Intern. J. Computer Math., 2003, Vol. 80(2), pp. 133–149

Taylor & Francis
Taylor & Francis Group

A PLACEMENT STRATEGY OF MULTIMEDIA OBJECTS BASED ON TEMPORAL RELATIONSHIPS

YIN-FU HUANG* and CHIH-CHIEN CHUANG

Institute of Electronics and Information Engineering, National Yunlin University of Science and Technology, 123 University Road, Section 3, Touliu, Yunlin, Taiwan 640, R.O.C.

(Received 7 February 2001; In final form 26 August 2001)

In an interactive multimedia information system, how to place multimedia objects on a disk to shorten the overall seek time for continuous retrieval is a critical issue. In the paper, based on the temporal relationships among media objects, we propose a placement strategy of multimedia objects to shorten the overall seek time for a multimedia presentation. First we derive the equations used to generate a storage pattern of media objects, and then propose an algorithm to place the objects according to the generated pattern. Next we use a de-clustering technique to support multimedia objects with high bandwidth requirements on multi-disks. Finally, a simulation model based on a practical disk is given, and several experiments are conducted to show the superiority of our algorithm.

Keywords: Multimedia information systems; Multimedia objects; Placement strategy; Seek time; Temporal relationships

C.R. Categories: H.3.2, H.5.1

1 INTRODUCTION

Due to the popularity of computers in recent years, Internet communication becomes a new trend, and World Wide Web (WWW) is now the most popular application on the Internet. While facing the upcoming multimedia world, more multimedia applications such as Video on Demand [2,11,12], Distance Learning etc., will be set up on WWW. For the data transmitted on the Internet, how to provide fast access is a critical issue, especially in an interactive multimedia information system.

The stored data in an interactive multimedia information system are mostly continuous media, such as videos, audio, animation, and images. Since their sizes are usually huge, they should be stored, accessed, and maintained efficiently [4,5]. The data management in the multimedia information system is not very similar to conventional data management. Multimedia data such as videos and audio are required to not only be presented very smoothly, but also be synchronized when two different types of media are displayed at the same time. These requirements have a close relationship with a data placement strategy

* Corresponding author. Tel: (+886)-5-5342601, Ext. 4314. Fax: (+886)-5-5312063.
E-mail: huangyf@el.yuntech.edu.tw

ISSN 0020-7160 print; ISSN 1029-0265 online © 2003 Taylor & Francis Ltd
DOI: 10.1080/0020716031000081 56

and the seek time of disks. Unless reasonable seek time is provided, the quality of services will not be satisfactory to users.

In the past, several data placement techniques were proposed to support continuous retrieval [3,8,13]. Though these data placement techniques expected to meet continuous retrieval by arranging data on an appropriate disk location, most of them only considered the parallel temporal relationships among media objects. To guarantee the continuous retrieval of multimedia objects, both parallel and sequential temporal relationships should be considered. Besides, since the bandwidth requirement of multimedia objects grows rapidly, a single disk cannot support the continuous retrieval of multimedia objects with high resolution and high bandwidth. Thus more researches have been addressing the data placement techniques on multi-disks [1,6,9,14,15].

In the paper, based on the temporal relationships among media objects [7,10], we propose a data placement strategy of multimedia objects to shorten the overall seek time for a multimedia presentation. First we derive the equations used to generate a storage pattern of media objects, and then propose an algorithm to place the objects according to the generated pattern. Next we use a de-clustering technique to support multimedia objects with high bandwidth requirements on multi-disks. Finally, we conduct several experiments to show the superiority of our algorithm.

The remainder of the paper is organized as follows. In Section 2, we describe the system model used in the paper. The temporal relationships among media objects and its corresponding directed connected graph are discussed in the section. In Section 3, we derive the equations for the storage pattern of media objects, and then propose the algorithm and de-clustering technique to place the objects on a single disk and multi-disks. In Section 4, a simulation model based on a practical disk is given, and several experiments are conducted to compare the placement strategies. Finally, we make conclusions in Section 5.

2 SYSTEM MODEL

In order to seize temporal relationships among media objects, a model called OCPN describing time dependencies and synchronization of media objects is adopted in our system. Then a corresponding directed connected graph transformed from an OCPN representation is proposed to facilitate our analysis. The problem explored here is how to place media objects on disks to shorten seek time, based on the temporal information in the directed connected graph.

2.1 OCPN Model

The OCPN (Object Composition Petri Net) proposed by Little and Ghafoor [10] is derived from Petri Net and can be used to represent the temporal relationships among media objects. Its definition is as follows:

$C_{\text{OCPN}} = \{T, P, A, D, R, M\}$

where

$T = \{t_1, t_2, \ldots, t_n\}$,

$P = \{p_1, p_2, \ldots, p_m\}$,

$A : \{T \times P\} \cup \{P \times T\} \rightarrow I, \quad I = \{1, 2, \ldots\}$,

$D : P \rightarrow R$,

$R : P \rightarrow \{r_1, r_2, \ldots r_k\}$, and

$M : P \rightarrow I, \quad I = \{0, 1, 2, \ldots\}$.

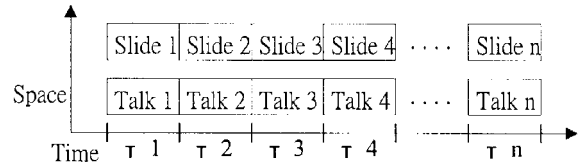

FIGURE 1 Slide presentation.

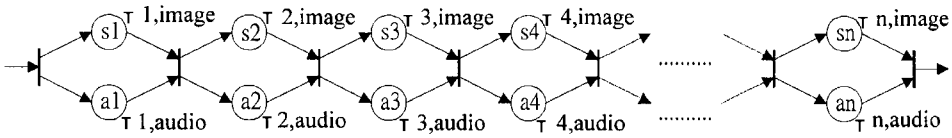

FIGURE 2 Description using OCPN.

