

視線判定機能によりタッチタイピング練習を支援するツールの開発

今村貴明^{†1} 永井孝幸^{†2} 中野裕司^{†2}

我々はタッチタイピングの練習の際、USBカメラを用いてキーボードを見ていないか判定するツールを windows アプリケーションとして開発した。練習前にキャリブレーションとして顔画像からテンプレート画像を作成し、その画像を用い、目の領域を検出する。その後、検出された領域に対し形状のマッチングを行うことで視線のずれの判定をリアルタイムに行う。このツールの有用性を確認するため実験を行った。その結果、キー毎ではなく、フレーズ毎に区切って判定を行うと、キーボードを見ながらタイプしていたフレーズのうちこのツールがキーボードを見ていたと判定したフレーズは約 90% になった。しかしキーボードを見ずにタイプしていたフレーズのうちキーボードを見ていないと判定したフレーズは 35% もあり、リアルタイムに練習者へフィードバックを行うには改善の余地が残されているという結果になった。今回用いた判定条件は暫定的なもので、よりキー毎の判定精度が高くなる条件を見つけることが今後の課題となる。

Development of Touch-Typing Training Tool with Eye-Gaze Detecting Function

TAKA AKI IMAMURA^{†1} TAKAYUKI NAGAI^{†2}
HIROSHI NAKANO^{†2}

We developed a touch typing training tool that can detect whether a trainee looks at display or keyboard during training. This tool runs on Windows PC with USB camera or web-cam. Eye-gaze is detected in two stages. First, eye region in captured image is detected by matching with template images of the trainee. Then, eye-gaze change is detected by similarity matching between the shape of detected eye region and that of template eye regions. This process is fast enough to detect eye-gaze in real-time. We assume template images are registered by trainee before training. We did an experiment to evaluate how accurate this tool can detect eye-gaze. The result is that this tool could determine eye-gaze 90% of phrase looking at the keyboard while typing in good condition, but also contains false-negative and false-positive cases. Failure that despite the subject did not look at the keyboard while typing, this tool determine eye-gaze change is about 35%. We need to improve the accuracy of eye-gaze detection.

1. はじめに

近年、パーソナルコンピュータ(PC)はインターネットをはじめとして、プレゼンテーション、文書作成、表計算など、さまざまな用途で使われている。PCへ入力を行うにはキーボードやマウスが使われているが、キーボードを使った入力のもっとも一般的な入力方法である。

このキーボード入力において、タッチタイピングと呼ばれる手法が存在する。タッチタイピングとは、キーボードを見ずにタイピングを行う手法のことである。

PCを扱い始めて間もないユーザーはタイピングする際、キーの位置を覚えていないため、キーボードを見ながらタイピングする傾向がある。このような場合、ディスプレイを見ていないため、タイプミスが起こった際、それに気づきにくく非常に非効率的であるといえる。

しかし、タッチタイピングを習得すれば、キーボードを見ずに画面を見ながらタイピングすることができ、タイプミスに気づきやすく、すばやく入力できるなどから、タッチタイピングはとても効率のよいタイピング手法といえる。しかし、タッチタイピングには欠点がある。それは、習得には練習が必要となることである。そのため、数多くのタッチタイピング練習ツールが開発されている[1]。

タッチタイピング練習ツールの一般的な画面構成は図

1[2]のように、問題文とキーボードと指の画像が表示され、次の入力文字と対応したキーが強調されるといったものである。これだけでは練習者の様子がわからないため本当に画面を見ずにタイピングできているかどうか判定することはできない。

我々の研究室では、練習者の興味のある単語から問題文を生成することで、学習意欲を保持させるウェブベースのタイピング練習ツールの開発を行っている[2][3]。我々はこのツールに練習者がキーボードを見ずにタイピングできているか判定を行う機能を加えようと試みた[4]。この機能を実現するためには、練習者がどこを見ているか判定するために視線の検出を行う必要がある。視線検出をタッチタイピング練習ツールへ応用した例は過去にあったものの[5]、



図 1: タッチタイピング練習ツールの画面

Figure 1: Typical user interface of touch-type trainer

1 熊本大学大学院
Kumamoto University
2 熊本大学
Kumamoto University

1枚の画像の処理に1秒ほどかかっているなど、実用化されたものは調査した結果、見つけることはできなかった。この視線判定機能は、リアルタイムでの処理ができ、練習環境を制限しないよう特殊な装置を使わず実現すべきであり、また一般的なPCユーザーであれば誰でも扱うことができるものであるべきだと考える。

2. タッチタイピング練習用の視線判定機能の開発

リアルタイムでの視線検出はすでに多くの研究がされているが[6][7][8][9][10][11]、カメラや視線検出を行う装置が特殊、あるいは高価であるため、本研究には適さない。今回は、簡易な装置で動作する必要があるため、一般的なUSBカメラやノートパソコンなどに内蔵されているカメラを使用する。また、高速な視線の判定を行うため、ここではキーボードを見ているか、ディスプレイを見ているかをリアルタイムに判定するという2択に限定した。そこで、カメラから取得できる顔画像の中からディスプレイを見ているとき(図2(a))、キーボードを見ているとき(図2(b))の目近傍の形状の違いに着目し、そこから視線の判定を行う。ここでは、この形状の違いを判別する為、2つのステージを用意した。まず第1ステージで、テンプレートマッチングを用いカメラから取得した顔画像から目の領域を検出する。第2ステージでは、第1ステージで検出した領域に対しさらに形状のマッチングを行うことで形状の違いを判別し、視線の判定を行う。これらの処理を行うため、オープンソースの画像処理ライブラリであるOpenCVを使用した[12]。



