

操作レンズ型デバイスのソフトウェア処理系の検討

A Software Design of Operation-Lens System
having Manipulation Support山口 巧^{†‡§}
Takumi Yamaguchi岡本 高幸[§]
Takayuki Okamoto島村 和典^{†‡}
Kazunori Shimamura

1. まえがき

日頃、情報通信ネットワークとの関わりの少ない人であっても、複雑な操作感を感じることなく、安心し、安全に情報通信ネットワークを利用できる環境を実現することが重要であり、人間とコンピュータとのインタフェースの課題解決が重要となっている [1]。しかし、年齢による格差が大きく、高齢になるほど利用率は低い。また、高精細なコンピュータディスプレイの普及により、Graphical User Interface (GUI) の高密度化が進んでいる。これは、視力の低下した高齢者は勿論、一般のユーザに対しても視認性の点で問題がある。これらの格差や問題に対して、筆者らは注視域を取り出して手元のデバイスに拡大表示することにより、ユーザの操作や視認を支援しながら撫でる動作で操作する操作レンズ型デバイス (OPR-LENS: Operation-Lens) システムを提案している [2]。本稿では、ポインティング操作機能と部分取り出しによる画像の拡大機能において、それぞれに要求されるソフトリアルタイム性には違いがあることを考慮し、実装システムのソフトウェア処理系の評価を行ったので報告する。

2. OPR-LENS システムの位置付け

本研究は、「視覚・操作支援をする」ことを実現しつつ、「押し付けがましくなく、ゆったりと操作する」簡便なユーザインタフェースを有するシステムの実現が目的である。OPR-LENS システムは、PC(Personal Computer) だけに留まらず、PCを含む広義の情報アプライアンスの設計公理に基づく [3]。OPR-LENS システムは、WIMP インタフェースに基づいた GUI に対して、能動的にアクセスできる半球形をした把持具型の入力インタフェースとその支援システムで構成される。ユーザ操作を容易に且つ支援するという観点から、対象ユーザの幅と操作インタフェースとしての自由度を軸にした本研究の位置付けを図1に示す ([4],[5] 他)。図1の中心点は、PCの操作インタフェースである汎用マウスとした。OPR-LENS システムは、ユーザ操作の自由度をあまり高くせずにユーザの対象範囲を広くするものである。多くのユーザは、複雑なボタンを操作したり、高度な機器そのものを使いたいわけでは決してなく、自然な形で情報や商品が得られ、快適に生活ができればいいはずだからである。また、注視域の OPR-LENS デバイス側への取り出しは、「ユーザの希望に合わせて手で確認したくなったら視認する」といった至ってユーザ任せなものであり、超小型携帯コンピュータの場合の「大きなオブジェクトの一部を覗き

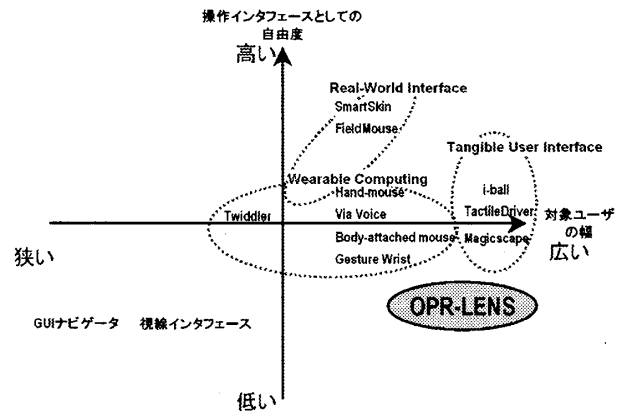


図1: OPR-LENS システムの位置づけ

込む」ようなユーザモデル [6] は想定していない。実世界指向インタフェースやウェアラブルコンピューティングとは違った、どちらかという、力覚あるいは触覚型ユーザインタフェースに近いインタラクションを狙えるものと位置づけている。