T, *P*, and *A* represent a set of transitions (bars), a set of places (circles), and a set of directed arcs, respectively. *D*, *R*, and *M* are mapping functions from the set of places to the real numbers (durations), and from the set of places to a set of resources, and from the set of places to the integers, respectively. Here an example is taken to illustrate the definition as follows:

Example 1 Given a multimedia slide presentation that consists of a sequence of synchronized audio and visual objects with various duration, each slide is accompanied with a voice annotation during the presentation as shown in Figure 1. The presentation can be described using OCPN as shown in Figure 2.

2.2 Temporal Relationships

There could be seven distinct temporal relationships between two media objects, as shown in Figure 3. In the figure, two media objects are represented as P_α and P_β with their duration τ_α and τ_β, respectively, and τ_δ is the finite delay of temporal relationships between them. The OCPN representations for these temporal relationships are shown in Figure 4. In the figure, the finite delay of temporal relationship between two media objects is denoted as a virtual media object P_δ with the duration τ_δ.

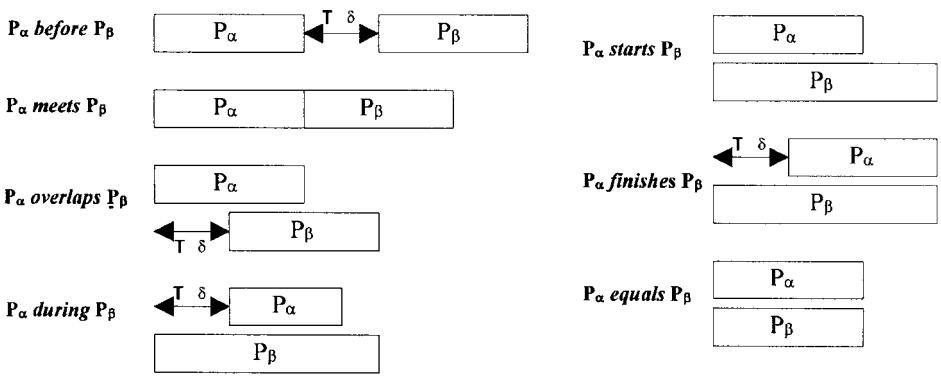

FIGURE 3 Seven temporal relationships.

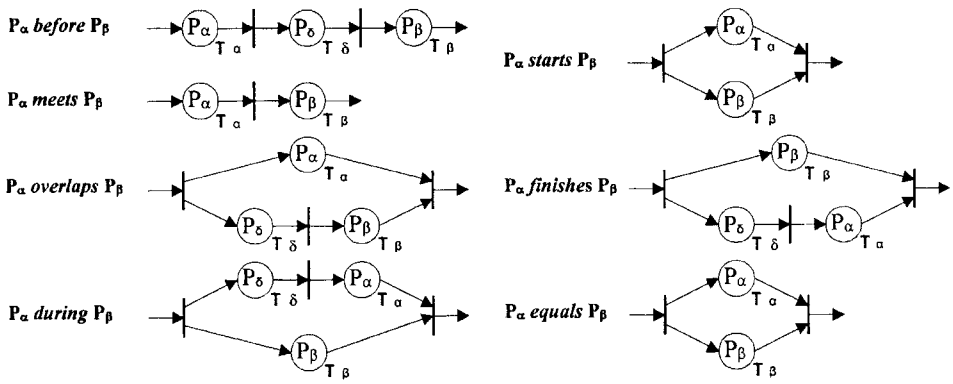

FIGURE 4 OCPN representations for seven temporal relationships.

2.3 Directed Connected Graphs

In our system model, the temporal relationships among media objects are expressed with OCPN. Then based on the OCPN representation, we transform it into a directed connected graph to facilitate our analysis. Here an OCPN representation and its corresponding directed connected graph are shown in Figure 5. A node in the directed connected graph represents a transition or a place in the OCPN representation. As shown in Figure 5(b), a shaded node

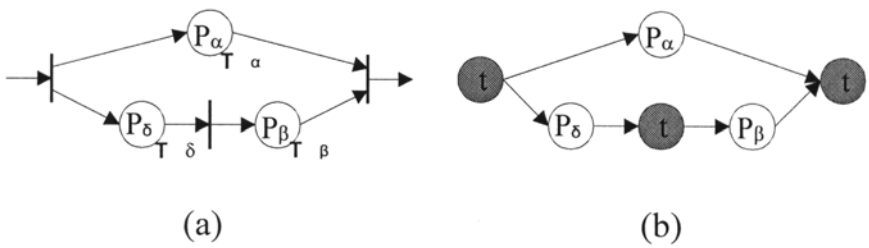

FIGURE 5 (a) OCPN representation. (b) Corresponding directed connected graph.

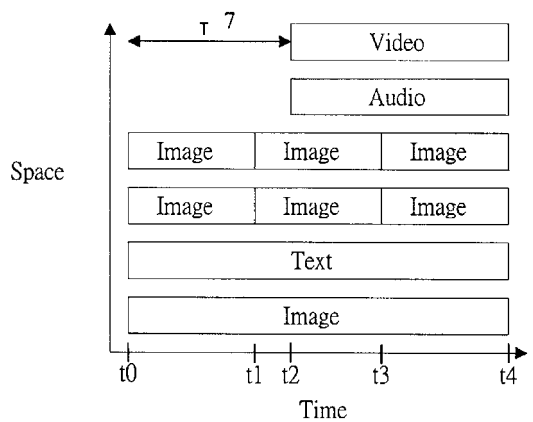

FIGURE 6 Multimedia presentation.

