

# ドライアイ軽減のための自動まばたきリマインド機能

東覚瑠菜<sup>†1</sup> 神場知成<sup>†2</sup>

**概要**：ドライアイを軽減することを目的とし、利用者のまばたき回数を自動的に検知し、その状況によってまばたきをうながすシステムを開発した。パソコンに内蔵されたカメラで取得する利用者の顔画像からまばたきを検知し、一定のルールにしたがって、画面上にまばたきのリマインドを表示する。①リマインドをしない場合、②1分に1回定期的にリマインドをする場合、③10秒間にまばたき回数が2回以下になった場合にリマインドをする場合、を比較し、現時点で明確な効果を確認するには至らないが、リマインド直後の10秒間は他の時間帯と比較して顕著にまばたき回数が増えるなどの観察を得た。従来このようなリマインド機能を実装した例はあるが、状況とは無関係に定期的にリマインドするものだけであり、利用者状況に応じて動的にリマインドを行うものはなかった。引きつづき適切なリマインド条件やリマインド方法を検討し、実用性の向上を目指す。

## 1. はじめに

目の疲れや不調を感じる人は多いが、十分な対策がとられているとは言えない。たとえば、ライオン株式会社がコロナ禍による外出自粛の状況の中で20～60代の男女合計300名に対して行った調査によると、目に対するケアは他の身体の部位と比べておろそかな傾向があったとされている[1]。

従来から、眼精疲労にはVDT作業の影響が大きく、VDT作業者の多くが眼精疲労の問題を抱えていることが知られている[2]。Thomsonによる調査では、コンピュータユーザの最大90%が、長時間コンピュータを使用した後で近くまたは遠くを見たときに、目の緊張、頭痛、目の不快感、目の乾燥、複視、視力障害などの視覚症状を経験する可能性があることが示され[3]、Dinesh J Bhanderiらが研究機関に勤務するコンピュータオペレータ419名を対象として行った調査では、194名(46.3%)が長時間のVDT作業後の眼の不快感を報告している[4]。

## 2. 従来の研究

ドライアイとは、涙液膜が崩れやすくなり目の不快感や見えにくさを生じる病気であり[5]、眼精疲労にはドライアイと関連する症状が多く含まれる。VDT作業におけるドライアイ症候群の割合は高く、原因の一つはVDT作業時にまばたきの回数が減ることであるとされている[2]。

Tsubotaらの調査[6]をもとに計算すると、コンピュータの使用中は、通常時と比べ30%ほどまばたきの回数が減少している。また同研究では104名のサラリーマンを対象に①リラックスしている時、②本を読んでいる時、③VDTでテキストを表示した時の3場面でまばたきの回数を比較

し、それぞれ①平均22回/分、②平均10回/分、③平均7回/分とVDT作業時のまばたきの回数が一番少ないという結果を示している[6]。

ドライアイでは目薬等も用いられるが、薬を用いない対策の一つとしてNoschらは、VDT作業時に定期的にまばたきをリマインドするアニメーションを使用することで、瞬きの回数が増加し、ドライアイ改善につながる可能性があることを示した[7]。ここでは1分間あたりのリマインドの回数を1回、2回、または個人の通常時のまばたきの回数などに応じて設定するとともに、リマインドの際にまばたきする回数を被験者に指示した。その結果、リマインドの度に2回まばたきするように指示した場合がもっとも効果的であり、ドライアイ改善に大きくつながると述べている。

しかし、利用者が長時間作業をしている最中にずっと定期的にまばたきを指示すると、結果的にリマインドの回数が非常に多くなり、長期的には利用者の慣れが生じて効果が薄くなる可能性がある。また、リマインド時に利用者に対してまばたき回数を明示的に指示することも、日常的な利用には適さない。このため本稿では、PCに内蔵されたカメラを用いて利用者のまばたき回数を自動的に検知し、まばたきの回数が減少してきたときだけリマインドを行う機能の提案をするとともに、利用者に対するまばたき回数の明示的な指示をしない状態での評価を行う。

