and median frequencies are below 10% and are comparable to the ones in [11]; however, the moments are not well preserved and present large variation. A reason for this behavior is that the MMP algorithm works on the time domain, and uses the mean-squared error as the distortion metric; this favors the maintenance of the shape (as can be seen from the good PRD results), but not the spectral features. One possible solution for it, that is a promising area for future work, is to add other distortion metrics to 1, taking into account spectral features.

Another promising research direction would be to use the proposed algorithm to compress isotonic signals. Since it uses an adaptive dictionary, it has the potential to capture and quickly adapt to the nonstationary characteristics presented by such signals.

The computational complexity of the MMP algorithm is reasonably high and a simplified analysis concerning this matter is available in [9]. Besides, we have not developed fast algorithms for its implementation yet, since the main goal currently is to identify its potentials and investigate enhancements for improving its performance for a variety of signals, like the EMG. Given these observations, we did not carried out complexity comparisons in the context of this paper, however, this is a main concern for future works.

V. Conclusion

We applied a one-dimensional version of the MMP algorithm to the compression of EMG data. The base algorithm and its extensions, composed of tools that allow better adaptation to smooth sources and more effective use of the dictionary, provided reconstructed signals with high quality. The results obtained for isometric signals were good, outperforming state-of-the-art schemes for EMG compression in terms of PRD $\times$ compression ratio. In brief, this paper enforces that the MMP algorithm, due to its universality, is an interesting alternative for biological signal encoding, and can be a viable alternative to compressing other data of the same class, like the EEG.

Acknowledgment

The authors would like to thank Prof. P. Berger, Universidade de Brasília, for providing the test signals used in this paper.

References

[1] A. P. Guerrero and C. Mailhes, "On the choice of an electromyogram data compression method," presented at the 19th Int. Conf.—IEEE/EMBS, Chicago, IL, Nov. 1997.

[2] E. Plevin and D. Zazula, "Decomposition of surface EMG signals using non-linear LMS optimisation of higher-order cumulants," in *Proc. 15th IEEE Symp. Comput.-Based Med. Syst.*, Jun. 2002, p. 49.

[3] D. Farina, R. Merletti, and R. M. Enoka, "The extraction of neural strategies from the surface EMG," *J. Appl. Physiol.*, vol. 96, no. 4, pp. 1486–1495, Apr. 2004.

[4] M. Nikolic, "Detailed analysis of clinical electromyography signals emg decomposition: Findings and firing pattern analysis in controls and patients with myopathy and amytrophic lateral sclerosis," Ph.D. dissertation, Faculty Health Sci., Univ. Copenhagen, Denmark, 2001.

[5] T. Hoshino, M. Tomono, R. Furusawa, T. Suzuki, M. Shimojo, and K. Mabuchi, "Development of a motion support system by using an electromyogram," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, The Hague, The Netherlands, Oct. 2004, vol. 5, pp. 4432–4437.

[6] P. A. Berger, F. A. O. Nascimento, J. C. Carmo, and A. F. Rocha, "Compression of EMG signals with wavelet transform and artificial neural networks," *Inst. Phys. Publishing: Physiol. Meas.*, vol. 27, no. 6, pp. 457–465, Jun. 2006.

[7] M. B. de Carvalho, E. A. B. da Silva, and W. A. Finamore, "Multidimensional signal compression using multiscale recurrent patterns," *Signal Process.: Image Video Coding Beyond Stand.*, vol. 82, no. 11, pp. 1559–1580, Nov. 2002.

[8] K. Nazarpour, A. R. Sharafat, and S. M. Firoozabadi, "Negentropy analysis of surface electromyogram signal," in *Proc. 13th IEEE Workshop Statist. Signal Process.*, Bordeaux, France, Jul. 2005, pp. 974–977.

[9] E. B. L. Filho, E. A. B. da Silva, M. B. Carvalho, W. S. S. Júnior, and J. Koiller, "Electrocardiographic signal compression using multiscale recurrent patterns," *IEEE Trans. Circuits Syst.—I: Reg. Papers*, vol. 52, no. 12, pp. 2739–2753, Dec. 2005.

[10] J. A. Norris, K. Englehart, and D. Lovely, "Steady-state and dynamic myoelectric signal compression using embedded zero-tree wavelets," in *Proc. 23rd Annu. EMBS Int. Conf.*, Oct. 2001, pp. 1879–1882.

