視線情報を用いた背後参照情報の透過提示システム

吉田 圭佑[1,a)]　大囿 忠親[1,b)]　新谷 虎松[1,c)]

概要：プログラミングなどの PC 作業では，複数の情報を参照しながら作業すること，すなわち参照作業がおこなわれる．しかし，アプリケーションウィンドウの大きさや位置により重なりが発生し，参照したい情報が見えなくなる場合がある．本研究では，このような情報を背後参照情報と呼ぶ．この場合，従来では背後参照情報を表示しているアプリケーションウィンドウを前面に出すなどのウィンドウ操作が必要であった．このようなウィンドウ操作は，主となる作業の中断につながり，作業効率の低下につながる恐れがある．本研究では，視線情報を用いることで主となる作業と並行して，背後参照情報を提示するシステムを開発した．本システムによって，背後参照情報が従来よりも少ないウィンドウ操作で可能になると期待される．

A Translucent Exposure of Backward Reference Information with Eye Tracking

Keisuke Yoshida[1,a)]　Tadachika Ozono[1,b)]　Toramatsu Shintani[1,c)]

Abstract:
Users performs tasks on refering information such as programming, we call it reference works. When a window is overlaid with another window, some information in the window become hidden. We call the information backward reference information. Then, users must manipulate windows such as activation to see backward reference information in the window. The window manipulation could decrease efficiency of works because of the interruption of tasks. In this study, we developed a translucent exposure of backward reference information with eye tracking. Our system may enable users to refer information of backward windows with less window manipulations.

1. はじめに

近年，主流となっている OS のウィンドウシステムでは，オーバーラッピングウィンドウレイアウトが採用されている．オーバーラッピングウィンドウレイアウトでは，アプリケーションウィンドウの重なりを許し，アプリケーションウィンドウの位置および大きさを自由に設定することができる．ユーザーが PC 上であるタスクを遂行する場合，利用するアプリケーションの組み合わせおよびデスクトップ上のアプリケーションウィンドウの配置によってタスク遂行環境を構築する．本稿では，このようなタスク遂行環境をワークスペースと呼ぶ．オーバラッピングウィンドウレイアウトでは，その自由度の高さから，ユーザの熟練度によりワークスペース内のアプリケーションウィンドウの可視性が高くなることが期待されている [1]．

我々は，オーバーラッピングウィンドウレイアウト上で，複数のアプリケーションを用いた作業の支援について研究をしている．特に，他のアプリケーションに表示された情報を参照しながら，主のアプリケーション上で作業すること（以降，参照作業）の支援のために，参照対象の情報（以降，参照情報）の提示手法について研究している．これまで，参照情報の提示方法および提示位置決定手法について，最前面，半透明かつクリックイベント透過による提示手法，画像処理技術により決定する手法を提案している [2]．また，ユーザの作業履歴と画像処理技術を併用する

[1] 名古屋工業大学大学院 工学研究科 情報工学専攻
Department of Computer Science, Graduate School of Engineering, Nagoya Institute of Technology
[a)] kyoshida@toralab.org
[b)] ozono@toralab.org
[c)] tora@toralab.org

ことで，ワークスペース内のアプリケーションウィンドウの可視性を向上させる自動ウィンドウマネジメントシステムについても提案している [3]．しかし，いずれもユーザが必要な参照情報の選択インターフェースについて改善の余地がある．

