Intelligent Prefetching and Buffering for Interactive Streaming of MPEG Videos

Susanne Boll, Christian Heinlein, Wolfgang Klas, Jochen Wandel
Databases and Information Systems (DBIS)
Computer Science Department, University of Ulm, Germany
{boll,heinlein,klas,wandel}@informatik.uni-ulm.de

ABSTRACT

Continuous delivery of media streams like video over IP networks so far is mainly handled by commercial approaches that deliver the stream forward-oriented in their own proprietary format. Though some existing streaming technologies are able to adapt to varying bandwidths, they do not provide smooth reactions to user interactions with the continuous stream.

We have developed the *MPEG-L/MRP* strategy, an adaptive prefetching algorithm for the MPEG-1 video format in combination with an intelligent buffering technique that allows for smooth and quick reactions to user interactions with the stream. With L/MRP [12] an approach already has been presented to deliver and buffer homogeneous continuous data streams like Motion-JPEG with special focus on fast reaction to user interactions. In contrast, the MPEG-1 encoding with its different frame types and the dependencies between frames opens the door to a more fine-grained adaptation of the continous stream. However, the complexity of MPEG-1 calls for comprehensive adaptation and special amendments of the L/MRP algorithm to make it an efficient preloading and buffering technique for MPEG-1 videos.

With the realization of *MPEG-L/MRP* in the context of a multimedia presentation engine on top of a multimedia repository we have an efficient means to deliver continuous streams of interactive multimedia presentations over existing IP infrastructure trying to minimize interaction response time and optimize loading/reloading portions of a video stream.

1. INTRODUCTION

In future, users of multimedia applications will no longer be satisfied with pre-packed presentations on stand-alone systems or proprietary compositions embedded in Web pages and rendered by browser plug-ins. Rather, personalized interactive multimedia presentations are needed, delivered on-demand from a multimedia server over an IP network to a user's flexible presentation environment. In this context, the delivery of continuous multimedia data as well as its presentation must be tailored to the specific requirements of this environment, i. e., the varying bandwidth, response time of the server, and the like.

The motivation of our work in the area of continuous delivery of interactive multimedia presentations over a network stems from our research project "Gallery of Cardiac Surgery" (Cardio-OP[1]) [8] which aims at developing an Internet-based and database-driven multimedia information system in the domain of cardiac surgery. The users of the system request multimedia content from different platforms over different network connections. Video streams are of high importance in this educational environment. During the learning process, it is indispensable for the user to interact on the stream so as to watch a scene again or jump to another interesting part of the video. Therefore the system must support interactions and, to be user-friendly, should react in a very responsive way. Hence, the presentation environment demands for streaming support for continuous media with suitable handling of user interactions.

In the project context, we developed a multimedia presentation engine which includes support for continuous MPEG video streams. For this, we developed the *MPEG-L/MRP* algorithm to continuously deliver MPEG-1 video streams over an IP network which we present in this paper.

Compared with, e. g., Motion-JPEG, the encoding of continuous video streams with MPEG-1 offers a significantly higher compression rate which is very important for a delivery over a network with potentially low bandwidth. We aim at continuously delivering the MPEG-1 stream in small units and at buffering these units in an intelligent way at the client such that the user is provided with a smooth and continous presentation though the user can possibly carry out VCR-like interactions on the stream like fast forward, reverse, or jumping to a bookmark in the video. The buffering technique should hide the request and buffering of units and rather deliver a continuous MPEG-stream of the best quality that can be currently provided to the application.

With *L/MRP* [12] we find a preloading and buffering strategy for continuous streams supporting interactions that has proven to perform better than "traditional" strategies like, e. g., LRU, FIFO, LFU, etc.This approach, however, aims at delivering and buffering *homogeneous* continuous data streams like Motion-JPEG with special focus on fast reaction to user interactions. The complexity of MPEG-1

[1] Partially funded by the German Ministry of Research and Education, grant number 08C58456. Our project partners are the University Hospital of Ulm, Dept. of Cardiac Surgery and Dept. of Cardiology, the University Hospital of Heidelberg, Dept. of Cardiac Surgery, an associated Rehabilitation Hospital, the publisher Hüthig-Verlag, Heidelberg, FAW Ulm, and ENTEC GmbH, St. Augustin. For details see also URL http://www.informatik.uni-ulm.de/dbis/Cardio-OP/

with its heterogeneous frame types of different importance, varying frame sizes, and inter-frame dependencies calls for comprehensive adaptation and special amendments of the original L/MRP algorithm to make it an efficient preloading and buffering technique for MPEG-1 videos. This paper presents our MPEG-1 specific preloading and buffer management strategy *MPEG-L/MRP* for MPEG-1 videos.