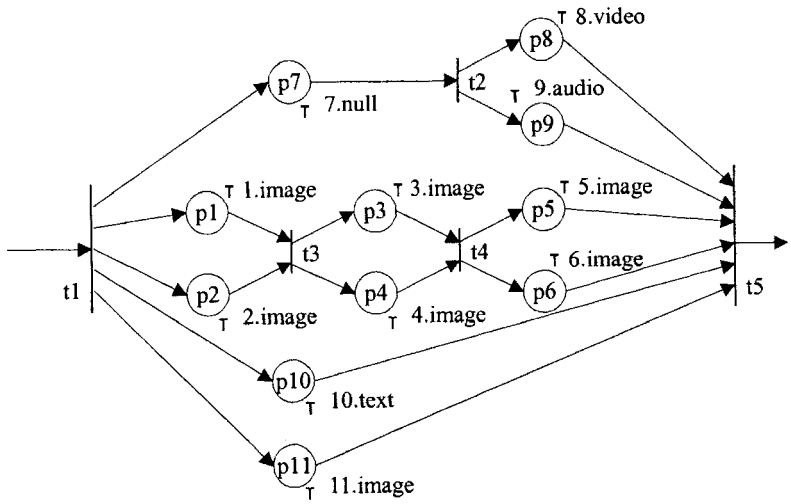

FIGURE 7 OCPN representation.

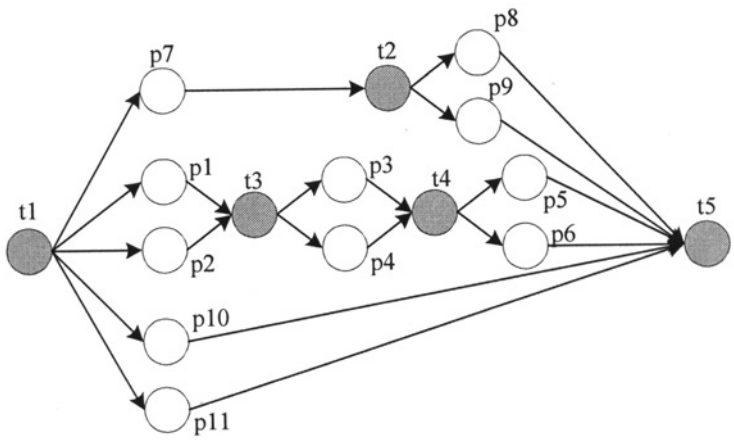

FIGURE 8 Corresponding directed connected graph.

and a blank one represent a transition and a place, respectively. Here an example is taken to illustrate the concepts as follows:

Example 2 Given a multimedia presentation as shown in Figure 6, its OCPN presentation and corresponding directed connected graph are shown in Figures 7 and 8.

3 DATA PLACEMENT STRATEGY

In this section, we propose placement strategies of multimedia objects on a single disk and multi-disks based on temporal relationships among them, respectively. Within the strategies, the equations used to define the storage patterns for data placement should be derived first. According to the storage patterns, we will understand how to place media objects on a disk to shorten seek time.

3.1 Storage Patterns

A storage pattern consists of several storage blocks that are referred to as the tracks of a disk in our model. Within a storage pattern, the number of blocks for each media object is different from each other, and is proportional to its consumption rate. Furthermore the storage patterns are not always fixed; they change dynamically with time. Here we classify media objects into two different types. One is dynamic media type such as video, audio, and animations; another is static media type such as images and texts. The equations used to define the storage patterns of both media types are different and described in the following subsections. The symbols used in the equations are outlined in Table I.

3.1.1 Storage Patterns of Dynamic Media Objects

The storage pattern of dynamic media objects consists of two parts, the media object area with size M and the gap area with size G. Both sizes are dependent on the data transfer rate of a disk and the playback rates of the media objects within the storage pattern. To display these continuous media smoothly, the data transfer rate of a disk should be faster than the playback rate of the media objects. By considering that one data block of a media object is read within one service cycle (*i.e.* $M = 1$), the condition for continuous retrieval can be formulated as follows:

$$\frac{1 + G_i}{\rho} \leq \delta_i \quad (i = 1, 2, \ldots, n) \tag{1}$$

The gap size that satisfies the condition for continuous retrieval can be derived and formulated as follows:

$$\begin{aligned} &\because \frac{1 + G_i}{\rho} \leq \delta_i \\ &\Longrightarrow G_i \leq \delta_i \times \rho - 1 \\ &\therefore G_i = \lfloor \delta_i \times \rho \rfloor - 1 \end{aligned} \tag{2}$$

TABLE I Symbols used in the equations.

Symbol	*Description*	*Unit*
ρ	The data transfer rate of a disk	blocks/sec
δ_i	Display time of one block of media object i	second
G_i	The gap size in the storage pattern for media object i	blocks
M_i	The data size of media object i in a merged storage pattern	blocks
N	The total blocks of a cylinder	
F	The moving frequency of a disk head	
T_{seek}	The seek time of cylinder-to-cylinder	second
T_d	The duration of a unit in an OCPN representation	second
S_m	The size of a media object	bytes
S_b	The size of a single block	bytes
B_m	The number of blocks used to store a media object	blocks
B_d	The number of blocks read during T_d	blocks

FIGURE 9 Storage pattern of each media object.

Using Eq. (2), the storage pattern of each media object can be represented as shown in Figure 9.

Since there is not always only one media object within one service cycle, we must merge several storage patterns as shown in Figure 9 into a real storage pattern. The media objects displayed within one service cycle should have some correlation, such as temporal relationships mentioned in Section 2. The goal of merging is to place the related media objects as close as possible, and the overall disk seek time will be shortened. The merging of storage patterns of related media objects could be formulated as follows:

$$G_{\max} : \text{Max}\ \{G_i, i = 1, 2, \ldots, n\}$$

$$M_i = \left\lceil \frac{1 + G_{\max}}{1 + G_i} \right\rceil \quad (i = 1, 2, \ldots, n) \tag{3}$$

$$G' = (1 + G_{\max}) - \sum_{i=1}^{n} M_i \tag{4}$$

Suppose media object 1 has the maximum storage pattern among all related media objects, $G_{\max}$ will be equal to G_1. Then after merging all storage patterns of related media objects, we obtain the storage pattern as shown in Figure 10.