参照作業では，主に入力作業をおこなう主アプリケーションおよび情報の参照など副次的な利用のための副アプリケーションを併用する．また，参照作業は作業フェイズおよび参照フェイズからなり，それぞれ交互に繰り返すことで遂行される．システム開発を例に挙げると，Web 上の API を閲覧しながらプログラムを記述することが参照作業にあたる．プログラムを記述するための統合開発環境および Web 上の API を閲覧するための Web ブラウザが，それぞれ主アプリケーションおよび副アプリケーションにあたる．システム開発の例での作業フェイズは，統合開発環境上でプログラムを記述することである．また，参照フェイズは，Web ブラウザで Web 上の API を閲覧することである．ここで，モニターの大きさおよび解像度が限られていることから，統合開発環境のウィンドウが Web ブラウザのウィンドウに重なる場合がある．この場合，Web ブラウザのウィンドウ内に表示されている情報の一部は，統合開発環境の背後に隠れてしまう．本研究では，ウィンドウの重なりによって隠れた参照情報を背後参照情報と呼ぶ．従来は，背後参照情報を閲覧するためには，背後にある副アプリケーションを一度前面に出すなどのウィンドウ操作が必要であった．また，主アプリケーションでの作業に戻るときには，背後にある主アプリケーションをもう一度前面に出す必要があった．このように，参照作業時に背後参照情報を閲覧するためには，複数のウィンドウ操作の必要があった．つまり，背後参照情報の閲覧のためには，一度作業フェイズから参照フェイズに移行しなければならず，これは作業の中断につながる．参照作業において，作業の中断は作業効率の低下につながる恐れがある．

本研究では，視線情報に着目し，背後参照情報の透過提示システムを開発した．本研究の目的は，作業フェイズを中断することなく，背後参照情報を提示することで参照作業を支援することである．本システムでは，視線によって透過提示する背後参照情報を指定する．これによって，背後参照情報閲覧のためのウィンドウ操作が減少し，参照作業の効率化が期待できる．本稿では，2 章で参照作業支援について関連研究の紹介および考察について述べる．3 章では透過提示する参照情報の指定手法について述べる．4 章および 5 章では，それぞれ開発システムについてシステム構成および実行例を示し詳細を述べる．6 章では本研究における考察および，考察を踏まえた今後の展望について述べる．7 章で，本研究をまとめる．

2. 関連研究

参照作業を支援するアプローチとして，ウィンドウ操作の減少およびワークスペースの可視性の向上がある．以下では，それぞれのアプローチについて，またそれについての考察を述べる．

ウィンドウ操作の減少することで，PC 上でのアプリケーション操作およびウィンドウ操作を支援する試みがされている．PC 上でのタスクの遂行のためには，複数のアプリケーションを利用する場面が想定され，多くのウィンドウ操作が行われることが知られているためである [4]．山中らは，マウスカーソルに z 軸方向の移動を加えることで，アクティブウィンドウの背後のアプリケーション操作を可能にした [5]．これにより，本来背後のアプリケーション操作のために必要なウィンドウ操作を削減することによって，背後のアプリケーション操作を伴うタスク遂行の時間短縮を実現した．柴田らは，作業毎にアプリケーションウィンドウをまとめることで，一括操作を可能にした [4]．これにより，複数のアプリケーションを利用する作業において，タスクの切り替え操作を支援しウィンドウ操作数を削減した．いずれも，ウィンドウ操作数を減らすことによって，あるタスクの完了時間を短縮している．このように，ウィンドウ操作数の減少は，PC 上での作業支援につながる．ここで，ウィンドウ操作はウィンドウの重なりによって隠れたウィンドウ内の情報を閲覧する場合にも必要である．上記の手法では，背後に隠れたウィンドウ内の情報を閲覧することについては着目していない．このことから少ないウィンドウ操作によって背後に隠れたウィンドウ内の情報を閲覧する手段が必要である．

また，アプリケーションウィンドウの透明度に着目して，ワークスペースの可視性を向上させる研究がされている [6][7]．Waldner らは，アプリケーションウィンドウ内の情報の重要度に着目し，透明度を変化させることで背後のウィンドウ内の情報の閲覧を可能にした．また評価実験により，背後のウィンドウ内の情報の閲覧を可能にすること，すなわちワークスペースの可視性を向上させることで，参照作業の完了時間を短縮できることを示した [6]．このように，背後のウィンドウ内の情報の閲覧を可能にすることは，PC 上での作業支援につながる．しかし，これらはユーザの意図とは関係なく透明度が変化するため，必ずしも背後のウィンドウ内の閲覧したい情報が閲覧できるとは限らない．この場合，従来と同じようにユーザはウィンドウの位置および大きさの変更といったウィンドウ操作を行う必要がある．このことから，ユーザが意図的に背後のウィンドウ内の情報を閲覧できることが必要である．