The remainder of this paper is organized as follows: Section 2 discusses related work. Section 3 revisits the original L/MRP algorithm and gives a short overview of the parts of MPEG-1 relevant to our approach. In Section 4, our new MPEG-L/MRP approach is presented which consists of a formal model and a corresponding algorithm. Section 5 sketches the implementation of the approach and Section 6 concludes the paper.

2. RELATED WORK

Related work, concerned with the delivery of multimedia content over the Internet, covers several research approaches dealing with the adaptive streaming of MPEG videos. As a part of the QUASAR project at the Oregon Graduate Institute [19] an MPEG player for adaptive MPEG streaming over the Internet has been developed which addresses resource scarceness in the end-to-end delivery. The focus lies on a quality of service (QoS) model and an adaptation mechanism of the player. To facilitate adaptive streaming, the MPEG video is provided by the server in different qualities. The stream is adapted in the temporal dimension by dropping B frames first, then P frames, and finally I frames. In addition, different spatial resolutions are provided as a second variable quality dimension. Buffering is applied to compensate network jitter but does not support fast reactions to user interactions. Another approach, the Media Streaming Protocol [4] developed at the University of Illinois, provides adaptive streaming of MPEG movies, too. On congestion, the protocol considers the different frame types of MPEG with their frame interdependencies and, similar to our approach, drops less important MPEG frames first. The client side buffer is employed only to smooth the jitter of arriving data but does not allow for minimizing interaction response time and reload of data after possible user interactions.

In the commercial area, many approaches can be found that deal very well with the streaming of videos, e. g., Quicktime [1] or Emblaze [2]. With VDOLive [17] and Real [14] approaches exist, that are furthermore able to adapt the video stream to fluctuations of the available bandwidth. For instance, with the introduction of the SureStream technology [15], Real allows to encode a video clip that serves for up to six different bandwidths. This stream can automatically be adjusted to compensate for network congestions. However, as this technique encodes multiple disjoint streams into one file, it leads to an inflation of the storage size and to redundancy. However, all the commercial approaches mentioned have in common that they operate on proprietary video formats and are neither designed to support minimization of the interaction response time nor to optimize the effort for reloading portions of a video stream.

With Q-L/MRP [3] an interesting application of L/MRP has evolved. Q-L/MRP extends L/MRP with additional interaction sets in order to support the specific QoS requirements of certain users. However, the approach does not deal with MPEG specific preloading and replacement strategies.

3. L/MRP AND MPEG-1 REVISITED

3.1 L/MRP

L/MRP (**L**east/**M**ost **R**elevant for **P**resentation) [12] is a buffer management strategy for interactive continuous data flows in a client/server environment. The client requests and receives a continuous medium in small units and buffers that part of the stream that is relevant for the current and future presentation. The main idea is to request, preload, and buffer those units that are *most relevant* to be presented in the near future. The speciality of the L/MRP strategy here is that the preloading and buffering takes into account the interactions a user possibly carries out on the stream, e. g., switch to fast forward playback or jump to a bookmark. By that means, the interaction response time compared to common buffer management and replacement strategies is reduced (cf. [12]). *Preloading* and *replacement* are the two tasks the buffer management strategy has to master. During preloading the next most relevant units of the continuous stream are determined, whereas the replacement strategy must decide which are the least relevant units as these are removed from the buffer to free space for more relevant units.

The L/MRP buffer management strategy treats the stream as a sequence of so called *Continuous Object Presentation Units (COPUs)* with an ascending numbering of the units. Looking at a sequence of COPUs from a specific presentation point p in time, the single COPUs are differently relevant for the current presentation which is expressed by assigning relevance values to each COPU. Consider Figure 1 for an illustration: The current presentation point is $p = 43$ and the user is watching the stream at double speed in forward direction. Then, every other COPU in forward direction close to the current presentation point is absolutely relevant for the upcoming presentation. These COPUs form the so called *referenced* set, as they are likely to be referenced in the near future. However, there are COPUs that already have been viewed. These belong to the *history* set of COPUs of the stream. As a user could change the direction of the playout at any time, these COPUs are still relevant for the presentation. Finally, the frames in forward direction which are *skipped* due to the double speed playout, are relevant, too, as the user could switch to normal speed playback at any time. These considerations can be continued for further interaction types such as fast backward, jumping to bookmarks, and the like.

