Design and Evaluation of a Distributed Intermedia Synchronization Algorithm based on the Fuzzy Logic

Sun-Jin Oh† · Ihn-Han Bae††

ABSTRACT

The basic requirements of distributed multimedia systems are intramedia synchronization which asks the strict delay and jitter for the check period of media buffer and the scaling duration with periodic continuous media such as audio and video media, and intermedia synchronization that needs the constraint for relative time relations among them when several media are presented in parallel. In this paper, a distributed intermedia synchronization algorithm that solves the intermedia synchronization problem by using variable service rates based on the fuzzy logic is designed and then the performance is evaluated through simulation.

퍼지 논리에 기반한 분산 인터미디어 동기화 알고리즘의 설계 및 평가

오선진† · 배인한††

요 약

분산 멀티미디어 시스템의 기본적인 요구 사항은 비디오와 오디오 같은 등시성 데이터의 생성 또는 검색 시점에서 표현 시점까지의 엄격한 지연과 지터를 요구하는 인트라미디어 동기화와 다수의 연속 미디어 스트림이 병렬로 표현될 때 그것들간의 상대적 시간 관계에 관한 제한을 만족시키는 인터미디어 동기화를 제공하는 것이다. 본 논문에서는 퍼지 논리에 기반한 가변 서비스 율을 사용하여 인터미디어 동기화 문제를 효율적으로 해결하는 분산 인터미디어 동기화 알고리즘을 설계하고, 그것의 성능을 시뮬레이션으로 평가하였다.

1. Introduction

With the development of multimedia processing techniques and high-speed communication network using computer technologies, studies on various distributed multimedia applications via computer communication network, such as on-line multimedia conference system and multimedia information search system, are now making a great progress in these research fields. There are, however, still several problems to implement such multimedia applications. One of them is the multimedia synchronization technique to maintain time relations among multimedia data. Such multimedia synchronization is achieved by controlling data transmission artificially to support the quality of service (QoS) of multimedia data properly. Since the requirements of QoS deeply depends on both applications and characteristics of media, an appropriate control of the artificial data transmission for synchronization should be allowed. The multimedia

† Department of Computer Science, Pohang Sunlin College
†† Department of Computer Engineering, Catholic University of Taegu-Hyosung

synchronization is involved in both intramedia and intermedia synchronization. The intramedia synchronization focuses on the temporal relationship within a single media stream when the periodic continuous media stream is presented on the distributed on-line multimedia system environment and it asks the strict delay and jitter for the check period of media buffer and the scaling duration. The intermedia synchronization, on the other hand, is involved in the temporal relationships among several media streams which are presented in parallel simultaneously and it asks the constraints for relative time relationships among them.

In this paper, we present a distributed intermedia synchronization algorithm based on the fuzzy logic. In order to solve the intermedia synchronization problem, receiver checks the differences between master and slave media's generation time and applies the variable service rates to the slave processor by using fuzzy logic based on the relative speed of slave media and the asynchronous time between master and slave media units. The performance of the algorithm is evaluated through simulation.

The remainder of this paper is organized as follows: Section 2 summaries related works for multimedia synchronization. Section 3 presents the proposed intermedia synchronization algorithm based on the fuzzy logic, and the algorithm is evaluated in section 4. Finally, section 5 concludes the paper.

2. Multimedia Synchronization

The primary objective of multimedia synchronization is to integrate continuous media such as audio and video into semantically synchronized multimedia objects in distributed multimedia systems. Since the temporal nature of continuous media, there are two kinds of multimedia synchronization techniques such as intramedia synchronization and intermedia synchronization. Intramedia synchronization is related to the transmission of each media object within the processing time as shown in Fig. 1(a). For the intramedia synchronization, transmission schedule is generated by computing the completion time of transmission from completion time of process with consideration of network delay. An appropriate scheme that can correct this problem is required at destinations. Intramedia synchronization asks the computation of the required maximum buffer size prior to transmit media stream. Buffer is used to synchronize the stream data by smoothening the effects of end-to-end network delay and clock drift. Thus, intramedia synchronization is a low level technique that processes the synchronization management among various data streams produced from communication channel. Intramedia synchronization is requested when there exists some differences of time in a multimedia object. On the other hand, intermedia synchronization maintains the time relations among different but related media streams as shown in Fig. 1(b). Intermedia synchronization addresses the temporal relationships that may exist among individual media streams, such as lip-synchronization between video and audio objects. The requirements can be originated based on a natural and contrived coupling of two or more streams, and must be vary depending on several factors including the contents of media stream and its intended use. Intermedia synchronization is requested when there exists some differences between master and slave media's generation time and applies the variable service rates based on the fuzzy logic

