looks at the same scene from different angles. In order to achieve a good coding efficiency, it is mandatory to take advantage of the correlation among the multiple views. Conventional multi-view coding (MVC) schemes assume that the encoder has simultaneous access to the uncompressed sequences captured by each camera. Therefore, by jointly encoding the different views, it is possible to accurately exploit the correlation between them.

In some application scenarios, communication between the cameras can be costly or even unfeasible, thus limiting the applicability of conventional MVC schemes. An interesting example is represented by cameras deployed as a sensor network, consisting of a large number of sensing devices that are low-power and with wireless communication capabilities. In order to reduce the amount of transmitted data, each node needs to locally compress its sequence, exploiting both spatial and temporal redundancies. When no communication is allowed between the encoders, the coding process cannot take advantage of inter-view redundancy.

Distributed source coding is a coding paradigm that enables to exploit the correlation among distributed sources at the decoder, by performing disjoint encoding but joint decoding. In the literature, there are several works in the area of multi-view image [1][2] and video coding [3][4][5][6] that take advantage of distributed video coding principles. The common goal is to generate at the decoder

a side information frame that optimally blends temporal and inter-view data. In [3] the temporal correlation within each image sequence is exploited locally at the encoder, while the inter-view redundancy is exploited at the central decoder. The paper proposes results for a two cameras setting, where only the second one is encoded using distributed source coding tools. References [4][5][6] propose similar schemes characterized by an asymmetric interleaved arrangement: even-indexed cameras are able to exploit spatio-temporal redundancy only, while odd-indexed cameras can also take advantage of inter-view redundancy. The proposed methods differ in the way the inter-view side information is generated, as well as in the fusion algorithm used to adaptively select either the temporal or the inter-view side information. In [5], the inter-view side information is obtained as a global homographic transformation between two adjacent side views. A simpler affine model is used in [4], while a locally adaptive block-based view interpolation is adopted in [6]. The fusion algorithms described in the literature try to optimally select the best source of side information for the current frame (temporal or inter-view), based on different criteria: [5] proposes one method based on monitoring the local motion activity and another one that uses previous and next decoded frames; [6] proposes a technique that projects reliability measures obtained for the intra-frame coded side cameras to the frame to be decoded.

In this paper we propose a new coding scheme based on distributed source coding principles for stereo sequences. The core of the architecture is based on the pixel domain Wyner-Ziv codec described in [7] and uses RCPT (Rate Compatible Punctured Turbo) codes to correct the side information into the decoded frame. Similarly to the aforementioned works, the proposed algorithm exploits the inter-view redundancy at the decoder only. Despite the previous works, it is fully symmetric since none of the two views needs to be chosen as the reference one. We also investigate the quality of the generated side information for different GOP sizes and target distortions. In addition, despite the recent literature on this topic (see, for example, [8], we do not assume that the original frames are available at the decoder to perform ideal error detection. In fact, we use simple statistics based on the log-likelihood ratios computed by the turbo decoder to determine when decoding is successful. This paper exploits some of the ideas originally presented by the same authors in [9]. Nevertheless, the exploitation of inter-view dependency is novel, as well as the study of the behavior of the fusion algorithm as a function of the GOP size and target distortion.

2. PIXEL DOMAIN WYNER-ZIV ARCHITECTURE

The video coding scheme that we present in this paper is based on the pixel domain Wyner-Ziv codec described in [7]. This coding architecture offers a pixel domain intra-frame encoder and inter-frame decoder with very low computational encoder complexity. The video sequence is partitioned in group of pictures (GOP) of size N . Let us denote with t the generic time instant within a GOP and with

*This work has been partially sponsored by MIUR (Italian Ministry of Education and Research) under the project PRIN - "Robust video coding techniques based on distributed source coding" and EU under Visnet II Network of Excellence

$X(t)$ the original frame at time t . The first frame of each GOP is intra-frame coded, and it is denoted as key frame. The frames $X(t)$, $t \in [1, N-1]$ are called Wyner-Ziv frames. Each pixel in the Wyner-Ziv frame is uniformly quantized. Bit-plane extraction is performed from the entire image and then each bit-plane is fed into a turbo encoder to generate a sequence of parity bits. At the decoder, the two key frames $\hat{X}(0)$ and $\hat{X}(N)$ ¹ are used by the motion-compensated frame interpolation module [10] to generate the side information $Y^T(t)$, which will be used by the turbo decoder and reconstruction modules. The decoder operates in a bit-plane by bit-plane basis and starts by decoding the most significant bit-plane. It only proceeds to the next bit-plane after each bit-plane is successfully turbo decoded.

