

# LookUp:視線移動情報の特徴と機械学習を用いた読書支援システム

中園 歩<sup>1</sup> 濱川 礼<sup>1</sup>

**概要:** 本論文では、視線移動の特徴からユーザの状況を推定し、適切な読書支援を行うシステム「LookUp」について述べる。既存の視線入力インターフェースはユーザが能動的に視線によるジェスチャを行なうものが一般的である。しかし、これらは他のインターフェースと比べて入力に要する時間が長く、また、視覚による情報収集機能が制限されてしまう。LookUp では、視認行動のための視線移動から特徴を分析し、ユーザが要求する支援動作を読み取り実行する。ここでは、LookUp を用いた読書支援システムの有効性について述べる。

## Reading support system using gaze movement feature

AYUMU NAKAZONO<sup>1</sup> REI HAMAKAWA<sup>1</sup>

**Abstract:** In this paper, we describe a system that estimates user's situation from the features of gaze movement and performs appropriate reading support. In the existing gaze input interface, it was common for the user to actively gaze by gaze. However, compared with other interfaces, these require longer input time, and the visual information collection function is restricted. The purpose of this system is to obtain features from gaze movement for viewing behaviors and to support user's intention reading. Here we describe the effectiveness of the reading support system using this system.

### 1. はじめに

近年、視線追跡技術が発展し、それと共に視線入力インターフェースの開発も増加している。一般向けのソフトウェアとしてリリースされるものも増えており、2017年には windows 10 のユーザ補助機能に、視線でキーボードとマウス操作が行える”Eye Control”機能が加えられた [1]。こうした視線入力インターフェースは、主に手足が不自由な障がい者向けのシステムとして活躍している。

一方で、現在の視線入力インターフェースは入力に時間がかかり、本来の目の役割である視覚情報の取得機能を一時的に制限してしまう問題がある。視線入力インターフェースでは一般的に、対象を注視することで選択動作として認識される仕組みが多い。Majaranta らの研究 [2] では、注視として認識されるまでに必要な視線の滞留時間は 150ms か

ら 1000ms とされており、また、Huckauf らの研究 [3] では同じタスクに対して視線入力インターフェースではマウスによる操作の 2 倍の時間がかかるというデータが存在している。さらに、この視線滞留時間中は他の部分に視線を向けることが不可能であり、そのため本来の作業に必要な情報から、視線を一時的に外さなければならなくなり、作業時間を遅延させる原因になってしまう。これらは、眼の本質的な役割が視覚情報の取得機能であるために、まずシステム側で視覚情報の取得なのか、あるいは視線入力なのか判定する仕組みを設けなければならないために発生する問題である。このように、視線入力インターフェースはキーボードやマウスと並ぶ利便性を持つとは言い難く、現状では視線入力インターフェースの用途は限定的である。

このような問題を解決するため、本研究では、視線が動いているときの特徴を分析し、ユーザが求める支援操作を推定、その結果から実際に支援操作を自動実行する仕組みと、それを元にした辞書ポップアップインターフェース

<sup>1</sup> 中京大学 工学部  
School of Engineering Chukyo University

「LookUp」を開発した。LookUpは、ユーザに対して意図的な視線入力を要求せず、本来の「見る」、「読む」という動作からユーザの状態を推定、ユーザにとって最適な読書支援を行う。

## 2. 関連研究

視線入力インタフェースを用いた研究は多数行われている。また、基礎研究として視線情報の分析手法の研究も多く存在する。ここでは、既存の視線入力インタフェース及び視線分析手法、本研究の位置づけを示す。

### 2.1 視線入力インタフェースの研究

視線入力インタフェースで問題となるのは、視線による入力をどこまで複雑化できるかという点である。視線追跡デバイスから取得できるデータは、主に視線がどこを向いているかという座標データのみであるため、マウスにおけるカーソル移動までは容易に実装できるが、その後のクリックやドラッグという操作は工夫が求められる。鈴木らは、視線を特定の手順で移動させるアイジェスチャを用いて、操作を指定するインタフェースの研究を行った[4]。このインタフェースでは、予めアイジェスチャを登録しておくことにより、視線操作のみで複数の操作を使い分けることが可能となっている。しかし、このインタフェースでは、依然として入力動作に時間がかかってしまう上、アイジェスチャ入力のために読書を一時的に中断する必要がある。本研究では、ユーザの行動を阻害することなく、ユーザが支援を欲するタイミングで実行される、レスポンスタイムが短いシステムの開発を目的とする。