When all blocks in the same cylinder are retrieved completely, the disk arm will move to the next cylinder and continues to retrieve data. This latency should be included in the merged storage pattern to refine the gap size. After calculating the moving frequency of a disk head within a service cycle, we adjust the merged storage pattern as shown in Figure 11. The adjustment is to detect whether a storage pattern is too late to service all embedded media objects within a service cycle. If it is (*i.e.* $G'' < 0$), some media objects not critical in the presentation have to be removed from the storage pattern to fulfill the service cycle.

$$F = \left\lfloor \frac{\sum_{i=1}^{n} M_i + G'}{N} \right\rfloor \tag{5}$$

$$G'' = G' - F \times T_{\text{seek}} \times \rho \tag{6}$$

Finally the adjustment on the sizes of the dynamic media objects in a storage pattern should be taken into consideration when one of the media objects has finished its presentation. Two situations for the adjustment are shown in Figure 12. At time t_1 as shown in Figure 12(a),

FIGURE 10 Merged storage pattern of related media objects.

FIGURE 11 Merged storage pattern including the seek overhead.

if the remainder size of Video 1 called k is not large enough for a full cycle (*i.e.* $B_{m(k)} < M_k$), then the size of Audio in the same storage pattern should be adjusted based on the formula as shown below. After time t_1, new storage patterns will be generated continuously for Audio and Video 2.

$$M_i = \left\lceil \frac{B_{m(k)} \times (1 + G_k)}{1 + G_i} \right\rceil \quad \forall i \tag{7}$$

Besides, at time t_1 as shown in Figure 12(b), if the remainder display time of Image 1 called l is shorter than a full cycle (*i.e.* $B_{d(l)} < (1 + G_{\max})$), then the sizes of Video and Audio in the storage pattern should be adjusted based on the formula as shown below. After time t_1, new storage patterns will be generated continuously for Video, Audio, and Image 2.

$$M_i = \left\lceil M_i \times \frac{B_{d(l)}}{1 + G_{\max}} \right\rceil \quad \forall i \tag{8}$$

3.1.2 Storage Patterns of Static Media Objects

It is not necessary to take into account the condition of continuous retrieval when generating the storage pattern of static media objects. The only thing we must consider is the size of static media objects. The storage pattern of related static media objects is the combination of themselves, and the position of each media object in the storage pattern is dependent on its duration T_d. The shorter T_d a media object has, the higher placement priority it will

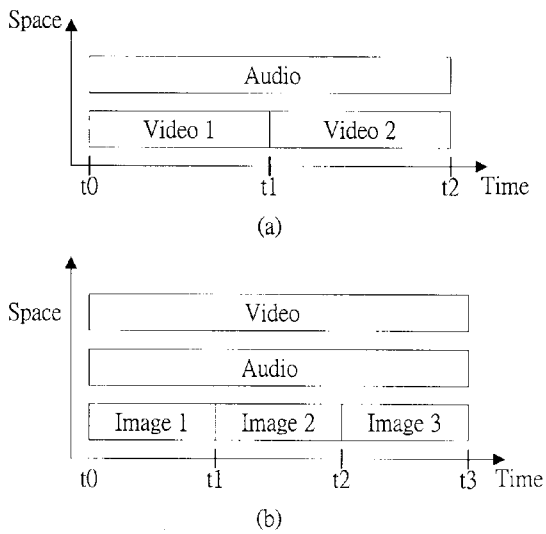

FIGURE 12 Two situations for adjusting a storage pattern.

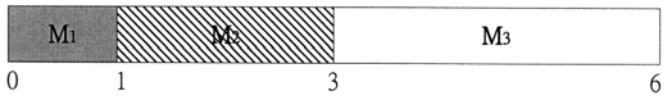

FIGURE 13 Storage pattern of static media objects.

have in the storage pattern. The reason is to avoid the media object with a shorter T_d being too late to be seen in the screen. The storage pattern of static media objects with different sizes is shown in Figure 13.

3.2 Algorithm for a Single Disk

For the seven temporal relationships mentioned in Section 2.2, we can classify them into two types, namely sequential relationships and parallel relationships. In the data placement algorithm for a single disk, we find out the media objects are related to each other in a parallel relationship and generate their storage patterns based on the equations defined in Section 3.1. Here the directed connected graph transformed from an OCPN representation and two parameters B_m and B_d of media objects are employed in the algorithm. These two parameters can be formulated as follows:

$$B_m = \frac{S_m}{S_b} \tag{9}$$

$$B_d = T_d \times \rho \tag{10}$$

In short, the procedure of the algorithm is as follows. At first, we have an initialization and then use a searching procedure to look for the media objects at a specific time point. According to the information of the searched media objects, we generate their storage patterns. The storage patterns could be adjusted as described in step 8 when the related cycle is not a full one. Also they could be merged together as described in step 10 when they are generated from different media types (*i.e.* dynamic and static ones). The procedure keeps on running for each advanced time point till the storage patterns of all the media objects are generated.

The details of the data placement algorithm for a single disk are formally given as follows:

/* t_{start}: the current starting node,
P_d: the set of dynamic media objects,
P_s: the set of static media objects,
D: the set of delay nodes,
T: the set of transition nodes,
Full_cycle: full cycle flag */

Step 1: Let t_{start} be t_1, and initialize P_d, P_s, D, and T being empty sets.
Step 2: Call ***Procedure Searching***.
Step 3: Repeat from step 4 to step 12 till P_d, P_s, D, and T are all empty.
Step 4: If P_d is empty, then
- *Step 4.1:* Generate the storage pattern according to the nodes with $B_m \neq 0$ in P_s, and then reset their B_m's to zero.
- *Step 4.2:* Get the minimum B_d (*i.e.* $B_{d(\min)}$) among all the nodes in P_s and D.
- *Step 4.3:* $B_d = B_d - B_{d(\min)}$, for all the nodes in P_s and D.

Step 5: If P_d is not empty, then
- *Step 5.1:* Full_cycle = on
- *Step 5.2:* Generate the storage pattern according to all the nodes in P_d, and repeat from step 6 to step 11 till one of the following conditions occurs:
  1. At least one node with $B_m = 0$ exists in P_d.
  2. At least one node with $B_d = 0$ exists in P_s or D.