3. PROPOSED ALGORITHM

Let us consider a scenario with two video sequences X_1 and X_2 representing the same scene taken from different viewpoints. The coding architecture described in Section 2 could be applied independently to each of the two views achieving very low encoding complexity. Nevertheless, in this case we would not exploit the inter-view redundancy.

Figure 1 illustrates the block diagram of the proposed algorithm. Since the algorithm is fully symmetric with respect to the two cameras, we only show the processing of the first view. The key frames $X_1(0)$, $X_1(N)$ are intra-frame coded using H.264/AVC and transmitted to the decoder. At the encoder, we split the Wyner-Ziv frames $X_1(t)$ into two subsets $A_1(t)$ and $B_1(t)$ based on a checkerboard pattern. The same operation is symmetrically performed on the corresponding frame of the second view $X_2(t)$, but inverting the role of subset A and B . At the decoder, subset $A_1(t)$ is decoded first, using the side information obtained by motion interpolation Y_1^T , thus exploiting temporal correlation. The decoded subset $\hat{A}_1(t)$ is used in two ways: 1) to improve the temporal side information to get $Y_1^T(t)$; 2) to perform view prediction and obtain $Y_1^V(t)$. A fusion algorithm is then used to generate the mixed temporal/inter-view side information $Y_1^{\tau V}(t)$ used to decode subset $B_1(t)$.

The proposed algorithm can be detailed as follows.

1. *Frame splitting*: Let (x, y) be the coordinate values of a pixel. For the first view $X_1(t)$:

$$\begin{aligned} &\text{If } [x \bmod 2] \text{ xor } [(y + 1) \bmod 2], \\ &\quad \text{then } (x, y) \in A, \\ &\quad \text{else } (x, y) \in B. \end{aligned} \quad (1)$$

Let us denote with $A_1(t)$ ($B_1(t)$) the pixel values assumed by $X_1(t)$ in the pixel locations belonging to the set A (B). The dual assignment, with sets A and B swapped, is performed for the second view $X_2(t)$.

2. *WZ encoding*: The encoder processes the two subsets $A_1(t)$ and $B_1(t)$ independently, generating parity bits for both of them.
3. *Motion interpolation*: The side information $Y_1^T(t)$ is obtained by motion interpolating $\hat{X}_1(0)$ and $\hat{X}_1(N)$ according to the algorithm described in [10].
4. *WZ decoding subset $A_1(t)$* , as described in Section 2, using $Y_1^T(t)$ as side information.
5. *Spatial interpolation*: The frame $\hat{X}_1(t)$ is spatially interpolated in the pixel locations corresponding to set B , using the

decoded set $\hat{A}_1(t)$. The output is the interpolated frame $Y_1^S(t)$. The simple non-linear, adaptive algorithm proposed in [9] is used for this purpose.

6. *Motion estimation*: The temporal side information is refined to obtain $Y_1^T(t)$. Bi-directional block based motion estimation is performed setting $Y_1^S(t)$ as the current frame and $\hat{X}_1(0)$ and $\hat{X}_1(N)$ as the reference frames.
7. *View prediction*: Frame $X_1(t)$ is predicted based on frame $X_2(t)$ to obtain $Y_1^V(t)$. To this end, block based motion estimation is performed setting $Y_1^S(t)$ as the current frame and $Y_2^S(t)$ as the reference frame. The geometric constraints of the camera setting are exploited by limiting the motion search along the epipolar lines, thus reducing the computational complexity of the motion search and the accuracy of the prediction.
8. *Fusion*: The improved temporal side information $Y_1^T(t)$ is combined with the view side information $Y_1^V(t)$ on a pixel-by-pixel basis, in order to find the best side information. In an ideal (but unrealistic) setting, if the original frame were available, the optimal side information should select either $Y_1^T(t)$ or $Y_1^V(t)$, depending on which is closest to $X_1(t)$, i.e.

