UDC 004.928

M.V.Davydov
Lviv Polytechnic National University

VISIBLE ARTICULATION SYNTHESIS FROM AUDIO STREAM FOR SIGN LANGUAGE TRANSLATION SYSTEM

Lips animation method for synchronous translation into sign language is described. As opposed to methods that work with a record of a full sentence, the proposed solution has a constant delay that does not depend of sentence length. The method is based on selection of video frames from a training video in a specific order to achieve natural look.

Keywords – speech driven lip animation, visible articulation, sign language translation

Introduction. Statement of the problem.

The problem of lips animation is studied in the context of sign language translation system development. Automatic translation of verbal language into sign language is an actual problem, which solution will expand the communication possibilities for people with hearing impairments.

This translation can be done from text, recorded audio or live audio stream. Sign language translation approaches have been studied for a long time abroad. For example, well known device iCommunicator [1] provides the possibility to translate English audio records or texts into Signed British. The main disadvantages of the known systems is the lack of translation into natural Sign Language, hand and lips animation inaccuracy.

Communication problems for people with reduced hearing arise in many situations. For example, when dealing with someone who does not speak sign language, in a telephone conversation or video phone call, listening to voice messages at railway stations, airports, watching TV where subtitles or sign language translation are not available.

To facilitate the perception of voice for people with reduced hearing, information should be supplemented by text messaging (subtitled), visible articulation, cued language [2], and sign language. The visible articulation plays an important role in the perception of voice for hard of hearing people. Accuracy of information understanding by people with hearing disabilities was 43% higher using voice and lip-reading comparing to voice-only communication in the studies conducted by Q. Summerfield [3]. Failure to see lips causes significant discomfort during a telephone conversation and significantly complicates the process of communication. Most systems for video telephony today do not solve the problem of articulation reproduction because transmission rate in such networks is not sufficient for a clear lip-reading.

The research conducted in the Sign Language Laboratory of Lviv Polytechnic National University is focused on development of a system for sign language translation with the least possible delay. Automatic sign language translation system being developed consists of software modules for speech recognition, translation, sign language synthesis, articulation synthesis and video encoding (Fig. 1). One of the important element of the sign language synthesis module is an articulation module that provides lip-reading possibilities along with sign language.

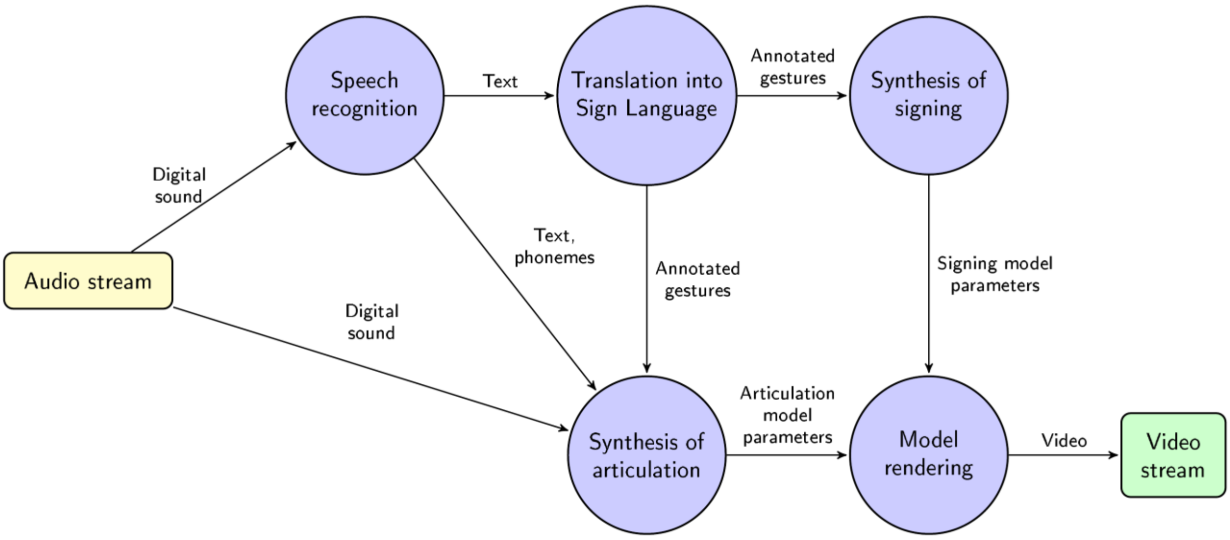

Fig. 1. Software modules and data flows in the system for translating into sign language

In the system for sign language translation animation lip can be performed in the following cases:

1. Lip animation from the audio stream to facilitate the perception of sound information by people with partial hearing loss.
2. Lip animation for the audio stream when the speech recognition module can't recognize words.
3. Animation lips while manually spelling words.
4. Lip animation as a part of sign language translation.