図 2: (a)ディスプレイ, (b)キーボードを見ているときの両目の形状

Figure 2: Shape of the eyes:(a)looking at display (b)looking at keyboard

2.1 テンプレートマッチングによる目の領域の検出

第1ステージでは、テンプレートマッチングによる目の領域の検出を行う。ここではOpenCVで用意されているMatchTemplate関数を使用した。この関数は対象の画像の上をテンプレート画像をスライドさせ比較していくことでもっとも似ていると判定された領域を検出する。このとき、指定した手法でマッチングを行っていくが、今回は以下の正規化相互相関を選択した。

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') \cdot I'(x+x', y+y'))}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I'(x+x', y+y')^2}}$$

式中の T と I はそれぞれテンプレート画像と対象の画像を表しており、 $R(x, y)$ は対象の画像のそれぞれのピクセルにおける相関値を表している。この相関値 R は-1から1までの値をとり、-1の場合はもっとも異なっており、1の場合はもっとも似ているということの意味する。よって、この結果の中からもっとも相関値が大きい座標を取り出すことでもっともテンプレート画像に似ているであろう領域を検出することができる。

2.1.1 本ツールでのテンプレートマッチングの適用方法

最初に、図2のような目近傍の領域だけをテンプレート画像とし、テンプレートマッチングを行っていた。しかし、このテンプレート画像ではキーボードを見たときに眉を誤検出してしまっていた。そこで、図3のようにテンプレート画像を目と眉を含む領域で作成するようにした。こうすることでディスプレイを見ているとき、キーボードを見ているとき、どちらの場合でも目と眉を含む領域を検出できるようになった。実際にテンプレート画像(図3)を用いてテンプレートマッチングを行い検出した領域の結果が図4である。図中の長方形がテンプレートマッチングで検出された領域となっている。

テンプレートマッチングを行うためにはテンプレート画像が必要となるが、これはタッチタイピング練習前に行うキャリブレーションフェーズで作成する。



図 3: (a)ディスプレイ, (b)キーボードを見ているときのテンプレート画像

Figure 3: Template images of eye region: (a)looking at display (b)looking at keyboard

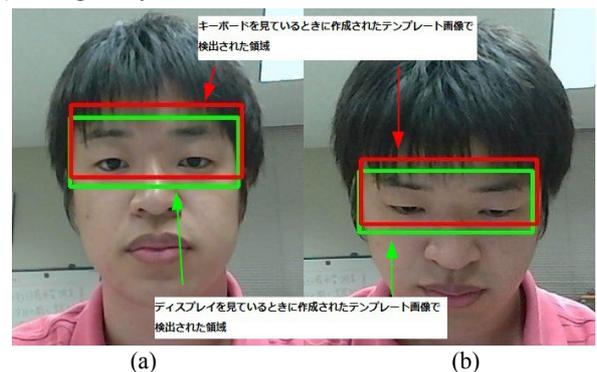


図 4: (a)ディスプレイ, (b)キーボードを見ているときのテンプレートマッチングの結果

Figure 4: Result of template matching: (a)looking at display (b)looking at keyboard

2.2 形状のマッチングによる視線の判定

第2ステージでは視線の判定を行うため、第1ステージで検出した領域に対し、さらにテンプレート画像との形状のマッチングを行う。形状のマッチングにはOpenCVで用意されているMatchShapes関数を使用する。この関数は指定した手法で2つの画像の形状を比較し、どれだけ形状が似ているかを示す数値(以下マッチ度)を返す。今回、我々は以下の手法を指定した。

$$I_2(A, B) = \sum_{i=1..7} |m_i^A - m_i^B|$$

$$m_i^A = \text{sign}(h_i^A) \cdot \log h_i^A, m_i^B = \text{sign}(h_i^B) \cdot \log h_i^B$$

式中の A, B は2つの画像を指しており、 h_i^A, h_i^B は A, B それぞれのHuモーメントを指している。Huモーメ

ントは回転, 相似, 並進などに対してロバスト性をもち, その画像の特徴を表すモーメントである[13]. この関数を用い, テンプレート画像と第1ステージで検出した領域がどの程度似ているかを調べる. 2つの画像の形状が似ている場合, マッチ度は小さい値となり, 2つの画像の形状が異なる場合, マッチ度は大きい値となる.

2.2.1 本ツールでの形状のマッチングの適用方法

このツールは, ディスプレイを見ているときのテンプレート画像で検出した領域とディスプレイを見ているときのテンプレート画像のマッチ度を計算, さらにキーボードを見ているときのテンプレート画像で検出した領域とキーボードを見ているときのテンプレート画像のマッチ度を計算し, 前者のマッチ度が小さかった場合ディスプレイを見ていると判定, 逆に後者のマッチ度が小さかった場合キーボードを見ていると判定する. しかし, 図3のテンプレート画像を使い, テンプレートマッチングを行うと, 目と眉を含んだ領域を検出することになる. 我々は, 図2で示した領域での形状の違いを判別したいため, 第2ステージではテンプレート画像, 検出した領域ともに目が含まれるであろう下半分のみを用いる. また, 2.3節でも述べるが, テンプレート画像は2値化画像を使用しているため, 検出された領域も同様に2値化し, その2つの2値化画像の形状のマッチングを行っている.