$$Y_1^{\tau V^*}(x, y, t) = \arg \min_{j=\tau, V} |X_1(x, y, t) - Y_1^j(x, y, t)|^2 \quad (2)$$

In practice, the decoder does not have access to $X_1(t)$. Therefore we need to infer $j = \tau, V$ for each (x, y) in the set B from the available information, i.e. the already decoded set $\hat{A}_1(t)$. For each pixel (x, y) in B , we denote its 4-adjacent neighbors as \mathcal{N}_{xy} . The fusion algorithm proceeds as follows:

- (a) Compute the difference between $Y_1^T(t)$ and $\hat{A}_1(t)$ to estimate of the temporal correlation noise.

$$e^T(x, y, t) = \sum_{(m, n) \in \mathcal{N}_{xy}} |Y_1^T(m, n, t) - \hat{A}_1(m, n, t)|^2$$

- (b) Compute the difference between $Y_1^V(t)$ and $\hat{A}_1(t)$ to estimate of the inter-view correlation noise.

$$e^V(x, y, t) = \sum_{(m, n) \in \mathcal{N}_{xy}} |Y_1^V(m, n, t) - \hat{A}_1(m, n, t)|^2$$

- (c) Generate the side information $Y_1^{\tau V}(x, y, t)$:

$$\begin{aligned} &\text{If } e^T(x, y, t) < \alpha e^V(x, y, t) + \beta, \\ &\quad Y_1^{\tau V}(x, y, t) = Y_1^T(x, y, t), \\ &\text{else } Y_1^{\tau V}(x, y, t) = Y_1^V(x, y, t) \end{aligned} \quad (3)$$

where α and β are properly defined constants discussed below. The idea is that both temporal correlation and inter-view correlation are not spatially stationary, but their statistics slowly vary across space. Therefore we can use the observed temporal (inter-view) correlation for neighboring, already decoded, pixels as an estimate of the actual temporal (inter-view) correlation for the current pixel. When $e^T(x, y, t) < \alpha e^V(x, y, t) + \beta$, the temporal side information is selected. This happens when no temporal occlusions and/or illumination changes occur. It also happens when the view prediction is of poor quality because of occlusions due to the scene geometry. When $e^T(x, y, t) \geq \alpha e^V(x, y, t) + \beta$, the inter-view

¹We denote with \hat{X} the decoded version of frame X .

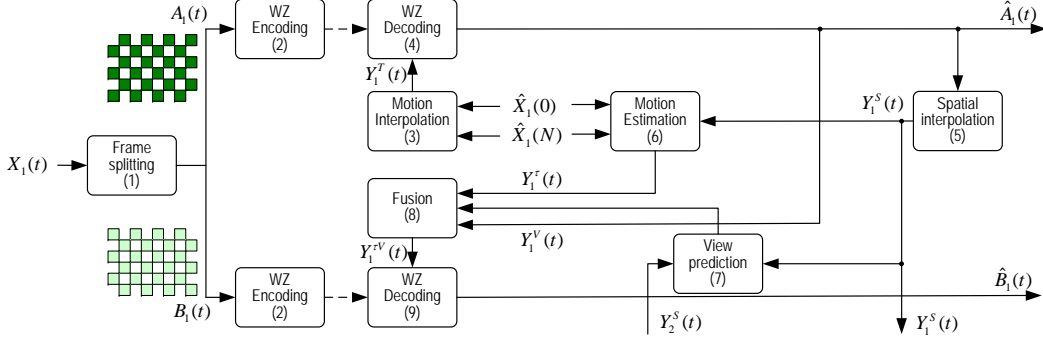


Fig. 1. Block diagram of the proposed algorithm.

side information is selected. This case occurs when the motion is complex and the temporal prediction is poor.