Step 6: If at least one node with $B_{m(i)} < M_i$ exists in P_d, then
- *Step 6.1:* Full_cycle = off
- *Step 6.2:* Select the node, say k, with the minimum $B_{m(k)} \times (1 + G_k)$.

Step 7: If at least one node with $B_{d(i)} < (1 + G_{\max})$ exists in P_s or D, then
- *Step 7.1:* Full_cycle = off
- *Step 7.2:* Select the node, say l, with the minimum $B_{d(l)}$.

Step 8: If $B_{m(k)} \times (1 + G_k) < B_{d(l)}$, then adjust the sizes of the dynamic media objects in the storage pattern, based on $B_{m(k)}$.

$$M_i = \left\lceil \frac{B_{m(k)} \times (1 + G_k)}{1 + G_i} \right\rceil \quad \forall i$$

else adjust the sizes of the dynamic media objects in the storage pattern, based on $B_{d(l)}$.

$$M_i = \left\lceil M_i \times \frac{B_{d(l)}}{1 + G_{\max}} \right\rceil \quad \forall i$$

Step 9: Let $B_{m(i)} = B_{m(i)} - M_i \ \forall i \in P_d$

Step 10: If P_s is not empty, then
- *Step 10.1:* Generate the storage pattern according to the nodes with $B_m \neq 0$ in P_s, and then reset their B_m's to zero.
- *Step 10.2:* Merge the storage pattern into that generated in *Step 5* or *Step 8*; that is the storage pattern for static media objects first, and then that for dynamic media objects.

Step 11: If (Full_cycle = on)
then Let $B_{d(i)} = B_{d(i)} - (1 + G_{\max}) \quad \forall i \in P_s \cup D$
else Let $B_{d(i)} = B_{d(i)} - B_{m(k)} \times (1 + G_k)$
or $B_{d(i)} = B_{d(i)} - B_{d(l)} \quad \forall i \in P_s \cup D$
according to the results of *Step 8*.

Step 12: Call ***Procedure Examination***.

/* Procedure Searching */

Step 1: For the ***directed connected graph***, search all the first nodes in the paths originated from t_{start}.

Step 2: If the node is a ***transition***, then insert it into ***T***.

Step 3: If the node is a ***dynamic*** or ***static*** media object, then insert it into $\boldsymbol{P_d}$ or $\boldsymbol{P_s}$, respectively.

Step 4: If the node is a ***delay***, then insert it into ***D***.

/* Procedure Examination */

Step 1: Repeat from step 2 to step 4 till none of the following conditions occurs:
1. At least one node with $B_m = 0$ exists in P_d.
2. At least one node with $B_d = 0$ exists in P_s or D.
3. T is not an empty set.

Step 2: If a node in P_d, say k, has $B_m = 0$, then
  Step 2.1: Let node k be t_{start}, and then delete it from P_d.
  Step 2.2: Call ***Procedure Searching***.
Step 3: If a node in P_s or D, say k, has $B_d = 0$, then
  Step 3.1: Let node k be t_{start}, and then delete it from P_s or D.
  Step 3.2: Call ***Procedure Searching***.
Step 4: If T is not an empty set, then
  Step 4.1: Select a node as t_{start}, and then delete it from T.
  Step 4.2: Call ***Procedure Searching***.

Here we take the directed connected graph as shown in Figure 8 as an example, which is transformed from the multimedia presentation as shown in Figure 6. The processes of the data placement algorithm and the generated storage patterns are shown in Figure 14. In iteration 1, the first storage pattern including the static media objects p_1, p_2, p_{10}, and p_{11} is generated at time t_0 as shown in Figure 8. In iteration 2, the second storage pattern also including the static media objects p_3 and p_4 is generated at time t_1. In iteration 3, the third storage pattern including the dynamic media objects p_8 and p_9 (*i.e.* video and audio) is generated at time t_2. During iteration 3, multiple storage patterns including the media objects p_8 and p_9 are generated continuously till time t_3. Since the last cycle immediately before time t_3 is not a full one, its storage pattern should be adjusted; that is the sizes of the media objects within the storage pattern should be reduced as described in step 8 of the algorithm. In iteration 4, two storage patterns, one for the dynamic media objects p_8 and p_9 and another for the static media objects p_5 and p_6, are generated at time t_3; then these two storage patterns are merged together as shown in Figure 14. Similar to iteration 3, multiple storage patterns including the media objects p_8 and p_9 are generated continuously during iteration 4 till time t_4 (*i.e.* the presentation is finished). Finally, according to the generated storage patterns, the placement of the media objects on a disk is shown in Figure 15. For simplicity, the gap field of each storage pattern is omitted here.

3.3 De-clustering Technique for Multi-Disks

In general, a single disk cannot support the continuous retrieval of multimedia objects with high resolution, such as a video object with HDTV (High Definition Television) quality

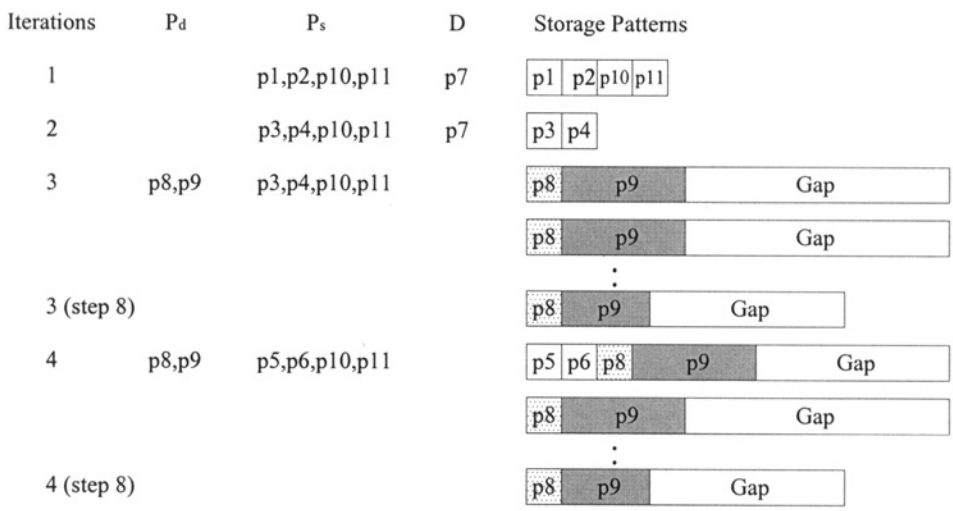

FIGURE 14 The generation of storage patterns.