The relevance of a COPU with respect to one of these sets is determined by a so called *distance relevance function* which expresses a COPU's relevance as a function of the distance of the COPU to the current presentation point p. For the referenced set, the distance relevance function is monotonously decreasing with value 1 for the next few COPUs to be presented. As the frames of the history and skipped sets are less likely to be presented, their distance relevance functions are decreasing more rapidly. Given one or more relevance functions for each COPU, an overall relevance function can be calculated, e. g., by taking the maximum relevance value for each COPU. This global relevance function is then used by the preloading and replacement of the buffer. The relevance value expresses which COPUs are likely to be presented when taking into account the different interactions a user could perform on the stream. L/MRP tries to keep those most relevant COPUs in the client buffer to achieve a quick and smooth reaction to the user interaction. Depending on the buffer size those COPUs above a certain relevance value are kept in the buffer and those below the threshold value are not loaded/are removed from the buffer to make room for the more/most relevant COPUs. Whenever the presentation point p proceeds, the relevance values are recalculated, the COPUs to be preloaded are determined and the COPU(s) with the least relevance value in the buffer are replaced.

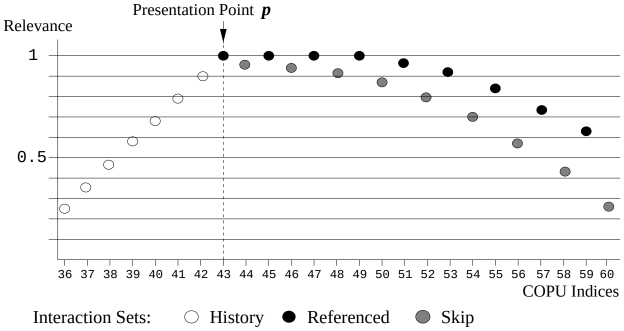

Figure 1: L/MRP: interaction sets and relevance values

3.2 MPEG-1

The MPEG-1 standard [6] is a coding format for audio and video streams. In this paper, we are concerned with video streams only [7]. The main feature of MPEG-1 that is interesting in this paper is that frames are no longer independent of each other as is the case with, e. g., Motion-JPEG which is a series of single JPEG [18] images. Figure 2 shows a sequence of MPEG frames and their interdependencies which are relevant for decoding the stream. An MPEG-1 video sequence in general consists of three different frame types, *I*, *B*, and *P*. Usually, the frames from one I frame up to the frame before the next I frame form a so called *Group of Pictures* (GoP). Since I frames (intra-coded pictures) are encoded similarly to JPEG images, their decoding is independent of other frames. The decoding of P frames (predictive coded pictures) depends on the preceding I or P frame of the same GoP. For B frames (bidirectionally coded pictures) decoding depends on both the preceding and the succeeding I or P frame. P and B frames allow a much higher compression rate than I frames by exploiting temporal prediction using motion vectors. It is important to note that the display order in which the frames are presented is different from the bitstream order in which the frames are decoded due to inter-frame dependencies. Figure 2 illustrates both the display order and the bitstream order of a stream. The order for decoding is very important as a preloading strategy must of course consider the order of decoding and not only of displaying the frames.

A preloading and buffer management strategy for MPEG-1 video must pay attention to the different frame types and their inter-frame dependencies, the bitstream order for decoding the stream, and the fact that the bitrate/data rate of the video and the size of the frames can heavily vary.