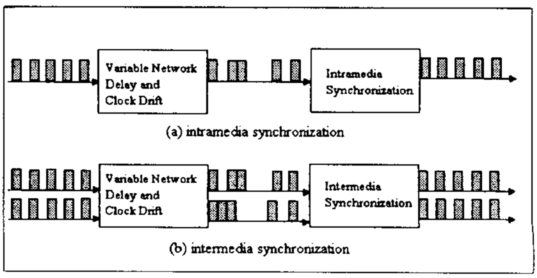


Fig. 1. Multimedia synchronications.

to the slave media to offset the asynchronization.

Related researches for multimedia synchronization are as follows: Escobar et al. [7] propose the scheme that realize the synchronization by using master clock shared by all nodes concerned with transmission of multimedia data. Playback time is decided by the given time contained in the media unit. Little et al. [8] use the scheme that controls the buffer level at receiver site. Synchronization problem is solved by dropping or duplicating frames according to the transmission rates in the buffer of receiver site. Anderson et al. [5] describe the local synchronization algorithm for recovering from asynchronization among interrupt-driven media I/O devices connected to a continuous media (CM) I/O server. The synchronization mechanism used is called a local time stamp (LTS). The devices having the most critical intramedia continuity is chosen as the master, and the other devices (slaves) are skipped or paused to maintain synchronization with master. Rangan et al. [2] present feedback techniques for synchronization in distributed multimedia systems. In order to maintain intramedia continuity and intermedia synchronization, feedback units are sent back to the server with the number of media units. These feedback units are used by the server to estimate the earliest and latest playback times of the media units sent by the master and slaves to determine if asynchronization exists. The master is chosen as the most critical intramedia continuity device. Scaling is accomplished by dropping or duplicating slave's video frames.

3. Distributed Intermedia Synchronization based on the Fuzzy Logic

In this paper, a distributed intermedia synchronization algorithm, operated in the environment of distributed multimedia system as shown in Fig. 2, is designed, where, each media unit is transmitted from media server, such as video or audio server, to clients. One practical application of this system

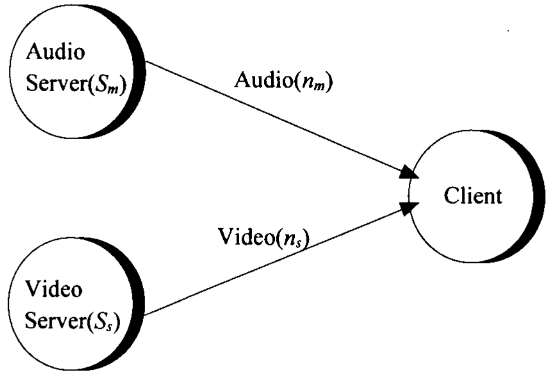


Fig. 2. The system configuration of a distributed multimedia system.

is VOD(video on demand) applications. In this section, the intermedia synchronization algorithm, which is executed under the distributed multimedia environment as shown in Fig. 2, is designed, where, each media unit is transmitted from media server to client. Media units which are generated at a constant rate with nominal period of generation p are transmitted to the client over a network whose minimum and maximum communication delays are $\Delta_{\min}$ and $\Delta_{\max}$, respectively. Let S_m be the master server, and S_s be a typical slave server. In general, human recognition of audio error is more sensitive than that of video error. Therefore, let audio be the master media and video be the slave media respectively.