2.2.2 予備実験

テンプレートマッチングと形状のマッチングを用いた視線判定機能を使い, 予備実験を行った. この予備実験は被験者に数秒間ディスプレイを見てもらい, その後数秒間キーボードを見てもらい, この動作を数回繰り返してもらうというもので, このときの視線を判定できるか調べる. 図5は得られたマッチ度の遷移を示したものである. (a)は, マッチ度の全体を, (b)は変化が見やすいようにマッチ度の上限を0.05としたものである. 点線のグラフはディスプレイを見ているときのテンプレート画像を用い, マッチ度を

算出したもの, 実線のグラフはキーボードを見たときのテンプレート画像を用いマッチ度を算出したものとなる. そして点線のグラフのマッチ度が小さいときはディスプレイを, 実線のグラフのマッチ度が小さいときはキーボードを見ていると判定している. 図6はおおよそ点線のグラフと実線のグラフのマッチ度が交互に小さい値を取っているため, 上手く判定できているといえる. 実際にこの遷移の仕方は被験者の視線の移動と一致していた. 点線, 実線ともにマッチ度が大きくなっているときがあるが, これはディスプレイとキーボード間での視線を動かしている最中を表しており, 2つのテンプレート画像で得られたどマッチ度どちらに対しても値が大きくなっている.

2.3 キャリブレーション

2.1節でも述べたように, このツールを使うためにはディスプレイを見ているとき, キーボードを見ているとき2つのテンプレート画像が必要となる. そのため, テンプレート画像を作成するために練習者には練習前にキャリブレーションを行ってもらう.

キャリブレーションは, まず, 練習者がディスプレイを見た状態で自身の顔が映っているカメラの画像を所定のキーを押すことでキャプチャし, その画像から図3のように目と眉を含むようにマウスをドラッグしてテンプレート画像となる領域を選択する. 次に, キーボードを見た状態で自身の顔が映ったカメラの画像を所定のキーを押すことでキャプチャし, 同様にその画像から目と眉を含むようにマウスをドラッグし領域を選択する. その後, 選択された2つの領域は自動的に2値化され, その2値化画像がテンプレート画像として用いられる. 2値化したテンプレート画像を用いる理由として, 2値化画像は照明条件に対しロバスト性を持つことがあげられる. 2値化には大津の手法を用いている[14]. このように目と眉を含む領域を2回選択するという単純な作業でキャリブレーションは完了となる.

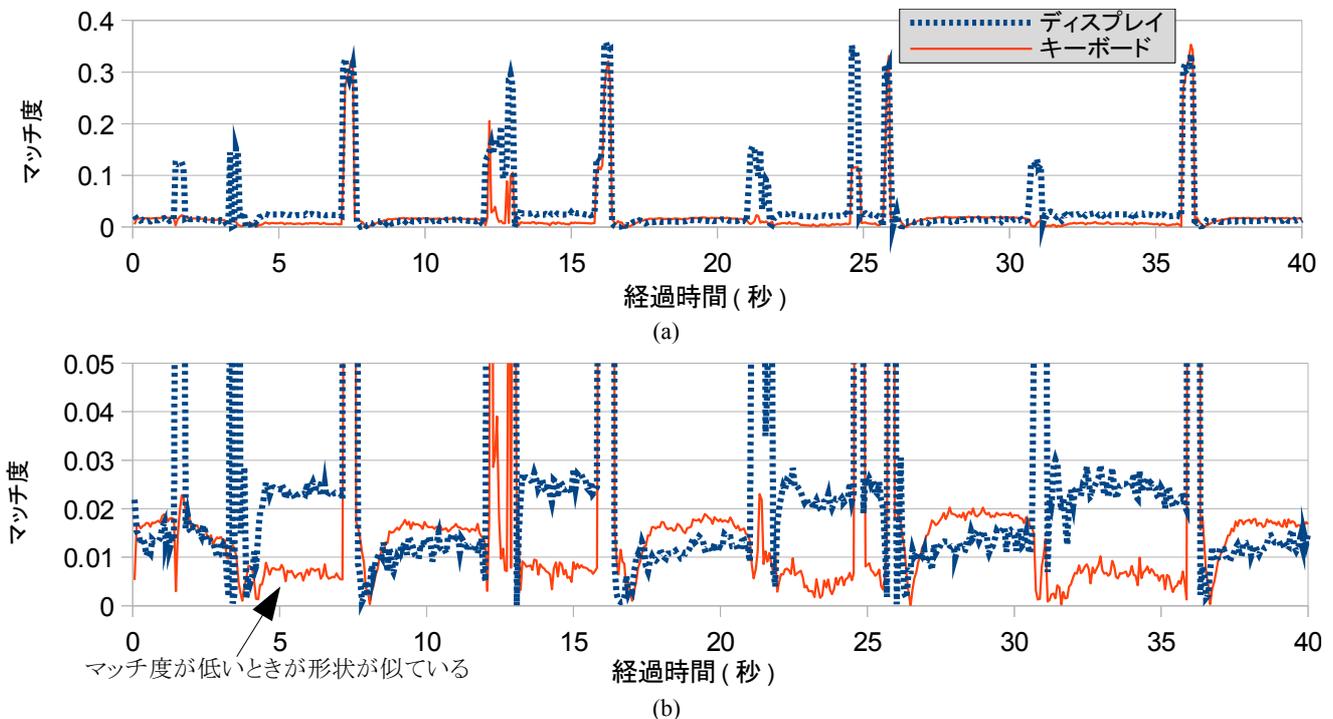


図 5: マッチ度の遷移

Figure 5: Change in the matching rate.