[11] E. Carotti, J. C. De Martin, R. Merletti, and D. Farina, "Compression of surface EMG signals with algebraic code excited linear prediction," *Med. Eng. Phys.*, vol. 29, no. 2, pp. 253–258, May 2006.

ECG Signal Compression Based on Dc Equalization and Complexity Sorting

Eddie B. L. Filho, Nuno M. M. Rodrigues*, Member, IEEE*,
Eduardo A. B. da Silva**, Member, IEEE*,
Sérgio M. M. de Faria*, Member, IEEE*, Vitor M. M. da Silva,
and Murilo B. de Carvalho

***Abstract*—In this brief, we present new preprocessing techniques for electrocardiogram signals, namely, dc equalization and complexity sorting, which when applied can improve current 2-D compression algorithms. The experimental results with signals from the Massachusetts Institute of Technology - Beth Israel Hospital (MIT-BIH) database outperform the ones from many state-of-the-art schemes described in the literature.**

***Index Terms*—Data compression, electrocardiogram (ECG), H.264/AVC, JPEG2000, preprocessing.**

I. Introduction

The electrocardiogram (ECG) is a very important tool for medical diagnosis. The need for remote diagnosis may demand for the transmission of the complete exam, which may be composed by as much as 12 derivations, over bandwidth-restricted communication networks. Furthermore, the need for databases of ECG exams from various patients, aiming at pathology development analysis or comparative diagnosis, makes efficient storage also an important issue. Therefore, both ECG storage and transmission demand suitable compression schemes, which

Manuscript received June 13, 2005; revised August 19, 2007. *Asterisk indicates corresponding author.*

E. B. L. Filho is with the Centro de Ciência, Tecnologia e Inovação do Pólo Industrial de Manaus, Manaus-AM 69057-040, Brazil (e-mail: eddie@ctpim.org.br).

N. M. M. Rodrigues and S. M. M. de Faria are with the Instituto de Telecomunicações and ESTG. Inst. Politécnico Leiria, Morro do Lena - Alto do Vieiro, 2411-901 Leiria, Portugal (e-mail: nuno.rodrigues@co.it.pt; sergio.faria@co.it.pt).

*E. A. B. da Silva is with the Universidade Federal do Rio de Janeiro, Rio de Janeiro-RJ 21941-972, Brazil (e-mail: eduardo@lps.ufrj.br).

V. M. M. da Silva is with the Instituto de Telecomunicações and Dep. Eng. Electrotécnica e de Computadores, Pólo II - Universidade de Coimbra, 3030-290 Coimbra, Portugal (e-mail: vitor.silva@co.it.pt).

M. B. de Carvalho is with TEC/CTC, Universidade Federal Fluminense, Niterói, RJ 24210-240, Brazil (e-mail: murilo@telecom.uff.br).

Digital Object Identifier 10.1109/TBME.2008.919880

must keep the integrity of the signal envelope as much as possible; otherwise, the medical team will perceive an erroneous or incomplete structure, compromising the diagnosis.

Many authors have tackled the problem of ECG compression by treating it as an image [1]–[5], allowing the encoder to efficiently exploit the inter- and intrabeat dependencies. These methods can be generally divided into three steps: QRS detection for segmenting the ECG into periods, preprocessing, and transformation. The preprocessing step is the main focus of this paper. Several authors have successfully used period normalization [1], [3], [4], which tends to lead to significant improvements. However, there are factors that prevent the normalized periods from having high correlations, which may compromise the performance of the compressor.

This brief proposes two new preprocessing methods that exploit signal structure and similarities more efficiently by increasing the correlation between adjacent rows in the ECG image: dc equalization, which clamps the signal segments to the same dc level, and complexity sorting, which rearranges the rows of the resulting 2-D array from the simplest to the more complex ones.

Some ECG compression methods use JPEG2000 as the encoder for the 2-D array of samples, with generally good results. In this work, we also investigate the use of the recently developed H.264/AVC intraframe encoder for this purpose.

This brief is organized as follows. In Section II, we discuss the main preprocessing techniques used throughout the literature, together with the ones proposed in this work. Section III provides simulation results for the chosen encoders. In Section IV, the proposed algorithm is discussed, and we present our conclusions in Section V.