以上より，少ないウィンドウ操作かつ意図的に背後のウィンドウ内の情報を提示するシステムが必要である．我々は

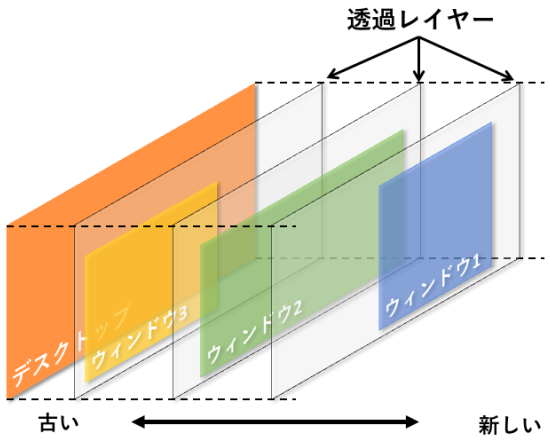

図 1　オーバーラッピングウィンドウレイアウトにおけるウィンドウの重なり

Fig. 1　Layers on The Overlapping Window Layout

視線情報に着目することで，少ないウィンドウ操作かつ意図的に背後のウィンドウ内の参照情報を提示するシステムの実現を目指す．つまり，視線情報によって背後参照情報を選択し透過提示する．特に，視線情報による背後参照情報の選択手法について 3 章で述べる．

3. 視線情報を用いた透過提示する参照情報の指定手法

参照作業では，キーボードおよびマウスでの入力作業と並行して情報を参照できることが望ましい．視線情報はキーボードおよびマウスによる入力作業と並行して利用ができる．視線情報を用いることで，入力作業を中断せず，閲覧したい参照情報の指定ができる．参照作業における参照フェイズと作業フェイズの並行が可能になるため，参照作業を支援できると考えた．本章では，視線情報を用いた透過提示する参照情報の指定手法について述べる．

背後のウィンドウ内の情報を透過して提示にするためには，視線情報によって任意のウィンドウおよび領域を指定できることが必要である．図 1 は，オーバーラッピングウィンドウレイアウトにおけるウィンドウの重なりのイメージ図である．オーバーラッピングウィンドウレイアウトでは，アプリケーションウィンドウは図 1 のように重なっている．図 1 が示すように，通常はアクティブになった履歴に従って手前から奥へ重なっている．このようなウィンドウからなる層構造において，それぞれの層を透過レイヤーとする．視線情報を用いたウィンドウの背後の情報の透過して提示することは，視線情報によって指定した透過レイヤー内の領域を半透明，最前面およびクリックイベント透過で提示することで実現する．参照情報を半透明かつ最前面で提示することにより，ユーザは参照情報を常に閲覧することおよび参照情報が主のウィンドウに重なる場合の可視性低下の抑制が可能になる．視線情報を用いて