In order to obtain the optimal value of α and β , we trained a linear classifier in the $[e^T, e^V]$ feature space for each frame of the sequences *Breakdancers*, *Exit*. The optimal value of β is always nearly equal to 0. Conversely, α assumes values in the range $[0, 6]$, with a strong peak in $\alpha = 1$. Therefore, we set this value for all the simulations reported in this paper.

We notice that the proposed fusion algorithm differs from those presented in [6][5]. In [6][5], the decision is taken based on the decoded key frames $\hat{X}_1(0)$ and $\hat{X}_1(N)$ or the intra-coded side views at the same time instant t . Thus, there is the problem of propagating the fusion decision to the actual frame to be decoded. In the proposed algorithm, part of the frame $\hat{A}_1(t)$ is already available. Therefore, the decision is more robust since it relies on data of the same frame, without requiring any propagation.

9. WZ decoding subset $B_1(t)$, as described in Section 2, using $Y_1^{\tau V}(t)$ as side information.

4. ERROR DETECTION IN TURBO DECODING

In the architecture described in Section 2, for each bit-plane, the Wyner-Ziv decoder sends a request for parity bits to the encoder via a feedback channel. The request loop is iterated until the bit error rate falls below a pre-defined threshold (i.e. 10^{-3}). In the recent literature [8], ideal error detection is assumed. This implies an unrealistic scenario where the decoder has a copy of the original frames.

The problem of error detection in turbo decoding has already been addressed in the communications literature. The work in [11] proposes a simple method for early stopping, in order to reduce the number of turbo decoder iterations, thus limiting the computational complexity. The method is based on monitoring the log a posteriori probability (LAPP) ratio, which is defined as:

$$L(u_i) = \log \frac{\Pr(u_i = 1|\mathbf{y})}{\Pr(u_i = 0|\mathbf{y})}, \quad (4)$$

where u_i is the i th bit to be decoded and \mathbf{y} is the received codeword of length n (k systematic bits and $n - k$ parity bits). In [11] it is proposed to early stop turbo decoding when $E[|L|] = 1/n \sum_{i=0}^{n-1} |L(u_i)| > T$, where T is a threshold that depends on n and on the channel statistics.

When turbo decoding is applied in the context of Wyner-Ziv coding of video, the main concern is to detect when no more parity bits need to be requested via the feedback channel. Based on

the results in [11], we propose the following stopping criterion. Let r denote the request index ($r \in [0, R]$), where $R = k/P$ and P is the puncturing period. For each request r , we compute L_r after the last I th iteration of the turbo decoder. Since in our context the parity bits are assumed to be error free, L_r is obtained as the expectation of the LAPP ratio over the systematic bits only, i.e., $L_r = 1/k \sum_{i=0}^{k-1} |L_r(u_i)|$. Decoding is declared successful if

$$\frac{L_r - L_{r-1}}{L_{r-1}} > K \quad (5)$$

where K is a properly defined constant. In fact, it can be observed that the value of L_r remains almost constant until the minimum number of parity bit requests needed for correct decoding is made. At this point L_r rapidly grows at a much higher value. As observed in [11], this criterion might suggest correct decoding also when few residual errors are still present. Nevertheless, the residual bit error rate is typically below 10^{-3} , and it is therefore adequate for the considered application.

5. EXPERIMENTAL RESULTS

We carried out several experiments in order to test the validity of the proposed algorithm. First, we tested the quality of the side information. The performance of the turbo decoding process heavily depends on the quality of the side information. Intuitively, a higher number of parity bits will be requested by the decoder when the correlation is weak, as more errors need to be corrected.

In this paper we provide results for the *Breakdancers* (BR, 100 frames) and *Exit* (EX, 250 frames) sequences, considering only two central views. The target distortion is determined by the quantization parameter $QB \in [1, 4]$, which indicates the number of bitplanes to be decoded. The temporal side information Y^T is obtained by motion-compensated interpolation of the lossy key frames. The key frames are intra-coded using H.264/AVC, setting a *QPISlice* parameter that depends on the target QB (the values 36, 35, 34, 33 are used in our experiments.)