## 3. システムの概要と実験手法

### 3.1 システム構成

システム構成を図1に示す。システムは「まばたき検知機能」と「リマインド機能」から構成され、それぞれ下記のように実装している。

<sup>†1</sup> 東洋大学 情報連携学研究所

<sup>†2</sup> 東洋大学 情報連携学部

### (1) まばたき検知機能

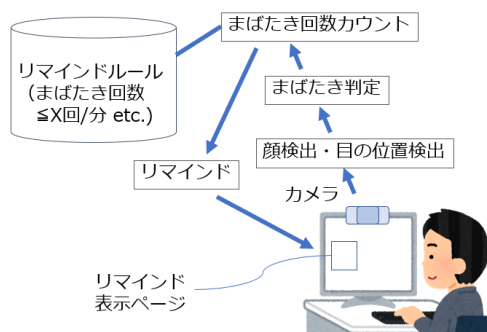


図 1. システム構成

ユーザが PC 上で作業をしている間、モニタ上部に設置された内向きカメラがユーザの顔画像を映し、画像処理によってまばたきを検知する。まばたきの検知は、ブラウザ上で、顔検知用の JavaScript API である `face-api.js landmarks` を利用している[8]。同手法ではライブラリにより検知された目の位置にもとづいて黒目付近の画素を判定し、その RGB 値が黒色からはずれた時 (RGB 値が閾値以上になった時) をまばたきと判定している。本実験ではまず、実験室環境において筆者等が実験用のノート PC を用いて 15 分程度の作業を行い、本検知機能でまばたき回数を測定したものと、該当時間に撮影した映像を目視確認して実際のまばたき回数とがなるべく近くなるように、黒目からはずれたことを判定するための閾値を調整した。次に、被験者数名で同様の確認を行い、結果的に実際のまばたき回数と、本検知機能による測定回数の「ずれ」が、ほぼ 10 分間に 3 回となることを確認した。

### (2) リマインド機能



(a) イラスト表示 (b) テキスト表示

図 2. リマインド表示機能

システムは、一定の条件が満たされたときにユーザにまばたきを促す機能を持つ。同機能は、JavaScript から Chrome ブラウザ上のリマインド表示ページを制御することで実装した。リマインド表示方法としては、図 2 の(a), (b)に示すように、イラスト表示とテキスト表示がある。

## 3.2 実験手法

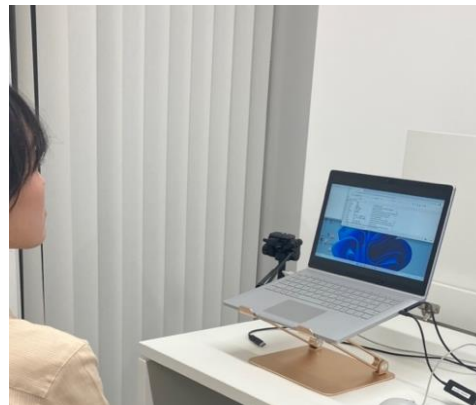


図 3. 実験風景 (PC 横のカメラは記録用)

上記のシステムを利用し、図 3 に示すような環境において下記のような実験を行った。なお、部屋が乾燥している際はまばたきの回数が増える可能性があるため、加湿器を用い部屋の湿度を最適とされる 40~50%に保った。マスクをはずした状態で実験を行っているが、これは顔検出ライブラリの精度が、マスクをした状態では低下することを確認したためである。

### 1) 実験概要

被験者 4 名に、画面上に表示した英単語と日本語訳、同意語が表示された Excel シートをスクロールしながら 15 分間見てもらう。単語数は十分あり、通常で 15 分間スクロールしていても読み終わることはない。

その間、システムは時々 Excel シートの左側に、まばたきのリマインドを表示する。表示は図 2(b)のテキスト表示を用いた。被験者に対して、まばたきのリマインドをすることはあらかじめ伝えてあるが、その際にまばたきを何回するかなどの指示はしていない。実験時の画面例を図 4 に示す。Excel シートやリマインド表示を画面上部の領域に設置しているが、これは被験者の視線を安定させ、まばたき検出の精度を安定させるためである。



図 4. 実験時の画面 (左側がリマインド表示)