II. Preprocessing Techniques for ECG Compression Using 2-D Encoders

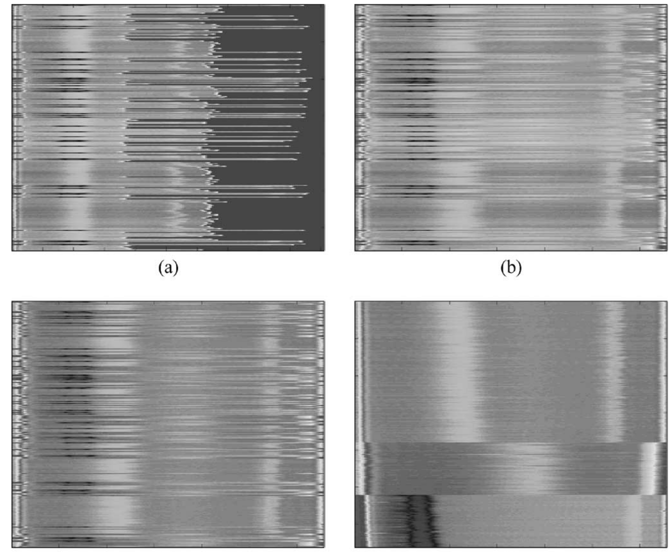
A. Main Techniques in the Literature

Processing the ECG signal as an image requires the transformation of the original record into a 2-D signal. This is accomplished by detection of each ECG period, followed by segmentation and row-oriented assembly. In this work, the QRS complex detection is executed with the algorithms available in [6]. Next, the signal is segmented and reassembled as an image by choosing the maximum value of the QRS complex as the segmentation boundary, leaving half peak at each end of the row. The 2-D array resulting from this process can be viewed in Fig. 1(a) for record 119 of the MIT-BIH ECG database. Although all periods are left aligned, they do not have the same length, usually due to the presence of some pathology or variations on patient's condition.

To correct this behavior and better exploit the interbeat dependencies, Wei *et al.* in [4] proposed the period normalization, also adopted by [3], which changes the length of all periods to a common value. Following this concept, an original ECG segment $X = [x(0)\ x(1)\ \ldots\ x(N_o-1)]$ can be converted into a normalized segment $X_n = [x_n(0)\ x_n(1)\ \ldots\ x_n(N_n-1)]$, which is computed using

$$X_n(m) = \hat{X}(h^*) \qquad h^* = \frac{m(N_o-1)}{(N_n-1)} \tag{1}$$

where $\hat{X}(h^*)$ is the decimated/interpolated version of $X(n)$, computed using cubic splines, N_o is the original period length, N_n is the normalized period length, and $m = 0, \ldots, N_n-1$. The decoder needs to know the sizes of each original period in order to reconstruct the original signal. This information is sent as side information, along with the header of the compressed file. Fig. 1(b) depicts the period-normalized version of the 2-D arrays in Fig. 1(a).



(a) (b) (c) (d)

Fig. 1. Effects of the proposed preprocessing techniques on record 119 of the MIT-BIH ECG data base. (a) Originally detected periods. (b) Length-normalized periods. (c) Dc equalization applied to (b). (d) Complexity sorting applied to (c).

In [1], another preprocessing technique was introduced, referred to as period sorting, which consists of a length-based ordering of all periods. Note that this relies on the supposition that periods with similar lengths tend to be highly correlated, which is not a very strong assumption and may not be valid for a large class of ECG signals (e.g., when some pathology is present).

B. Dc Equalization and Complexity Sorting

A brief evaluation of the length-normalized assembled image in Fig. 1(b) reveals that adjacent lines may have different dc levels; this creates high frequencies along the vertical direction of the resulting image, which tend to decrease the efficiency of most compressors. In order to tackle this problem, we devised a dc equalization procedure. In it, the original dc levels of all ECG periods are calculated according to

$$\mathrm{dc}_k = \left\lfloor \frac{1}{L_N} \sum_{i=1}^{L_N} x_k(i) \right\rfloor \tag{2}$$

where $x_k(i)$ are the signal samples of the kth period and L_N is their normalized length. Based on that, all periods are clamped to the minimum possible dc level, resulting in a smoother image. This process is described by

$$x_k^{\mathrm{dc}}(i) = x_k(i) - (\mathrm{dc}_k - \mathrm{dc}_{\min}) + \delta \tag{3}$$

where $x_k^{\mathrm{dc}}(i)$ are the dc-equalized period samples, $\mathrm{dc}_{\min}$ is the minimum dc level, and δ is an offset that guarantees that the minimum sample value is greaten than or equal to zero. The result of this preprocessing step is illustrated in Fig. 1(c).

