5ZA-6

ウィンドウの表現性を拡張するインターフェースの提案

持田光将†，深井佑希†，伊藤永悟‡，藤田光治‡，佐野勇司†,‡，藤本貴之†,††

東洋大学大学院工学研究科†　東洋大学工学部‡　東洋大学総合情報学部††

１．研究の概要

近年,三次元表示を可能とする３D ディスプレイが急速に普及している．極めて安価に３D コンテンツを楽しむ環境が整備されており,対応するソフトウェアの数も日々,増加している．

しかしながら,現在の技術では,三次元表示を行なうコンテンツは原則として,「画面全体」である．局部的な箇所だけを「立体化」する場合は,事前にコンテンツ／ソフト自体がそうなるように加工しておく必要がある．すなわち,リアルタイムな映像で局部的な立体化を実現することは難しい．そこで本研究では,３D ディスプレイを用いて,リアルタイムに画面全体ではなく,局区部的な箇所のみを立体視するシステムを提案し,試作した．

２．３D ディスプレイの現在的課題

一般に３D ディスプレイを用いて立体視用を行なう場合,ディスプレイに映し出されたコンテンツは,そのまま全てが立体化する（図１）．

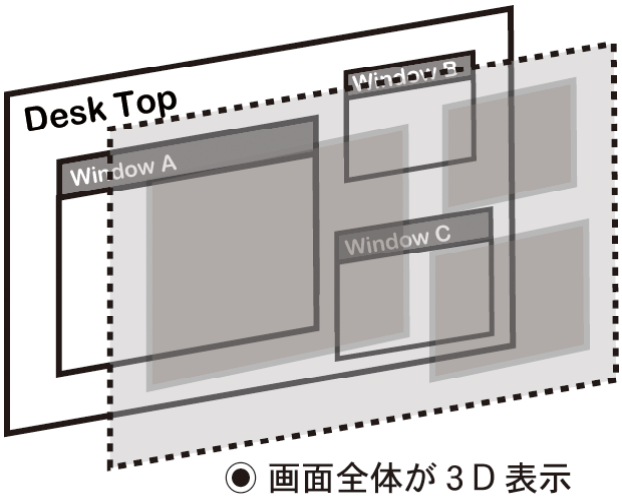

図１．一般的な３D ディスプレイでの立体視

そのため,ディスプレイ内の一部だけを局部的に立体化するといったことは困難である．画面の一部や必要箇所のみを立体化するように事前に３D 用映像として制作されたコンテンツであれば可能であるが,リアルタイムな映像や作業の中で,それを実現させる事は困難である．例えば,現在作業中のウィンドウやソフトウェアの部分だけを立体視さえ,その他を通常のフラットな画面状態にさせておく,といった３D 表示の運用を実現させるシステムや商品は著者らが知る限り存在していない．本研究で開発したシステムでは,コンピュータ操作を行なう場合に,特定の作業ウィンドウ（あるいはソフトウェア）のみを立体視させることが可能である（図２）．これにより,デスクトップ上に複数開いたウィンドウ（あるいはソフトウェア）のうち,映像のみを３D 表示にし,ワープロソフトやブラウザは通常のフラットな常時のままにしておくなど,３D 表示を計算機操作の効率化や表現性の拡張に利用することができる．

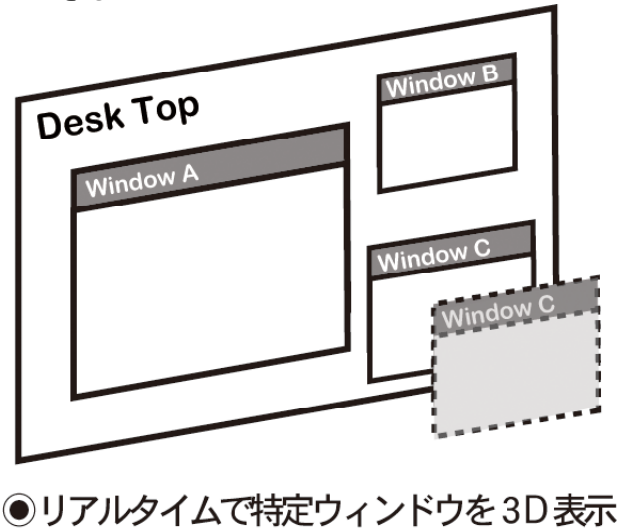

図２．本研究で提案・実装する３D 表示

３．立体視表示

３D ディスプレイを用いた立体映像の生成と取得には,一般に当該コンテンツを左右にズラして表示し,それをディスプレイと同期させた専用眼鏡で閲覧することで実現させる．立体映像を生成するために必要となる元コンテンツの左右のズレに関しては,一般に,快適な立体視を可能とするには,左右の映像のズレ量を 2.36 ㎝以下に,不快でない立体視を可能とするためのズレ量は 4.45cm 以下であるとされている（手前への立体視の場合）．また,３D ディスプレイと立体視用眼鏡の同期に関しては,現在市販されている一般的な３D システムの同期周波数は 120Ｈｚである．左右を 120Hz で切り替えるため,片方の映像では 60Hz となる．すなわち,ディスプレイと眼鏡の同期間隔,左右映像それぞれ 1 秒間に 60 回を交互に表示させることとなる．よって本システムでは,Windows OS のデスクトップ上で立体視をしたい当該オブジェクトを左右 2.36cm 未満の最適な距離でズラして表示させつつ,その左右の表示を１秒間に 60 回交互での表示とすることで,立体表示を実現する．

A proposal of Interface to expand Window's representation.