## 2) リマインド表示ルール

被験者 1 名あたり、まばたきを促すタイミングを下記に示す 3 つのルールで行った。目の疲労を考慮し、同一被験者に対する異なるルールによる実験は別の日に行った。

R1) リマインドなし

R2) 1 分間に 1 回、定期的にリマインド

R3) まばたきが 10 秒あたり 2 回以下になった場合のみリマインド

R3)については、一般の人の通常時のまばたき回数の平均が 1 分間あたり 10-16 回前後とされているため[7]、10-16 回を大幅に下回ることを防ぐよう 10 秒に 2 回以下と設定した。

## 4. 結果と考察

### 4.1 実験結果

表 1. 被験者ごとのまばたき回数

被験者 A

ルール	0-5 分	5-10 分	10-15 分	合計
R1	83	145	183	411
R2	106	93	93	292
R3	123	171	161	455

被験者 B

ルール	0-5 分	5-10 分	10-15 分	合計
R1	190	191	158	539
R2	160	165	222	547
R3	189	159	137	485

被験者 C

ルール	0-5 分	5-10 分	10-15 分	合計
R1	101	80	66	247
R2	136	154	124	414
R3	149	188	157	494

被験者 D

ルール	0-5 分	5-10 分	10-15 分	合計
R1	142	141	119	402
R2	129	104	121	354
R3	157	168	156	481

表 1 に、被験者ごとに、各リマインドルールに基づき制御した際のまばたき回数を示す。実験時間の 15 分間のうち、最初の 5 分、次の 5 分、最後の 5 分に分けて示している。

また、リマインドの表示回数は結果的に表 2 のようになった。R3 の場合だけ、被験者によりリマインド表示回数異なる。

表 2. リマインド表示回数

ルールによる違い

	R1	R2	R3
被験者 A	0	15	19
被験者 B	0	15	20
被験者 C	0	15	7
被験者 D	0	15	6

R3 におけるリマインドタイミング

	0-5 分	5-10 分	10-15 分	合計
被験者 A	8	6	5	19
被験者 B	2	7	11	20
被験者 C	3	1	3	7
被験者 D	2	2	2	6

(1) 総まばたき回数の比較 (15 分間)

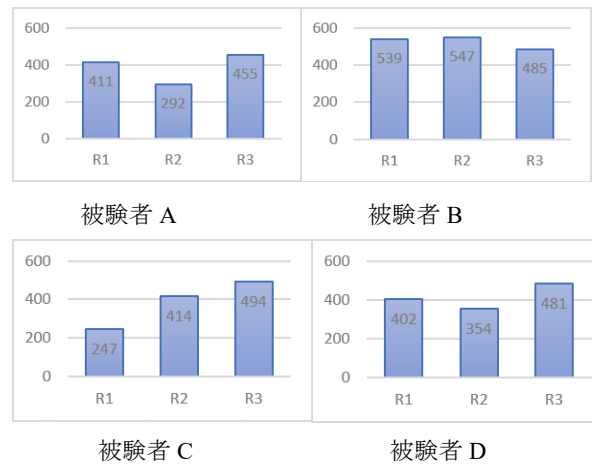


図 5. 総まばたき回数

各被験者に対してそれぞれのリマインド表示ルールを用いた時の総まばたき回数を図 5 に示す。左から順に R1, R2, R3 における総まばたき回数である。R1 と R2 を比較すると、被験者 A と D では R2 の方が総まばたき回数が少なく、被験者 B と C では多くなっている。R2 では継続的にまばたきのリマインドを表示しているのやや意外な結果である。また、R1 と R3 を比較すると、被験者 B を除いた 3 名については、R1 の総数と比べ R3、すなわちまばたきが減ったタイミングでリマインドされた際の方が、まばたきの総数が多くなっている。被験者 B 以外の 3 名については、R2 と R3 を比較するとまばたきの総数は R2 < R3 である。全体としては、ある程度リマインド機能の効果は見られるが、ばらつきが大きい。また、定期的にリマインドを促している R2 において R1 よりもまばたきが少なくなっている点がある点は説明がつかないが、総合的に見ても定期的にまばたきを促すことは、あまり効果が高くないとみなすこともできる。