Table 1 indicates the quality of the side information measured in terms of PSNR (dB), averaged over the two views. The PSNR is computed only for pixels belonging to subsets $B_{1,2}$, since for subsets $A_{1,2}$ the temporal side information Y^T is the only available. In order to study the behavior with different GOP sizes, we provide results for GOPs of length 2 and 8.

A number of interesting conclusions can be drawn. The refined temporal side information Y^τ is always better than the original side information Y^T ($\Delta^{\tau-T} = Y^\tau - Y^T > 0$). This is especially true

BR	GOP 2								GOP 8							
	QB	Y^T	Y^τ	Y^V	$Y^{\tau V}$	$Y^{\tau V^*}$	$\Delta^{\tau-T}$	$\Delta^{\tau V-\tau}$	Y^T	Y^τ	Y^V	$Y^{\tau V}$	$Y^{\tau V^*}$	$\Delta^{\tau-T}$	$\Delta^{\tau V-\tau}$	
QB	1	25,88	26,59	25,42	27,49	29,11	0,71	0,90	23,53	24,21	24,21	25,09	26,42	0,68	0,88	
	2	25,90	27,95	26,46	29,53	31,18	2,05	1,58	23,56	25,93	25,70	27,59	29,11	2,37	1,66	
	3	25,96	28,90	27,12	31,43	33,05	2,94	2,53	23,53	27,16	26,71	29,83	31,40	3,63	2,67	
	4	26,11	29,65	27,62	33,08	34,62	3,54	3,43	23,86	28,18	27,52	31,96	33,42	4,32	3,78	
EX	GOP 2								GOP 8							
	QB	Y^T	Y^τ	Y^V	$Y^{\tau V}$	$Y^{\tau V^*}$	$\Delta^{\tau-T}$	$\Delta^{\tau V-\tau}$	Y^T	Y^τ	Y^V	$Y^{\tau V}$	$Y^{\tau V^*}$	$\Delta^{\tau-T}$	$\Delta^{\tau V-\tau}$	
	1	32,70	34,11	26,78	33,10	36,08	1,41	-1,01	29,21	30,10	26,27	29,81	32,26	0,89	-0,29	
	2	33,27	35,00	26,91	34,07	37,03	1,73	-0,93	29,42	31,52	26,64	31,33	33,88	2,10	-0,19	
3	33,85	35,85	27,03	35,06	37,84	2,00	-0,79	29,65	33,05	26,96	33,05	35,49	3,40	0,00		
4	34,40	36,69	27,14	36,19	38,79	2,29	-0,50	29,83	34,10	27,16	34,49	36,75	4,27	0,39		

Table 1. Distortion (in dB) of the side information generated at the decoder. Temporal Y^T , refined temporal Y^τ , inter-view Y^V , fused $Y^{\tau V}$.

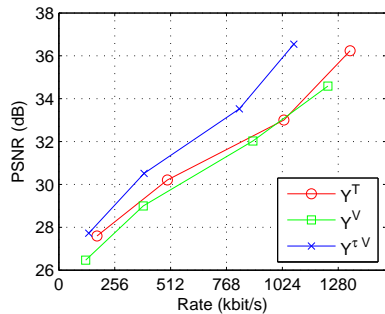


Fig. 2. Rate/Distortion graph for *Breakdancers* sequence, GOP = 2

for low distortion levels (i.e. $QB = 4$). In fact, in this case, the high frequency content of the image is retained, and the additional accuracy of the refined motion field is fully exploited. Keeping the QB values fixed, $\Delta^{\tau-T}$ increases for longer GOP sizes. This is due to the fact that motion-compensated interpolation is not accurate when key frames are spaced apart in time. For both sequences, the gain obtained by motion refinement can be as large as $+4dB$.

The quality of the inter-view side information Y^V is strongly dependent on the cameras baseline. In the *Breakdancers* sequence the two views are highly correlated, and Y^V improves over Y^T (but, by itself, it is worse than Y^τ). In the *Exit* sequence the baseline is wider, and the quality of Y^V obtained with our simple block-matching algorithm is rather poor.