3. タッチタイピング練習ツールの開発

2節で説明した視線判定機能を備えたタッチタイピング練習ツールの開発を行った。そこでこの節では、このツールの概要を述べる。

3.1 練習ツールについて

このツールは Windows アプリケーションとして開発した。その画面表示は、図 6 のように問題文、入力した文字、キーボード画像からなる。次に入力する文字に対応したキーが赤く強調され、今押しているキーは黄色で強調される。このように簡易なものではあるがタッチタイピング練習ツールとしては一般的な画面表示となっている。このツールは、現時点では、正誤問わずキーが押されたときにそのキーコードと時刻、入力した文字が正しいかどうか、そしてそのときの視線判定の結果を記録する。

3.2 練習ツールと視線判定機能の連携

これまでに説明した視線判定機能とこの練習ツールとの連携について説明する。視線判定機能は別プロセスで動作し、ディスプレイを見ているときのテンプレート画像を用い算出したマッチ度、キーボードを見ているときのテンプレート画像を用い算出したマッチ度、この 2 つのマッチ度とそれらが算出された時の時刻を保持し、記録する。また、練習中にもキーボードを見ていると判定されたときはアラートとして赤い画像をタッチタイピング練習画面の下に表示することでリアルタイムに練習者にフィードバックするようにしている。

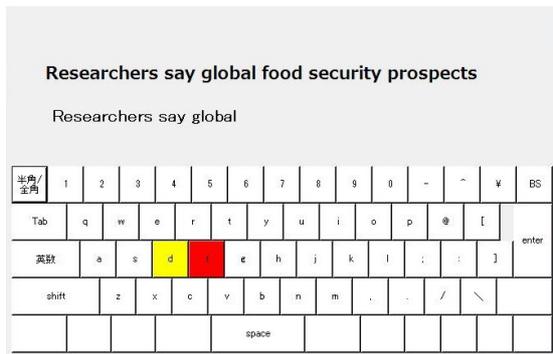


図 6: タッチタイピング練習ツールの画面表示

Figure 6: Screenshot of the developed touch-typing training tool.

4. 実験

2.2.2 節の実験のように視線の動きが激しくなければ、この視線判定機能は練習者がディスプレイを見ているかキーボードを見ているか判別することができることがわかった。次に、この機能がタッチタイピング練習の際の素早い視線の動きでも判定を行えるかどうか確認するため、実験を行った。

4.1 実験内容

本実験は、被験者にこのタッチタイピング練習ツールを使ってもらい、そのときのツールの振る舞い、そして記録されたデータから被験者がキーボードを見たときに視線の判定を行えているかどうか確認する。この実験では、DELL LATITUDE E6420 を使用する (図 7)。このノートパソコンは web カメラを内蔵しており (図中丸部分)、最大 1920x1080 の解像度で 30fps のフレームレートでの録画を行える。本実験ではこの内蔵カメラを使用する。



図 7: DELL LATITUDE E6420 と内蔵カメラ

Figure 7: Laptop with internal camera (DELL LATITUDE E6420)

また、被験者にはキャリブレーションから自身で行ってもらい、以下の図 8 に示す文を問題文としてタイピングしてもらおう。この問題文には、オープンソースのタイピング練習ツールである KLAVARO の問題文を使用した[15]。本実験では、カメラ解像度は 640x480、フレームレートは 15fps に設定した。そして、押されたキーについてキーボードを見ていたかどうか判定するために暫定的にそこから 8 フレーム分マッチ度を遡って比較し、4 フレーム以上キーボードを見ていたときに作成されたテンプレート画像から算出されるマッチ度が小さかった場合、そのキーはキーボードを見ながらタイプされたと判定するように設定した。図 9 に本実験で作成されたテンプレート画像を示す。(a) がディスプレイを見ているときに作成されたテンプレート画像、(b) がキーボードを見ているときに作成されたテンプレート画像となる。

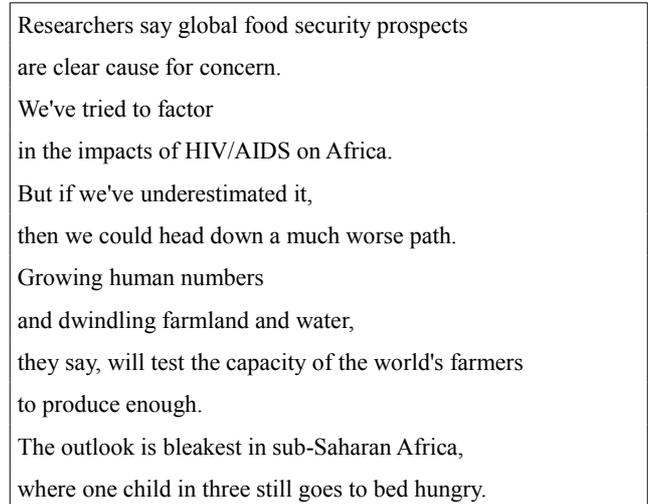


図 8: 問題文

Figure 8: Sentences used in experiment

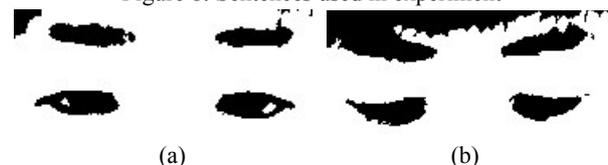


図 9: 本実験で作成されたテンプレート画像

Figure 9: Template images used in experiment.