(2) 時間ごとのまばたき回数の変化

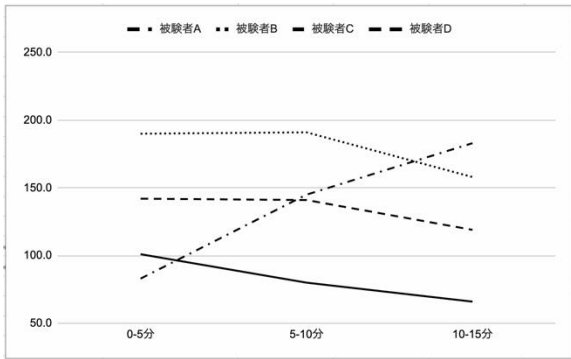


図 6. R1 における 5 分ごとのまばたき回数の変化

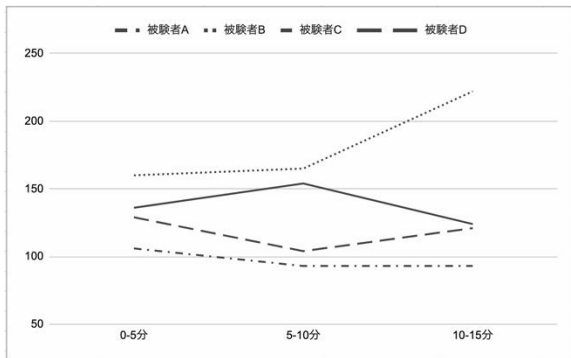


図 7. R2 における 5 分ごとのまばたき回数の変化

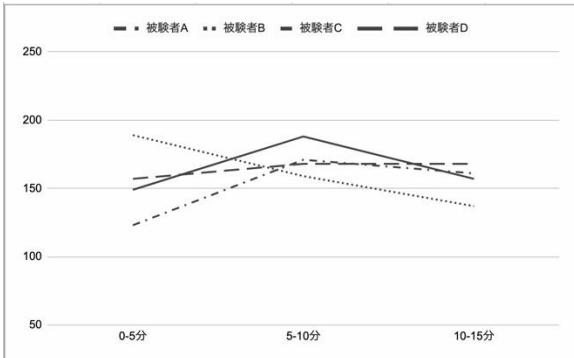


図 8. R3 における 5 分ごとのまばたき回数の変化

次に、15分間の実験を5分ごとに分けた時の各時間枠におけるまばたき回数の変化を図6から図8に示す。

図6によれば、リマインドがない状態で被験者Aを除く3名は、初めの5分と比べ最後の5分のまばたき回数が少なくなっており、時間の経過とともにまばたきの回数は減少傾向にある。被験者Aだけは時間とともに顕著にまばたきの回数が増加している。

図7によれば、R2の手法では明確な傾向はみられない。

図8によれば、R3の手法では全体のまばたきの回数が

安定する傾向が強まり、個人差も減ってまばたきの回数は5分間に150回程度の中に収まっている。

また、全体を通してR1で5分間あたりまばたきの最小値は66回であったのに対し、R3では123回と、最小値が増加している。R3でリマインドを出すのは10秒間あたりのまばたきの回数が2回を下回ったときであり、これは5分間に換算すると60回なので、まばたき回数が一定値を下回ったときにリマインドをすることは一定の効果を出していると推測できる。

#### 4.2 考察

##### (1) リマインド表示とまばたきについて

上記のように、まばたきの総数だけを見るとリマインドの効果はやや不明確だが、あらためてリマインド直後の10秒間のまばたき回数と、その他の時間におけるまばたき回数の比較をすると、次のようなようになった。

##### 実験全体を通じた10秒当たりのまばたき回数

平均：4.7 標準偏差：1.0

##### R2のまばたき回数

リマインド直後の10秒間

平均：5.4 標準偏差：0.8

その他の10秒間：4

平均：4.4 標準偏差：1.3

##### R3のまばたき回数

リマインド直後の10秒間

平均：5.5 標準偏差：1.5

その他の時間

平均：4.6 標準偏差：0.3

R2, R3ともに、リマインド直後の10秒間のまばたき回数の平均は、その他の時間の平均と比べて約20%高くなっており、被験者はリマインド表示に気づき、すぐに意識的にまばたきを行ったことが推測される。