In this paper, by using fuzzy logic based on the relative speed of slave media compared with master media's speed and the asynchronous time between master and slave media units, intermedia synchronization algorithm that plays back the slave media with variable service rates is designed. The asynchronous time between master and slave media units is calculated from Rangan[2]'s approach. If τ_m and τ_s are the arrival times of media unit n_m and n_s at client from media server S_m and S_s respectively, the earliest and the latest generation times of n_m and n_s represented as $g^e_m(n_m)$, $g^l_m(n_m)$, $g^e_s(n_s)$, and $g^l_s(n_s)$, respectively, can be derived as follows:

$$g^e_m(n_m) = \tau_m - \Delta_{\max} \quad (1)$$
$$g^l_m(n_m) = \tau_m - \Delta_{\min} \quad (2)$$
$$g^e_s(n_s) = \tau_s - \Delta_{\max} \quad (3)$$
$$g^l_s(n_s) = \tau_s - \Delta_{\min} \quad (4)$$

The generation time of media unit n_m and n_s, $g_m(n_m)$ and $g_s(n_s)$, are included in generation intervals $[g^e{}_m(n_m), g^l{}_m(n_m)]$, $[g^e{}_s(n_s), g^l{}_s(n_s)]$ respectively.

Theorem Media units n_m and n_s belong to a synchronization set if

$\delta g_{\max}(n_m, n_s) \leq \varepsilon$,

where,

$\delta g_{\max}(n_m, n_s) = max(g^l_m(n_m) - g^e_s(n_s), g^l_s(n_s), - g^e_m(n_m))$ and ε is the maximum tolerable asynchronous time.

Proof Since $g^e_m(n_m) \leq g(n_m) \leq g^l_m(n_m)$ and $g^e_s(n_s) \leq g(n_s) \leq g^l_s(n_s)$, we get

$g^l_m(n_m) - g^e_s(n_s) \geq g(n_m) - g(n_s)$

and $g^l_s(n_s) - g^e_m(n_m) \geq g(n_s) - g(n_m)$.

Therefore,

$max(g^l_m(n_m) - g^e_s(n_s), g^l_s(n_s) - g^e_m(n_m)).$

$\geq | g(n_m) - g(n_s) |$

As a result, if

$\delta g_{\max}(n_m, n_s) = max(g^l_m(n_m) - g^e_s(n_s), g^l_s(n_s) - g^e_m(n_m)) \leq \varepsilon$

then

$| g(n_m) - g(n_s) | \leq \varepsilon$

and hence media units n_m and n_s belong to a synchronization set[2]. □

Since the exact generation time of media unit is not known to client, the asynchronous time can be calculated based on the generation interval of media units transmitted from master to slave servers as in Eq. (5), where, function *dif* gets the maximum value between two numbers without considering sign.

$$\delta g_{dif}(n_m, n_s) = dif(g^l_m(n_m) - g^e_s(n_s), g^e_m(n_m) - g^l_s(n_s)) \quad (5)$$

The membership function for the degree of synchronization between master and slave media units is shown in Fig. 3, and the fuzzy mapping function for the degree of synchronization between master and slave media units is as in Eq. (6).

$$\mu_S(\delta g_{dif}(n_m, n_s)) = S(\delta g_{dif}(n_m, n_s), \varepsilon)$$
$$= \begin{cases} 0 & \delta g_{dif}(n_m, n_s) \leq -\varepsilon \\ \dfrac{\delta g_{dif}(n_m, n_s) + \varepsilon}{\varepsilon} & -\varepsilon < \delta g_{dif}(n_m, n_s) < 0 \\ \dfrac{\varepsilon - \delta g_{dif}(n_m, n_s)}{\varepsilon} & 0 \leq \delta g_{dif}(n_m, n_s) < \varepsilon \\ 0 & \delta g_{dif}(n_m, n_s) \geq \varepsilon \end{cases} \quad (6)$$

where, S is the fuzzy mapping function that represents the degree of synchronization.