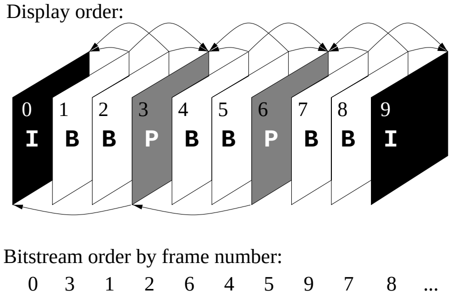

Figure 2: MPEG frame types and their interdependencies

4. MPEG-L/MRP MODEL

4.1 Overview of MPEG-L/MRP

Basic idea

So far the L/MRP approach has proven [12] to be superior to traditional preloading and buffering strategies especially when it comes to fast reaction to user interactions. The basic idea of *MPEG*-L/MRP is to provide the same interaction responsiveness as achieved with L/MRP but in particular to take into account the specific features of the MPEG video stream. The different frame types with their inter-frame dependencies and their different importance for the presentation are the main issue when adapting L/MRP to MPEG. The MPEG-L/MRP strategy exploits the knowledge about the importance and dependencies of the frames such that the video can be optimally presented under the available network bandwidth. Therefore, the interaction sets and the associated relevance functions of the L/MRP strategy are adapted such that they reflect this specific importance of frames for the presentation. When frames do not arrive in time at the client, temporal adaptation is used in order to maintain a continuous presentation.

Choosing the appropriate COPU size

The first issue of adapting L/MRP to MPEG streams is the kind and size of the data that forms a COPU. The COPUs are the basic units for transportation of the stream. Looking at MPEG-1 there are different possibilities to define a COPU:

A COPU corresponds to a GoP. The rather big size of the COPU might be a problem. If such a COPU cannot be delivered to the client, in average half a second of the video is missing. This size is also unsuitable for, e. g., a fast forward presentation of the video, since all frames had to be loaded to the client though only a subset of them would be needed.

A COPU corresponds to a part of a GoP. [5] proposed to use IBB or PBB groups. However, the groups and therefore the COPUs are then dependent on each other. And this restricts the supported coding scheme of the MPEG stream to IBBPBB...PBB patterns.

A COPU corresponds to a frame. Here, still the COPUs are dependent on each other like the frames of the MPEG stream are. However, this granularity allows for fast and targeted reaction to varying network bandwidth and user interactions.

We decided to use the third alternative as it offers the most appropriate possibility to compensate fluctuations in the available network bandwidth and, at the same time, offers support for fast and smooth reactions to user interactions on the stream. This decision serves as the basis for the formal model to follow.

4.2 The MPEG-L/MRP Model

Overview

In this subsection, the MPEG-L/MRP model will be developed step by step. Following some preliminary definitions, we introduce *presentation sets* as a means to collect those frames which have to be displayed for a particular kind of presentation of a video, such as normal playback, double speed presentation, and so on. Since P and B frames cannot be decoded independently, additional I or P frames might be necessary to actually decode and display the frames of a specific presentation set. These inter-frame dependencies are captured by *dependency sets*, leading to the notion of *closed presentation sets.*

Afterwards, static and dynamic *relevance functions* are defined as a means to quantify the relevance of frames contained in a particular presentation set. While static relevance functions are used to assign relevance values to frames surrounding a static *reference frame* (representing, e. g., a bookmark), dynamic relevance functions are needed to compute the relevance values of frames surrounding the *current presentation point* which is constantly moving in time during a normal presentation of the video. Both static and dynamic relevance functions are based on *generic relevance functions* which define relevance values independent of a particular reference frame or the current presentation point.

Finally, a *global relevance function* is introduced which combines the relevance values of static and dynamic relevance functions into a single overall relevance value for each frame of the video which will be used by the MPEG-L/MRP algorithm to determine preloading candidates and replacement victims.

Remark: For readers familiar with the details of the original L/MRP model [12], it should be noted that the formal model evolved in several aspects in order to adapt it to the special requirements of the MPEG video format. In particular, the notion of *interaction sets* containing *pairs* of frames (or COPUs) and relevance values (determined by so called *distance relevance functions*) has been split into two orthogonal concepts: *presentation sets* containing frames only on the one hand, and *relevance functions* assigning relevance values to frames on the other hand. By that means, interframe dependencies can be captured quite easily by introducing dependency sets which are completely independent of the concept of relevance values. Furthermore, *generic relevance functions*, which are translated to a particular frame and restricted to a particular presentation set in order to obtain static and dynamic relevance functions, are somewhat easier to use than the corresponding *distance relevance functions* of the original model, especially when frames are not equidistantly distributed within a presentation set.

