

# EffecTV: メガデモ技術のリアルタイムビデオイフェクトへの応用

福地 健太郎                      Ed Tannenbaum  
東京工業大学 情報理工学研究科  
fukuchi@is.titech.ac.jp    et@et-arts.com

概要: ソフトウェアによるリアルタイムビデオイフェクトソフトウェア「EffecTV」を開発した。本ソフトウェアでは、「(メガ)デモ」と呼ばれる、高速なアニメーション生成技術を応用して、リアルタイムに入力される画像の動きを反映した映像効果を即時に生成する。その結果、インタラクティブ性とエンターテインメント性の高い、新しい映像効果を提供できた。EffecTVはこれまでに、VJ(Video Jockey)や劇中映像、インターネット放送等に使用されている。本報告では実装技術の概要と、これまでの使用事例について述べる。

## EffecTV: an application of DEMO techniques to realtime video effect

Kentaro Fukuchi                      Ed Tannenbaum  
Tokyo Institute of Technology  
fukuchi@is.titech.ac.jp    et@et-arts.com

**Summary.** We developed a realtime software video effect processor named “EffecTV”. This software employs various hacker-side techniques of computer graphics known as “DEMO”, and it enables to output video images reflects the motion of the input images in realtime. As a result, we achieved to produce highly interactive entertaining visual experience. EffecTV has ever been used for VJ works, theatrical play, internet broadcasting, etc. In this paper, we describe some use case of EffecTV.

### 1 はじめに

「デモ (demo)」とは、元々は一般のユーザーが作成した、家庭用コンピュータ上で動作するデモンストレーションソフトウェアを意味し、多くは映像効果の出来を競うものである。古くは Amiga や Commodore 64 上で動作するデモが数多く発表され、近年では PC にとどまらず、Dreamcast や PlayStation2 までもがその対象となっている。デモの基本精神は「人を驚かせるようなソフトウェア技術を誇示する」事にある。一般にデモは安価で入手できるホームコンピュータ上で動作するように作られていたため、演算や表示能力の制限が厳しいものが多く<sup>1</sup>、いきおい、そうしたコンピュータでは実現が難しいと考えられた映像効果を達成する事が目標となった。例えばポリゴンによる 3DCG のリアルタイム表示がその課題であった。現在では家庭用の PC でも凄まじい演算能力と 3DCG 表示能力が実装されているため、新たな方向性が模索されている。なお、デモ文化について

<sup>1</sup> 日本では一般的な「メガデモ」という呼称は、1 枚のフロッピーディスク (1 メガバイト) に収まるように作られたデモを指している。



は、Web サイト [4][5] に詳しい。

筆者は 1994 年頃から富士通の家庭用コンピュータ「FM-TOWNS」を用いて、リアルタイムに動作するビデオイフェクトソフトウェアを製作していた。ビデオカメラやビデオデッキからの入力画像をデジタル化し、ソフトウェアで加工して、すぐにビデオ信号として出力する、というものである。当時行なっていた加工は、画像の座標変換や色変換といった単純なものであった。

EffecTV[1] は、筆者が 2001 年に発表したリア

ルタイムビデオイフェクトソフトウェアで、前述のデモ技術とリアルタイムビデオイフェクトとを融合させたものである。デモ技術を応用することで、入力画像をリアルタイムに加工し、斬新な映像効果を付与して出力することができるようになった。また、入力画像の動きを反映した映像効果を出力することで、インタラクティブ性とエンターテインメント性を併せ持つことがわかった。

## 2 EffecTV

EffecTV は筆者が 2001 年 2 月に、Linux 上で動作するソフトウェアとして発表した。発表当初は 8 種のイフェクトを提供していたが、発表後すぐに様々な人々から改良パッチや新イフェクトが寄せられるようになり、現在では 50 種近くのイフェクトを提供している。また、当初は開発者は筆者 1 人のみだったが、現在は 9 人となっており、その国籍はアメリカ・ドイツ・オランダ等様々である。

