

HDTV動画像処理システムにおける ビデオサーバとAPIの開発

7U-8

小野 泰志 馬場 条太郎 沼尻 裕 村山 正之

(株) 東芝

1. はじめに

近年、ウィンドウに動画を表示し、その内容をデジタル情報として保存、処理、再生する機能を有するハードウェア、ソフトウェアが出現している。今回、大画面、高精細を特徴とするハイビジョンモニタを使用したフルカラー表示可能(24ビットプレーンのフレームバッファ)な動画像処理システムを開発した。本システムは、ハイビジョンモニタ上でウィンドウシステムを実行可能とし、容易に各種の応用実験ができるようアプリケーションプログラミングインタフェースもあわせて開発した。本稿では、ハイビジョンモニタ動画像処理システムにおけるウィンドウシステムの動画拡張方式とそのアプリケーションプログラミングインタフェースについて述べる。

2. ハイビジョン動画像処理システムの概要

本システムは、エンジニアリングワークステーションAS4080にハイビジョンモニタ用フレームバッファボードを実装し、OS/AS V4.13で動作する。このフレームバッファボードは動画、静止画をフルカラー表示する。また、静止画像圧縮の標準であるJPEGを採用し、画像データの圧縮/伸長が可能である。例えば、カメラから動画を表示しながらJPEG圧縮が可能である。また、JPEG圧縮データを伸長して表示することも可能である。(図1参照)

平均1メガバイト/秒の書き込み能力を持つディスクに、640x480 NTSC画像を30フレーム/秒で録画する場合、画像データを1/20に圧縮することで実現できた。

3. ウィンドウシステムのビデオ機能拡張

ハイビジョン動画像処理システムは、X Window上で実現した。標準のX Windowは、動画表示、制御機能を持っていないため、動画機能を付加する必要があった。一般にX Windowに動画機能を付加する場合、Xの拡張仕様に従ってXサーバとXプロトコルを拡張をしているものが多い。しかし、我々はXサーバとXプロトコルを拡張せずに、X Windowに動画機能を付加する方法を取った。

Xプロトコルを拡張せずに動画機能を実現する単純な方法の一つに、アプリケーションが動画表示・制御をすべて行うという考え方がある。しかし、一つのアプリケーションが動画表示・制御をすべて行うとすると、アプリケーション間の動画関連のリソースの競合やネットワークへの対応が難しくなる。

そこで我々は、動画関連の表示・制御、リソース管理機能を一つの独立したプロセスとした、サーバ/クライアントモデルで実現した。このプロセスをビデオサーバと呼ぶことにする。ビデオサーバとクライアント間は今回規定したプロトコルに従って通信を行う。ビデオサーバは直接フレームバッファにアクセスし、Xサーバとは接続しない。ビデオサーバはウィンドウ関連のリソースを保持せず、動画関連のリソースのみを管理する。