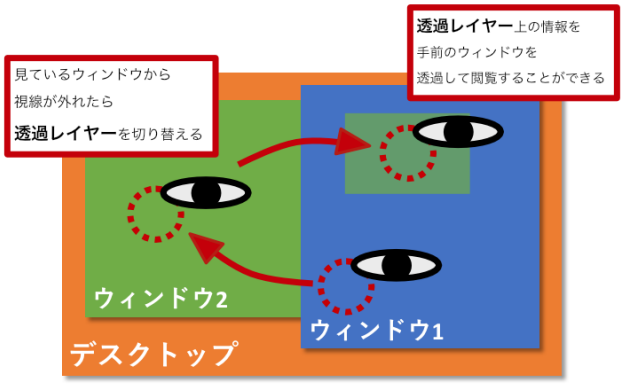

図 2　視線移動による透過レイヤーの移動

Fig. 2　Changing Transparent Layers with Eye Movements

任意のウィンドウの情報を透過して提示するためには，さらに a) 透過レイヤーの変更，および b) 透過情報の固定の二つの機能が必要である．

a) 透過レイヤーの変更とは，透過して閲覧するウィンドウが存在する層の変更である．透過レイヤーの変更は，透過して閲覧するレイヤーを特定の操作によって変更する．本研究では特定の操作として，視線を対象ウィンドウへ移動およびキーボードショートカットキー押下の二つを用意した．図 2 は，視線を対象ウィンドウへ移動する操作の概要である．図 2 では，視線がウィンドウ 1 からウィンドウ 2 へ移動し，その後ウィンドウ 1 に移動している．図 2 のように，視線を見ているレイヤー上のウィンドウから見たいウィンドウへ移動することで，透過レイヤーが奥に切り替わる．これにより，ウィンドウが多層に重なっている場合も，目的の透過レイヤーに切り替えることができる．視線を対象ウィンドウへ移動する操作では，背後のウィンドウの情報を閲覧するために，マウス操作およびキーボード操作を必要としない．これにより，背後の情報の参照がキーボードおよびマウス操作での入力作業と並行して行えることが期待される．

キーボードショートカットキー押下では，設定したキーボードショートカットキーを押下するたびに，透過レイヤーが奥に切り替わる．キーボードショートカットキーを用いて，背後のアプリケーションウィンドウの情報を確認する方法として，アプリケーションスイッチによる方法がある．アプリケーションスイッチによる方法とは，参照するアプリケーションウィンドウをアクティベイトし情報を確認した後に，作業するアプリケーションウィンドウをもう一度アクティベイトする方法である．この方法では，背後に隠れた情報は一時的にしか閲覧できない．キーボードショートカットキー押下によって透過レイヤーを切り替える方法では常に情報を参照することができる．この方法は，視線を対象ウィンドウへ移動する操作に比べ，キーボード入力操作を必要とするが，誤操作の減少が期待できる．

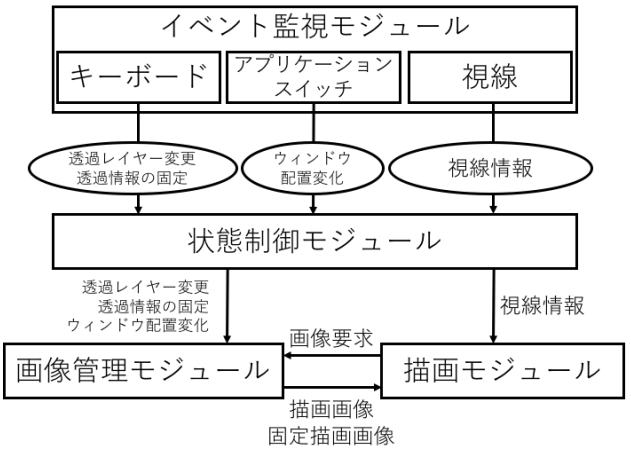

図 3 システム構成図
Fig. 3 A System Architecture.

b) 透過情報の固定とは，透過して提示している情報を指定した場所に固定する機能である．視線はユーザが注視している場合でも，固視微動により不安定である．そのため，確実に閲覧したい情報を表示させておくことが必要である．また，半透明の情報が別の情報と重なった時，視認性が低下する場合がある．文字と文字が重なった場合は，特に情報の視認性が低下する．視認性を改善するために，情報を表示する位置を変更できることが必要である．これらのことから，透過して閲覧している情報を固定し，任意の位置に表示させつづけることができる機能が必要であると考えた．本システムでは，キーボードショートカットおよび視線移動を併用することで実現する．

4. システム構成

本システムは，アプリケーションウィンドウの視線情報を用いて背後参照情報を透過提示するシステムである．参照情報の透過提示は，指定した参照情報を半透明，最前面かつクリックイベント透過で提示することによって実現する．参照情報を半透明かつ最前面で提示することにより，ユーザは参照情報を常に閲覧することが可能になり，また参照情報が主のウィンドウに重なる場合の可視性低下の抑制が期待できる．参照情報の透過提示により，ユーザは参照情報の上から作業ウィンドウの操作が可能になる．これらにより，参照作業における参照情報閲覧を支援する．

本システムの構成を図 3 に示す．本システムは，イベント監視モジュール，状態制御モジュール，画像管理モジュールおよび描画モジュールから構成される．本節では，各モジュールについて説明する．