### 2.1 基本構成

#### 2.1.1 ハードウェア構成

ここでは PC を使用した構成を説明する。

本体は一般的な構成の PC を使用し、OS は Linux を用いる。CPU の演算能力はそれ程高いものは要求されない。筆者は Celeron(300MHz) を搭載した PC で開発している。入力及び出力画像のサイズが 320×240 ピクセルの場合、ビデオ入力のフレームレートと同じ毎秒 30 フレーム処理を達成している。より解像度の高い画面、例えば 640×480 ピクセルの画面を使用する場合や、複数のイフェクトを重ねてかける場合には、より高速な機器が必要となる。Pentium4 1.6GHz を搭載した PC では、4 種のイフェクトをかけた場合でもフレームレートが落ちる事はなかった。なお、EffecTV の処理においてはメモリアクセスがその処理時間の大半を占めるため、CPU の処理能力以上にバスの帯域幅が重要となる。

ビデオ入力には、ビデオキャプチャーカードにビデオカメラやデッキ等を接続するか、USB カメラあるいは IEEE1394 カメラを使用する。IEEE1394 接続でのデジタルビデオカメラによる入力は現在はサポートされない。

出力は画面にそのまま表示される。他のビデオ機器に出力する場合は、PC のグラフィックスカードの出力信号をビデオ信号(コンポジット又は S)に変換する機器が必要となる。

#### 2.1.2 ソフトウェア構成

EffecTV では、各イフェクトは入力画像をビットマップとして受け取る。各イフェクトはそれを加工してビットマップとして出力部に渡す。出力に OpenGL を用いる場合は、各イフェクトが GL 画面に対して出

力する。この処理を入力フレーム毎に行う。ビットマップは 32bit の RGB 値(各 8bit)列からなる。サイズは起動時に決定される。また、カメラをスクリーンと同じ方向に向けて設置した場合は、鏡であるかのような感覚を提供することができるよう、入力画像を左右反転させる機能を持つ。

イフェクトはキーボード操作によって切り換えられる。現在の実装では同時には一種類のイフェクトしかかける事はできない。しかし、EffecTV は vloopback[9] という仮想ビデオデバイス機構を利用し、それ自身が別のビデオ入力デバイスであるかのように振舞う機構を実装しているため、EffecTV の出力を別プロセスとして動作している EffecTV に渡す事ができ、イフェクトを多層的にかけ事が可能となっている。また、この機構を利用すると、ビデオ入力をうける他のソフトが EffecTV の出力画像を受け取れるようになる。例えば Web カメラソフトと接続すると、加工された映像を Web サーバーから供給できる。

プログラムは移植性を考慮して大部分が C で書かれており、他の環境に移植するのが容易となっている。例えば、EffecTV 自身は PlayStation2 Linux でも動作している。また、各イフェクトは様々なソフトウェアのプラグインとして移植されている。例えば BeOS 用ソフトウェアの stampTV、Windows 用ソフトウェアの visualJockey 等がある。いずれも入力部と出力部の API をそれらのソフトウェア用書き換える作業が移植作業の大部分であり、それ以外の部分はほぼ変更を必要としない。ただし、エンディアンが異なるプラットフォームへの移植をする際には、一部コードの書き換えが必要である。

### 2.2 イフェクト

EffecTV は約 50 種のイフェクトを提供している。本節では、その中から特にインタラクティブ性の高いイフェクトを取り上げて紹介する。なお、個々のイフェクトの細かいアルゴリズムや技法については、紙面の都合上詳しくは取り上げない。デモで使用される映像効果のアルゴリズムについては、[6] のサンプルコード群を参照されたい。一般的な CPU 向けの高速度化技法については、[3] が詳しい。[2] では、EffecTV で採用している、拡張命令群を使用しない高速度化技法について述べている。

#### 2.2.1 FireTV

FireTV は、画面中の前景部分を燃やしているような映像効果を作る。本イフェクトでは背景差分の技法を使用している。まず、背景画像をあらかじめキャプチャーしておく。その後は、入力画像からこの背景画像を引き、前景画像を得る。EffecTV では輝度成分の差分を利用している (1)。この前景画像に対して炎を生成する事で、画面内の人間のみを燃やした