However, although the dc equalization has helped in reducing the vertical high-frequency content of the resulting image, adjacent lines may still be very dissimilar after its application. This can be perceived in Fig. 1(c) as sharp discontinuities along the vertical direction. One way to alleviate this problem is to use the period sorting described in [1]; however, as it has been previously discussed (see Section II-A), this preprocessing technique, in some cases, still fails to guarantee the maximum similarity across adjacent periods.

Given the mentioned restrictions, we have devised a simple approach to deal with this problem: to rearrange the rows/periods using a metric that exploits the similarity among periods more efficiently than just the period length. After the completion of the dc equalization, the variances of all periods are computed, and the segment with the smallest variance is moved to the first row of the image matrix. The other rows are then occupied by the remaining segments in descending order of similarity with the first one; it has been verified that a good similarity metric in this case is the mean squared error, given by

$$\mathrm{MSE}_k = \frac{1}{L_N} \sum_{i=1}^{L_N} (x_k(i) - x_{\sigma_{\min}}(i))^2 \tag{4}$$

where $x_{\sigma_{\min}}(i)$ are the samples of the period with the smallest variance and $x_k(i)$ are the samples of the kth ECG period. We refer to this step as complexity sorting because the periods are ordered according to a correlation metric with the more "well-behaved," i.e., less complex, period. Its effect on the dc-equalized 2-D array of Fig. 1(c) is illustrated in Fig. 1(d).

When compared to the 2-D arrays generated by the other preprocessing techniques [see Fig. 1(a), (b), and (c)], the 2-D array that results from dc equalization and complexity sorting tends to increase the correlation between adjacent rows, favoring the performance of the 2-D encoder.

III. Experimental Results

To assess the coding efficiency of the proposed algorithm, we ran tests with the first 216 000 samples (10 min) from records 100, 117, and 119 from the MIT-BIH ECG database. These records have been chosen because there are compression results reported for them in the literature [1]–[5]. The signals were sampled at 360 Hz with 11 b of resolution. The dimensions of the resulting ECG images are variable and depend on the number of detected periods and their average length. The records 100, 117, and 119 were converted into images with dimensions 284 × 761, 427 × 506, and 328 × 660, respectively. The original length of each period, the original dc levels, the original ordering and the clamp level are arithmetically encoded and sent as side information to the decoder. The quality of the reconstructed signals was evaluated by using the percent root mean square difference (PRD), commonly adopted in the literature and defined as

$$\mathrm{PRD} = \sqrt{\frac{\sum_{i=0}^{N-1} (x(i) - \hat{x}(i))^2}{\sum_{i=0}^{N-1} (x(i) - \mu)^2}} \times 100\% \tag{5}$$

where $x(n)$ and $\hat{x}(n)$ are the original and the reconstructed signals, respectively, N is their length, and $\mu = 1024$. The compression ratio (CR) is evaluated as

$$\mathrm{CR} = \frac{B_o}{B_c} \tag{6}$$

where B_o is the total number of bits in the original signal and B_c the total number of bits in the compressed format, including side information. It is worth noticing that, for each of the test signals, $B_o = 11 \times 216\,000$, corresponding to the 11-b resolution and samples from the first 10 min of each ECG. The results are summarized in Table I. The compression method with length normalization, dc equalization, and complexity sorting is referred to as Type 3. Note that we also provide some performance figures for two other cases: one, where we use length normalization with dc equalization (Type 1) and another where we use length normalization with complexity sorting (Type 2), for records 100 and 119. Note that, although we compare methods by analyzing their PRDs for fixed CRs, we can infer that a similar performance is obtained when fixing the PRD. If one method has larger PRDs than the others for the same CRs, then the fact that the PRD × CR curve is monotonically increasing implies that, for a given PRD, it would also have a larger CR than the others. It can be seen that Type 3, where both dc equalization and complexity sorting are used with period normalization, provides better performance than in the cases when either only dc equalization (Type 1) or only complexity sorting (Type 2) are used. Also, the proposed algorithm outperformed all the other tested methods for all the records. The only exception is Chou *et al.* [1] with record 119 at a CR of 20.9 (for a CR of 10:1, the proposed method outperformed the one in [1]).