イベント監視モジュールでは，アプリケーションスイッチイベント，キーボード操作イベント，および視線移動イベントを監視する．アプリケーションスイッチすると，最前面に表示されているウィンドウが変化する場合が多い．このことから，イベント監視モジュールは，アプリケーションスイッチイベントが発生すると，ウィンドウ配置の状態が変化したことを状態制御モジュールに送信する．キーボード操作イベントは，透過レイヤー変更機能および透過情報の固定機能を呼び出すために監視する．それぞれの機能に設定されたショートカットキーが呼び出されると，機能が呼び出されたことを状態制御モジュールに送信する．本システムでの視線移動イベントの監視には，Tobii Eye Tracker 4C[*1]を用いた．視線移動イベントは，視線追跡デバイスから送られた視線情報を状態制御モジュールに送信する．

状態制御モジュールでは，イベント監視モジュールから得た情報を用いてシステムを制御する．アプリケーションスイッチイベントが発生したとき，新しくウィンドウ配置情報を取得し，画像管理モジュールに送信する．透過レイヤー変更機能が呼び出されたとき，透過レイヤーを奥に変更することを画像管理モジュールに送信する．透過情報の固定機能が呼び出されたとき，その時に透過している情報を固定情報として登録することを画像管理モジュールに送信する．視線移動イベントが発生した時，視線の座標を描画モジュールに送信する．また，視線の移動によって透過レイヤーが変更したとき，透過レイヤーを変更することを画像管理モジュールに送信する．

画像管理モジュールでは，描画する透過情報の生成および保持をおこなう．状態制御モジュールから新しいウィンドウ配置情報を受信したとき，その情報からウィンドウ画像を生成する．画像管理モジュールはウィンドウ画像を透過レイヤーにより合成し，描画画像として保持する．また，透過レイヤーが変更されたとき，ウィンドウ画像を透過レイヤーにより合成し描画画像を更新する．状態制御モジュールから，透過情報を固定している情報として登録することを受信したとき，現在の描画画像，描画の位置および大きさを固定描画画像として保持する．

描画モジュールは，画像管理モジュールが保持している描画画像および固定描画画像を描画する．状態制御モジュールから視線情報を受信したとき，画像管理モジュールに描画画像および固定描画画像を要求する．画像管理モジュールから描画画像および固定描画画像を受信したとき，それぞれの画像について視線位置の周辺に該当する部分を描画する．このとき描画画像および固定描画画像は，最前面および半透明かつクリックイベント透過で描画，すなわち透過提示される．

5. システム実行例

図 4 および図 5 は，本システムを利用している様子およびその時のスクリーンの状態を示している．図 4 および

[*1] https://tobiigaming.com/eye-tracker-4c/

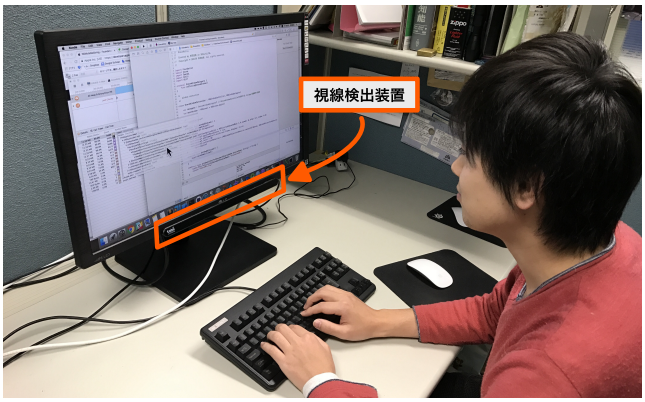

図 4　本システムを利用しているユーザの様子
Fig. 4　A State of Using The Our System.

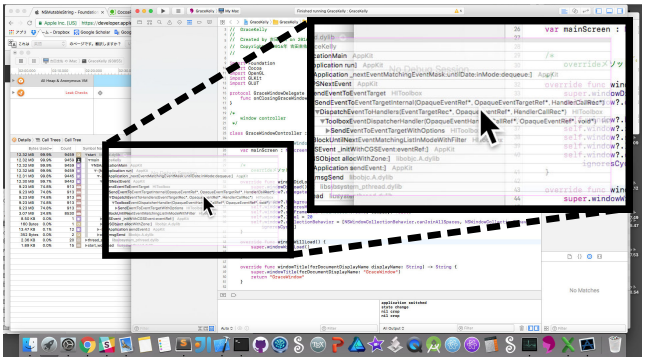

図 5　本システム利用時のスクリーンショット
Fig. 5　A Screenshot of Using The Our System.