また、Biedertらが開発したThe eyeBook[5]は、本の中にトリガを設定し、設定箇所を読むと挿絵が切り替わるといったコンテンツを視線追跡技術によって実現している。しかし、The eyeBookの手法では静的に埋め込まれたコンテンツを動かすことは出来るが、リーダーとしての機能には柔軟に対応することが困難である。LookUpでは、読むコンテンツに関わらずユーザに対して適切な支援が行える汎用的なインタフェースを目指す。

### 2.2 視線分析技術及び手法についての研究

視線データは人間の興味や注意などを分析するための要素となっており、マーケティングやスポーツ、プログラミングなど、多くの分野で用いられている。Charlesらは日本語の文章の理解度を視線から分析できないか研究を行った[6]。その結果、読解を努力した箇所には視線の滞留時間に特徴がみられることが判明した。しかし、こうした手法の多くは数分から数十分かけて行なわれたデータから検出することが多く、本研究のようなリアルタイム性が求められる場合では活かすこと難しい。一方、短時間の入力データによって視線から特徴を分析することを目的とした研究

も存在する。上坂らは、自動車の運転手の視線情報から直後の運転行動の予測が可能であるか研究を行った[7]。自動車の運転手は右折や車線変更に際してミラーを確認するが、実際にこの研究では、高確率で右左折、車線変更といった各行動の予測が可能であることが判明している。さらに、短時間での視線分析結果をリアルタイムで反映しているシステムとして、山本らが開発した、視線特徴からゲームの難易度を最適なものに調整するシステム「百人eye首」がある[8]。このシステムにおける視線分析はゲーム中に行われ、分析結果がリアルタイムにゲームの難易度として反映されている。したがって、短時間での視線分析は状況を制限することによって、行動の予測が可能であると考えられる。

## 3. コンセプト

LookUpでは、人間の自然な視線移動を入力データとして扱う。自然な視線移動とは、図1のように、文字や画像といった視覚情報を得るため視線を動かすことである。

### 3.1 既存の視線入力インタフェース

従来の視線入力インタフェースでは、主にアイジェスチャを用いたコマンド入力を行う手法と仮想コントローラを用いた手法の2種類が存在した。

#### 3.1.1 アイジェスチャを用いた視線入力インタフェース

アイジェスチャとは、スマートフォンなどに搭載されているスワイプ、フリックといったタッチジェスチャのように、特定の軌道で視線を動かすことによってシステム側で予め登録された機能を呼び出す手法である。図2は、ジェスチャを入力するために、視線をZ字の形に動かしている例である。このようなジェスチャを予めシステム側で実装することにより、ユーザは視線移動のみで様々なコマンドを実行することが可能になる。しかし、ジェスチャが単純なものであった場合、自然な視線移動と間違えかねない。図2のような背景に文章があるような場面では、上から下へ視線を動かすようなジェスチャでは文面を読んでいる状態でも誤作動を起こしてしまう可能性がある。そうした問題を解決しようとする、ジェスチャを複雑なものにしなければならず、それにより入力に時間がかかってしまう。さらに、ユーザは事前に複雑なジェスチャと対応するコマンドを把握しておく必要があるため、負担がかかってしまう。

#### 3.1.2 仮想コントローラを用いた視線入力インタフェース

仮想コントローラとはスクリーンキーボードのように画面上にコマンドを実行するためのボタンやパッドといったGUIツールである。視線入力インタフェースでは、まず画面上に表示したコントローラに視線を注視することでコマンドを選び、次に操作対象を注視することでコマンドの実行を行う。図3の例では、まずコントローラから右クリッ