##### (2) リマインド表示方法について

今回はテキストを用いたリマインド表示を行っており、図2(a)に示したようなイラストによるリマインドは行っていない。これは、イラストでは絵の図柄の影響が大きいと考えたためだが、テキストはイラストよりも目立たず、被験者がやや気づきにくい。被験者からの感想としても、途中リマインドの存在を忘れテキストに気づかない時もあったというものもあった。今後、イラスト表示による効果なども検討する予定だが、それ以外にも、用いるテキスト文、フォントサイズ、点滅やアニメーションの有無など、リマインドの表示位置などさまざまな要素が考えられ、それら

については本稿で述べたような表示タイミングとは別の観点からの検討となる。

### (3) まばたき検知機能について

実験を行うにあたっては 3.1 に述べたように事前にできる限りの調整を行ってまばたき検知の誤差を 10 分間に 3 回程度となるようにしたが、実験をすべて終えた後でビデオ映像を用いて一部の再確認を行ったところ、これを超える誤差が出ていた場合もあるようだった。すべての確認を行うことはできなかったため部分的に見たところでは、被験者 B についてややまばたき判定がうまく行っていない場合が見受けられた。うまく行っていない場合の原因として、カメラに顔全体が写っておらずフレームアウトしている場合、手で顔を覆っている場合、横を向いているなどが見られた。実験としては、これらを制御したうえで再度行うことも検討しているが、本稿で述べたような機能を実用化するためにはむしろ、それらの自然な状況でもまばたき検知を判定することができたり、あるいは検知ができなくなった場合には自動的に、定期的なリマインド機能を出すように切り替えたりするなどの工夫が必要であろう。

## 5. おわりに

ドライアイの軽減を目的とし、PC 利用者の作業中のまばたきを検知し、適切なタイミングでリマインドを出すシステムの構築と実験を行った。従来、常時一定タイミングでリマインドを出す機能や、リマインド時に行うべき行動（まばたき 2 回など）を指示した際の効果はある程度示されていたが、本稿では、利用者のまばたき状況を検知して状況に応じたリマインドをする試みを行った。

効果としては、現段階ではまばたき検知機能が正確とは言えなかったことや、被験者数が十分でないため結論を出すことはできないが、リマインド直後のまばたきの回数が他の時間の回数と比べ多いことなど、一定のリマインド効果は確認できた。

今後、長時間ディスプレイを見ることによる目への負担はさらに増え、それに伴いドライアイのような症状も増えると想定される。さらに改良を行うことで実用的なドライアイ対策へとつなげていきたい。

**謝辞** 本研究は、東洋大学重点研究推進プログラムにより助成を受けたものです。同助成に感謝いたします。

## 参考文献

- [1] ライオン・外出自粛下の”目の疲れ・不調”に関する実態調査、<https://www.atpress.ne.jp/news/213544> (2020.5.25)
- [2] Erdinest N, Berkow D. [COMPUTER VISION SYNDROME]. Harefuah. 2021 Jun;160(6):386-392. Hebrew. PMID: 34160157.
- [3] Thomson WD. Eye problems and visual display terminals--the facts and the fallacies. Ophthalmic Physiol Opt. 1998

Mar;18(2):111-9. PMID: 9692030.

(<https://pubmed.ncbi.nlm.nih.gov/9692030/>)

- [4] D.J. Bhanderi et al. : A community-based study of asthenopia in computer operators, Indian J Ophthalmol. 2008 Jan-Feb; 56(1): 51-55. doi: 10.4103/0301-4738.37596
- [5] ドライアイ：日本眼科学会ホームページ (<https://www.nichigan.or.jp/public/disease/name.html?pdid=9>)
- [6] Kazuo Tsubota and Katsu Nakamori: Dry Eyes and Video Display Terminals, New England Journal of Medicine 328 (8), 584-584 (1993).
- [7] Nosch DS, Foppa C, Tóth M, Joos RE. Blink Animation Software to Improve Blinking and Dry Eye Symptoms. Optom Vis Sci. 2015 Sep;92(9):e310-5. doi: 10.1097/OPX.0000000000000654. PMID: 26164310. (<https://pubmed.ncbi.nlm.nih.gov/26164310/>)
- [8] Javascript でまばたき検知 face-api.js landmarks (<https://g-llc.co.jp/blog/programming012/>)