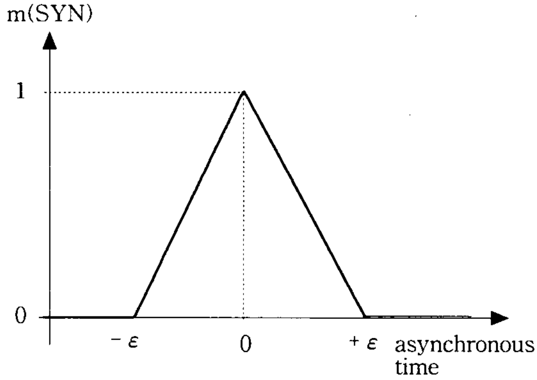


Fig. 3. The membership function for the degree of synchronization among media.

The normalization of fuzzy set for the degree of synchronization between master and slave media is represented in table 1. Rules for variable service rate control, which is based on the asynchronous time between master and slave media units and the relative speed of slave media compared with master media's speed, is represented in Table 2, where, if $\delta g_{dif}(n_m, n_s) > 0$ then the state of slave media is SLOW, and FAST,

Table 1. The normalization of fuzzy set for the synchronization

membership function	normalized full set	normalized regions	basic fuzzy set
$\mu_S(\delta g_{dif}(n_m, n_s))$	[0.00~1.00]	[0.00~0.24] [0.25~0.49] [0.50~1.00]	WSYN MSYN SSYN

Table 2. The rules for variable service rate control

SMS \ SYN	SSYN	MSYN	WSYN
SLOW	NSR	LHSR	VHSR
FAST	NSR	LLSR	VLSR

the number of rules = 6

(Input Variable)
SYN - Synchronization
SMS - Slave Media State
(Output Variable)
VSR - Variable Service Rate
(Linguistic Variable)
SLOW - A slave media is slower than the master media
FAST - A slave media is faster than the master media
SSYN - Strong Synchronization
MSYN - Medium Synchronization
WSYN - Weak Synchronization
NSR - Normal Service Rate
LLSR - Little Low Service Rate
VLSR - Very Low Service Rate
LHSR - Little High Service Rate
VHSR - Very High Service Rate

otherwise.

Now, we define an algorithm executing the proposed intermedia synchronization based on the fuzzy logic in Fig. 4.

4. Simulation and Evaluation

In this paper, the performance of proposed intermedia synchronization algorithm is evaluated through simulation, where the asynchronous time between master and slave media units is examined and the number of occurrences of variable service rates are counted. Simulation parameters used in this paper are listed in Table 3.

Table 3. Simulation parameters

parameter	value
maximum asynchronous time(ε)	150 ms
synchronous jitter(j_s)	5 ms
NSR	30 frames/sec
LLSR	28 frames/sec
VLSR	26 frames/sec
LHSR	32 frames/sec
VHSR	34 frames/sec

Fig. 5 shows the asynchronous time between master and slave media units with time. No fuzzy synchronization doesn't achieve intermedia synchronization because the asynchronous time between master and slave media is large. Specially, asynchronism occurs at 110~120 seconds, 170 second, and 270~290 seconds. However, the fuzzy synchronization works quite well with intermedia synchronization because the asynchronous time is less than the maximum tolerable asynchronous time between audio and video media.

Fig. 6 shows the number of occurrences of variable service rates. As shown in Fig. 6, 93.6% of the total processed video frames are processed with normal service rate(NSR), 2.8% are with little low service rate(LLSR), and 3.5% are with little high service rate(LHSR). So, no video frames are processed with either very low service rate(VLSR) or very high service rate(VHSR). Therefore, QoS degradation due to the variable service rate is almost none since the variable service rates are within the range of ±5% of the nominal service rate in video media.