OPR-LENS システムの PC 操作を想定した構成を図2に示す。OPR-LENS モジュールのマウスカーソル周辺に注視域としてフォーカス領域を設定し、その領域内のデスクトップ画面を OPR-LENS デバイスに実装する表示部に部分拡大表示する。OPR-LENS デバイスは通常のマウスと同等の機能項目を持たせる。拡大画像は操作支援の効果を高めるように注視域を非線形拡大 [7] させる。OPR-LENS モジュールと OPR-LENS デバイスを接続するインタフェースは通常のマウスが GUI 上で実現している描画時間間隔レベルを満足するよう決定する。OPR-LENS システムは、OPR-LENS モジュールと OPR-LENS デバイスに搭載するサーバ・ビューアソフトウェア部、および OPR-LENS デバイスのハードウェア部で構成する。ソフトウェア部の全体構成を図3に示す。注視域の画像取り出しとその転送機能、ならびに操作イベントの通知と接続対象の自動検索機能を持たせる。OPR-LENS モジュールは主にサーバ機能を有するソフトウェアで構成する。それ以外のユーザ系の支援系ソフトウェアと操作するハードウェアは OPR-LENS デバイスが受け持つことで機能構成を簡明にできる。OPR-LENS モジュールと OPR-LENS デバイスとのデータの送受信は、TCP/IP 上の独自プロトコルで実現する。OPR-LENS モジュールと OPR-LENS デバイスの接続インタフェースは IEEE802.x, USB もしくは IEEE1394 など TCP/IP に適合する系を用いる。

[†]高知工科大学, KUT[‡]情報通信研究機構, NICT[§]高知工業高等専門学校, KNCT

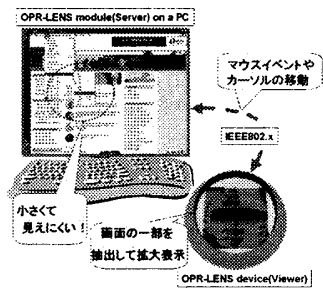


図 2: PC 操作を想定した OPR-LENS システムの概要

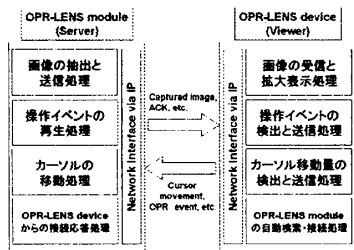


図 3: OPR-LENS システムのソフトウェア構成

3. 画像取出し手法

PC のデスクトップを遠隔操作するリモートデスクトップシステムは種々あるが [8, 9], 転送データ量やデータ圧縮・伸張時間が大きく, デスクトップ全画面をキャプチャするとき画面の殆どの部分が急激には変化しないことを前提として設計されている。一般的に PC のデスクトップ画面は動画再生などの場合を除いて静的表示が多く, リモートに取り出したキャプチャ画面の再描画がゆっくりでも操作に大きな支障の出ないものが多い。リモート操作の用途はサーバ機の設定変更やログデータの閲覧などがその代表例である。これに反して, 本研究の提案するシステムでは, ユーザ操作に連動し, 且つその操作を妨げない高速性と追従性が要求される。本システムの画像取出しは, OPR-LENS デバイスで表示される領域サイズに合わせて, ユーザの拡大倍率に応じた OPR-LENS モジュールの取り出す画像サイズを切り替えてキャプチャする方法を採用した。この方法は OPR-LENS モジュールで拡大倍率に応じた処理が必要になるものの, 拡大倍率が上がるとともに転送データ量が減少し, 本システムの転送系全体の処理時間を向上させ得る。

4. 設定プロトコル

前節で選択した画像取出し手法と本システムの必須機能ならびに自動検索機能を実現するための設定プロトコルを図 4 に示す。OPR-LENS モジュール検索処理を行った後, 画像データ送受信用コネクション (Port:8081), マウスイベントデータ送受信用コネクション (Port:8082), カーソル移動データ送受信用コネクション (Port:8083) を設定する。初期設定後, Port 毎に独立してデータを送受信する。画像データとカーソル移動データは周期的に