Two image compressors were used in the tests: the JPEG2000 (the Kakadu version, available at http://www.kakadusoftware.com) and the H.264/AVC intraframe encoder, version 12.1 of the reference software, available at http://iphome.hhi.de/suehring/tml/index.htm. We have used all the default parameters for JPEG2000, and varied only the quantization step, which was fixed to 0.00005. The main and tile headers were removed from the resulting file because they can be regenerated at the decoder. The adaptation of the H.264/AVC video encoder to the coding of the normalized ECG image requires the use of the FRExt High 100 profile (that uses YUV 4:0:0 chroma sampling format) [7], associated with 11 b luminance samples. The input file passed to the encoder is a YUV file with the preprocessed ECG samples (represented with 16 b) organized in a standard raster order. Rate-distortion (RD) optimization, deblocking filter, and context-based adaptive binary arithmetic coding (CABAC) were enabled, as well as 8 × 8 blocks for both the transform and the prediction steps. For a better exploitation of the H.264/AVC prediction process, the dimensions of the ECG image were changed to multiples of 16; the image rows were extrapolated by repeating the last line and the normalized row length was set to

$$N_n = 16 \left\lceil \frac{N_n}{16} \right\rceil . \tag{7}$$

An overall evaluation of the records reveals that if the target CR is high, H.264/AVC is preferred (at high ratios, residues are simple and easier to compress); otherwise, it is suggested to use JPEG2000. We have observed that when the H.264/AVC intraframe encoder is used, the complexity sorting is not so advantageous and can be skipped. Therefore, complexity sorting has been turned on only when the standard deviation of period lengths was greater than 10%.

IV. Discussion

A. Transmission Delay and Buffering

Like most ECG compressors that operate on 2-D arrays, the proposed scheme relies on the assumption that the periods are gathered and reassembled as an image. This will cause some delay, depending on the size of the ECG image. For example, in our tests, the transmission of the partial records would require a buffering time of 10 min. It is clear that this behavior compromises real-time data transmission; however, if one can tolerate the extra overhead, buffering time can be greatly reduced by using smaller images. Despite this, the proposed scheme is useful in cases when real-time transmission is not required, and also, when one is only concerned with data storage. One should note that the amount of memory needed for buffering ECG periods is not high. For example, if we use three consecutive bytes for storing two samples at 12 b each, the test signals would just require approximately 317 KB of memory.

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT ECG COMPRESSION SCHEMES

Algorithm	Record	CR	PRD
Lee et. al [5]	100	24 : 1	8.10
Chou et. al app. 2 [1]	100	24 : 1	4.06
Norm - JPEG2000	100	24 : 1	6.01
Type 1 - JPEG2000	100	24 : 1	4.10
Type 2 - JPEG2000	100	24 : 1	5.16
Type 3 - JPEG2000	100	24 : 1	3.95
Type 1 - H.264	100	24 : 1	3.47
Type 3 - JPEG2000	100	10 : 1	2.12
Type 1 - H.264	100	10 : 1	2.08
Type 3 - JPEG2000	117	24 : 1	1.72
Type 1 - H.264	117	24 : 1	1.64
Chou et. al app. 2 [1]	117	13 : 1	1.18
Type 3 - JPEG2000	117	13 : 1	1.07
Type 1 - H.264	117	13 : 1	1.14
Wei et. al [4]	117	10 : 1	1.18
Bilgin et. al [3]	117	10 : 1	1.03
Chou et. al app. 1 [1]	117	10 : 1	0.98
Type 3 - JPEG2000	117	10 : 1	0.86