To date, the most investigated are cases 3 and 4 – the animation along with sign language or fingerspelling [4]. The focus of this article is on animating lips to enable lip-reading for people with partial hearing loss and animation in case when phonemic information is not available.

Analysis of recent research and publications

Automatic animation of realistic speaking characters remains a challenging problem in the field of computer graphics because of the complexity of facial expressions and human sensitivity to imperfections in visual articulation models. There are two main approaches to the synthesis of visible articulation for virtual characters [5]. The first one is based on phonemes. Phonemes can be obtained from the input text using dictionary or from speech recognition engine. The resulting video is created on the basis of phonemic information. The second approach is based on audio stream features without phoneme recognition.

The first approach can be used for synthesis of visible articulation from a text or from an audio record. The main advantage of this approach is the use of a dictionary that helps in distinguishing visemes with similar pronunciation but completely different visible articulation. However, the use of dictionary limits the set of words being animated properly. It leads to completely inappropriate lip animation in the case of speech recognition mistake. This is a big drawback provided that modern speech recognition software does one mistake for every 8 words on the average. Besides that, the transition to the phonetic representation removes emotional component of the sentence, laughter, intonation.

Modern automatic text-to-speech synthesis software is not able to arrange the logical stress properly, and, therefore, the phoneme-based approach has limitation for creation of realistic animation. The use of a dictionary causes delays in real time articulation synthesis, because the system has to waits for the pronunciation of complete word or sentence.

The approach based on articulation synthesis from audio stream features may preserve emotional component of the sentence, logical stress, laughter, yawn, and other non-phonemic phenomena. The animation can partially represent sounds of other languages, even if they are not contained in the training set. The transfer of such systems into another language is as easy as creation of a training video in another language.

The method based on audio features has its own disadvantages. For example, it is difficult to distinguish some visemes that are very similar in audio features, but differ in articulation. These are, for example, the sounds / m / and / n /, / b / and / v /.

The known solutions for visual articulation synthesis that are based on audio stream features use hidden Markov models and their modifications, Kalman filter, and neural networks for viseme selection.

The method of hidden Markov models was used by Simons and Cox [6] to create a synthetic speech-driven head. Fifty sentences were processed in order to create training samples, accompanied by video. There were extracted 10,000 vectors describing audio stream and 5000 vectors that describe facial image. These vectors were clustered to obtain 16 centroids for facial image and 64 centroids for sound features. The fully connected Markov model was created with 16 states. Each state of the model is responsible for one visual state of the face and in every state one frame of the video is rendered. The Markov model was trained using the Baum-Welch algorithm.

Kalman filter method, that was used in [7], leads to better smoothing of lip animation comparing to animation synthesized using hidden Markov models. Despite the fact that the animation has less ragged movements, it was, according to the authors, more different from natural.

Authors of the system named 'Picture my voice' [8] used recurrent neural network for transforming the space of 11x13 cosine Fourier transform coefficients into 37 parameters of facial animation. The training data was obtained from the system of facial animation based on phonemes. Phonetic representation was not used during animation synthesis.

A method based on coarticulation model was used in [9]. The method used intermediate phonetic representation of sentences. The trajectory of face control points was determined at the training stage for pronunciation of phoneme triples and pairs. Dynamic programming algorithm was used on the stage of synthesis to determine the optimal trajectory of control points. The method shows the best quality for videos for synthesized in English. The disadvantage of this method is the need to use large training database and the inability to synthesize video with small delay.

Article goals

The article is focused on the first two cases of articulation synthesis when the animation is done directly from audio stream. The stated problem is in the synthesis of visible articulation of virtual character according to the incoming audio stream. Parameter ΔT is set to limit the maximum delay of the synthesized animation relatively to input stream.