Preliminary Definitions

Let, as usual, $\mathbb{N} = \{1, 2, 3, \dots\}$ be the set of natural numbers and $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$ the set of integer numbers. For $k \in \mathbb{Z}$, let $\mathbb{Z}_k$ denote the set of integers from 0 to k, i. e.,

$$\mathbb{Z}_k = \begin{cases} \{0, 1, \dots, k-1, k\} & \text{for} \quad k \geq 0, \\ \{k, k+1, \dots, -1, 0\} & \text{for} \quad k < 0. \end{cases}$$

For a subset $M \subseteq \mathbb{Z}$ of integers, let $\chi_M : \mathbb{Z} \to \{0, 1\}$ be the *characteristic function* of M assigning a value of 1 to all members of M and 0 to all other numbers:

$$\chi_M(x) = \begin{cases} 1 & \text{for } x \in M, \\ 0 & \text{otherwise.} \end{cases}$$

Presentation Sets

For a particular video comprising $n \in \mathbb{N}$ frames, let

$$F = \{0, 1, \dots, n-1\}$$

be the set of its *frame numbers* in display order. Furthermore, let I, P, and B be pairwise disjoint subsets of F representing the set of all I, P, and B frames of the video, respectively. Assuming that a video does not contain other frame types (in particular, D frames), it holds:

$$F = I \cup P \cup B.$$

A *presentation set* is a subset $S \subseteq F$ of frames which have to be displayed for a particular kind of presentation of the video. For instance, the presentation set

$$F_s = \{f \in F \mid f = i \cdot s, i \in \mathbb{N}_0\} = \{0, s, 2s, \dots\}$$

specifies the set of all frames f which have to be displayed for a forward or backward presentation of the video with a relative speed (or *skip factor*) of $s \in \mathbb{N}$.

Dependency Sets

Due to inter-frame dependencies, in order to be able to decode and display the frames of a particular presentation set S, it might be necessary, however, to decode additional frames. These inter-frame dependencies are captured by the *dependency set* $D(f) \subseteq F$ containing all frames $g \in F$ which are directly or transitively needed to decode and display frame $f \in F$. Using the auxiliary definitions

$$I(f) = \max\{g \in I \mid g \leq f\}$$

and

$$P(f) = \min\{g \in I \cup P \mid g \geq f\}$$

specifying the *closest preceding I frame* of frame f and the *closest succeeding I or P frame* of frame f, respectively, $D(f)$ can be defined as follows:

$$D(f) = \{f\} \cup \{g \in I \cup P \mid I(f) \leq g \leq P(f)\}.$$

Since $I(f) = f = P(f)$ for an I frame $f \in I$, it holds $D(f) = f$ in that case, which means that no additional frame is needed to decode an I frame. For a P frame $f \in P$ it holds $I(f) < f = P(f)$, and thus $D(f)$ contains f and all preceding P frames up to and including the closest preceding I frame. The same holds for a B frame $f \in B$, but since $I(f) < f < P(f)$ in that case, $D(f)$ contains the closest succeeding I or P frame of f, too.

Remark: For $I(f)$ and $P(f)$ to be well-defined for all frames $f \in F$, the first frame of a video must be an I frame and its last frame must be an I or P frame. Without these restrictions, the video would not be standard-conforming, however.

Closed Presentation Sets

The *closure* $\bar{S}$ of a presentation set $S \subseteq F$ can be defined as the set

$$\bar{S} = \bigcup_{f\in S} D(f)$$

containing all frames which are actually needed for a particular kind of presentation, either directly because they have to be displayed or indirectly due to inter-frame dependencies. A presentation set S is called *closed*, if $S = \bar{S}$ holds.

Given the definition of F_s above, the closure $\bar{F}_s$ comprises, for example, all frames which are actually needed for a presentation of the video with a relative speed of s. Intersecting $\bar{F}_s$ with one of the sets I, P, or B, yields the pairwise disjoint sets $I_s = \bar{F}_s \cap I$, $P_s = \bar{F}_s \cap P$, and $B_s = \bar{F}_s \cap B$ containing all I, P, or B frames, respectively, necessary for such a presentation.

