

平成21年 5月31日現在

研究種目：基盤研究（C）
 研究期間：2007～2008
 課題番号：19500094
 研究課題名（和文） マルチディスプレイによるメディア表現のための
 アプリケーション・フレームワーク
 研究課題名（英文） AN APPLICATION FRAMEWORK FOR MULTI-DISPLAY SYSTEM
 AS MEDIA REPRESENTATION PLATOFORM
 研究代表者
 太田 高志（OHTA TAKASHI）
 東京工科大学・メディア学部・准教授
 研究者番号：30386768

研究成果の概要：CG コンテンツの表現手段の拡大を目的として、表示プラットフォームとしてディスプレイ配置を柔軟かつ動的に構成出来るマルチディスプレイ環境を実現するための、アプリケーション・フレームワークの設計と実証実験を行った。共有空間を介して、対称関係にある独立した表示プロセスが、同一環境にアクセスすることにより、動的にマルチディスプレイ表示を構成することが出来た。また、表示しながらディスプレイの配置構成を変更させながら、表示内容がその変更に応じて対応することを実現した。派生研究として、プログラムの独立性を出来るだけ高めながらも、互いに連携して一つの処理を行うためのアプローチについての研究を行い、今後の目指す方向性についての示唆を得た。

交付額

(金額単位：円)

	直接経費	間接経費	合計
2007年度	2,300,000	690,000	2,990,000
2008年度	1,100,000	330,000	1,430,000
年度			
年度			
年度			
総計	3,400,000	1,020,000	4,420,000

研究分野：総合領域

科研費の分科・細目：情報学・メディア情報学、データベース

キーワード：グラフィクス、ネットワークコンピューティング

1. 研究開始当初の背景

(1) 我々は、水族館のVRによる再現を目指すAqua Projectを通じて、水槽を泳ぐ3Dの魚のモデル化とインタラクティブなシステムを開発してきた。しかしながら、PCの画面というプラットフォームの限界から、個人による閲覧という使用形態の限定や、実際の水族館のような広大な空間の意識を持たせることの困難を感じており、それらを解決する新たなプラットフォームとしてマルチデ

ィスプレイのもたらす広大な表現空間というものを利用したい希望を持っていた。

(2) マルチディスプレイに関する研究は、表示領域の物理的な面積の拡大と高解像表示の実現のためや、多人数の共同作業や情報共有のための大きなデスクトップ環境の実現等の用途などで、複数のディスプレイを上下左右に隣接配置して全体で一つの画面とする研究が多く行われている。これらは、流

体やタンパク質構造分析等の科学計算可視化結果の詳細表示や、多人数による可視化の分析、巨大なデスクトップによる共同作業環境の提供などを目的としているものが多い。我々は、マルチディスプレイを、単純に表示面積の拡大に利用するだけではなく、メディアアートやCGなどのコンテンツ作品を提示するための新たな表現を可能とすることに利用出来るのではないかと考えた。

(3) 単純な画面拡大ではない使用としては、例えば、円周状に外向き、もしくは内向きにディスプレイを配置し、ディスプレイをまたいで回遊する魚や鳥の表示や、サーキットを走行するレースカーの様子を表示するなどのアプリケーションが考えられる。また、ディスプレイ毎に異なる背景や環境を表現し、ディスプレイ間をまたがって移動する鳥や魚のモデルが環境の違いによる影響を受けて、異なるディスプレイ領域では異なる行動や反応が行われるような作品も出来るだろう。また、箱状にディスプレイを外側に向けて並べることで、立体をそれぞれの方向から見るにあたって、実際の感覚と合致した表示インターフェースを提供することも可能となるだろう。これらの例のように、ディスプレイの配置構成の自由度を持たせることで、画像的なりアリティだけではなく、空間的な配置や関係性を利用した表示形態を作り出せるのではないかと考えた。

(4) 複数のプロセスがネットワークを介して協働処理を行うシステムに関して、始めから特定の目的に合わせた設計で全体の設計を行うのではなく、一つ一つは独立して設計されたプロセス（プログラム）が共通の環境に参加することで、動的かつ自律的に全体のシステムが組み上がるようなアプローチに関心があり、そのような機構による実アプリケーション例の構築を行いたいと考えていた。