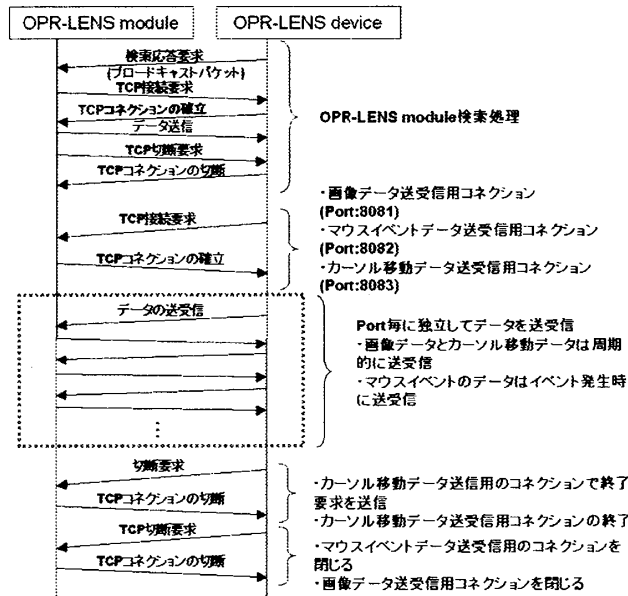


図 4: 設定プロトコル全体の流れ

送受信し, マウスイベントのデータはイベント発生時に送受信する。

OPR-LENS モジュールは OPR-LENS デバイスのリクエストに応じて ACK か PC 上の GUI の部分画像を OPR-LENS デバイスに送信する。拡大倍率 1 倍時に解像度 XGA に対して同アスペクト比の 1/12 領域, 256 × 256 ピクセルをデフォルトで送信する。リクエストは, OPR-LENS デバイス接続時のみ発生する初期化通知, OPR-LENS モジュールから取り出し画像を送信要求する画像データ要求, OPR-LENS デバイスでユーザが指示したマウスイベント通知, カーソル移動量通知の 4 つである。このうち画像データ要求とカーソル移動量通知は OPR-LENS デバイスのタイマ処理によって一定周期 T ごとに OPR-LENS モジュールに通知される。マウスイベント通知はユーザ操作によって不定期に随時発生する。本システムでは, TCP ポートを 3 つ使用しており, OPR-LENS モジュールと OPR-LENS デバイス間のリクエストのやりとりと OPR-LENS モジュールからの画像データ送信 (Image) は異なるポートを使う。これにより, コネクションは制御用とデータ転送用に分けられている。

5. サーバ・ビューアソフトウェア

OPR-LENS モジュールのサーバプログラムの流れを図 5 に示す。起動後, 参加ネットワークから割り当てられる IP アドレスの取得, TCP ポートなどの初期化処理をしたのち, OPR-LENS モジュールからの接続要求待ちとなる。OPR-LENS モジュールから接続要求を受け取るとサーバはビューアに接続し, 通信を開始する。接続完了後, OPR-LENS モジュールは GUI 上のマウスカーソルを中心としたフォーカス領域を取得できるようにタスクを常駐化する。ビューアからのイベント受信待機状

