

X Visual Effect Extension(2)

～インプリメントと評価～

松浦 宏[†] 高野 元[‡](株)NEC 情報システムズ[†] NEC C&C 研究所[‡]

2 V-6

1 はじめに

プレゼンテーションにおいて、視覚的に聴衆にアピールしたい場合や、デモンストレーション等でのユーザの操作性の快適さを求めて、筆者は X ウィンドウシステム上に視覚効果拡張を行っている [1]。拡張作業としては、イフェクトコンテキスト (以下 EC) の追加、視覚効果処理ルーチンの作成、ユーザ提供ライブラリ関数の作成が挙げられる。本稿では、特に視覚効果処理ルーチンにおいて非同期処理の問題に対処する実装方法を中心に述べる。

2 視覚効果拡張の方法と問題点

2.1 X の拡張方法

X のリクエスト処理

X ウィンドウシステムは、サーバ-クライアント型のウィンドウシステムである。サーバは、クライアントから受け取ったリクエストを一旦リクエストキューに蓄え、キューに蓄えられた順にリクエストに対応する処理ルーチンを起動して各処理を行う。よってプロトコルリクエストを追加することによって新規機能の拡張を行うことが可能である。EC の操作に関してはこの方法により機能拡張を行っている。

ラッピングによる機能拡張

X サーバ内では、ウィンドウ構造、スクリーン情報を示す構造体の中に描画処理などの関数へのポインタが定義されている。上位関数からは、これらの下位関数をポインタを通じて呼び出すことになる。既存の描画ルーチンを改造する際には、これらの関数へのポインタを、新規作成の描画ルーチンへのポインタに置き換えることによって簡単に付け替えることが可能である。このような方法を関数のラッピングと呼ぶ [2]。視覚効果処理ルーチンはラッピングによる機能拡張を行っている。

2.2 非同期処理の問題

X はリクエストを順次処理していくが、大抵のリクエスト処理は瞬時に終了してしまうので見掛け上は複数のウィンドウ処理が同時に行われているように見える。しかし視覚効果処理は、例えばウィンドウをマップする際にワイプ及びフェード処理を行うため、1つのリクエスト処理は一定の処理時間が必要である。これは、視覚効果処理中には、他のリクエスト処理が行えないことを意味する。また、仮に視覚効果と他のリクエストの同時処理を可能にした場合、矛盾するリクエストを同時に処理しなければならない状況が発生し得る。例えば、マップリクエストを処理中のウィンドウに対してアンマップリクエストを受け付けた場合である。そこで視覚効果処理には、複数のウィンドウを非同期的に同時に描画出来るようになると共に矛盾する処理をブロックするような機構が必要となる。

X Visual Effect Extension(2)

Hiroshi Matsuura, Hajime Takano

NEC Informatec Systems and NEC Corporation, Ltd.

3 視覚効果処理ルーチンの実装

視覚効果中に他のリクエストを処理するためには、視覚効果描画処理ルーチンが一定時間 CPU を占有するのではなく、周期的に起動されて描画処理を行うようにすればよい。

また、視覚効果処理中のウィンドウに対する矛盾したリクエストをブロックするには、サーバの持つリクエストキューとは別に、リクエストをキューイングする機構を用意すればよい。

周期的割り込み処理

視覚効果機能が付与されたウィンドウに対して表示制御リクエスト (MapWindow, UnmapWindow, ConfigureWindow) を受け付けた時、サーバはそのリクエストの情報を視覚効果処理テーブルに登録 (キューイング) し、割り込み処理ハンドラ (Block/Wakeup ハンドラ) を起動する (図 1)。

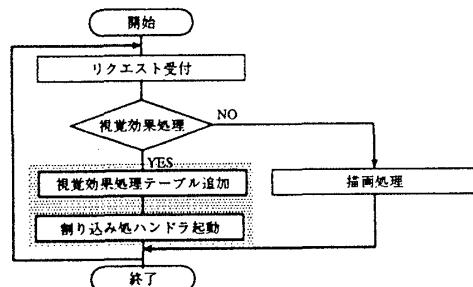


図 1: 表示制御ルーチン

視覚効果処理テーブルは、描画処理時間や EC 属性等を保持するテーブルである。視覚効果処理テーブルにデータが格納されている場合、周期的に起動される割り込みハンドラを作動させる。

割り込みハンドラが起動されると、このテーブルに登録されているリクエスト情報にしたがって描画処理を行う。

この割り込みハンドラは、キューイングされた視覚効果処理をテーブル内から順次検索し、その時間で行わなければならぬ描画処理を行うと、次の割り込み時間まで休眠する (図 2)。

このように、視覚効果処理をキューイングし、周期的割り込みによって描画を行うことによって、複数のウィンドウが同時に視覚効果を実行することが可能になる。

描画リクエストのキューイング機構

視覚効果処理テーブルの構造を、階層的リスト構造で実現する。同時に処理可能な描画リクエストをリスト構造で保持し、割り込みハンドラはこのリストを検索し、描画処理を行う。既にテーブルに登録されているウィンドウに対する描画リクエストを受け取った場合、そのウィンドウのリストにキューイングする。図 3 は、Wi