Algorithm	Record	CR	PRD
Type 1 - H.264	117	10 : 1	0.95
Bilgin et. al [3]	117	8 : 1	0.86
Type 3 - JPEG2000	117	8 : 1	0.75
Type 1 - H.264	117	8 : 1	0.81
Bilgin et. al [3]	119	21.6 : 1	3.76
Tai et. al [2]	119	20 : 1	2.17
Chou et. al app. 2 [1]	119	20.9 : 1	1.81
Norm - JPEG2000	119	20.9 : 1	3.90
Type 1 - JPEG2000	119	20.9 : 1	3.77
Type 2 - JPEG2000	119	20.9 : 1	2.90
Type 3 - JPEG2000	119	20.9 : 1	1.92
Type 3 - H.264	119	20.9 : 1	1.78
Chou et. al app. 2 [1]	119	10 : 1	1.03
Norm - JPEG2000 [1]	119	10 : 1	1.37
Type 3 - JPEG2000	119	10 : 1	0.93
Type 3 - H.264	119	10 : 1	1.00
Type 3 - JPEG2000	119	8 : 1	0.74
Type 3 - H.264	119	8 : 1	0.83

B. Computational Complexity

The computational complexity of the QRS detection algorithm, length normalization, and JPEG2000 or H.264/AVC intraimage compressors is similar to the one of the schemes in [1] and [3]. The dc equalization consists only of one division per period, and a number of summations that is proportional to the period length [see (2) and (3)]; thus, the computational complexity associated with it is low. The complexity of the sorting algorithms consist of the operations described by (4) followed by a sorting procedure. The computational effort associated to (4) is one multiplication per input sample. The sorting procedure is similar to the one in [1].

C. Memory Usage

The total amount of memory needed by the proposed scheme consists of buffering memory and one ECG frame memory. The former has been verified to be approximately 317 KB; the latter depends on the image dimensions, that is, normalized-period length and number of detected periods. For example, taking into account the record 100 and considering a storage unit composed of 16 b integers, the amount of memory required would be approximately 422 KB.

V. CONCLUSION

We presented new preprocessing techniques that can be directly applied to the compression of both regular or irregular ECG signals. They can be incorporated by any compressor or combined with existing schemes, leading to an enhancement in the quality of the reconstructed signal. The performances of the proposed techniques were assessed by using the state-of-the-art JPEG2000 and H.264/AVC intraframe encoders; the resulting compressors were shown to outperform the methods described in the literature.

The results presented in this brief reveal new directions in ECG compression, showing that it is worthwhile to further develop techniques that adapt the ECG signal to the chosen compressor. With simple adaptations, one can obtain relevant improvements in compression performance using existent and well-established algorithms, accelerating and reducing the costs of the deployment of ECG compression methods.

REFERENCES

[1] H.-H. Chou, Y.-J. Chen, Y.-C. Shiau, and T.-S. Kuo, "An effective and efficient compression algorithm for ECG signals with irregular periods," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 6, pp. 1198–1205, Jun. 2006.

[2] S.-C. Tai, C.-C. Sun, and W.-C. Tan, "2-D ECG compression method based on wavelet transform and modified SPIHT," *IEEE Trans. Biomed. Eng.*, vol. 52, no. 6, pp. 999–1008, Jun. 2005.

[3] A. Bilgin, M. W. Marcellin, and M. I. Altbach, "Compression of electrocardiogram signals using JPEG2000," *IEEE Trans. Consum. Electron.*, vol. 49, no. 4, pp. 833–840, Nov. 2003.

[4] J.-J. Wei, C.-J. Chang, N.-K. Chou, and G.-J. Jan, "ECG data compression using truncated singular value decomposition," *IEEE Trans. Inf. Technol. Biomed.*, vol. 5, no. 4, pp. 290–299, Dec. 2001.

[5] H. Lee and K. M. Buckley, "ECG data compression using cut and align beats approach and 2-D transforms," *IEEE Trans. Biomed. Eng.*, vol. 46, no. 5, pp. 556–565, May 1999.

[6] M. Aboy, C. Crespo, J. McNames, J. Bassale, L. Jenkins, and B. Goldsteins, "A biomedical signal processing toolbox," in *Proc. Biosignal 2002*, Jun., pp. 49–52.

[7] G. Sullivan, P. Topiwala, and A. Luthra, "The H.264/AVC advanced video coding standard: Overview and introduction to the fidelity range extensions," in *Applications of Digital Image Processing XXVII (Proc. SPIE)*, A. G. Tescher, Ed., vol. 5558, Nov. 2004, pp. 454–474.