図 1 自然な視線移動の例



図 2 アイジェスチャの例

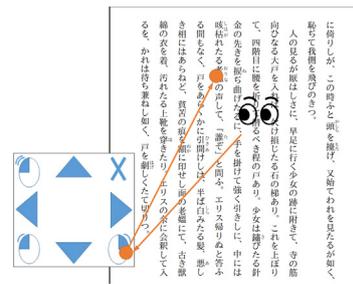


図 3 仮想コントローラの例

クを選択するため、コントローラの右下のアイコンを注視、その後、操作対象の箇所に視線を移動させている。この手法はアイジェスチャとは異なり、使用可能なコマンドが GUI によって示されているのでユーザが訓練する負担は少ない。しかし、こちらも誤操作を防止するため、注視を検出する時間が取られてしまう。また用意するコマンドの分だけコントローラの UI が肥大化し、画面のスペースを取ってしまうという問題がある。

### 3.2 自然な視線移動を使用する利点

LookUp で実装する自然な視線移動を入力データとして使用する手法では、既存の視線入力インタフェースと違い、入力のための視線移動の時間が発生せず、視覚情報の取得が困難になるという問題も発生しない。また、システムが自動でユーザが求める支援動作を推定してくれるため、事前に訓練が必要といった負担もかからなくなる。しかし、LookUp で利用が想定される読書では、文面上のどこでユーザが支援を求めるか不明である。そこで、LookUp では視線の移動量を機械学習によって分析することで、どこを見ているユーザが求める支援動作を推定できるようにした。

## 4. システム構成

LookUp は、ディスプレイ下部に設置する視線追跡デバイスとシステムを制御する PC からなる。システムの利用イメージを図 4 に示す。なお、LookUp における読書機能部分は Amazon.com 社の電子書籍リーダー Kindle for PC を利用している。ユーザはディスプレイに表示された Kindle の画面を見て読書を行い、その際に視線追跡デバイスによって視線情報を取得される。取得された視線情報は、システムによって分析され、ユーザが求める支援動作を推定する。例えば、推定結果が、「辞書が見たい」となった場合、図 5 に示す Kindle の辞書機能が実行される。

### 4.1 視線追跡機能

LookUp では、ユーザの視線分析のために、視線追跡デバイスとして The Eye Tribe 社 [9] のアイトラッキングシステムを使用した。このアイトラッキングシステムはディ