4.2 実験結果

本実験で設定した条件で得られた結果、そして異なる条件でマッチ度の遷移を検証した結果を述べる。また、新たに設定した条件で判定に成功したケース、成功しなかったケースについて示す。

4.2.1 視線判定結果

まずは被験者がキーボードを見たときにこのツールがキーボードを見ていると判定できたかどうかを述べる。被験者はキーボードを見ながらタイプしたキーが多数あったにも関わらず、今回用いた条件ではキーボードを見ていると判定されたキーはなかった。その理由については5節で検討する。そこで、数フレーム遡って判定するのではなく、キーボードを見ながらキーを押した瞬間の判定ができていたかを確認した。被験者が1回の視線移動で複数のキーを打っていたこともあり、視線移動とキー入力一対一に対応しなかった。そこで入力単語毎に視線判定結果を照合することにした。なお、1単語で10打以上しなければならぬ場合は前半と後半で分ける。例えば、1行目の3単語目にあたる「global」であれば、この単語中であればどこで何回キーボードを見ようと1回キーボードを見たときと数える。1行目の1単語目にあたる「Researchers」はシフトキーも含め、12打しなければならぬので、「Resea」、「rchers」の2つに分けて数える。つまり「Researchers」の場合であれば、最大2回キーボードを見たときと数えることになる。また、コンマやピリオドはその直前の単語に含むものとする。ここではこの区切りをフレーズと呼ぶ。図10に実験映像によりキーボードを見ていると判断したフレーズに下線をつけたものを示す。

<u>Researchers</u> say global food security prospects	1
are clear cause for <u>concern</u> .	2
<u>We've</u> tried to factor	3
in the impacts of <u>HIV/AIDS</u> on <u>Africa</u> .	4 5 6
<u>But</u> if <u>we've</u> underestimated it.	7 8 9
then we could head down a much worse <u>path</u> .	10
<u>Growing</u> human numbers	11
and dwindling farmland and <u>water</u> .	12
they <u>say</u> will test the capacity of the <u>world's</u> farmers	13 14
to produce <u>enough</u> .	15
<u>The</u> outlook is bleakest in <u>sub-Saharan Africa</u> .	16 17 18 19
where one child in three still goes to bed <u>hungry</u> .	20

図 10: キーボードを見ながらタイピングしていた単語

Figure 10: Underline words are typed while looking at keyboard.

図10に示したとおり、「HIV/AIDS」と「sub-Saharan」がそれぞれ10打以上になり、前半と後半それぞれでキーボードを見ていたので2回と数える。これらキーボードを見ながらタイプしたと確認された20フレーズの中で、そのフレーズの最初のキーを押してから、そのフレーズの最後のキーを押すまでの間に、1フレームでもキーボードを見ていると判定されたものを○、そうでないものを×とした照合結果を表1に示す。

表 1: キーボードを見たフレーズと照合結果

Table 1: Phrases types during looking at the keyboard.

Resea	○	concern.	○	We've	×	HIV/	○
AIDS	×	Africa.	○	But	○	we've	○
it,	○	path.	○	Growing	○	water	○
say,	○	world's	○	enough.	○	The	○
sub-S	○	aharan	○	Africa.	○	hungry.	○

4.2.2 被験者がキーボードを見たときにマッチ度の逆転が起こったケース

今回の被験者は問題文(図8)をすべて打ち終わるまでにおよそ2分10秒かかった。これをフレーム数に換算するとおよそ2000フレームほどになり、マッチ度をすべて記載するには多すぎるのでその一部を抜粋して示す。被験者がキーボードを見たときに作成されたテンプレート画像から算出されたマッチ度の方が低い値をとったケース、すなわち表1中の判定に成功した(○)中から、ここでは「Growing」というフレーズを取り上げる。

ここではその中から「Grow」という部分について説明する。Growとタイピングしている間の2つのマッチ度の遷移、そしてキーが押されたタイミングを表すグラフを図12に示す。グラフの縦軸はマッチ度、横軸はタイピングを始めてからの経過時間を表しており、逆三角でプロットした点はそのタイミングで押されたキーを表している。このグラフでは、シフトキーを押す直前、そしてrキーを押している間にキーボードを見ているときに作成したテンプレート画像から算出されるマッチ度のほうが小さい値をとっている。このときの被験者の様子がそれぞれ図11である。このように被験者がキーボードを見ているときに、キーボード側のマッチ度が低くなるという、つまりマッチ度の逆転という想定していた挙動が確認された。

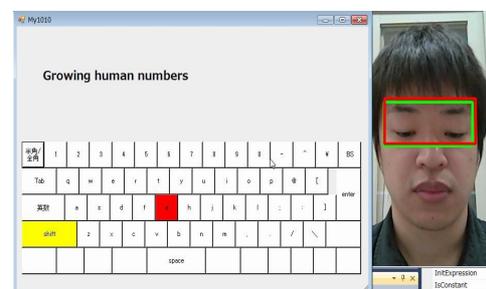


図 11: 「G」をタイピングしているときの様子

Figure 10: Screenshot of typing "G"