図 5 では，ユーザはスクリーン上のマウスカーソル付近を注視している．図 5 の黒枠は，ユーザがマウスカーソル付近を注視している際の情報の透過閲覧の様子およびその拡大図を示している．図 4 では，ユーザはキーボードで入力しながら，背後の情報を閲覧している．視線移動の検出は，図 4 のモニター下部に設置されたデバイスによっておこなわれている．図 5 は，Instruments 上のデバッグ情報を見ながら Xcode で作業している様子である．このとき Xcode のウィンドウは Instruments のウィンドウの前面に重なり，Instruments のウィンドウ内の情報の一部が見えなくなってしまう．これまで，このようにウィンドウの重なりにより隠れた情報を閲覧するためには，ウィンドウを前面に出すなどのウィンドウ操作が必要だった．本システムを用いることで，図 5 のように，ウィンドウの重なりにより隠れた情報を閲覧しながら作業することができる．

図 6 および図 7 は，背後のアプリケーション内の情報を確認しながら Xcode 上でシステム開発をしている例である．図 6 および図 7 はそれぞれ，透過情報の固定および移動の例を示している．図 6 の A は，透過情報の固定をしている．これによって，視線が移動した場合でも継続して背後の情報を閲覧することができる．また，図 7 の B は，透過情報の固定機能を用いて図 6 の A の透過情報の位置を移動し固定した様子である．透過情報の重なりにおいて，特

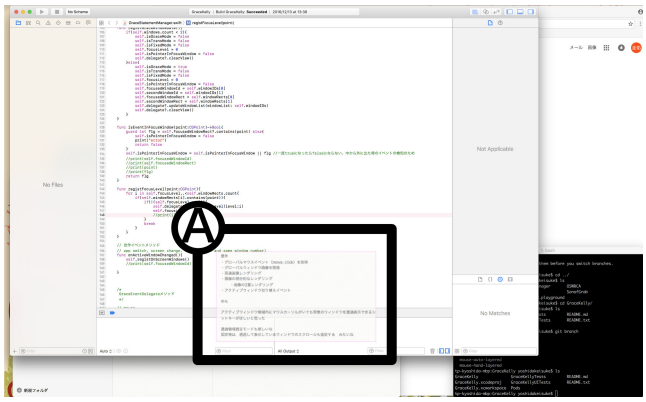

図 6　透過情報の固定
Fig. 6　Fixing Transparent Information.

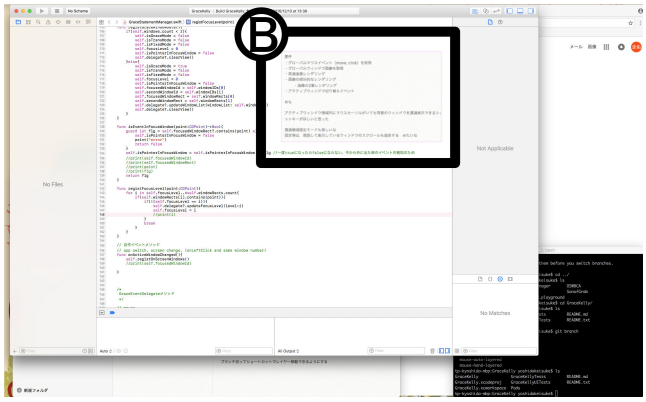

図 7　固定した透過情報の移動
Fig. 7　Moving The Fixed Transparent Information.