2. 研究の目的

(1) 本研究は、メディアアートやCGなどのコンテンツ作品を表示するための新たな表現プラットフォームとして、柔軟で動的にディスプレイの数や配置を構成出来るマルチディスプレイ環境を実現することを目的とする。マルチディスプレイ環境として、単純に大きな画面を構成することのみを目標とするのではなく、多様な配置や動的な変更が可能な環境を目指すものである。マルチディスプレイの実現は、ハードウェア、OSやウィンドウシステムなどのシステムソフトウェアによるアプローチを主として多様なアプローチで行なわれているが、我々は、柔軟

なシステムの実現ということ念頭に置き、分散コンピューティング環境とアプリケーション・フレームワークの提供によるアプローチを採用することとした。また、ディスプレイの配置を静的に固定されたものとするだけではなく、アプリケーションの表示を行っている途中でも動的に変更出来るようにし、その変更に表示内容も反応するようにしたいと考えた。それによって、表示環境としてのバリエーションを拡げることが出来るばかりでなく、新たなインタラクティブティを持った作品を制作するプラットフォームとして、従来とは異なったマルチディスプレイ環境の実現を目指すことを目標とする。

(2) マルチディスプレイの実現という直接的な目標の他に、複数のプロセスが協働して一つの環境や処理を形成するにあたって、システム全体の管理機構を用意するのではなく、独立した個々のプロセスが同一の環境に参加することによって、動的に連携動作が実現されるような「ゆるい」連携システムの実例としての位置づけも期待した。そのような個々のプロセス及びプログラムの独立性を目的とするのは、作品の作成を第一義とする者のためのプログラム作成の難易度を下げる目的が一つであるが、その他に、日常的な動作を通じて様々なデジタル情報とのインタラクションを提供するような、「環境」としての情報環境を構築するにあたって、始めから全ての機能の連携を設計するのではなく、個々に設計された新たな機能が、随時追加出来るようなシステム構築のアプローチに向かったプロトタイプとしたい希望があったためである。

(3) プログラミング・フレームワークとして、連携表示のためにネットワークを介した分散プログラミングを意識することなく、表示内容のプログラムに注力するだけで、動的なマルチディスプレイ環境を利用出来るような設計を目指す。

3. 研究の方法

(1) 2年間の研究期間で、当研究で目指すマルチディスプレイ方式の実現に向け、まずアプリケーション・フレームワークに必要な機能要件の確定と、初期設計を行う。それに基づき、プロトタイプの実装、及び、実証実験を行って、フレームワーク設計への知見を得ることと初期設計の課題の洗い出しを行う。最後に、プロトタイプ作成によって得られた課題点などを考慮して最終的なフレームワークの設計を行い、それに基づくアプリケーション例の作成を行う。全体として、この研究には二つの段階があると考えている。

まず分散コンピューターにより、互いのディスプレイの表示を情報のネットワークによる通信で連携させるための仕組み、CGのモデルが分散コンピューター間を移動し動的に実行されるための仕組みを構築することがまず一つある。次に、その仕組みが有効な具体的なシナリオに基づくコンテンツの創造が、もう一つの課題である

(2) まず初期設計として、連携表示を行うための分散コンピューティング構成を実現、管理する仕組みと、連携表示時の領域情報の管理機構、およびCGモデルの動作の処理と表示の担当の振り分け管理、等の処理について、等を考案しなくてはならない。ところで、コンテンツをマルチディスプレイで表示する主なアプローチとして二通りのものが考えられるだろう。一つは、一つの大きな仮想空間がディスプレイとは独立して存在しており、その部分領域を個々のディスプレイが担当して表示するものである(図1)。これをケース1と呼ぶことにする。もう一つは、仮想空間が個々のディスプレイの表示領域の集合として構成されるものであり、ケース2と呼ぶ(図2)。ケース1は、タイルディスプレイと呼ばれる、大画面を構成するためのマルチディスプレイ・システムで行われているようなものが相当する。必ずしも仮想空間全体を全て覆い尽くす必要は無く、間隔を空けたいくつかのディスプレイで仮想空間を覗き見るような表現も可能であろう。ケース2は、個々の空間を隣接して配置することによって、仮想空間をブロックのように組み立てていくようなものと言えるだろう。当面、本研究ではケース2について追求するものとする。