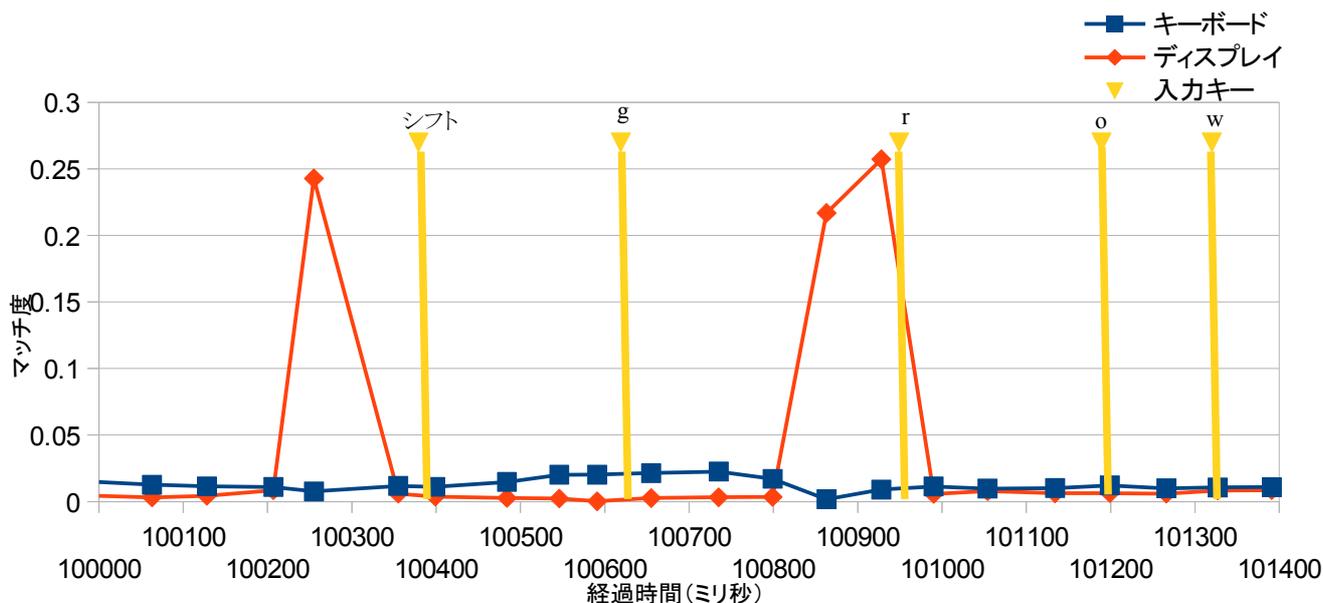


図 12: Grow 打鍵時の視線のマッチ度の遷移

Figure 13: Change in the matching rate during typing "Grow".

4.2.3 被験者がキーボードを見たときにマッチ度の逆転が起こらなかったケース

逆に被験者がキーボードを見ていたにも関わらず、このツールがキーボードを見ていたと判定できなかったケースについて紹介する。そのケースとは3行目の「We've」になる。このケースについても4.2.2節と同様に、We'veとタイピングしている間の2つのマッチ度、そしてキーが押されたタイミングを表すグラフを図14に示す。そしてこのときの被験者の様子を図13に示す。このように被験者はキーボードを見ていたにも関わらず、ディスプレイを見ていたときに作成したテンプレート画像から算出されたマッチ度の方が小さい状態が続いている。vキーを押したときにはキーボードを見ていたときに作成したテンプレート画像により算出されたマッチ度が小さくなり、逆にディスプレイを見ていたときに作成したテンプレート画像により算出されたマッチ度は大きくなっているが、それでも逆転するまでには至らなかった。このように、被験者がキーボードを見ていたにも関わらず、マッチ度の逆転が起こらなかったケースも確認された。

5. 考察

実験を行った結果から、判定方法、キーボードを見るタイミング、判定精度について考察を行う。

5.1.1 判定条件について

タッチタイピング練習の際の素早い視線の動きを正確に検出することはできず、そのため、キー毎にキーボードを見ていないかどうかを判定することはできなかった。本実験では、練習者がキーボードを見るときはキーを押す前に視線を動かし、その後、数フレーム分はキーボードを見た状態が維持されると予想し、そこで、そのキーが押されたときから8フレーム分遡り、その内キーボードを見ていたときに作成したテンプレート画像から算出されるマッチ度のほうが低かったフレームが4フレーム以上あった場合、そのキーはキーボードを見ながらタイプしたという判定条件を設定することで、視線の動きを正確に検出することができなくてもキー毎に正しい判定ができることを期待した。しかし、図12のように、視線を動かしても1~2フレームの間しかマッチ度が変化しなかった。そのため、何れのキーについても、キーボードを見ながらタイプしていたと判定することができなかった。そこで、4.2.1節で、キー毎ではなくフレーズ毎にマッチ度の遷移を比較した。4.2.1節では、キーボードを見ながらタイプしていたフレーズのうち90%のフレーズでマッチ度の逆転が起こっていたという結果が得られた。この条件ではフレーズ毎でしか判定できず、どのキーをキーボードを見ながらタイプしたのか判定できないという問題点がある。そのため、今後どのような判定条件がよりよい精度で、フレーズ毎ではなくキー毎に判定を行えるのか検討する必要がある。

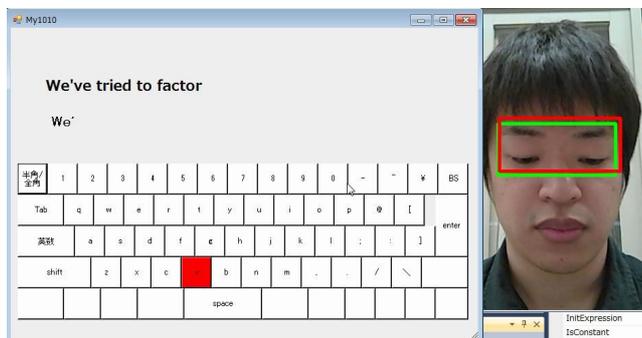


図 13: 「We've」とタイピングしているときの被験者の様子

Figure 14: Screenshot of typing "We've".