†Mitsumasa MOCHIDA, Yuki FUKAI: Graduate School of Eng., Toyo University.

‡Eigo ITO, Koji FUJITA, Yuji SANO: School of Eng., Toyo University.

††Takayuki FUJIMOTO: Dep. of Information Science and Arts,Toyo University

４．アクティブな２つの映像の生成について

アクティブとなっているウィンドウ（あるいはソフトウェア）を 2.36cm 未満の適切距離にダブらせて表示させ,それを３D ディスプレイ内で専用眼鏡と同期させた状態で閲覧することで,当該ウィンドウ（あるいはソフトウェア）をリアルタイムに立体視させる.

アクティブウィンドウがズレた 2 つの映像の生成フローを以下のステップで示す.

〔Step.1〕 マルチディスプレイで複数のデスクトップを以下の図３のように作成.

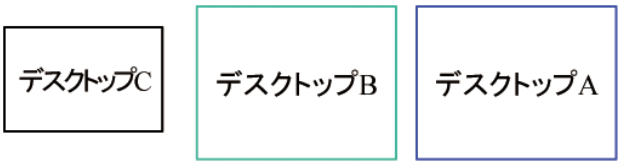

図３．デスクトップの設定

ここでは,デスクトップ A と B は全く同じ解像度・壁紙とし,アイコンなどは付加的なコンテンツは存在しないと仮定する．デスクトップ C を通常の作業領域用のデスクトップ(アイコンやタスクバーなどがある)とする.

〔Step.2〕 アクティブウィンドウをデスクトップ A に表示（図４）.

図４．アクティブウィンドウの表示①

〔Step.3〕 アクティブウィンドウをデスクトップ B にデスクトップ A にあった位置よりも 2.36cm 以内程度ズラして表示（図５）.

図５．アクティブウィンドウの表示②

〔Step.4〕 アクティブウィンドウをデスクトップ A に表示した後,デスクトップ B に 2.36cm 以下程度ズラして表示.

〔Step.5〕 上記２～4 のフローを交互に高速で繰り返して,アクティブウィンドウがズレた 2 つの映像を生成する（図６）.

図６．ズレた２つの映像の表示

５．アクティブウィンドウの操作について

同一ウィンドウを左右にズラして表示させるためには,アクティブになっているウィンドウの位置情報の取得し,視覚的に適正な速度で高速移動させることで,実現させる．その位置情報の取得と反映には Windows の API を用いて操作する.具体的には,C 言語を用い user32.dll の関数を利用する.

具体的には以下の手順で操作を実現する.

（１）GetForegroundWindow 関数で,アクティブウィンドウ(ユーザが操作するもののみ)のハンドルを取得する.

（２）GetForegroundWindow 関数で取得したハンドラと移動させるディスプレイ A の位置を SetWindowPos 関数に設定し,ウィンドウを移動させる.

（３）GetLocalTime 関数で時間を取得し,その時間に応じて, Sleep 関数で待ちの処理を行い,ディスプレイ同期させる.

（４）再度 GetForegroundWindow 関数でハンドラを取得し,SetWindowPos 関数にそのハンドラと移動させるディスプレイ B の位置設定し,ウィンドウを移動させる.

（５）再度 GetLocalTime 関数で時間を取得し,その時間に応じて,Sleep 関数で待ちの処理を行い,ディスプレイ同期させる.

（６）以上の(1)から(5)までを繰り返すことでそれぞれの位置に同じウィンドウが表示されている状態となる.

５．試用実験と考察

実装したシステムを用いて,試用実験を行なった．実装したシステムでは,ディスプレイとの同期タイミングなどを考慮した結果,ウィンドウのズレ幅は 2 ドットから 4 ドット,左右間の高速移動の待ち時間は８ミリ秒で行なった.

尚,試用実験には,プログラムの実行用に acer 社製のノート型計算機・ASPIRE AS3810T-H22（Windows OS ／ Core 2 Duo SU9400 1.4GHz（3MB）／メモリ：DDR3 PC3-8500 2GB）を利用し,３D ディスプレイには,ZALMAN 社の ZM-M215W（21.5inch／16:9 ワイド）という一般的な民政機を用いた.

結果から,概ね特定ウィンドウのみの立体視を実現することができた．しかしながら,現状では,アクティブなオブジェクトを高速での短距離移動をループさせることで擬似的にズレた状態を生成しているため,利用環境によっては,それぞれのウィンドウにチラつきが発生する場合がある．また,複数のウィンドウを利用する場合にスムーズな利用する場合の処理にも課題があるので,今後,検討したい.

参考文献

[1]狩野育史，“運動視差による立体映像：残像式立体映像”，三次元映像のフォーラム，2006
[2]西川善司，“各立体方式の基本技術とその課題,業界の流れを理解する 立体視対応ディスプレイのしくみと技術的問題”,インターフェース 37(1)，44-55，2011-01，CQ 出版社