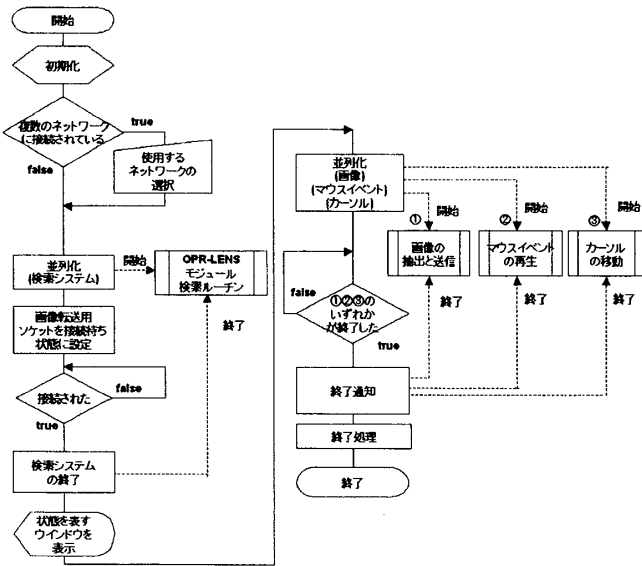


図 5: OPR-LENS モジュールのサーバプログラムの流れ

態となる。待ちうけスレッドは3つで、画像抽出、マウスイベント、カーソル移動量の各イベントに応じたマルチスレッド処理になっている。これ以降は終了要求するまで設定プロトコルに従い通信を行う。

OPR-LENS デバイスのビューアプログラムの流れを図 6 に示す。サーバ側が接続待機中であること確認し、通信を開始する。同時に、サーバ側とのカーソル位置座標を一致させるための初期化を行う。これは、ビューアからカーソル移動量の通知をサーバに行い、連動させるための準備である。同時に、マウスクリックの不定期なマウスイベントの割り込み処理を開始する。スレッド処理を行い、カーソル移動量と拡大倍率をユーザが変化させた場合のスレッド待ちとなる。その後、ビューアは予め設定された時間間隔でタイマ割り込みを行いながら設定プロトコルに従い処理を行う。ビューアは操作ユーザからのイベント要求以外は、サーバから受信した画像データをデコードし、拡大処理を施してビューア上の指定領域に描画を続ける。サーバとビューア間の転送画像の画像形式は、ファイルサイズと圧縮・伸長時間のトレードオフから JPEG 形式を採用している。

OPR-LENS システムの自動検索機能はビューア側の検索プログラムとサーバ側の応答プログラムの2つによって成り立っている。本機能のフローを図 7 示す。以下、流れを説明する。①サーバ側で応答プログラムが起動されブロードキャストパケットの待ち受け状態になる。②ビューア側で検索プログラムを起動され検索が実行されると、検索プログラムは応答プログラムが待ち受けを行っているポート、デフォルトは 18000 ポートに対してブロードキャストでデータを送信し、すぐに TCP の接続待ち受け状態に入る。ブロードキャストでは相手に確実に届くことは確認できないがネットワーク上のすべてのノードに対して同時に通信をすることができる。③ブロードキャストパケットを受け取った応答プログラムは