If the coding scheme of a video is a constant repetition of the pattern illustrated in Figure 3 (i) (followed by a final I frame), the presentation set F_2 contains all frames depicted as shaded boxes. Since this set comprises all I and P frames of the video, it is already closed, i. e., it holds $\bar{F}_2 = F_2$ in that case. The presentation set F_3 on the other hand, illustrated in Figure 3 (ii) is not closed, since additional P frames identified by black arrows are needed to decode the B frames that are to be presented at a skip factor of 3. That means, that the closure $\bar{F}_3$ contains 6 instead of 4 frames out of each 12-frame pattern IBBBPBBBPBBB resulting in an overhead of roughly 50%.[2]

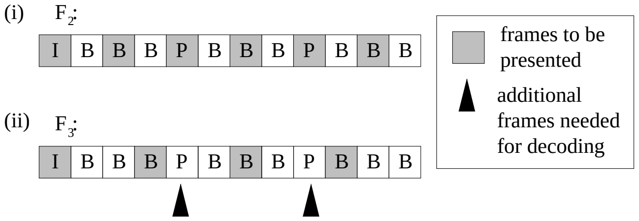

Figure 3: Presentation sets at different skip rates

Relevance Functions

A *generic relevance function* is a function $\rho : \mathbb{Z} \to [0,1]$ assigning a *relevance value* $\rho(x) \in [0,1]$ to each integer number $x \in \mathbb{Z}$. Typically, a generic relevance function is either *monotonously increasing* for $x \le 0$ and zero-valued for $x > 0$ or zero-valued for $x < 0$ and *monotonously decreasing* for $x \ge 0$. For instance, the linear functions

$$\lambda_a^b(x) = \begin{cases} \max(b \cdot (1 - x/a), 0) & \text{for } x \in \mathbb{Z}_a, \\ 0 & \text{otherwise,} \end{cases}$$

with a peak value of $b \in [0,1]$ for $x = 0$ and positive values for $x \in \mathbb{Z}_a \setminus \{a\}$ $(a \in \mathbb{Z})$ are typical examples of generic relevance functions (cf. Figure 4 (i) for $a \ge 0$ and (ii) for $a \le 0$).

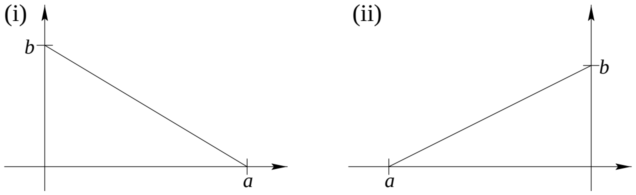

Figure 4: Typical generic relevance functions

A *static relevance function* is a function $\sigma : F \to [0,1]$ assigning a relevance value $\sigma(f) \in [0,1]$ to each frame $f \in F$. Typically, a static relevance function σ is constructed by *translating* the peak of a generic relevance function ρ to a specific *reference frame* $r \in F$ and *restricting* its domain to a particular presentation set $S \subseteq F$:

$$\sigma(f) = \rho(f - r) \cdot \chi_S(f).$$

A *dynamic relevance function* is a function $\delta : F \times F \to [0,1]$ assigning a relevance value $\delta(f,p) \in [0,1]$ to each frame $f \in F$ taking into account the current presentation point $p \in F$. Similar to a static relevance function, a dynamic relevance function is usually constructed by translating and restricting a generic relevance function ρ, where the current presentation point p replaces the static reference frame r:

$$\delta(f,p) = \rho(f - p) \cdot \chi_S(f) \quad \text{for some} S \subseteq F.$$

As noted above, static relevance functions are typically used to describe bookmarks where the reference frame r specifies the position of the bookmark in the video, while dynamic relevance functions are needed to model dynamic presentations of the video like normal playback, reverse playback, fast forward, etc., where the current presentation point p is constantly moving in time.

Global Relevance Function

Given a set of static relevance functions $\sigma_1, \ldots, \sigma_k$ and a set of dynamic relevance functions $\delta_1, \ldots, \delta_m$, the *global relevance function* $\gamma : F \times F \to [0,1]$ is defined as an appropriate combination of these functions, e. g., by computing a *weighted maximum value* for each frame $f \in F$:

$$\gamma(f,p) = \max\left(\max_{i=1,\ldots,k} \omega_i \cdot \sigma_i(f), \max_{j=1,\ldots,m} \pi_j \cdot \delta_j(f,p) \right).$$

Here, the weighting factors $\omega_1, \ldots, \omega_k \in [0,1]$ and $\pi_1, \ldots, \pi_m \in [0,1]$ can be used as global regulators similar to the slide controls of a sound mixer.

[2] Since the sizes of I, P, and B frames are usually quite different, this is indeed only a rough estimation.

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**