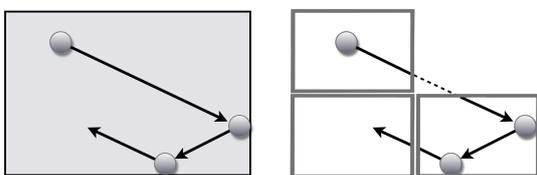


図0 仮想空間とディスプレイの関係(1)

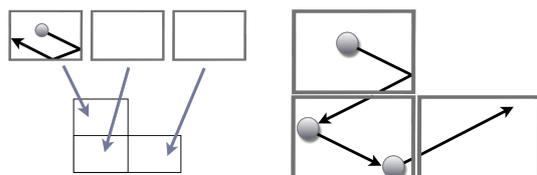


図0 仮想空間とディスプレイの関係(2)

(3) 初めからフレームワークを設計、開発

するのではなく、(2)で挙げた概念設計の機構がうまく機能するかどうかプロトタイプ・アプリケーションを作成し、その動作検証によって課題を確認し、フレームワーク設計の知見を収集する手順とする。まず、設計の検証のために、簡単な情報共有と、空間管理機構を埋め込んだ形で2DのCGを使ったアプリケーションを用意し、2、3台のディスプレイによる連携表示を試す。それによって、必要となる機能や、共有、交換すべき情報、PC間を移動するCGオブジェクト内に保持すべき属性情報の洗い出しを行う。

(4) (3)で作成したプロトタイプの動作実行の結果から、機能的な課題点やプログラム構成を再検討するなどを行い、最終的なフレームワークの設計指針を作成する。ただし、ここでも2年の研究期間ということもあり、ある程度限定した範囲における「最終設計」とならざるを得ない。ただし、それ自体、今後の拡張に繋がる基盤となるものの提案までを実現する。

(5) 設計したフレームワークを基に、プロトタイプで行ったアプリケーション例について再実装し、実行検証を行う。これによって、今後につなげる研究指針をまとめることが最後の課題である。本研究期間で行った対象は、2Dの自律的に動作するCGオブジェクトによるアプリケーションであったが、今後は、3Dや背景などの扱い、さらにはCGオブジェクトだけではなく、画像なども含めたものへの展開も考えられる。そのようなものを扱うための知見をまとめることは重要な成果となる。

4. 研究成果

(1) まずプロトタイプを作成した後、その検証結果の検討から次のようなフレームワークの設計指針をまとめた。

個々のPCは、それぞれの担当領域分のみ処理を行うように設定した。これは、あらかじめマルチディスプレイで表示することを考え全体を用意するのではなく、個々のPCは単体で完結するようにプログラムを設計することである。また、それらが互いに連携出来るためには、隣接の表示内容の影響が連携することが必要である。位置の管理も含めて、異なるPCの数だけ異なるプログラムを用意するのは面倒であるし、システムが固定化されたものになってしまう。そこで、個々のPC上で動作させるプログラムに及ぼす影響を最小限とするために、以下のようなプログラムの設計とした。

まず、個別の表示のための個々のPC上で動作するプログラムは基本的に同一のものとして

し、その処理は自分自身が表示する領域だけを担当することとする。マルチディスプレイとしてのシステムは、個々の PC がこの共有空間に参加することによって動的に形成される。個々のディスプレイの位置関係に合わせた CG オブジェクトのディスプレイ間の移動は、個々の表示領域に対する境界条件という形で設定することによって扱う。隣接ディスプレイへと移動していくようなオブジェクトに関しては、隣接への移動を指定する境界に達したときに、ローカルのオブジェクトリストから削除し、移動先の実行プロセスのリストに追加することで実現する。また、そのような情報のやりとりを行うために各プロセスがアクセス出来る共有の情報空間を用意する (図 3)。