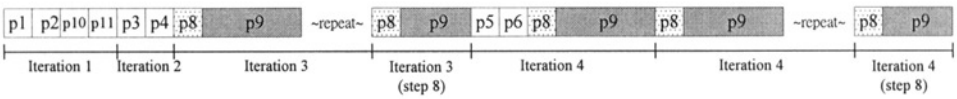

FIGURE 15 The physical storage organization of a disk.

requiring approximately 860 Mb/s bandwidth. Thus we have to extend the algorithm from a single disk to multi-disks to meet the high bandwidth requirement of multimedia objects. In a multi-disks environment, we first generate storage patterns according to the aggregated bandwidth of multiple disks. Thus Eq. (1) derived in Section 3.1.1 has to be modified as follows:

$$\frac{1 + G_i}{\rho_t} \leq \delta_i \tag{11}$$

$$\rho_t = \rho_1 + \rho_2 + \cdots + \rho_n$$

where ρ_t is the aggregated bandwidth of all disks. With the Eq. (11), we can generate the storage patterns for multi-disks, similar to those for a single disk.

After a storage pattern is generated, we de-cluster it across all disks in a round-robin manner. The number of blocks placed on each disk in a round is based on the ratio of its disk bandwidth to the minimum disk bandwidth. It can be formulated as follows:

$$\rho_{\min} : \text{Min}\{\rho_i, \quad i = 1, 2, \ldots, n\}$$

$$B_{di} = \left\lfloor \frac{\rho_i}{\rho_{\min}} \right\rfloor \quad (i = 1, 2, \ldots, n) \tag{12}$$

Here we take an example to illustrate the concepts. Given five disks and their B_{di}'s 1, 2, 2, 2, 1, a storage pattern is de-clustered across the five disks as shown in Figure 16.

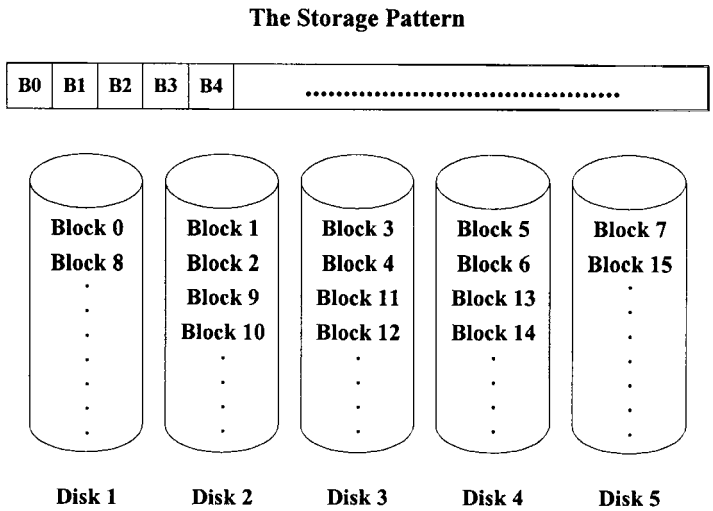

FIGURE 16 The storage pattern de-clustered across five disks.

4 PERFORMANCE SIMULATION

In this section, we show the performance simulation of our algorithm and other placement strategies. In the experiments, the performance of these algorithms is measured based on total seek time. First we describe the simulation model used in the experiments, and then show the experimental results.

4.1 Simulation Model

Here Seagate Cheetah 18LP-ST39103LC is simulated in the experiments. Its parameters are shown in Table II. The total seek time measured in the experiments is based on the parameter track-to-track seek. In general, the total seek time consists of the seek time of moving disk head to the initial track and the seek time of crossing track during data retrieval.

To show the superiority of our algorithm, we compare it with three placement strategies. The first one is Chen and Little's placement [3] that also considers both parallel and sequential temporal relationships among media objects, but more than one media object could be within one track. The second one is Rangan and Vin's placement [13] that only considers the parallel temporal relationships among media objects. If the multimedia objects are presented at the same moment, they will be merged together and placed on disks as closely as possible; otherwise, they are placed on disks randomly. The last one is the random placement that places multimedia objects on a disk randomly and does not consider the temporal relationships among media objects. In addition, we also explore the influences resulting from the ratio of different multimedia object types in a presentation. Thus three different ratios of static multimedia objects to dynamic multimedia objects are defined for nine cases in each experiment, as shown in Table III.

4.2 Experimental Results

Experiment 1: Different presentations

In the experiment, the number of multimedia objects within a presentation is 20 for each case. The average sizes of dynamic and static multimedia objects are 1000 and 5 tracks, respectively. To facilitate the observation, the scale of the y-axis is represented in logarithm with base 10. As shown in Figure 17, the total seek time of our algorithm is the least among all placement strategies. The reason is that our algorithm considers both parallel and sequential temporal relationships among media objects, thereby placing them as closely as possible on a disk. Chen's placement performs a little worse than our algorithm, since more tracks are required on a disk, thereby incurring the combination overhead. As for Rangan's placement, it only considers the parallel temporal relationships among media objects, and thus performs worse than ours and Chen's.