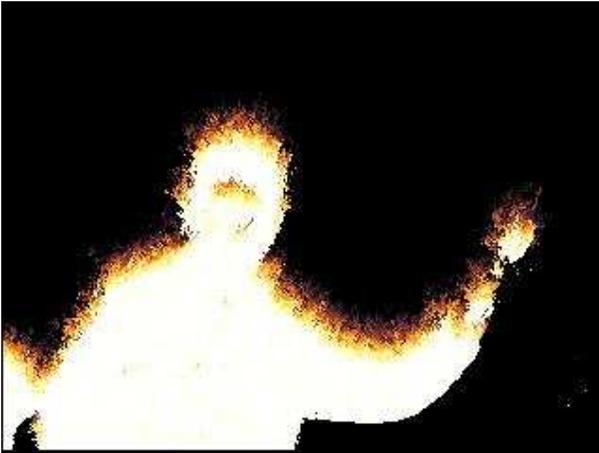


図 1. FireTV



図 3. QuarkTV



図 2. BurnTV

りする事が可能となる。

炎の生成手法はデモ技術として古くから知られているもので、火種となるピクセルの上方向に炎を成長させる。前述の方法で抽出した前景画像を火種として与えると、図 1 のような画像を得る事ができる。

図 2 は、上述の方法で生成した炎の画像を、入力画像と合成するイフェクトである。このイフェクトでは炎を生成する際に、前景画像の輪郭線のみを火種として用いる事で、炎が前景画像の全てを覆ってしまう事を防いでいる。

### 2.2.2 QuarkTV

QuarkTV は、動いている物体を粒子状に加工して表示する。図 3 では、動かしている手のみがドット単位でバラバラになっているように見えている。

本イフェクトはタイムバッファと我々が呼ぶ手法を用いている。まず、8 フレーム分の領域を持つリングバッファ(タイムバッファ)に最近 8 フレーム

の画像を置く。出力画像を生成する際に、画面上の各ピクセル毎にタイムバッファ中からフレームをランダムに選択し、そのフレームから同座標のピクセルを取得し、描画する。擬似アルゴリズムで表わすと以下ようになる。

```
for(y=0; y<height; y++) {
    for(x=0; x<width; x++) {
        output[y][x] =
            timebuffer[rand()%8][y][x]
    }
}
```

このように、本手法は動体検出の為の特別な手法は使っていない。静止している物体はタイムバッファ上ではどのフレームをとっても同じように表われているため、動いている物体のみがイフェクトの影響を受ける。この手法は動体の色や明るさ等の影響を受ける事がなく、計算コストも低い。

本イフェクトでは、各ピクセル毎にタイムバッファからランダムにフレームを選択しているが、この選択方法を変える事で、まったく違った印象を与えるイフェクトを作る事ができる。例えば EffecTV のイフェクトである SpiralTV は、画面中心からの距離に応じてフレームを選択する事で、同心円状に動体に変化するイフェクトを作り出している。

### 2.2.3 RippleTV

RippleTV は、画面全体が水面であるかのように振舞い、画面中で動きが検出されたところで波立つようにプログラムされている(図 4)。動き検出には背景差分の手法を応用している。取得フレームからその前フレームを引き、閾値以上の差があった部分が、動きのあったところであるとみなす。

水面のシミュレーションは、やはり古くからデモ技術としては知られているもので、有限要素法を応用したアルゴリズムを採用した。全てを整数演算



図 4. RippleTV



図 5. RadioacTV

で計算するために、プロセッサを問わず高速に動作する。

#### 2.2.4 RadioacTV

RadioacTV は、画面中の動いている物体の周囲に、拡散する青白い光を発生させる。イフェクト名は、これを放射能に見立てた事に由来する。

拡散する光の生成手法は、デモ技術では“Blur & Zoom”等の名前で知られるもので、画面全体を convolution matrix でぼかした後、拡大する事で簡単に得られる。RadioacTV では、動きのあった部分に白色のピクセルを出力し、その画面を左記の方法で拡散させ、入力画像に合成して出力している。動きの検出は、RippleTV と同じ方法を用いている。