Main part

The developed articulation synthesis module consists of audio feature extraction and viseme synthesis sub-modules (Fig. 2).

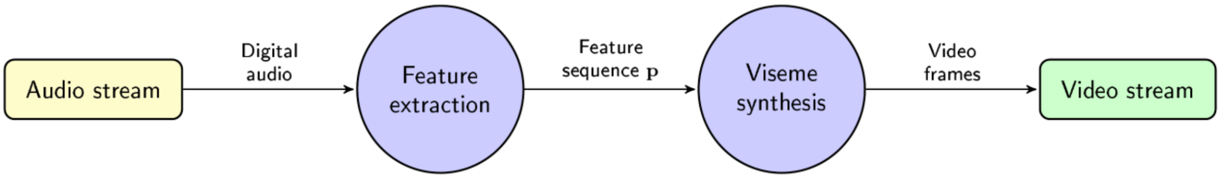

Fig. 2. The structure of articulation synthesis module

The digitized sound is coming from microphone or network as a sequence of samples $\{x(i)\}$ with sampling frequency F. Feature extraction unit processes the sequence $\{x(i)\}$ in order to extract its local characteristics. The energy of the sound at different frequencies and its change over the time is used as sound features instead of using sequence $\{x(i)\}$ itself. It was done because the sequence of values itself is not a useful representation of sound features [10, p. 614]. Audio stream is considered to be a sequence of samples $\{x(i)\}$ of the sampling frequency $F = 8kHz$. Any audio stream $x_1(i)$ with another sample rate F_1 is converted to sampling frequency F utilizing linear interpolation:

$$x(i) = x_1\left(\left\lfloor i\frac{F_1}{F}\right\rfloor\right)\left(i\frac{F_1}{F} - i\frac{F_1}{F}\right) + x_1\left(i\frac{F_1}{F}\right)\left(1 - i\frac{F_1}{F} + i\frac{F_1}{F}\right).$$

The following convolution is used to extract audio energy indicator at frequency f in the neighborhood of sample q:

$$E(q, f) = \sqrt{\left(\sum_{i=q-s}^{q+s} x(i)\sin\left(2\pi\frac{f}{F}i\right)e^{-\alpha^2\left(\frac{i-q}{F}\right)^2}\right)^2 + \left(\sum_{i=q-s}^{q+s} x(i)\cos\left(2\pi\frac{f}{F}i\right)e^{-\alpha^2\left(\frac{i-q}{F}\right)^2}\right)^2},$$

where s is a half size of a window, α is a parameter of Laplace window function (α is a value about 400, proposed in [11]).

The energy was calculated for frequencies in range 80…800 Hz with step 40 Hz:

$$E_i(q) = E(q, 40i + 40),\ i = 1,...,19.$$

The obtained 19 values if energy at frequencies 80, 120, …, 800 Hz where smoothed to 8 values using averaging formula:

$$E_i(q) = \sum_{j=(2i+1)-2}^{(2i+1)+2} E_j(q)e^{-\left(\frac{j-(2i+1)}{4}\right)^2},\ i = 1,...,8,$$

where $E_i(q)$ is a weighted sum of signal energies at frequencies $80i...80i + 160Hz$.

For the purpose of noise reduction the energy was averaged over time using 4 points with 128 sample steps:

$$\overline{E}_i(q) = \frac{E_i(q-192) + E_i(q-64) + E_i(q+64) + E_i(q+192)}{4},\ i = 1,...,8,$$

$$D_i(q) = \frac{-3E_i(q-192) - E_i(q-64) + E_i(q+64) + 3E_i(q+192)}{4},\ i = 1,...,8,$$

where $\overline{E}_i(q)$ as an average energy in the neighborhood of sample q at frequencies $80i...80i + 160Hz$, $D_i(q)$ is an average speed of energy change in the neighborhood of q at frequencies $80i...80i + 160Hz$.

To describe the characteristics of the audio signal in the time neighborhood of q 16 values $< \overline{E}_1(q),..., \overline{E}_8(q), D_1(q),..., D_8(q) >$ where used. The characteristics of the signal were determined with step of $8000/FPS$ samples, where FPS is a frame rate per second.