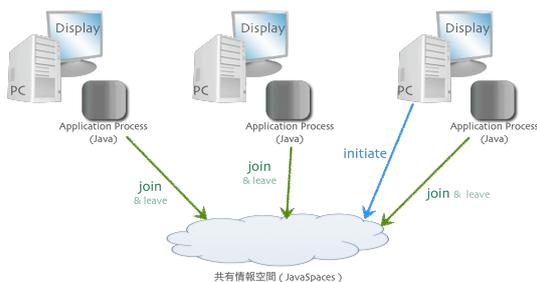


図 3 共有メモリ空間を介した連携

境界条件の設定については、例えば図 4 に示すように行う。図 4 の上図は、単独で実行されるプロセスに対する境界条件の設定である。表示領域が全領域と同一であるため、画面の境界は全て跳ね返る「壁 (Wall)」と設定する。他のディスプレイと隣接した場合には、図 4 の下に示すように、隣接するディスプレイ (2 番のディスプレイ: d#2 と指定することとする) との接続を指定する境界情景を設定する。このように、連携状況の違いを境界条件の指定に集約することで、CG コンテンツの主たる処理のプログラム部分を同一のものとする事が可能となる。

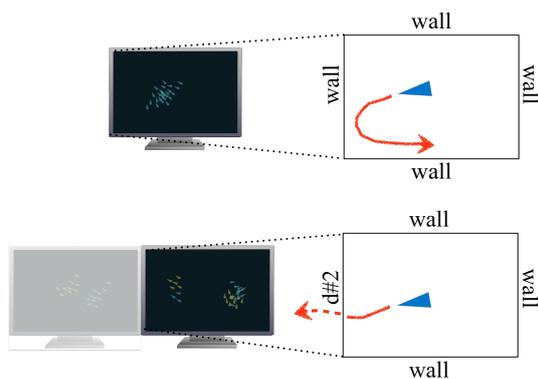


図 4 境界条件による隣接ディスプレイの位置関係

(2) 上で述べた仕組みを用意し、まずマルチディスプレイの基本的な機能として、複数のディスプレイを並べ、連携して一体となった表示が出来ることを確認した。最小構成ではあるが、3 台のディスプレイ (PC) を並べ、CG の円がディスプレイを超える際に、位置やタイミングがずれることなく、複数のディスプレイを全体の表示領域として、動作することを確認した。また、マルチディスプレイ環境の構築にあたっては、始めから隣接して配置し、プロセスを共通の管理下の下に一斉に動作させるのではなく、一つのディスプレイと PC を動作させておき、その隣に一つずつディスプレイを追加することによって動的に構成することが出来ることを確認した。これは、プロセスが共有メモリ空間に参加することによって達成されている。

(3) 表示する CG のオブジェクト数についての検証を行った。初期のプロトタイプ作成時は、個々の CG オブジェクトについて、初期にそのオブジェクトを生成させたプロセスが、表示されるディスプレイ領域に関わらず管理を続ける設計となっていた。この方法では、CG オブジェクトがディスプレイを移動した後に、その位置などの属性情報を、共有空間を介して通知し続けなければならなかった。そのため、オブジェクトの数が増加するにつれ、ネットワークの負荷により、動作が遅くなってしまい、数個以上のオブジェクトを扱えなかった。それを受けて、設計を変更し、個々のオブジェクトはディスプレイを移動する際に、担当する PC も変更することとした。移動に際して、個々のオブジェクトのその瞬間の属性を、共有空間を介して送信し、移動先で新たなオブジェクトとして再生するようにした。ちょうど CG オブジェクトを移動エージェントのように扱うこととしたのである。この設計変更の結果、図 5 に示した例では、200 の CG オブジェクトを同時に扱うことが出来たことを確認した。

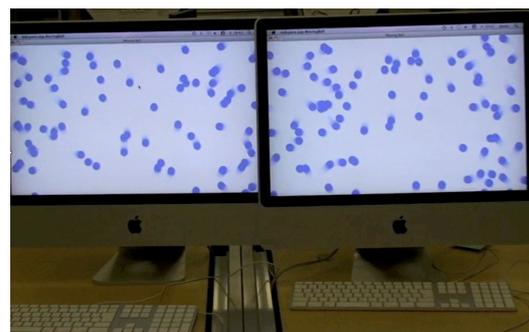


図 5 多数の CG オブジェクトの表示

(4) 新たなインタラクティブ性を実現するという目的であった、表示を行いながら動的にディスプレイの位置構成を変更する機能について、センサーによる簡単な位置検出と合わせたもので実証実験を行った。検証を行ったデモのシナリオは以下のようであった。