#### 2.2.5 NervousTV

NervousTV は、フレームの順番をランダムに入れ替える事で、動体の動きをコミカルに変えるイフェクトである。

NervousTV の仕組みは非常に単純で、最近 32 フレームをバッファに蓄えておく。出力の際には、そのフレームの中からランダムに 1 フレームを選択し、それを表示する。これだけで、例えば、人が歩いている様子は、前後に行きつ戻りつを高速に繰り返しているように見える。フレームの選択を完全なランダムにするのではなく、時間軸方向で前か後に数フレーム再生すると、音楽でいうところの“スクラッチ”をしているような映像が得られる。

### 3 使用事例

EffecTV はこれまでに様々な場面での使用を試みてきた。ここではその使用事例について紹介する。

#### 3.1 常設 EffecTV

EffecTV 一式を、オフィスやクラブのフロアに常設する事を試みた。カメラを部屋の隅に、中心に向けて設置しておき、カメラ脇の液晶画面に EffecTV の画面を表示する。イフェクトは数分毎に切り替わるように設定した。これら一式を、オフィスでは共有の談話室に、クラブではダンスフロアにある VJ ブースの脇にそれぞれ設置し、観客の反応を観察した。

いずれの場合も、多くの観客がカメラの前で体を大きく動かして、画面上の反応を楽しんでいる様子が見られた。各イフェクトはいずれも、動きが大きければ大きい程、映像効果もそれを反映して激しくなるので、観客は始めは単に指先などを小さく動かしていても、次第に腕や上体全体を動かすようになった。ディスプレイの前に長居して、いろんな動きを模索する者も多かった。時には格闘技の型を披露したり、ダンスフロアに設置した場合には、音楽のビートに合わせて体を動かすなどして楽しむ様子が見られた。

#### 3.2 ライブ VJ での使用

筆者は映像作家の奥秀太郎氏と組んで VJ (Video Jockey または Visual Jockey) をやっている。我々が取り組んだのは、バンドのライブ演奏向けの映像生成である。ライブ VJ では、ステージ上のバンドの演奏風景を取り込んでの映像送出手が重要となる。そこで、EffecTV を用いてのリアルタイムイフェクトを VJ 作業に応用した。ここでは、映画「風花」記念イベントでの、大友良英のライブでの使用事例について述べる。

使用したイフェクトは主に AgingTV (フィルム映画のような画像劣化をリアルタイムに施す) と RadioacTV であった。RadioacTV を使用する際には、ステージ上のカメラでターンテーブルやギターを操作する手を接写し、そこにイフェクトをかける、という使い方を試みた。なお、ステージは暗かったので、赤外線カメラを使用した。赤外線カメラだと、

ステージのみならず、フロア上の観客の様子も撮影できる。

映像効果としては、今日の前で展開している演奏風景が何十年も前の映像のように劣化して見えたり、あるいは奏者の手の動きがイフェクトによって誇張されて提示されているのが面白い。しかし、EffecTVの本来の面白さであるところの、観客が能動的に楽しむという面白さはなかった。

### 3.3 演劇での使用

近年は演劇の舞台上にスクリーンやテレビを設置し、劇中に映像を流す演出が多く為されている。これまでに EffecTV は、二つの舞台で使用されている。ここではその使用事例について述べる。

ケラリーノ・サンドロビッチ作「室温～夜の音楽～」(2001年7月初演、青山円形劇場)では、RadioacTV と BurnTV が使用された。RadioacTV を使用した場面は、舞台上の演者の一人が実は幽霊であるという設定で、その演者は舞台上ではこれまでと変わりなく見える。しかし、舞台上部に設置されたスクリーンでは、その演者の周囲に青白い光が付加されて見えるので、幽霊である事がわかる、という仕掛になっていた。BurnTV は、舞台となっていた館が炎上するという劇の最後の場面で使われた。舞台上では主役の二人が抱き合っているだけだが、やがて音響効果とともに、スクリーン上には BurnTV を使用した、二人が燃えている画面が写し出され、やがて舞台の照明が暗くなると炎上する映像のみが見える、という仕掛であった。