The output of feature extraction unit is a sequence $\mathbf{p} = p_1p_2...p_N$ of vectors $p_i \in P$ that describe characteristics of the audio stream at regular time intervals $1/F_v$, where F_v is a frame rate of video to be created. The number of samples N that are available when the next frame of video has to be created is determined using the parameter of maximum delay ΔT:

$$N = \Delta T \cdot F_v.$$

The sequence $\mathbf{p}$ is an input to the viseme synthesis unit. The output of this unit is a sequence $\mathbf{v} = v_1v_2...v_i...$ of vectors $v_i \in V$ that uniquely describe each frame of video to be created. The unit creates a video stream and combines it with sound. At each step, a single video frame is synthesized, and then the next element of a sequence $\mathbf{p}$ is obtained and its first element is removed.

In order to generate vectors that describe the output video a hidden Markov model $\mathrm{M} = < \mathrm{T}, \mathrm{P}, \mathrm{A}, \mathrm{B} >$ is utilized, where T is a set of Markov model states, P is a set of possible observations, $A: T \times T \times P \to R$ is the probability matrix of transition between states, and $B: T \times P \to R$ is the probability matrix of obtaining observations in the states of the model. Unlike the existing approaches, the probabilities of transitions between states of a hidden Markov model are calculated based on the received observations that helped to narrow down the search to the optimal sequence of states and get a more natural animation.

States T of the hidden Markov model are all frames $q_1 q_2 \ldots q_m$ of the training video. Let's denote $Q: T \to P$ the map from training video frames into audio features extracted in their neighborhood and $W: T \to V$ – the map from video frames to vectors that describe face model in it.

The problem of visible articulation synthesis is considered as a problem of searching optimal sequence of states $t_1 t_2 \ldots t_n$, $t_i \in T$ given input audio stream, so that the synthesized sequence of frames from a sequence of vectors $W(t_1), W(t_2), \ldots, W(t_n)$ that encode face images in the scene in the most natural way.

The search of optimal sequence of Markov model states is performed at every frame selection stage that maximizes their posterior probability given previous frame t_0 and sequence of observations p:

$$\langle t_1, t_2, \ldots, t_N \rangle = \underset{t_1, t_2, \ldots, t_N}{\arg\max} P(t_1, t_2, \ldots, t_N \mid t_0, p), \qquad (1)$$

where t_0 is the state of Markov model, that was used for synthesis of previous video frame, that was already presented to user, and p is a sequence of extracted features from available part of audio stream. The probability of state sequence in the modified Markov model is calculated using formula

$$P(t_1, t_2, \ldots, t_N \mid t_0, p) \sim P(p \mid t_1, t_2, \ldots, t_N) \cdot P(t_1, t_2, \ldots, t_N \mid t_0) = \prod_{i=1}^{N} B(t_i, p_i) \cdot A(t_{i-1}, t_i, p_i). \qquad (2)$$

The probability of Markov model output is defined as a normal distribution with mode obtained from training video:

$$B(t_i, p_i) = \exp\left(-k_1 \cdot |Q(t_i) - p_i|^2\right),$$

where k_1 is a coefficient of audio stream influence.

The calculation of transition probabilities is done by search within C training video frames where extracted audio feature vector was the most similar to obtained in sequence. Besides this the next frame in the training video is also considered as possible transition. For a set of states $\{t_j\}$ the transition probability is set proportional to the probability of observations and the similarity of frames

$$A(t_{i-1}, t_j, p_i) \sim B(t_j, p_i) \cdot \exp\left(-k_2 |W(t_j) - W(t_c)|^2\right),$$

where k_2 is a coefficient that takes into account similarity of adjacent frames of synthesized video and reduces the visible jumpy change in articulation.

Fig. 3 illustrates the process of selecting possible states of the model for further synthesis of the video. In this example t_0 is the last frame that has been synthesized. This frame corresponds to frame q_2 of the educational video. q_2 is the last known state of the Markov model. The set of possible transitions q_3, q_5, q_6 limits the frame t_1 that can be rendered next to t_0. State q_3 is chosen as the frame next to t_0 is the training video, states q_5 and q_6 are selected using the minimum distance search from audio stream feature vector p_1. In the example $C = 2$.