まず始めに一つの PC のプロセスが動作しており、円はそのディスプレイ内でのみ移動している。次に、始めの PC の右隣に他の PC を隣接して配置すると右側に表示領域が拡張され、CG の円はそちらのディスプレイとの境界を越えて運動するようになる。新たに追加された右の PC のディスプレイ領域に円が入っている間に PC を始めの PC の隣から取り除くと、二つのプロセスの連携が中断され、円は新たな PC のディスプレイ領域内のみ移動が制限されるようになる。この状態から再び隣り合う配置に戻すと、円は二つのディスプレイを一つの領域として移動することを再開する。さらに、右に配置した PC を今度は始めの PC の左隣に隣接配置すると、先に右側の領域として動作していた PC が、今度は左側の領域を担当して機能するようになる (図 6)。

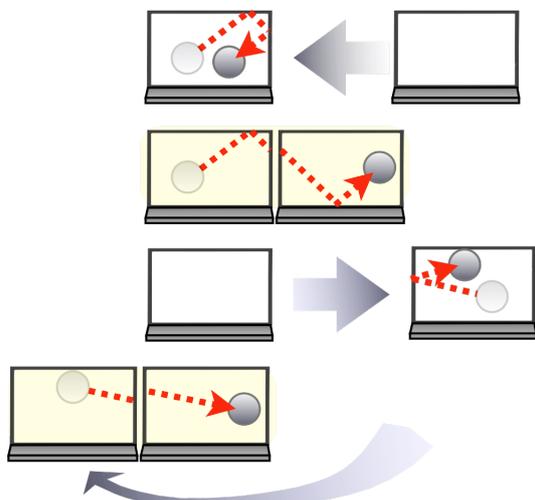


図 6 動的な配置変更の実行シナリオ

PC の位置検出には、RFID タグと検出器を利用した (図 7)。始めの PC の両隣の平面状に RFID タグを配置しておき、後から追加する PC に検出器を接続しておき、どの RFID タグを検出したかにより、始めの PC のどちらの隣に配置されたかを判断出来るようにしたのである。また、タグの認識が消えた時点をもって隣接から移動したことを判断した。これらのセンサー情報を利用し、共有メモリ空間へのアクセスを動的に管理することによって、マルチディスプレイ環境の動的な構成を制御した。また、位置の変更が検出された

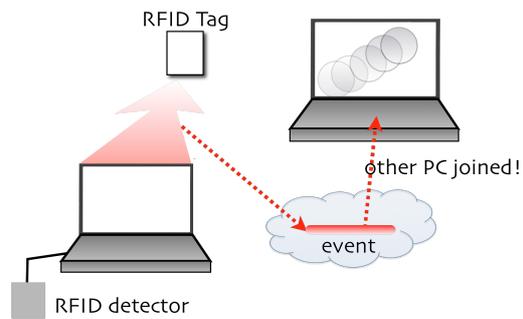


図 7 センサーを利用した位置の検出

場合には、それによって境界条件を変更することによって、新たな位置関係における連携に対応した。

この検証によって、我々の想定していた、動的なインタラクションを持つ、新たな CG コンテンツ表示のプラットフォームとしてのマルチディスプレイ環境という基本的なコンセプトの実効性について、ある程度の確認が出来たと考えている。また、独立に単独実行を行うプロセスが動的に協働する環境を創り上げるといったソフトウェア連動のアプローチについても、具体的な事例として提示することが出来たと考えている。

(5) アプリケーション例の構築から得た知見を基に今後の研究課題を以下のようにまとめた。

① 機能面として、表示の連携のための基本的な仕組みは実現出来たが、それ以外に以下の要素が、より自然な連携表示のために必要である。

- ・ ディスプレイのベゼル部の中の考慮
- ・ ディスプレイを移動するときに、CG が片側から徐々に消え、別の側に徐々に現れるようにするための一般的な工夫
- ・ 3DCG に対応した、視点情報の取り扱い
- ・ 異なる解像度やサイズのディスプレイの調整

また、表示媒体として PC ディスプレイだけを想定するのではなく、室内の環境を作るような目的を想定し、プロジェクター投影も含めて考慮して、上記の課題を含めた総合的な表示補正機構が望まれる。