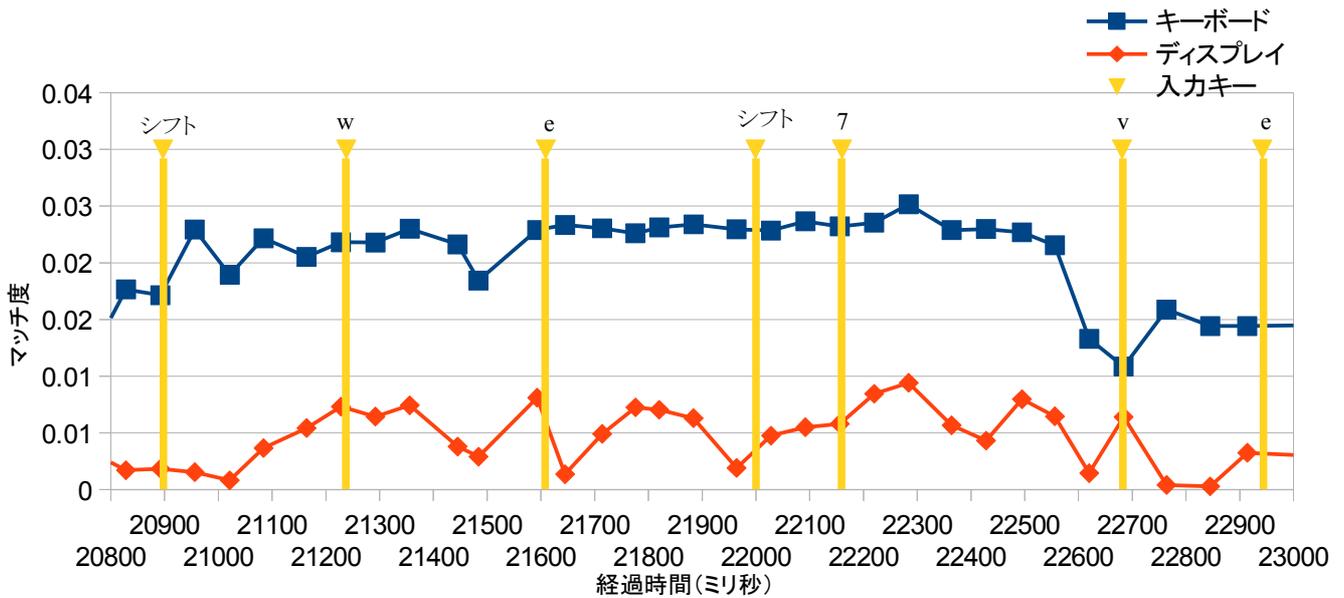


図 14: We've 打鍵時のマッチ度の遷移

Figure 15: Change in the matching rate during typing "We've".

5.1.2 視線移動の不規則性について

キー毎にキーボードを見ていないかどうか判定する条件を決める際、キーボードを見るタイミングが一定ではないということも考慮しなければならない。本実験の被験者は1名であったが、それでもキーボードを見るタイミングが一定ではなかった。つまり、キーボードを見ながらタイプするケースもあれば、キーボードを見て、キーの位置を確認した後、視線を戻してタイプするというケースも観測された。一般的な視線の動かし方として、図12のシフトキー、rキーのようにキーを押す前にキーボードを見ると考えられる。つまり、そのキーをタイプした瞬間の視線の動きと、キーボードを見たかどうかということは必ずしも一致しない可能性があると考えられる。そのため、判定条件と密接に関わってくるが、そのキーを押した瞬間のマッチ度のみを判定に用いるという条件では、正しい判定が行えない可能性がある。本実験では観測されなかったが、初心者だった場合、1回のキー入力に対し、複数回キーボードを見てしまうケースも考えられる。これらのケースは、視線の動きに対応したマッチ度が算出されていれば基本的に問題ないはずだが、キーボードを見ていないか判定する条件を設定する上で、このような被験者の動きもあることを考慮すべきだろう。

5.1.3 判定精度の問題

視線を動かした際に、我々が想定していた通りのマッチ度の遷移が見られないケースもあった。そのため、判定精度の向上が必要である。本実験では、図14のようなマッチ度の遷移の他に、キーボードを見ていないにも関わらず、マッチ度の逆転が起こったケースも観測された。このケースは、4.2.1節と同じ条件でフレーズ毎に区切り数えると、実験映像に基づきキーボードを見ずにタイプしていたと確認された58フレーズのうち、20フレーズでキーボードを見たときに作成されたテンプレート画像から算出されたマッチ度の方が低い値をとるという間違った結果となっていた。つまり、キーボードを見ずにタイプしていたフレーズのおよそ35%は、被験者の視線の動きと、我々が想定していたマッチ度の遷移が一致しなかったことになる。そのときの一例を図15に示す。図15のように検出された領域の

下半分に当たる目の形状はディスプレイを見ているときに作成したテンプレート画像の下半分の形状と似ているように見えるが、マッチ度はキーボードを見ているときに作成したテンプレート画像から算出されたマッチ度の方が低いという結果になった(2.2節で示したように、2つの画像の形状が似ているほどマッチ度は低い値となる)。このような結果となった原因は現在調査中である。本ツールは、練習中に練習者がキーボードを見たときにリアルタイムでキーボードを見ているとフィードバックすることを目指している。そのため、キーボードを見ていないときのマッチ度の遷移も重要であると考えている。本実験では、キーボードを見ていないときにタイプしたフレーズのうち約35%に対してキーボードを見ていたというアラートを表示した。リアルタイムフィードバックを行うためにも、より高い精度で練習者の視線の動きを検出する必要がある。

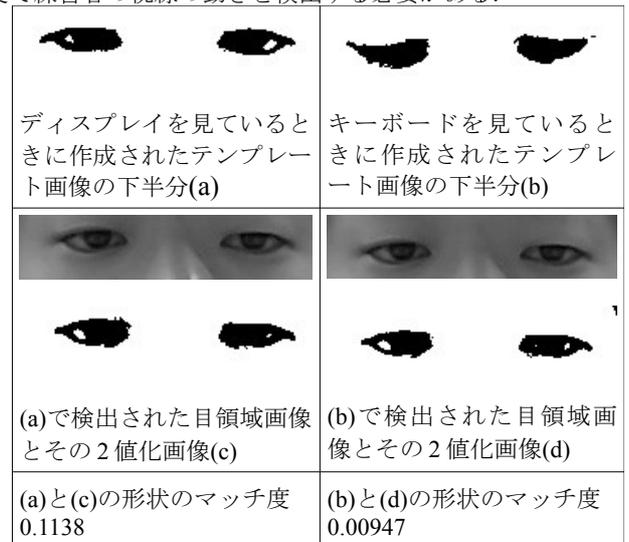


図 15: ディスプレイを見ているにも関わらずキーボードを見ていると判定されたときのマッチ度算出過程

Figure 16: Screenshot and binary image of looking at the keyboard.