TABLE II The parameters of Seagate Cheetah 18LP-ST39103LC

Model number	ST39103LC
Model name	Cheetah 18LP
Capacity	9.1 Gbytes
Average transfer rate	24.5 Mbytes/s
Average seek time	5.2 ms
Trace-to-track seek	0.6 ms
Disks	3
Heads	6
Total tracks	117,612

TABLE III The ratio of static objects to dynamic objects in nine cases

Cases	*Ratios (static:dynamic)*
Case 1 ~ 3	1:1
Case 4 ~ 6	7:3
Case 7 ~ 9	3:7

Experiment 2: Different object sizes

The experiment explores which placement strategy will become worse more rapidly when object sizes are increased. Here we double the average size of dynamic multimedia objects (*i.e.* 2000 tracks). The other factors remain the same as Experiment 1. As shown in Figure 18, the total seek time is definitely affected by object sizes. However, our algorithm is less sensitive to object sizes than the other three placement strategies.

Experiment 3: A multi-disks environment

The experiment explores the performance of different placement strategies in a multi-disks environment. Each placement strategy consists of storage pattern generation techniques and de-clustering techniques. The storage pattern generation techniques considered here are ours, Chen's [3], Rangan's [13], and random, whereas the de-clustering techniques are round-robin (RR) and random. Since PWM [15] was proposed based on Rangan's placement in a multi-disks environment, it is applied directly in the experiment. We have three disks in the experiment and their disk bandwidths are 1:2:2. The average sizes of dynamic and static multimedia objects are 500 and 10 tracks, respectively. The other factors remain the same as Experiment 1.

Since the experimental results are clustered into two groups, we depict them in two figures. As shown in Figures 19(a) and 19(b), the total seek time of ours and Chen's is much less than

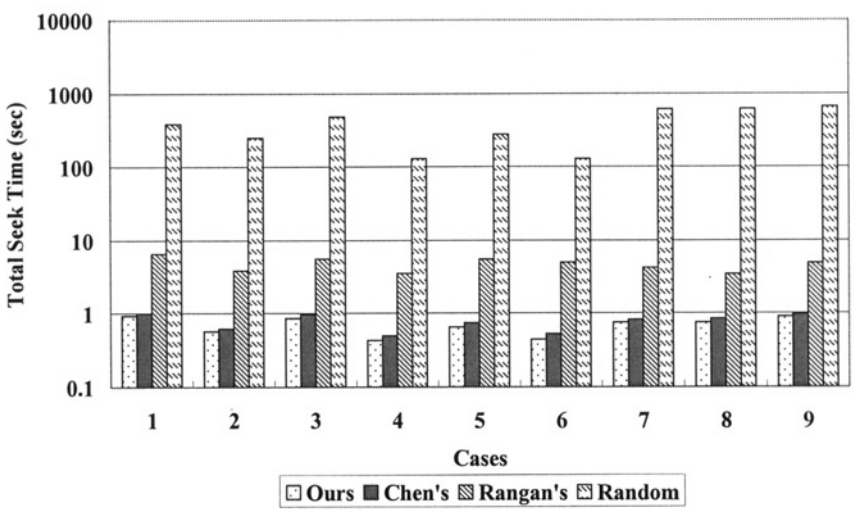

FIGURE 17 Total seek time for different presentations.

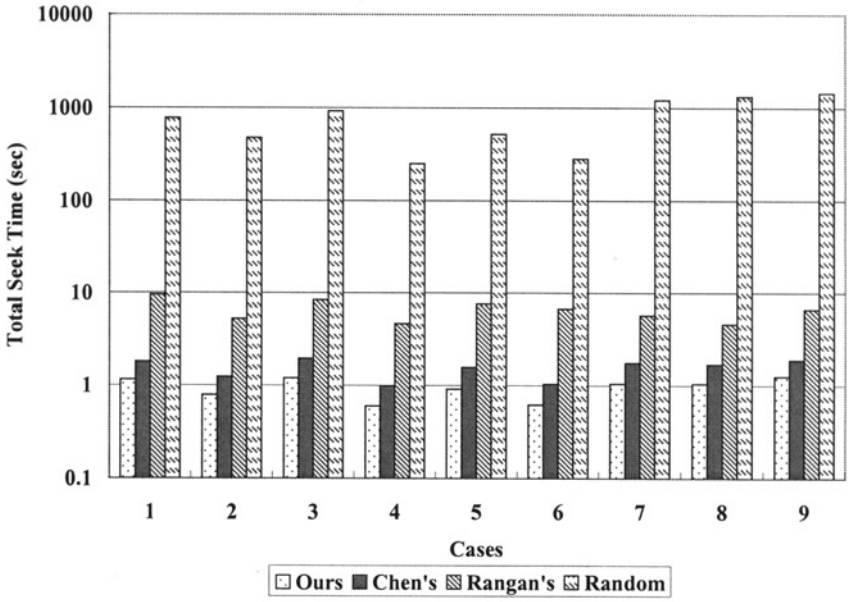

FIGURE 18 Total seek time for different object sizes.

the other two placement strategies, since our and Chen's placement consider both parallel and sequential temporal relationships among media objects. We also find that if temporal relationships are considered when generating storage patterns, the total seek time is not apparently different for different de-clustering techniques.

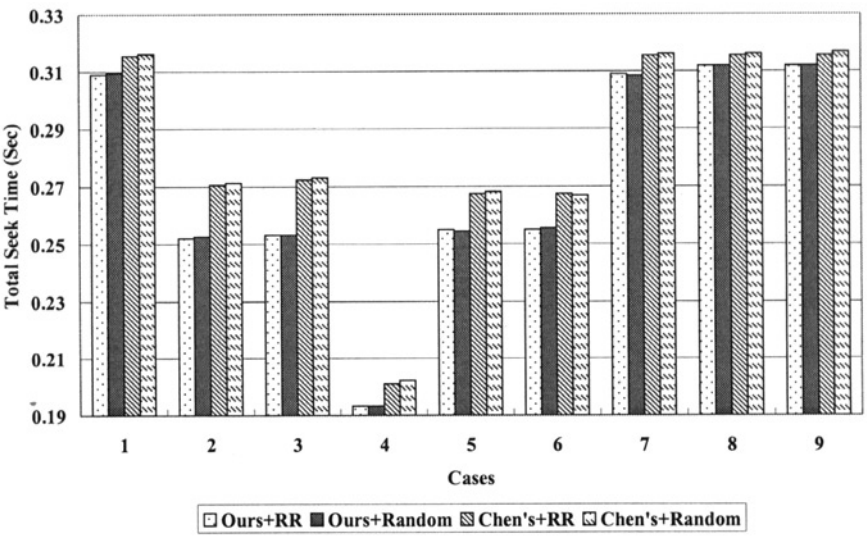

FIGURE 19(a) Total seek time in a multi-disks environment.