② フレームワークの設計として、現状では単なる機能の提供に留まっているが、最終的に目指すのは、コンテンツ作成のユーザーへの便宜のため、単純な CG オブジェクトの表示と動作のプログラム部分だけを行い、それを組み込むだけで良いようにすることである。

③ 実用性を考慮した場合、台数やグラフィックス要素におけるスケーラビリティの検証

を行う必要がある。今回、PCの台数としては3台まで、またCGオブジェクト数は数百までの範囲で実験したに留まっている。10台程度のPCと、CGオブジェクトの数については、より系統的な調査が必要である。

④ソフトウェア連動のアプローチとして、現状では共有メモリ空間を設定し、そこへの情報の書き込みと読み出しを連携の手段とした。これは、プロセス間の直接的なロジックの呼び出し(RPC:リモートプロシージャコール)のような方法ではなく、より「ゆるやかな」ソフトウェア連携の手法を模索したものである。ここまでも、連携する個々のプロセスの独立性、対称性を念頭において設計を行って来ており、同一のプログラムによるプロセスが、管理機構の別プロセス無しで連携出来るようにしている。今後は、同一のプログラムだけでなく、異なる処理のプロセスも同様の「ゆるやかな」連携によって動作する機構を一般化したい。これは、多くの機構によって人をとりまく「環境」を構築していくにあたって、始めから全体を計画し構築するのではなく、個々の機能を一つ一つ部分毎に作成していったものが、徐々に大きな環境となっていくようにすることをサポートするためである。

⑤動的な構成を目指した理由として、CGコンテンツの表現の拡張があった。従って、表現の拡張をアピール出来るようなシナリオの実コンテンツの作成が重要である。多様なコンテンツ内容に対応した異なるディスプレイ配置のそれぞれに対して固有のシステムを用意することなく、同じ仕組みによって柔軟に対応出来るということが一つの利点として主張出来ると考えている。しかしながら、一つのコンテンツの内容自体で、動的な位置変更などの要素が従来の静的なディスプレイ環境では出来なかった表現を与えるような例を作ることが必須の課題である。

⑥生活の様々な場面において、「コンピューター」を直接使用するのではなく、食事、読書などの日常の動作をきっかけとして、コンピューターの機能を、コンピューターの操作を直接意識せずに利用するような、環境としての情報提供機能のような仕組みが今後の人と情報機器との関わり方として期待される。その際には、情報の表示機能もディスプレイに局在するのではなく、様々な形態と場所に偏在することが予想される。そのような拡張に向けての展開が、この研究の今後のより大きな枠組みとしての目的として考えられる。その際にはディスプレイの連携に留まらずに、様々な環境状態のモニター機能と、提示情報の多様性などを含めて、今回提示したような個々のプロセスの連携を可

能とするようなフレームワークの研究として発展すべきである。

5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[学会発表] (計6件)

① 太田高志、位置情報を利用した複数ディスプレイの動的な連携表示、第14回計算工学会講演会、May、2009、東京

② Takashi OHTA, “Dynamically Reconfigurable Multi-Display Environment for CG Contents,” ACE2008: International Conference on Advances in Computer Entertainment Technology, Dec. 2008, Yokohama

③ 太田高志、動的に構成可能なCG表現のためのマルチディスプレイ環境、日本バーチャルリアリティ学会第13回大会、Sep. 2008, 奈良

④ 太田高志、白山晋、モジュールの「ゆるい」結合によるプログラムの構築、第13回計算工学会講演会、May、2008、仙台

⑤ 太田高志、若林尚樹、高橋里奈、メディア表現プラットフォームとしてのマルチディスプレイ環境、第13回計算工学会講演会、May、2008、仙台

⑥ Takashi OHTA, and Susumu SHIRAYAMA, “Loosely-Coupled Construction of Scientific Computation Code”, APCOM' 07, December, 2007, Kyoto

6. 研究組織

(1) 研究代表者

太田 高志 (OHTA TAKASHI)
東京工科大学・メディア学部・准教授
研究者番号: 30386768

(2) 研究分担者

若林 尚樹 (WAKABAYASHI NAOKI)
東京工科大学・メディア学部・教授
研究者番号: 40254586

高橋 里奈 (TAKAHASHI RINA)
東京工科大学・片柳研究所・助教
研究者番号: 50386786