ビデオサーバはウィンドウシステムに依存することなく一ウィンドウシステムが動いていなくても一ビデオ表示が可能となる。

サーバ/クライアントモデルとしたことで、ビデオサーバはフレームバッファ固有のインタフェースをアプリケー

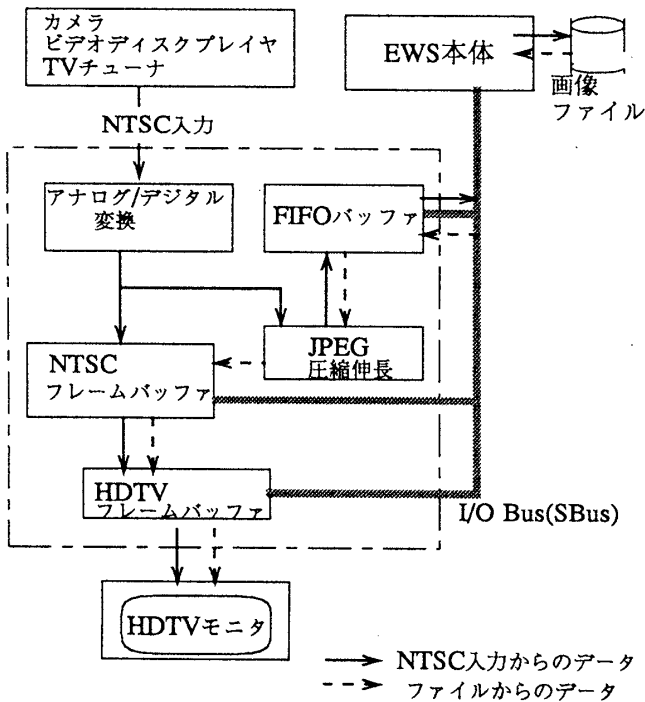


図1 ハイビジョン用フレームバッファボード

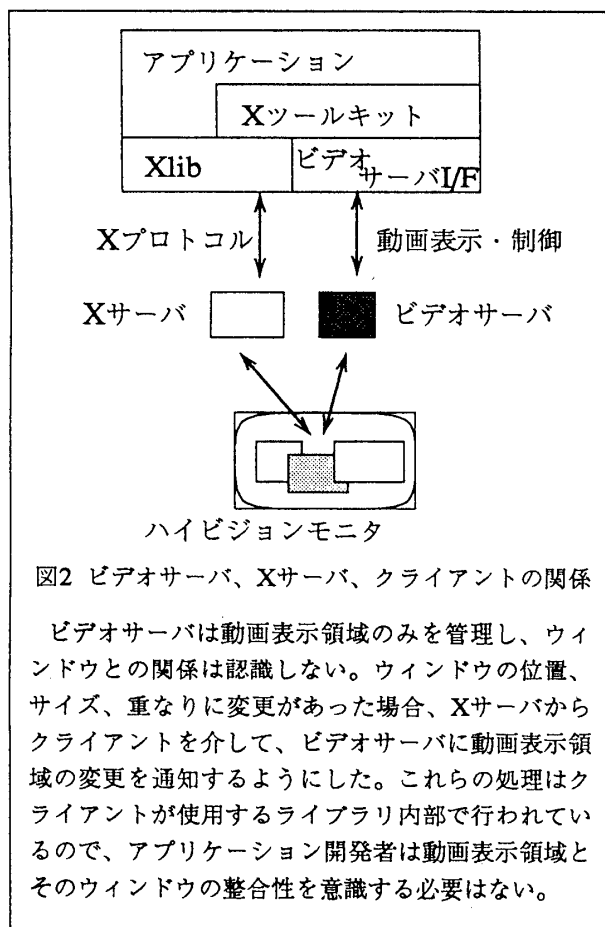
Development of a Video Server and API for HDTV Moving Picture Processing System
Yasushi Ono, Joutaro Baba, Yutaka Numajiri, Masayuki Murayama
TOSHIBA corporation

ションから切り離れた。アプリケーション開発者は、動画機能の詳細なインプリメントを知る必要はない。

また、新しい動画機能を持つフレームバッファを開発した場合、そのフレームバッファ用にビデオサーバを開発すれば、アプリケーションの変更を必要としない。(図2参照)

<ビデオサーバ方式の利点>

- ・ウィンドウシステムに依存せずに動画機能を実現
- ・ネットワーク透過の動画アプリケーションが開発可能
- ・固有のフレームバッファに依存しないアプリケーション開発が可能



4. API

本システムは、ウィンドウ環境で動画処理を行うアプリケーションを作成するためのアプリケーションプログラミングインタフェース(API)を提供している。本システムでは、APIとして新しい動画処理用関数群を提供するのではなく、ウィンドウプログラミングの延長として、動画処理機能を使用できるようにした。

X Windowのアプリケーションを開発する場合、Xツールキットを使用するのが一般的である。Xツールキットはオブジェクト指向の考え方が取り入れられて

おり、我々はクラス拡張の可能なOLIT(OPEN LOOK Intrinsic Toolkit)に動画処理用クラスを追加することで、アプリケーションから動画処理が行えるようにした。

5. VideoWidget

OLITはユーザーインタフェースのオブジェクト(Widget)を提供する。ボタン、スクロールバー、ポップアップのような部品が、オブジェクト指向のクラスで実現されている。

OLITに、動画を扱う部品としてVideoWidgetを追加した。VideoWidgetは、グラフィック描画などに用いられるStubWidgetのサブクラスとして実装し、Xのライブラリとビデオサーバインタフェースライブラリを使って、動画表示、制御を行う。

VideoWidget特有のリソースとして、フレーム/秒、JPEGパラメータ、NTSC入力チャンネルなどがある。これらのリソースはツールキット標準の関数で扱うことができる。

ここでも、既存のライブラリや構造体に変更を加えることなく、VideoWidgetとビデオサーバインタフェースライブラリを追加することにより、ビデオ機能を使用するためのプログラミング環境を提供している。

VideoWidgetは、アプリケーションプログラマから、ビデオサーバインタフェースを隠す。アプリケーションプログラマはOLITのプログラミングを知っていれば、ボタンやメニューと同じようにビデオウィンドウを簡単に作成できる。

単純な動画機能(再生、録画のみ)を行うプログラムなら、C言語で200ステップ以下で作成可能である。

6. おわりに

Xプロトコルを拡張せずビデオサーバプロセスを導入することで、ウィンドウシステムに動画処理機能を追加した。この結果、ウィンドウシステムと整合性をとりながら、固有のウィンドウシステムに依存しない動画処理システムが実現できた。また、APIとして、OLITにビデオ部品を追加することで、本システムを用いた各種の応用実験やアプリケーション開発を容易にできるようにした。

[参考文献]

[1] 河合 利彦、大森 啓輔、田辺 充:「ビデオ信号をXウィンドウ上に表示するワークステーションボード」日経エレクトロニクス、1991 No.530

[2] 「OPEN LOOK Intrinsic ツールキット Widgetセット プログラマーズガイド」日本サン・マイクロシステムズ社

[3] X Toolkit Intrinsic-C Language Interface, MIT,1991