従来の劇中映像は、役者の生の芝居を反映するものではなく、作り置き映像を流すものであった。この二つのシーンはいずれも、舞台上で芝居をする役者の振舞いを、リアルタイムに加工して舞台上のスクリーンに投影するという意味で、画期的な演出であった。

ただしいずれも、舞台とスクリーンは離れて設置されており、またスクリーンで映像が展開されている間も、舞台上では役者が動いているため、観客は注意を二分されるか、あるいは舞台に集中してしまうため、映像が効果的に働いたかどうかは疑問が残る。

サモ・アリナズ「ルーシー 26」(2002年7月初演、下北沢本多劇場)では、舞台冒頭の役者紹介の場面で EffecTV が使用された。舞台両袖前に、役者が一人一人入れ換わりで立ち、パフォーマンスをする。カメラはその姿を撮影し、その映像を EffecTV で加工して舞台中央の大スクリーンに投影した。役者にはあらかじめ EffecTV の効果を説明し、事前に練習期間を設けていた。この演出では、EffecTV のインタラクティブ性を充分活用できた。また、役者は公演期間中に様々な動きを試みており、役者のパフォーマンスを効果的に引き出す事ができた。

また、映像は舞台中央に大きく投影され、役者は袖に立っているの、観客は意識をスクリーンに集中する事ができた。

### 3.4 その他の応用

本節では、前節までの使用事例以外に試みた実験について述べる。

EffecTV の出力画像をビデオ会議システムに接続して、ビデオ会議に参加してみた。普段はイフェクトをかけない通常の画面を表示させておくが、自分の発言に注意を引きたい場合などで、いくつかのイフェクトをかけてみた。あるいは、他者の発言中に、自分の感情を反映させるために自分を燃やしてみたり、モザイクをかけて匿名を装ってみたりと、いろいろな遊びを発見した。参加者には大変好評だったが、イフェクトをかけているのが自分一人だったため、参加者全員がイフェクトをかけるようになったらまた違った印象が残るだろう。将来、インターネット上でのビデオチャットが一般的になったら、面白い応用が考えられるかもしれない。

いくつかのサイトでは、EffecTV を Web カメラに接続する実験を実施している。初めてイフェクトのかかった画像を観せられた観客には受けが良いのだが、一般に Web カメラの画像はそれ程動きがなく、また帯域幅の制限からフレームレートも高くないものが多いため、興味深い結果は得られなかった。

## 4 考察

遊園地や公園等で、遊具として設置されている曲面鏡の前で子供達が自分の姿を写して遊んでいる姿を見かける事がある。それは、自分の姿が歪められて見えるのを楽しむだけでなく、どのように動くとどんな変化をもたらすかを、自分の体をもって試してみているようにも見受けられる。

EffecTV の面白さは、入力画像にカメラを用い、自分の振舞いを加工して即時出力した場合に強く発揮される。このとき、EffecTV は上記の遊具と同じで、言わば「魔法の鏡」として機能している。観客はこの魔法の鏡の前で、いろいろと動きを試しながら、どんな動きをするとさらに面白い映像が作れるかを模索し始める。特に EffecTV を公共の場に設置した場合では、観客は単に EffecTV を楽しむだけでなく、その場にいる他の観客を楽しませるために、様々な工夫を凝らす姿を観察できる。逆に言えば、観客は、自分の表現をするために、EffecTV を道具として用いる事ができる。

EffecTV のもう一つの能動的な楽しみ方は、イフェクトを自分の手で作る点にある。入力・出力とも単純なビットマップとして扱えるようになっており、C 言語でのプログラミング経験さえあれば、イフェクトを作成するのは容易となっている。かつては CPU の演算能力が低かったため、高速なイフェク

トを実現するために様々なプログラミングテクニックを必要としたが、現在ではそうした点はほとんど気にする必要はない [2] ため、イフェクト製作の敷居は低くなったと言える。しかし、フレームレートが低下してくるとフレーム間の連続性が低下し、多くのイフェクトでは映像効果が失われる。また、応答性が悪くなり、「鏡」として機能しなくなる。従って、ある程度以上に複雑なイフェクトを製作する場合は、高フレームレートを達成するために何らかの形で最適化を施す必要がある。