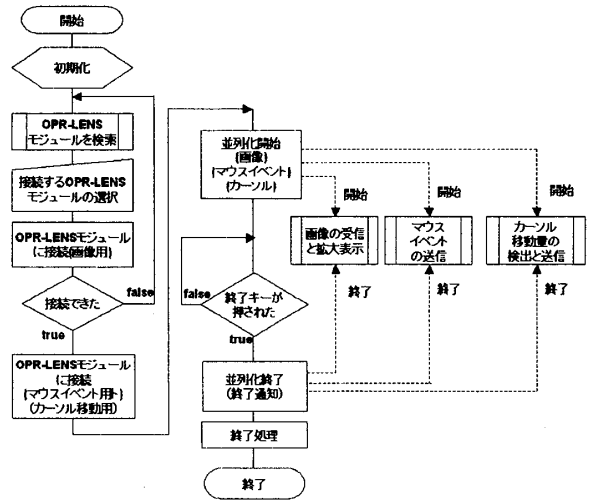


図 6: OPR-LENS デバイスのビューアプログラムの流れ

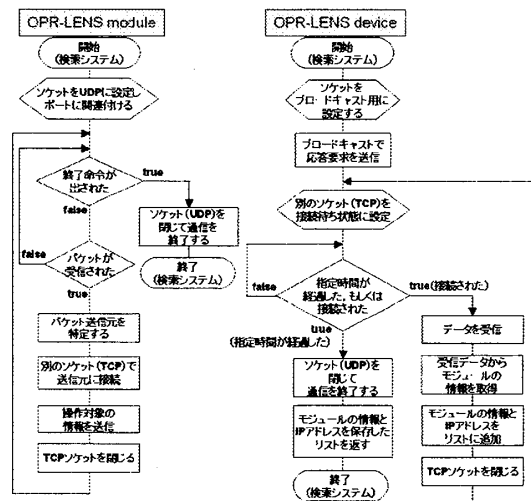


図 7: 自動検索処理の流れ

そのパケットから送信元を検出し、送信元に対して TCP の接続要求を行う。要求を行うポートはあらかじめ決められておりデフォルトでは 18002 ポートに設定している。④TCP での接続が確立されると応答プログラムはそのプログラムが起動されているサーバの情報を送信して通信を終了する。検索プログラムは接続要求を出してきたサーバの IP アドレスと送信されてきた情報をリストとして得ることができる。

6. ソフトウェア部の要求性能と評価

リアルタイム性を満足することは、応答速度が速いことではなく、厳密には、システムに課せられる時間的制約が満足できることを意味する。よって、リアルタイム性が要求されるシステムの設計ではその時間的制約が具体的にどの程度の時間であるかを知る必要がある。OPR-LENS システムには、ポインティングデバイスと

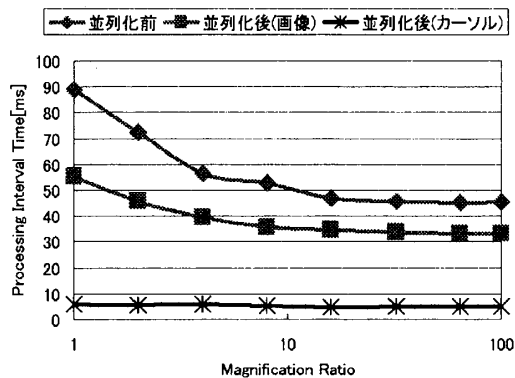


図 8: インターバル時間の実測値 (Web ページ背景)

としての側面と拡大表示機能をもったディスプレイとしての側面がある。ここでは、要求される最大処理時間の具体的な基準値として次の設定をした。ポインティングデバイスとしては、USB マウスの最大サンプリング周波数 125[Hz] を、拡大表示機能を持つディスプレイとしては、VGA サイズで 30[fps] を目標とする。インターバル時間として、ポインティングデバイスとしての処理は 8[ms] 以内に、ディスプレイとしての処理は 33[ms] 以内に終了しなければならないことになる。

一方、OPR-LENS システムのソフトウェアでは、画像の処理、カーソルの移動処理、マウスイベントの処理をすべて並列に行う。これは、それぞれの処理の完了に要する時間に大きな隔たりがあるためであり、また、それぞれに要求されるリアルタイム性 (時間的制約) にも違いがあるためである。アプリケーションレベルで表現できる並列処理は、CPU のアーキテクチャによる並列化とは違い、OS によって CPU 消費時間がそれぞれの処理に分配され、すべての処理が少しずつ順番に処理されていくという仮想的な並列処理である。そのため、並列化を行うとそれぞれの処理に要する時間は長くなってしまふ。しかし、カーソルとマウスイベントの処理は、処理を行う周期が短くなる。これは、画像の処理中に待機していた時間を省略できたからである。このように、並列化を行うことにより、短時間で完了する処理は短い周期で実行することが出来るため、カーソルの移動やマウスイベントのリアルタイム性が大きく向上する。

6.1 リアルタイム性の評価

並列化を行う前後で、システムのリアルタイム性がどのように変化したかを実験により測定した。拡大倍率を 1 倍から 100 倍まで変化させて、並列化前のシステムでは、画像、カーソル、マウスイベントのすべてを合わせた処理の周期を測定し、並列化後のシステムでは、画像の周期とカーソルの周期をそれぞれ測定した。カーソルの動き方によって、背景画像が変わってしまい、画像データの大きさが変化してしまう可能性がある。カーソルの移動はプログラムによって制御しながら画面上の 20 箇所測定し、背景画像は同一 (Yahoo!Japan の Web ページ) にした。拡大倍率を横軸にしたインターバル時間の測定結果を図 8 に示す。並列化前と並列化後ではカーソ