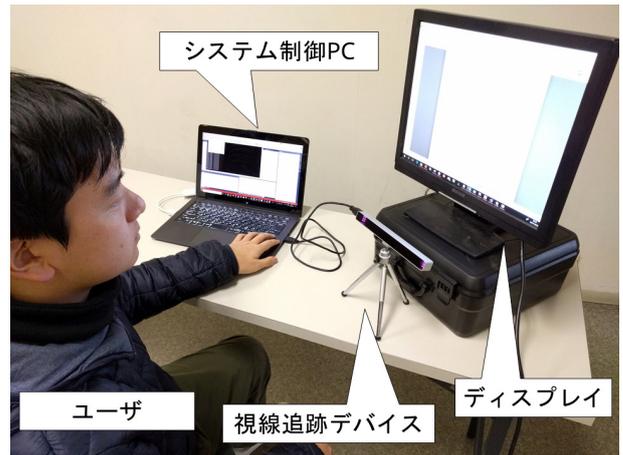


図 4 システムの使用イメージ

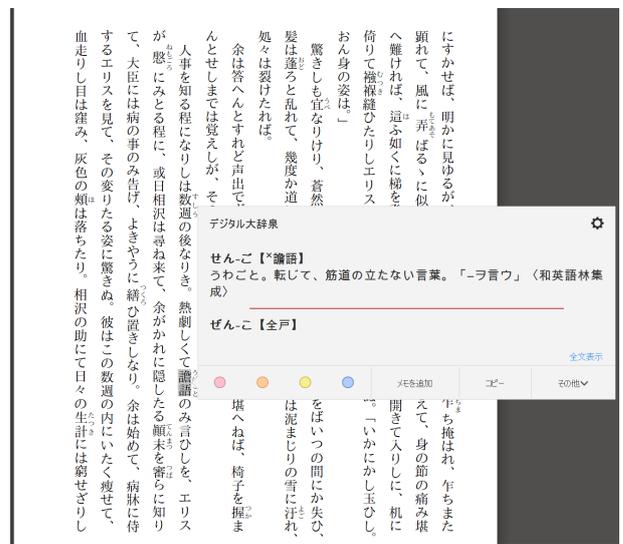


図 5 Kindle の辞書機能

スプレイ下部に設置して使用するタイプで、30Hz の速度 (毎秒 30 件のデータ) で視線の動きを追跡することが出来る。アイトラッキングシステムを使用するにあたり、データ取得サーバの準備及びキャリブレーション機能が必要であるが、これは同社が提供しているクライアントアプリを使用した [10]。また、LookUp の内部実装においても、同様に提供されている SDK を使用して、開発を行っている。このアイトラッキングシステムで得られる視線情報には、右目、左目、両目が画面上のどこを見ているかを示す座標、

瞳の大きさなどがあり、LookUp では両目の中心の座標を使用している。ここで、システム開始からの経過時間  $t$  に対する両目の中心の座標  $x, y$  座標は、それぞれで  $x_t, y_t$  とする。

## 4.2 視線分析部

今回、視線分析部では Deep Belief Network(DBN) を用いて、視線の分析を行っている。深層学習の実装には、.NET 向けの機械学習ライブラリである Accord.NET Framework を使用した。

### 4.2.1 学習用データセット作成

学習用データセットには、実際に筆者の中園が Kindle を使用して読書を行なった際の視線データと、マウス及びキーボードの操作ログを用いた。LookUp ではユーザの読書支援として、難解な単語を読んだ際に辞書を表示させる機能を実装しているため、学習データにはそのような読解が困難な状況のデータが含まれることが望ましい。そこで、データセット作成作業では、Kindle ストアで公開されている森嶋外の「舞姫」[11]の原文を用いた。舞姫の原文は文語体や旧字体となっており、読み物として難易度が高いと判断したためである。データは15分間取得し、csv ファイルとして出力、その後システム側で入力データの形式に変換している。

### 4.2.2 学習モデル

LookUp の目的は、視線情報を元にしたユーザが求める支援動作の推定であるため、学習データはデバイスが取得した視線情報を時系列で並べたデータとユーザの状態を表したデータとなる。システム開始からの経過時間  $t$  に対する入力データ  $X_t$  は、視線追跡デバイスから取得した直近5秒分(毎秒30件  $\times$  5秒 = 150件)の視線の座標データを元に求めた各フレームごとの移動量式(1)を分解して並べた式(2)となる。文章をなぞる様に視線を動かした場合、図6のようになる。

$$(\Delta x_t, \Delta y_t) = (x_t - x_{t-1}, y_t - y_{t-1}) \quad (1)$$

$$X_t = [\Delta x_{t-149}, \Delta y_{t-149}, \Delta x_{t-148}, \Delta y_{t-148}, \dots, \Delta x_{t-1}, \Delta y_{t-1}, \Delta x_t, \Delta y_t] \quad (2)$$

学習データ及び出力データはエミュレートに使用するマウスとキーボードの操作とする。学習データとして認識させる際には、予め各操作が出力層のノードに対応するようにしている。表1に主な操作と支援動作を示す。出力データのノードには、各操作が求められている確率が格納され、学習データとして引き渡す際には、0,1の2値が渡される。また、操作ログに含まれているマウスとキーボードの操作は、一度の操作に対し1件しか含まれておらず、このままではデータ全件に対して、操作回数が極端に少なくなってしま

う。そのため、入力データ  $X_t$  に対する学習データ生成時には、1秒先となる  $t+1$  から  $t+30$  までの操作ログに含まれる操作全てを格納するようにした。この処理を行うことによって、ユーザが経過時間  $t$  の直後にどの支援を求めるとかという推定と、各操作の件数の増加させ学習モデルに認識させやすくした。