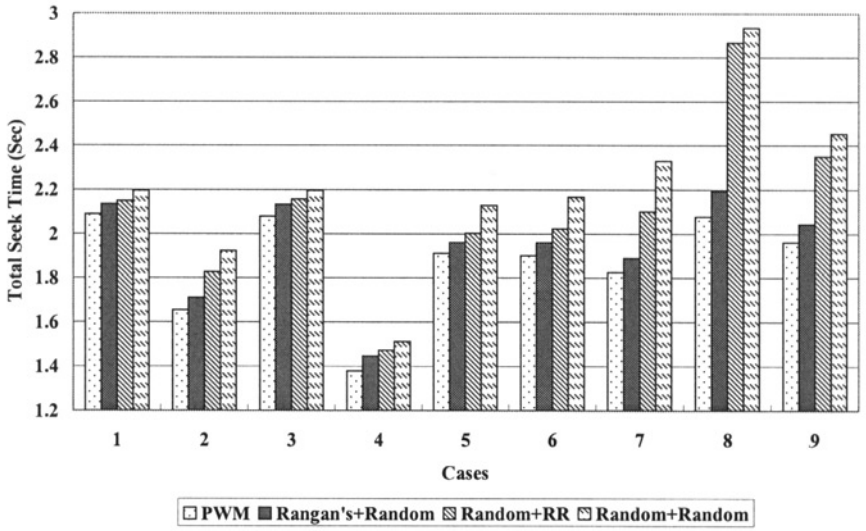

FIGURE 19(b) Total seek time in a multi-disks environment.

5 CONCLUSIONS

In an Interactive Multimedia Information System (IMIS), how to place multimedia objects on a disk to shorten the overall seek time for continuous retrieval is a critical issue. In the paper, we propose placement strategies of multimedia objects on a single disk and multi-disks based on temporal relationships among them. Through simulation using a practical disk model, we found that the total seek time of our algorithm is less than the other three placement strategies. Our algorithm is also less sensitive to object sizes than the other three placement strategies. Besides we also find that if temporal relationships are considered when generating storage patterns, the total seek time is not apparently different for different de-clustering techniques.

References

[1] Berson, B., Ghandeharizadeh, S., Muntz, R. and Ju, X. (1994) Staggered striping in multimedia information systems. *Proc. ACM International Conference on Management of Data*, pp. 79–90.
[2] Chen, H.J., Krishnamurthy, A., Venkatesh, D. and Little, T.D.C. (1995) A scalable video-on-demand service for the provision of VCR-like functions. *Proc. Second IEEE International Conference on Multimedia Computing and Systems*, pp. 65–72.
[3] Chen, H.J. and Little, T.D.C. (1996) Storage allocation policies for time-dependent multimedia data. *IEEE Transactions on Knowledge and Data Engineering* **8**(5), 855–864.
[4] Christodoulakis, S. and Zioga, F.A. (1999) Database design principles for placement of delay-sensitive data on disks. *IEEE Transactions on Knowledge and Data Engineering* **11**(3), 425–447.
[5] Gemmell, J. and Christodoulakis, S. (1992) Principles of delay-sensitive multimedia data storage and retrieval. *ACM Transactions on Information Systems* **10**(1), 51–90.
[6] Grandeharizadeh, S. and Ramos, L. (1993) Continuous retrieval of multimedia data using parallelism. *IEEE Transactions on Knowledge and Data Engineering* **5**(4), 658–669.
[7] Hamblin, C.L. (1972) Instants and intervals. *Proc. First Conference International Society for the Study of Time*, pp. 324–331.
[8] Kwon, T.G. and Lee, S. (1995) Data placement for continuous media in multimedia DBMS. *Proc. International Workshop on Multi-Media Database Management Systems*, pp. 110–117.

[9] Lee, C.I., Chang, Y.I. and Yang, W.P. (1996) A new storage and retrieval method to support editing operations in a multi-disk-based video server. *Proc. International Conference on Parallel and Distributed Systems*, pp. 132–141.
[10] Little, T.D.C. and Ghafoor, A. (1990) Synchronization and storage models for multimedia objects. *IEEE Journal on Selected Area in Communication* **8**(3), 413–426.
[11] Lougher, P. and Shepherd, D. (1992) The design and implementation of a continuous media storage server. *Proc. Third International Workshop on Network and Operating System Support for Digital Audio and Video*, pp. 63–74.
[12] Rangan, P.V., Vin, H.M. and Ramanathan, S. (1992) Designing an on-demand multimedia service. *IEEE Communications*, 56–64.
[13] Rangan, P.V. and Vin, H.M. (1993) Efficient storage techniques for digital continuous multimedia. *IEEE Transactions on Knowledge and Data Engineering* **5**(4), 564–573.
[14] Tsai, W.J. and Lee, S.Y. (1994) Storage design and retrieval of continuous multimedia data using multi-disks. *Proc. International Conference on Parallel and Distributed Systems*, pp. 148–153.
[15] Watkins, A.R. and Omiecinski, E. (1995) Storing continuous media objects using parallelism with merging, *Proc. 19th International Conference on Computer Software and Applications*, pp. 339–345.

Copyright © 2003 EBSCO Publishing

Copyright of International Journal of Computer Mathematics is the property of Taylor & Francis Ltd and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.