Finding a set of C possible transitions is done by means of R-tree search algorithm, that guaranties search time complexity $O(C \log(|T|))$. Expression (1) is calculated using dynamic programming. Given that at each step the number of sates is incremented not more than by C, the number of all possible states in dynamic programming does not exceed $1 + TC$. Thus the overall search complexity $O\left(T(1+TC)^2\right)$ or

$O\left(T^3C^2\right)$. To reduce the computational complexity of finding maximum (1) only Q the most probable states where selected at each step, which helped reduce the search time to acceptable results without visible drawbacks. In the experiments the value of T>30 did not lead to improved articulation synthesis because it was more than the average time of pronunciation of a word. The value of Q is selected in the range 10…20, that makes it possible to apply the algorithm in real time.

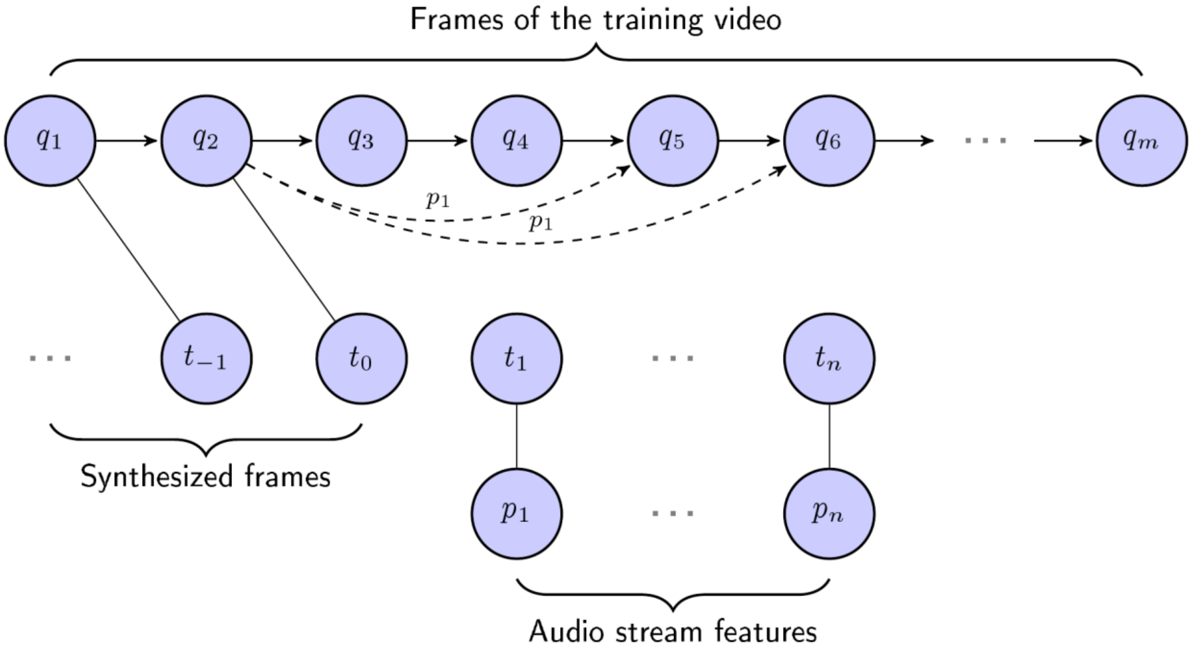

Fig. 3. The set of possible transitions of the Markov model implied by audio stream feature vector p_1.

Experimental results

The developed system was evaluated on the basis of video mailbox service. The service converts regular audio message in the mailbox into a properly articulated video.

The training video for the service was created using manually animated virtual character who utters training monologue. This video was created using Poser with the phrase in German «Guten Tag lieber Anrufer! Sie sind verbunden mit meiner Video-Mailbox. Zur Zeit bin ich leider nicht persönlich erreichbar. Bitte hinterlassen Sie eine Nachricht nach dem Signalton ». The length of the training video was 10 seconds recorded at 30 FPS.

The visual articulation for Ukrainian, German, and English phrases was synthesized using the developed method. The example sequence of frames for a word “Hello” in English is shown in Fig . 4. The time for synthesis of a single video frame was 20 ms for $N = 5$ and $C = 10$ on a quad core computer with 3.1 GHz Intel i5 processor.