表1 操作と支援動作の例

操作	支援動作
無操作	-
左矢印キー	ページ送り
右クリック	辞書表示
Escape キー	辞書非表示

学習に使用するネットワークはノード数50の隠れ層4層とした。

## 4.3 読書支援部

LookUp は windows フォームアプリとなっており、動作中は Kindle のウィンドウの上に被さる様に透明なウィンドウが配置される。これにより、背面にある Kindle の利用を妨害をせずに描画処理を行うことが可能である。

Kindle では、マウスやキーボードを用いることで、前述の表1の機能を利用することが可能である。これを視線入力のみで利用するため、読書支援部では、視線分析の結果を元に PC 操作をシミュレートする機能を実装している。例えば、ユーザが辞書表示機能を必要とした場合には、システムによりカーソル位置を視線の座標位置とした右クリックの押下をシミュレートすることで Kindle の辞書機能がポップアップされる仕組みとなっている。これにより、ユーザはマウスやキーボードの操作を一切必要とせずに、Kindle の機能が利用できるようになる。

## 5. 実験結果

### 5.1 学習モデルの評価

まず生成した学習モデルの評価を行った。評価用データは、学習用データセット作成時とは異なる書籍として中勘助の「銀の匙」[12]を使用し、学習データを収集する手順と同様にして行なった。表2にその結果を示す。なお、ここでは学習モデルが99%以上の確率と判定された推定データのみをカウントしている。表2のデータから、学習モデルが各機能を推測することはほとんどないことが分かる。

表2 推定結果と実際の結果の組み合わせの比率

		予想			
		通常	ページ送り	辞書表示	辞書消去
結果	一致	100.00%	0.93%	0.62%	12.29%
	不一致	0.00%	0.65%	4.95%	16.55%
実データ数		23062	1819	1272	1244



り、それを活かすためにはより多彩な機能を使い分けられるようにする必要がある。今後は読書に限らず、様々な作業において有効性を示せることを目指す。

## 参考文献

- [1] Windows 10 視線制御 - マイクロソフト アクセシビリティ, 2017.
- [2] Päivi Majaranta and Kari-Jouko Rähä. Text entry by gaze: Utilizing eye-tracking. *Text entry systems: Mobility, accessibility, universality*, pp. 175–187, 2007.
- [3] Anke Huckauf and Mario H. Urbina. Object selection in gaze controlled systems: What you don't look at is what you get. *ACM Trans. Appl. Percept.*, Vol. 8, No. 2, pp. 13:1–13:14, February 2011.
- [4] 鈴木若菜, 外山託海, Kunze Kai, 岩村雅一, 黄瀬浩一. 読書アノテーション支援システム実現に向けたアイジェスチャの提案 (パターン認識・メディア理解). 電子情報通信学会技術研究報告 = IEICE technical report : 信学技報, Vol. 113, No. 493, pp. 85–90, mar 2014.
- [5] Ralf Biedert, Georg Buscher, and Andreas Dengel. The eyebook using eye tracking to enhance the reading experience. *Informatik-Spektrum*, Vol. 33, pp. 272–281, 2009.
- [6] Charles Lima Sanches, Koichi Kise, and Olivier Augereau. Japanese reading objective understanding estimation by eye gaze analysis. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers, UbiComp '17*, pp. 121–124, New York, NY, USA, 2017. ACM.
- [7] 上坂竜規, 野田雅文, 目加田慶人, 出口大輔, 井手一郎, 村瀬洋. ドライバの視線情報を利用した運転行動予測. 電子情報通信学会技術研究報告. MI, 医用画像, Vol. 111, No. 49, pp. 105–110, may 2011.
- [8] 山本倫也, 米田宗弘, 長松隆, 渡辺富夫. 百人 eye 首 : 視線と札取り動作の予測に基づくテーブルトップ対戦型百人一首ゲーム. 情報処理学会論文誌, Vol. 54, No. 4, pp. 1551–1562, apr 2013.
- [9] The eye tribe, 2017.
- [10] *Basics — eyetribe-docs*, 2017.
- [11] 森嶋外. 舞姫, 9 2012.
- [12] 中勘助. 銀の匙, 9 2017.
- [13] 谷崎潤一郎. 陰翳礼讃.