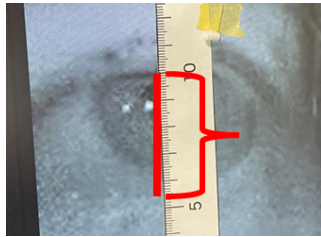


お世話になります。
お忙しいところすいませんが、以下内容について、ご依頼は可能でしょうか
ご検討宜しくお願い致します。
提案期限と費用について、相談可能です。

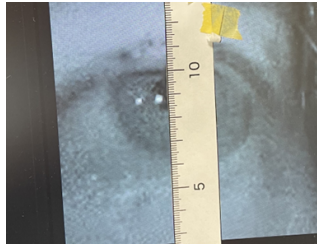
- 希望納期: 1~2週間以内であればうれしいですが、時間がかかる場合には遅れても構いませんので、可能な納期をご提示いただけるでしょうか。
 - 動作環境: Windows10か11
 - 言語: 相談可能であればしたい。
 - 費用: 1~3万程度でお願いしたいです。可能な範囲をご提示ください。
-

- 開発物
 - 以下の作業を半自動化するソフト
 - 作業: 目のカメラ映像1フレームごとに、上瞼と下瞼の位置を手動検出。
 - 詳細は、点線以下の作成ソフトのイメージの動画を閲覧ください。
- 作成ソフトのイメージ
 - 以下からダウンロードお願いできるでしょうか
 - <https://70.gigafile.nu/1203-h602ed2fbf1cf1c6e056c2f07c1303e36>
 - 以下の内容のファイルがあります。
 - 説明動画
 - 動画での利用資料
 - 補足:
 - 説明資料の以下の説明箇所の内容は実行困難と考えているため、無視してください。
 - スライド: 拡張可能か伺いたい点(画像解析で自動検出の例のスライド)
 - 動画: 14:49~
 - 内容: 画像解析で自動検出の例
- 解析素材の動画
 - 人が瞬きをしている際の目のカメラ映像があります。
 - <https://70.gigafile.nu/1203-d91df36f667b22e9763f4ed3756e1588d>
 - 上のURLには以下の5種が入ってます。カメラ映像のフレームレート(100~200HZ程度)、画質は動画ごとに異なります。以下について、片方の目だけを対象にします。両目の動画についても片目だけに限り行います。
 - ①
 - eye0 120HZ
 - 横400、縦400、片方の目が映っている
 - ②
 - eye0 240HZ
 - 横192、縦192、片方の目が映っている
 - ③
 - パタン1 123HZ
 - 横192、縦192、片方の目が映っている
 - ④
 - パタン2 199HZ
 - 横:384、縦192、両目が映っている(片目ごとに横192、縦192の動画を、結合していると思う)
 - ⑤
 - パタン3 197HZ
 - 横:384、縦192、両目が映っている(片目ごとに横192、縦192の動画を、結合していると思う)
 - 出力内容
 - ソフトでは以下の2種のファイルを出力します
 - CSV
 - 上瞼と下瞼の距離の出力をしたいというものです。
 - これに関わるデータを全部取得したいので、本ドキュメントの最終ページに記載の情報が欲しいです。
 - ※次の設定がほしい
 - モード①: 検出点として打った点だけ数値が埋まっていて、その他は空欄。例えば、10フレームごとに瞼の位置を記録した場合、1,10,20,30,...フレーム目については、csvに数値が入っているが、間のフレーム(例えば、2~9フレーム目)は空欄
 - モード②: 検出点として打っていない点は補完: 手前のフレームの数値を入れておく、あるいは点と点の間を線形や2次曲線の補完をする、等があります。Pythonでは容易にできますが、利用言語で難しい場合は、モード①だけでいいです。
 - 出力画像
 - 本件は上記の説明動画を参照願います。
 - 上のcsv出力に利用した画像の出力。例えば以下を想定しています。
 - 新規フォルダを生成し、その中に画像を出力する
 - フォルダ名は「読み込みファイル名」+「出力時の現在時刻」
 - 画像の名前は「Index」+「読み込みファイル名」
 - その他

- 開眼程度(上瞼と下瞼の開き幅)とは
 - これは、以下画像で示す赤色の直線です。
 - 画像では黒目の中の瞳孔の中央から縦線を引いていますが、厳密にそのラインでなくても構いません。



-
- 開眼程度(上瞼と下瞼の開き幅)の数値化とは
 - カメラ映像をもとに、上瞼と下瞼の間の距離を数値化するとは、例えば、以下画像のようにメモリ付き定規などをもとに、上瞼と下瞼の間が、フレーム1は400mm、フレーム2が390mm、といったように、計測して記録していくことを指します



-
- ※画像に区切ってから数値化でも構いません。以下は参考までに動画をフレームごとの画像にするコードです。

csvには次の内容が出力されるイメージです

1. **index**
 - 動画の何フレーム目か
2. **TimeStamp1**
 - 動画の開始フレームを0とし、フレーム1つ毎の時間幅を加算した値。例えば、動画が100HZなら1フレーム毎に10ミリsecごと加算する。
3. **TimeStamp2**
 - 読み込み動画の録画開始時刻+TimeStamp1の値。読み込み動画のメタ情報で録画開始時刻が含まれているものを利用する。例えば、録画開始時刻が13:06:00なら、これに加算していく。
4. **上瞼と下瞼の点の間の距離_1**
 - 下で記載している「上瞼の位置_1、Y軸」と「下瞼の位置_1、Y軸」の間の距離
5. **上瞼と下瞼の点の間の距離_2**
 - 下で記載している「上瞼の位置_2、Y軸」と「下瞼の位置_2、Y軸」の間の距離
6. **上瞼の位置_1、Y軸**
 - 補助線ベースのY(説明動画の例なら-50~130)
7. **上瞼の位置_2、Y軸**
 - 画像ピクセルベースのY軸の値。(例えば、左下から上にY軸正方向、右にX軸正方向)
8. **上瞼の位置_2、X軸**
 - 画像ピクセルベースのX軸の値。
9. **下瞼の位置_1、Y軸**
 - 補助線ベースのY(説明動画の例なら-50~130)
10. **下瞼の位置_2、Y軸**
 - 画像ピクセルベースのY軸の値。
11. **下瞼の位置_2、X軸**
 - 画像ピクセルベースのX軸の値

	A	B	C	D	E	F
1	index	TimeStamp1	TimeStamp2	上瞼と下瞼の点の間の距離_1	上瞼と下瞼の点の間の距離_2	\\
2		動画の開始フレームを 0 とし、フレーム1つ毎の 時間幅を加算した値	動画の録画開始時間 + TimeStamp1			
3	1					
4	2					
5	3					
6						
7						
8						