に文字が重なった場合に透過情報および前面のウィンドウ内の情報の視認性が悪くなることがある．この場合に，透過情報の固定機能を利用することで，透過情報の重なりを解決することができる．図??の B では，透過情報を文字などの情報がない位置に配置している．これによって，情報の重なりによる視認性の低下を抑制することができる．これにより，アプリケーションウィンドウ背後の情報の透過閲覧を支援する．

6.　考察・展望

本システムによって，参照作業時の情報閲覧が支援できると期待できる．このことから，それぞれのアプリケーションウィンドウが重なってしまう場合がある．

マウスおよび視線を併用したインターフェースの評価実験 [8] では，操作効率を向上させることを示しているが，一方で操作ミスの数が増減するというデータもある．本システムにおいても，固視微動による認知的不可の向上が懸念される．固視微動を軽減するために，視線移動の座標を平滑化することが必要である．

また，3 章で述べた透過レイヤーの変更方法である視線を対象ウィンドウへ移動およびキーボードショートカットキー押下について，どちらの方法が早く目的の情報を見

つけることができるかを検証する必要がある．前者の方法は，視線移動のみでアプリケーションウィンドウ背後の情報を透過して閲覧することができる．一方で，後者の方法は，視線移動とキーボード操作を併用することで，アプリケーションウィンドウ背後の情報を透過して閲覧することができる．ここで，前者の方法は，キーボードおよびマウスでの入力作業を中断しないため，作業効率が向上すると期待できる．しかし，視線移動のみで操作するため，操作の習得が難しいと考えられる．そのため，本システムにおける情報の透過閲覧の習得難易度について調査する必要がある．

今後の展望として，本システムの有効性を示すため，評価実験を行う必要がある．本システムでは，視線情報を併用することで，キーボードおよびマウスによるウィンドウ操作を減らし，参照作業効率を向上させることが目的である．そのため，本システムを使う場合と使わない場合における参照作業タスク完了時間を計測する必要がある．

7. おわりに

本研究では，視線情報に基づくウィンドウ背後の情報の透過提示システムを開発した．本稿では，参照作業の支援のために視線情報に基づく情報の透過提示手法を提案した．本システムを用いることで，視線移動によりアクティブウィンドウ背後の情報を閲覧することができる．本研究の特筆すべき点は，ウィンドウの重なりによって隠れてしまった情報の参照が，従来よりも少ないウィンドウ操作で可能になったことである．これによって，参照作業におけるウィンドウ操作時間が削減され，作業効率の向上が期待できる．

参考文献

[1] Bly, S. A. and Rosenberg, J. K.: A comparison of tiled and overlapping windows, *ACM SIGCHI Bulletin*, Vol. 17, No. 4, ACM, pp. 101–106 (1986).
[2] 吉田圭佑，丹羽佑輔，大囿忠親，新谷虎松：重畳 Web ブラウジングにおける視認性改善について，人口知能と知識処理研究会 2015， Vol. 115, No. 381, pp. 25–30 (2015).
[3] Keisuke, Y., Tadachika, O. and Toramatsu, S.: FoXpace: Manipulating Windows Based on the User's Work History, *Proceedings of the 4th International Conference on Smart Computing and Artificial Intelligence*, IIAI AAI, pp. 698–703 (2016).
[4] Hirohito, S. and Keingo, O.: Docking Window Framework: Supporting multitasking by docking windows, *Proceedings of the 10th asia pacific conference on Computer human interaction*, ACM, pp. 227–236 (2012).
[5] 山中祥太，宮下芳明：重なりあったウィンドウ間を移動可能なマウスカーソル操作手法とその評価，ヒューマンインタフェース学会論文誌 The transactions of Human Interface Society， Vol. 15, No. 1, pp. 313–326 (2013).
[6] Waldner, M., Steinberger, M., Grasset, R. and Schmalstieg, D.: Importance-driven compositing window management, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, pp. 959–968 (2011).
[7] Ishak, E. W. and Feiner, S. K.: Interacting with hidden content using content-aware free-space transparency, *Proceedings of the 17th annual ACM symposium on User interface software and technology*, ACM, pp. 189–192 (2004).
[8] 大和正武，神代知範，門田暁人，松本健一ほか：視線・マウス併用型インタフェースのドラッグ& ドロップ操作への適用，情報処理学会論文誌， Vol. 44, No. 1, pp. 166–175 (2003).