ルの移動処理の周期が大きく短縮されているのがわかる (並列化前のカーソル移動の周期は全体の周期にほぼ等しい)。各倍率におけるカーソル移動時間の周期の平均値を求めると 5.4[ms] であり、前述の目標値、8[ms] よりも短い。よって、ポインティングデバイスとして要求されるリアルタイム性は満たしているといえる。画像部の処理間隔について見てみると、並列化後の方が並列化前よりも処理時間が短くなっている。これは、並列化後のプログラムでは並列化以外の改良も加えられているためである。ユーザに対して別途報告を行った主観量評価結果 [10] で最も良いとされた拡大倍率 5 倍前後では、画像の周期は 40[ms] 以下であることがわかった。これは前述の目標値、33[ms] を超えてしまっているが、フィルム映画の 1 フレームの時間である 40[ms] とは同程度もしくはそれよりも短い時間であり、十分に滑らかに表示されていると考えられる。

7. まとめ

本稿では、OPR-LENS システムにおけるポインティング操作イベントと画像転送のソフトウェア処理系に関する実装と性能評価を行った。ソフトウェア構成を示し、GUI 画像の部分取出しの手法と転送プロトコルについて明示した。拡大倍率に応じて OPR-LENS モジュール上のキャプチャサイズを変化させ、OPR-LENS モジュールと OPR-LENS デバイス間の転送量を少なくした。また、OPR-LENS デバイス側で OPR-LENS モジュール上のイベントを一元的に並列化制御し、OPR-LENS モジュール側のソフトウェア部の処理を効率化できた。OPR-LENS システムを構築し、ソフトリアルタイム性が確保されていることを確認した。今後は処理の改良による CPU 負荷の軽減などが求められる。

今後はユーザの操作条件緩和を目的とした適応型システムへの拡張や文字入力装置としての可能性について研究を進める。

参考文献

- [1] 総務庁: 情報通信白書平成 15 年版 (2003).
- [2] 山口巧, 芝治也, 島村和典: 情報アプライアンスを遠隔操作支援する操作レンズ型デバイス, 情報処理学会全国大会, Vol. 66,4A-3, pp. 45-46 (2004).
- [3] D.A.Norman(岡本, 安村, 伊賀訳): *The Invisible Computer*, 新曜社 (2000). (パソコンを隠せ, アナログ指向でいこう).
- [4] Ikeda, H., Naemura, T., Harashima, H. and Ishikawa, J.: i-ball: Interactive Information Display like a Crystal Ball, *Conference Abstracts and Applications of ACM SIG-GRAPH 2001*, p. 122 (2001).
- [5] Division, H. D.: Magicscape, *HITACHI IT convention* (2003).
- [6] Siio, I.: Scroll Display: 超小型情報機器のための指示装置, 情報処理学会, Vol. 39, No. 5, pp. 1448-1454 (1997).
- [7] Keahey, T. A. and Robertson, E. L.: Techniques for non-linear magnification transformations, *IEEE Symposium on Information Visualization*, pp. 38-45 (1996).
- [8] Richardson, T., Stafford-Fraser, Q., Wood, K. R. and Hopper, A.: Virtual Network Computing, *IEEE Internet Computing*, Vol. 2, No. 1, pp. 33-38 (1998).
- [9] <http://www.6.ibm.com/jp/pspjinfo/javadesk/>: IBM Corp. Desktop On-Call.
- [10] Yamaguchi, T. and Shimamura, K.: A proposal of a palm-pointing device using lens-effect with partially capturing of PC's GUI, *Proceedings of APSITT2003*, pp. 25-30 (2003).