一部のイフェクトは、画面の大きさとパラメータ調整が密接な関係にある。例えば RippleTV では、波の伝わる速さは、320×240 ピクセルの画面で、約 1 秒かけて波頭が端から端へと到達するように調整してある。しかし同じイフェクトを 640×480 ピクセルの画面に適用すると、波の速度は左記の目標を達成できない。同様に、炎の動き等、画面の大きさに適応してパラメータを自動的に調整する機構が必要である事がわかった。

## 5 関連研究

富士通大分ソフトウェアラボラトリが FM-TOWNS 用に発表した「おもしろ TV」は、ソフトウェアによるリアルタイムビデオイフェクトを、CPU に 80386 を採用した FM-TOWNS 上で実現したものである。画像反転や色調変換等の基本的なイフェクトが中心で、入力画像の動きを反映するものではないが、タイムバッファと同じ手法を使用したイフェクトが含まれており、斬新であった。

Ed Tannenbaum[7] の作品「Do-Undo」は、カメラとディスプレイが観客に向けられており、3 秒間観客の動きを録画すると、次の 3 秒間はその動きを逆再生して表示する、というものである。観客は逆再生のおかしさそのものをまず楽しむが、やがてさらなるおかしさを作り出すために、逆再生を積極的に利用ようになる。例えば録画中に物を落してみたり、画面外に出てみたりなど、様々な試みをするようになる。そのうち、みんなで協力しあったり、一人が自発的に監督役になったりして、より面白い演出を試みるようになるなど、興味深い知見が報告されている。

GStreamer[8] は、フリーのマルチメディアパイプラインシステムで、映像や音声等のデータを、様々なモジュールを組み合わせて加工・出力することができる。UNIX 系のプラットフォームでは事実上の標準になりつつあり、入力・出力とも様々な規格に対応するために、豊富なプラグインが実装されている。現在、EffecTV のイフェクトのうちのいくつかは GStreamer 用に移植されている。GStreamer では、音声や MIDI 信号等も統合的に扱えるため、リアルタイムインタラクションの幅をさらに広げることができる。

## 6 まとめと今後の課題

デモ技術とリアルタイムビデオイフェクトを融合したソフトウェアを実装した。入力画像の動きを反映した映像効果により、高いインタラクティブ性とエンターテインメント性を提供できた。

現在、ソフトウェアのアーキテクチャを改め、イフェクトをより作りやすくするように改良している。現状では C プログラミングの高度な理解が必要だが、ビジュアルプログラミング言語を採用する等して、イフェクト製作の敷居を低くしたい。また、イフェクト製作を支援するために、動体認識・画像処理のためのライブラリを標準で用意する事を予定している。

今後は、近年の CPU が提供するマルチメディア拡張命令や、グラフィックスボードの処理能力を反映し、イフェクトの質の向上を目指す。また、イフェクト選択やパラメータ調整は、現状ではキーボードで行うが、これを入力画像や他の情報を用いるようにする事を課題とする。特に、音声情報を反映したイフェクトによって、VJ やダンス映像向けの新しい映像効果が期待できる。

## 参考文献

- [1] EffecTV <http://effectv.sourceforge.net/>
- [2] 福地健太郎. メガデモに見る小手先高速化技術. 夏のプログラミングシンポジウム「プログラミングの鉄人 - プログラミングの技」, 2001.
- [3] Henry S. Warren, Jr. Hercker's Delight. Addison-Wesley, 2002.
- [4] ORANGE JUICE - The demoscene information center <http://www.ojuice.net/>
- [5] Demoscene Outreach Group <http://www.scene.org/dog/>
- [6] Programmers Heaven Graphics & Sound zone <http://www.programmersheaven.com/zone10/>
- [7] Ed Tannenbaum's web page <http://www.et-arts.com/>
- [8] GStreamer <http://www.gstreamer.net/>
- [9] vloopback <http://motion.technolust.cx/vloopback/>