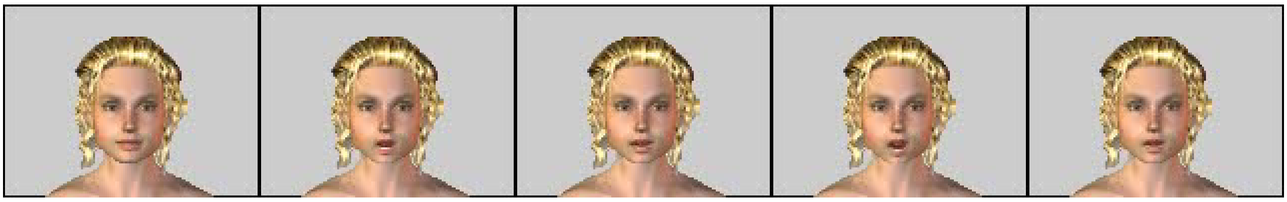

Fig. 4. Frames of viseme sequence for English word “Hello” synthesized on the basis of German training video. The frames are given in 5 frames interval.

The created prototype was used for the synthesis of video on a server that serves visual mailbox for videophones. According to the experiments, it was easier for users with reduced hearing to understand the synthesized message accomplished with lip animation then original voice message that contained only sound. In the absence of programs that synthesize lip animation in real time, comparison was carried out with the program LipSync developed by Annosoft, which is one of the world leaders in the development of lip sync animation software. LipSync program performs preprocessing of a file to select phonemes. In experiments the synthesized articulation for the German, English, and Ukrainian phrases was compared to animation created with LipSync. The animation made by LipSync was smoother, but there was no significant advantage for understanding the video in real time using lip-reading.

Conclusions

In the development of this work the prototype of lip animation synthesis system was developed and evaluated to produce articulation to help people with partial hearing loss. Virtual characters, who speak, can also be used for interactive web-pages in videophone systems, gaming systems and more. Further research will be aimed at improving the audio-visual transformation that can animate not only the language but also other sounds such as call, cry, laugh, and so on.

1. iCommunicator Features and Benefits / ел. ресурс. - реж. доступу http://www.myicommunicator.com/productinfo/features_benefits.shtml. - перевірено 10.10.2013. 2. Heracleous P. Gestures and Lip Shape Integration for Cued Speech Recognition / P. Heracleous, N. Hagita, D. Beautemps // Pattern Recognition (ICPR), 2010 20th International Conference on. – 23-26 Aug. 2010. – pp. 2238-2241. 3. Summerfield Q. Lipreading and audio-visual speech recognition / Q. Summerfield // Phil. Trans. R. Soc. Land. B. – vol. 335. – 1992. – pp. 71-78. 4. Krak Yu.V. Animation of virtual models of human face in speech synthesis (in Ukrainian) / Yu.V. Krak, O.V. Barmak // Artificial intelligence – 2002. Proceedings of International Scientific and Technical Conference. – Taganrog. TRTU. - Vol. 2. - 2002. - с. 138-142. 5. Ezzat T. Trainable videorealistic speech animation / T. Ezzat, G. Geiger, T. Poggio // ACM Transaction on Graphics (Proceedings of ACM SIGGRAPH' 02). – 2002. – pp. 388–398. 6. Simons A. Generation of mouth shapes for a synthetic talking head / A. Simons, S. Cox. // Proc. Inst. Accoust. – vol. 12. – 1990. – pp. 475-482. 7. Lehn-Schioler T. Mapping from speech to images using continuous state space models / T. Lehn-Schioler, L. K. Hansen, J. Larsen // Book Section. Springer Berlin Heidelberg: Machine Learning for Multimodal Interaction. – Lecture Notes in Computer Science. – 2005. – pp. 136-145. 8. Massaro D. W. Picture my voice: Audio to visual speech synthesis using artificial neural networks / D. W. Massaro, J. Beskow, M. M. Cohen, C. L. Fry, and T. Rodriguez // Proc. AVSP 99. – 1999. 9. Deng Z. Synthesizing speech animation by learning compact speech co-articulation models / Z. Deng, J. P. Lewis, and U. Neumann // Proc. of Computer Graphics International (CGI) 2005. – Long Island, NY: IEEE Computer Society Press. – June 2005. 10. Allen J. Natural Language Understanding / James Allen. – 2nd ed. The Benjamin Comings Publishing, Inc. – 1995. 11. Sorokin V., Tsyplikhin A. Segmentation and recognition of vowels (in Russian). Informational processed. Vol 4. №2, pp. 202-220 – 2004.