6. 今後の課題

練習者がキーボードを見ながらキーをタイプしているときにキーボードを見ていると判定すること、練習者がキーボードを見ていないときにキーボードを見ていないと判定すること、これらの判定精度を向上させることが必須である。そのためには、練習者の視線の動きをより高い精度で検出しなければならない。マッチ度の算出については図 15 のように、見た目から予想されるマッチ度と実際に得られるマッチ度には差がある。つまり、いかに2値化を上手く行い形状が似ている画像が作成されても、我々が望んでいるマッチ度が得られない可能性がある。判定精度向上のため、より多くの被験者で実験を行い、より多くのデータを集計し、適切なキーボードを見ていないか判定する条件を導き出したいと考えている。また、このツールは現在フレームレートが 15fps に設定されているが、より高いフレームレートで動作させ、実験を行いたいと考えている。これは、フレームレートを上げることでより細かい時間間隔でマッチ度を算出することができ、より多くの判定材料を得られるためである。また、キーボードを見ていないのにキーボードを見たときと判定されるケースとキーボードを見ていないときにキーボードを見たときと判定されるケースの区別が容易になるのではと期待している。

7. まとめ

我々は、視線判定機能によりタッチタイピングの練習を支援するツールを開発した。このツールは、特別な装置や特別な環境を必要とせず、一般的な USB カメラやノートパソコンなどに内蔵されているカメラを使って、一般的な PC ユーザーであれば誰でも使用することができることを目指した。このツールは解像度 640x480、フレームレート 15fps で動作し、リアルタイムで処理を行うことができた。このツールを使い、タッチタイピングの練習のときのようなすばい視線の動きでもキーボードを見ていないか判定できるかどうか確認するため実験を行った。練習者は、キーを押す前にキーボードを見、キーを押し終わるまでキーボードを見ているという予想から設定した暫定的な判定条件では、結果として、キー毎にキーボードを見ていないか判定を行うことはできなかった。そこでキー毎ではなくフレーム毎にマッチ度の遷移を確認したところ、実験映像からキーボードを見ながらタイプしたと確認されたフレームのうち、90%のフレームでキーボードを見ているというマッチ度の遷移が観測された。しかし、キーボードを見ずにタイピングできていると確認されたフレームのうち、キーボードを見ながらタイプしたというマッチ度の遷移が確認されたフレームが約 35%に及んだ。そのため、判定精度を向上させること、特に適切な判定条件を見つけることが非常に重要であると考え。今後、より多くの被験者で実験を行い、より多くのデータを収集する必要がある。また、フレームレートを上げることで、判定精度を向上できるのではないかと考えている。

謝辞 タッチタイピング練習ツールの技術資料を提供していただいた龍弘大氏（現 NTT コムウェア）をはじめ、ご協力頂いた皆様に、謹んで感謝の意を表す。

参考文献

- 1) タッチタイピング練習ツール一覧, Vector
<http://www.vector.co.jp/vpack/filearea/win/edu/comp/typing/>
- 2) K. Ryu, Y. Kondo, T. Nagai and H. Nakano, "Developing a

multilingual online typing tool that uses a sentence from web search results", IEICE Tech. Rep., vol. 110, no. 85, ET2010-13, pp. 37-42, June 2010.

3) Y. Kondo, H. Nakano and T. Nagai, "SCORM compliant typing tool that generates exercises from Wikipedia", IPSJ SIG Technical Reports, 12, pp.96-103, Sept. 2009.

4) T. Imamura, T. Nagai, H. Nakano, "A simple eye-gaze detecting function for touch-typing training system", Global Engineering Education Conference (EDUCON), 2012

5) M. Arai, H. Shimosakai, H. Watanabe and S. Takei, "Recognition of Eye Direction Using Separability Filters in Touch-Type Training", International Conference on Computers in Education (ICCE'02), pp.67-68, December 2002.

6) R. Valenti, J. Staiano, N. Sebe and T. Gevers, "Webcam-Based Visual Gaze Estimation" Image Analysis and Processing – ICIAP 2009, vol. 5716/2009, pp. 662–671, 2009.

7) R. Valenti, T. Gevers, "Accurate Eye Center Location and Tracking Using Isophote Curvature", IEEE Conference on Computer Vision and Pattern Recognition, 2008.

8) M. Chau, M. Betke, "Real Time Eye Tracking and Blink Detection with USB Cameras", Boston University Computer Science Technical Report No. 2005-12, 2005.

9) J. Zhu, J. Yang, "Subpixel eye gaze tracking", Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on, pp.124 - 129, 2002.

10) Hennessey, C. Nouredin, B. Lawrence, P., "Fixation Precision in High-Speed Noncontact Eye-Gaze Tracking", Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, pp289-298, 2008.

11) R. Khilari, "Iris tracking and blink detection for human-computer interaction using a low resolution webcam", ICVGIP '10 Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing, 2010.

12) 画像処理ライブラリ OpenCV

<http://opencv.willowgarage.com/wiki/>

13) M. K. Hu, "Visual Pattern Recognition by Moment Invariants", IRE Trans. Info. Theory, vol. IT-8, pp.179-187, 1962.

14) N. Otsu, "A threshold selection method from gray level histograms", IEEE Trans. Systems, Man and Cybernetics, 1979, Vol.9, pp.62-66

15) オープンソースタッチタイピング練習ソフト KLAVARO
<http://klavaro.sourceforge.net/>