5. Conclusion

In this paper, we present a distributed intermedia synchronization algorithm based on the fuzzy logic. The intermedia synchronization is achieved by

Distributed Intermedia Algorithm:

Step 1) Compute the generation interval of each media unit that is received from media server.
Step 2) Compute the asynchronous time, $\delta g_{dif}(n_m, n_s)$ between master and slave media units from the generation intervals, periodically. The period of synchronization check is $\frac{\varepsilon}{j_s}$, where, j_s denotes synchronization jitter that is the maximum allowable drift between the constituent streams of presentation. This drift is the difference between the presentation progress of the fastest and slowest streams.
Step 3) Get the degree of membership for synchronization between master and slave media using the fuzzy mapping function of Eq. (6).
Step 4) Get the basic fuzzy set which corresponds to the degree of membership.
Step 5) Decide the control rule for variable service rate according to the relative speed of slave media compared with master's speed and the synchronization degree between mater and slave media from Table 2.
Step 6) The slave processor plays back the $n = \lfloor \frac{\delta g_{dif}(n_m, n_s)}{\delta t} \rfloor$ frames of the slave media with the selected display rate, where $\delta t = | f_s - f_s' |$, f_s and f_s' are the display times of one frame by normal rate and selected variable service rate, respectively.

Fig. 4. The proposed distributed intermedia synchronization algorithm based on the fuzzy logic

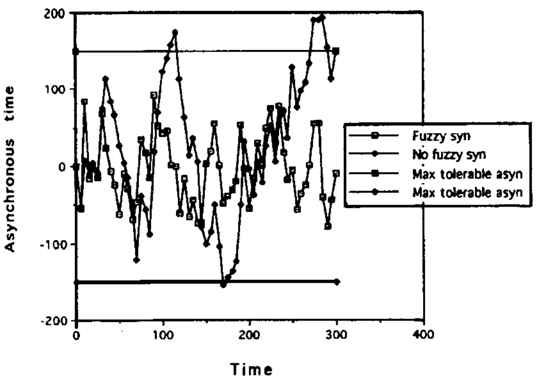


Fig. 5. The asynchronous time between master and slave media units

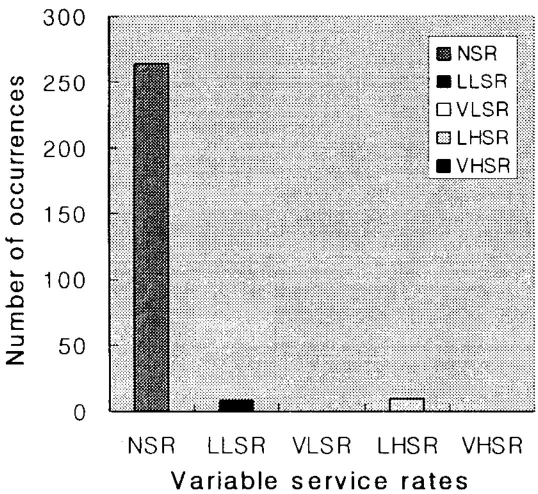


Fig. 6. The frequency of occurrences of variable service rates.

using variable service rates based on the fuzzy logic for the relative speed of slave media and the asynchronous time between master and slave media units, at receiver site.

The proposed algorithm is evaluated with respect to synchronization accuracy and the number of occurrences of variable service rates through simulation. Simulation results shows that the proposed algorithm is performed quite well with regards to intermedia synchronization. No fuzzy synchronization could not achieve intermedia synchronization since the asynchronous time between master and slave media is too large. On the other hand, the fuzzy synchronization can achieve intermedia synchronization since the asynchronous time is much less than the maximum tolerable asynchronous time between audio and video. Furthermore, the QoS degradation due to variable service rates is relatively trivial since video frames are processed with neither very low service rate(VLSR) nor very high service rate(VHSR) in any circumstances. On the whole, it has been proved that the variable service rates are at most within the range of ±5% of the nominal service rate in video media

A realization of the distributed intermedia synchronization using variable service rates based on fuzzy logic and effective interaction between intramedia and intermedia synchronization are our future works

References

[1] E. Stoica, H. Abdel-Wahab and K. Maly, "Synchronization of Multimedia Streams in Distributed Environments", *Proceedings of the Int. Conf. on Multimedia Computing and Systems*, pp.395~402, 1997.

[2] P. V. Rangan, S. Ramanathan, H. M. Vin, and T. Kaepper, "Techniques for Multimedia Synchronization in Network File Systems", *Computer Comm.*, Vol. 16, No. 3, pp. 168~176, March 1993.