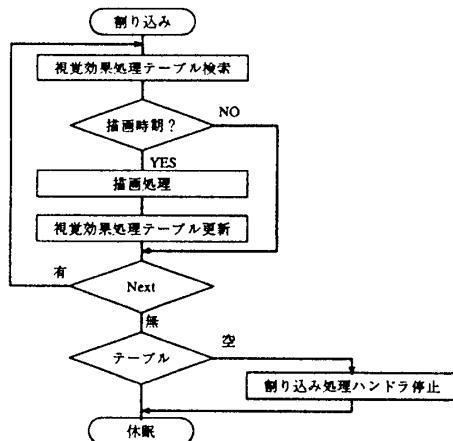


図 2: 割り込み処理ルーチン

ンドウ A, B, C の 3 つのウィンドウに対するリクエストのキューイングの例である。

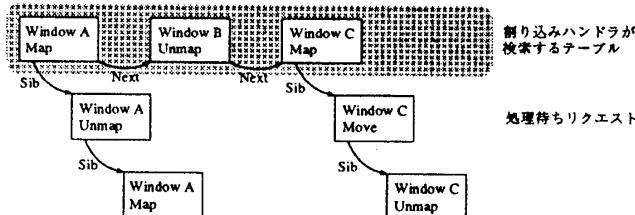


図 3: 視覚効果処理テーブル構造

これによって、視覚効果処理中のウィンドウに対する矛盾した処理をブロックすることが可能である。

4 評価と課題

NEC EWS4800/330(MIPS R4400, 67MHz, BitBlt 機構搭載) 上に視覚効果拡張をインプリメント、以下のようないきさつを得た。

- 視覚効果処理速度

サーバ内で視覚効果処理を行うことにより、600 × 400 ピクセル程度の ウィンドウに対して、ほぼビデオレートのなめらかな視覚効果処理が 実現できた。

- 非同期描画処理

Block/Wakeup ハンドラの利用により、視覚効果描画処理中に他のリクエストに対する処理が可能となつた。また、複数ウィンドウに対して同時に視覚効果処理を行うことも可能である。

- 改造規模

EC 操作部 (プロトコル拡張部)	約 3KL
描画処理ルーチン部	約 8KL
既存部改造箇所	15 箇所

関数のラッピングという方法を用いることで、既存部分への改造を少なくでき、移植性の高いプログラムとなつた。

- フルカラー及び動画ウィンドウへの対応

他の拡張機能との共存が可能であり、MVEEX 拡張との組み合わせによって ビデオウィンドウにも視覚効果を利用することが可能である。また、TrueColor へも対応しており、フルカラー ウィンドウに対しても視覚効果が利用できる。

- 既存アプリケーションへの対応

既存アプリケーションへの EC の設定及びバインドを行うツールを試作した。その結果、既存アプリケーションにも問題なく視覚効果が実現出来た。

既存のイメージ表示アプリケーション上で視覚効果処理を行っている例を図 4 に示す。

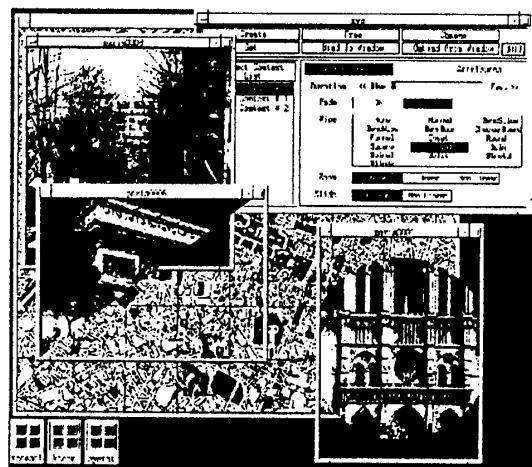


図 4: 視覚効果処理画面

X では、ウィンドウのオーバーラップによって隠されている領域が表示される際 リージョンという矩形領域の単位で描画を行う。複数のウィンドウをオーバーラップすると複数のリージョンに対する描画処理が発生することになる。そこで、アンマップリクエストの際にウィンドウのオーバーラップの状態によって 視覚効果処理の効果が一定にならないという問題がある。

5 おわりに

ここでは、視覚効果の描画処理ルーチンの実装方法について述べた。描画処理を非同期に実行するための仕組みとして Block/Wakeup ハンドラによる周期的な割り込み処理の利用と描画リクエストのキューイングを用い、スムーズな視覚効果描画処理を実現することができた。

参考文献

- [1] 高野元、松浦宏、的場ひろし X Visual Effect Extension(1)～視覚効果を用いたウィンドウ操作とそのアーキテクチャ～、情報処理学会第 49 回全国大会, No. 2V-5. 情報処理学会, 1994.
- [2] 加藤洋太郎訳、実装と移植 X Version11 Release 5