In Section 3, we described an algorithm that allows to fuse the temporal (Y^τ) and inter-view (Y^V) side information to obtain $Y^{\tau V}$. For the *Breakdancers* sequence, the gain $\Delta^{\tau V-\tau} = Y^{\tau V} - Y^\tau$ increases with QB , being in the range $[+0.9dB, +3.8dB]$. By increasing the GOP size, $\Delta^{\tau V-\tau}$ becomes slightly larger $[0dB, +0.3dB]$, since the temporal side information quality decreases. For the *Exit* sequence, we already observed that the inter-view side information is of poor quality. Therefore, the proposed fusion algorithm, is unable to outperform the refined temporal side information Y^τ , apart for the case of long GOP sizes and low distortion levels. Besides the inter-view redundancy is not exploited in this case, the side information generated by the proposed algorithm is largely better than Y^T , thanks to the temporal side information refinement.

Figure 2 shows the rate-distortion curves obtained with the proposed coding scheme, using the side information is Y^T , Y^V , and $Y^{\tau V}$ respectively. By combining the enhanced temporal and inter-view side information, we can achieve a coding gain between $0.5dB$

and $4dB$, with larger gains at high bit-rates. The error detection algorithm presented in Section 4 is used to determine the number of requests needed for successful turbo decoding.

6. CONCLUSIONS

In this paper we propose a coding scheme for stereo video sequences characterized by low encoding complexity and lack of communication between the cameras at the encoder. We describe an algorithm that improves the side information in two ways: refining the temporal side information and exploiting the inter-view redundancy. We are currently improving the generation of the inter-view side information and studying more sophisticated fusion algorithms by using non-linear classifiers.

7. REFERENCES

- [1] Anne Aaron, P. Ramanathan, and Bernd Girod, "Wyner-Ziv coding of light fields for random access," in *Proceedings of the IEEE Workshop on Multimedia Signal Processing*, Siena, Italy, September 2004.
- [2] Nicolas Gehrig and Pier Luigi Dragotti, "On distributed compression in dense camera sensor networks," in *Picture Coding Symposium*, San Francisco, CA, December 2004.
- [3] Markus Flierl and Bernd Girod, "Coding of multi-view image sequences with video sensors," in *Proceedings of the International Conference on Image Processing*, Atlanta, GA, October 2006.
- [4] Xun Guo, Yan Lu, Feng Wu, Wen Gao, and Shipeng Li, "Distributed multi-view video coding," in *Proceedings of SPIE-IS&T Electronic Imaging*, vol. 6077.
- [5] Mourad Ouaret, Frederic Dufaux, and Touradj Ebrahimi, "Fusion-based multi-view distributed video coding," in *ACM International Workshop on Video Surveillance and Sensor Networks*, Santa Barbara, CA, October 2006.
- [6] Xavi Artigas, Egon Angeli, and Luis Torres, "Side information generation for multiview distributed video coding using a fusion approach," in *7th Nordic Signal Processing Symposium*, Reykjavik, Iceland, June 2006.
- [7] Bernd Girod, Anne Aaron, Shantanu Rane, and David Rebollo Monedero, "Distributed video coding," *Proceedings of the IEEE*, vol. 93, pp. 71–83, January 2005.
- [8] Catarina Brites, Joao Ascenso, and Fernando Pereira, "Feedback channel in pixel domain wyner-ziv video coding: Myths and realities," in *European Signal Processing Conference*, Florence, Italy, September 2006.
- [9] Marco Tagliasacchi, Alan Trapanese, Stefano Tubaro, Joao Ascenso, Catarina Brites, and Fernando Pereira, "Exploiting spatial redundancy in pixel domain Wyner-Ziv video coding," in *Proceedings of the International Conference on Image Processing*, Atlanta, GA, October 2006, Submitted.
- [10] Joao Ascenso, Catarina Brites, and Fernando Pereira, "Interpolation with spatial motion smoothing for pixel domain distributed video coding," in *EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services*, Slovak Republic, July 2005.
- [11] Fengqin Zhai and I.J. Fair, "Techniques for early stopping and error detection in turbo decoding," *IEEE Transactions on Communications*, vol. 51, pp. 1617–1623, October 2003.