[3] J. Sandvoss, J. Winckler and H. Wittig, "Network Layer Scaling Congestion Control in Multimedia Communication with Heterogeneous Networks", *IBM European Networking Center Technical Report* 43.9401, 1994.

[4] L. Delgrossi, C. Halstrick, D. Hehmann, R. G. Herrtwich, O. Krone, J. Sandvoss and C. Vogt, "Media Scaling for Audiovisual Communication with the Heidelberg Transport System", *Proceedings ACM Multimedia 93*, pp. 99~104, Aug. 1993.

[5] D. P. Anderson and G. Homsy, "A Continuous Media I/O Server and Its Synchronization Mechanism", *IEEE Computer*, Vol. 32, No. 10, pp. 51~57, Oct. 1991.

[6] L. Ehley, M. Llyas and B. Furht, "A Survey of Multimedia Synchronization Techniques", *IEEE Computer Society Press*, pp. 230~256, 1995

[7] J. Escobar, D. Deutsch, C. Partridge, "Flow Synchronization Protocol", *Proc. of the IEEE Globecom*, Vol. 3, pp. 1381~1387, 1992.

[8] T. D. C. Little, F. Kao, "An Intermedia Skew Control System for Multimedia Data Presentation", *Proc. of the 3rd Int. Workshop on Network and Operating System Support for Digital Audio and Video*, 1992.

[9] M. M. Gupta and T. Yamakawa, *Fuzzy Computing*, Elsevier Science Pub., p. 500, 1988.

[10] G. J. Killer, T. A. Folger, *Fuzzy Sets, Uncertainty, and Information*, S. G. Osbourne, p. 355, 1988.

[11] 노홍태, 하숙정, 배인한, "스케일링과 가변 서비스 율을 사용한 분산 멀티미디어 동기화 알고리즘의 설계 및 평가", 한국정보과학회 학술발표 논문집, 제 23권 1호, pp. 399~402, 1996.

[12] 오선진, 이경숙, 배인한, "퍼지 논리에 기반한 분산 인터라미디어 동기화 알고리즘", 제9회 산·학·연 멀티미디어 산업기술 학술대회 논문집, pp. 37~42, 1997.

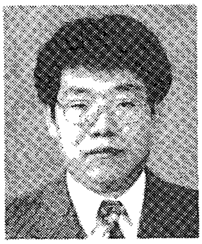
[13] 오선진, 배인한, "퍼지 논리에 기반한 분산 인트라미디어 동기화 알고리즘의 설계 및 평가". 한국 퍼지 및 지능 시스템 학회 논문지 Vol. 8, No. 1, pp. 78~85, 1998.

[14] 변증남, 이한오, 류형근, "퍼지 논리를 이용한 발전소 기동 및 정지의 자동화", 정보과학회지, 제 10권, 제 1호, pp. 56~64, 1992.

[15] 이광형, 오길록, 퍼지 이론 및 응용(II권: 응용), 홍릉과학출판사, 1992.

오 선 진

1981년 한양대학교 공과대학 졸업(공학사)
1987년 미국 Wayne 주립대학교 전자계산학과 (Post Bachelor)
1990년 미국 Detroit 대학교 대학원 전자계산학과 졸업 (이학석사)
1993년 미국 Oklahoma 주립대학교 대학원 전자계산학과 박사과정 수료
1994년~현재 포항선린대학 조교수/학과장
1995년~현재 대구효성가톨릭대학교 대학원 전산통계학과 박사과정
관심분야 : 운영체제, 분산시스템, 멀티미디어시스템, 퍼지시스템, 인공지능, 객체지향 데이터 베이스 시스템



배 인 한

1984년 경남대학교 전자계산학과 졸업(공학사)
1986년 중앙대학교 전자계산학과 졸업(이학석사)
1990년 중앙대학교 전자계산학과 졸업(공학박사)
1996년~1997년 Dept. of Computer and Information Science, The Ohio State University, USA(Post Doctor)
1989년~현재 대구효성가톨릭대학교 전자정보공학부 부교수
관심분야 : 운영체제, 분산시스템, 멀티미디어시스템(특히, 고성능 주문형 비디오 서버 설계), 이동 컴퓨팅, 전자상거래