

極楽画像計測 (measure, merge, visualize) 取扱説明書 Ver. 3.4.1用 2023.03.24

gokuraku_measure

Gokuraku measure (@tatsuva) Ver.3.4

[I] : 画像ファイルをフォルダーから読み込む
(右クリック) : 単一ファイルを開く / 終了

[S] : サブメニューを開く
[L] : 線分定義ファイルの読み込み (none)
[M] : 画像のプレビュー (ESC:中断 / スペース:一時停止)
(左クリック) : 計測
(右クリック) : 計測完了
Shift+[+] : 倍率アップ / ダウン / リセット (=1.0)

gokuraku_merge

Gokuraku merge (@tatsuva) Ver.3.4

[I] : 座標ファイルをフォルダーから読み込む
(右クリック) : 単一の座標ファイルを読み込む
[C] : 全データ消去
[S] : サブメニューを開く
[F] : 出力フォルダーを選ぶ
[E] : 統合データファイルを出力
[ESC] : プログラム終了

フォルダー: 未選択
読み込み済み: なし

gokuraku_visualize

Gokuraku visualize (@tatsuva) Ver.3.4

[I] : 結合データファイルをフォルダーから読み込む
(右クリック) : 単一のデータファイルを読み込む
[S] : サブメニューを開く
[D] : データリストを表示 (コンソール)
[L] : 線分定義ファイルの読み込み (none)
[P] : 画像のプレビュー (ESC:中断 / スペース:一時停止)
(←, →, +SHIFT, +CTRL) : 拡大縮小
[F] : 出力フォルダーの選択
[E] : 画像ファイルの出力
[G] : ガイドライン表示の反転 (true)

鈴鹿工業高等専門学校 機械工学科

白井 達也

(shirai@mech.suzuka-ct.ac.jp)

【概要】 画像計測用のプログラムである

デジタルカメラで撮影した静止画や、ビデオカメラで撮影した動画より生成した静止画像（jpeg, png, tiff, tga）上をマウスでクリックして、その画像上のx, y座標をテキストファイルに記録するソフトウェアです。1 画像あたり 1 個のテキストファイルが作成されます。Processing 上で動作します（3.5.4 and 4.2）。WindowsやLinux上で動作する実行形式も生成可能です。

ともかく大量の画像データの画像計測を素早く行いたい人向けです。

3つのソフトウェアで構成されます。

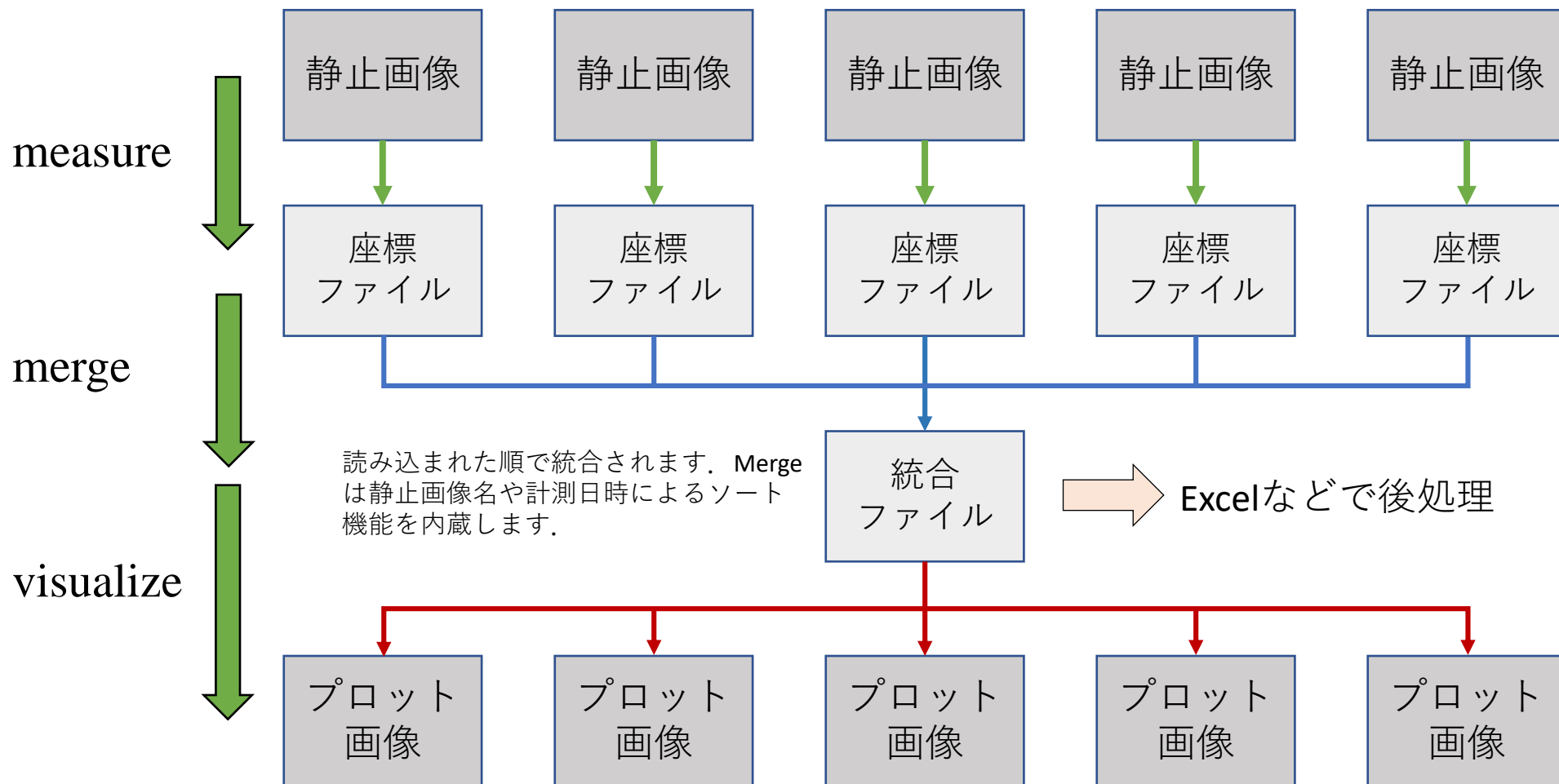
1. Gokuraku measure : 静止画像上の座標値をテキストファイルに記録
2. Gokuraku merge : measureで出力した複数のテキストファイルを一つに統合
3. Gokuraku visualize : 統合されたテキストファイルを画像に変換

measureで計測，それをmergeして，visualizeする。そういう流れです。

※長さを求める，角度を求める，面積を求めるのであれば，mergeで生成したmerged.txtをExcel等に読み込んで，各人で求めて下さい。あくまで座標値のみを計測することに特化したプログラムです。

【イメージ】

静止画像 : オリジナルの画像 (単発の静止画像, 動画から生成した連番静止画像)
座標ファイル : 計測して得られた点群のX,Y座標と色情報 (画像データファイル名+“.txt”)
統合ファイル : 座標データを1ファイルにまとめたもの (merged.txt)
プロット画像 : 統合データから生成したプロット画像 (merged-?????.jpg)



※プロット画像にはオリジナルの静止画像を合成可能です。

Stick pictureの描画にも対応 (線分定義ファイルの読み込み) しています。

複数枚のプロット画像から, Processing IDE上のムービーメーカーでMOV(QuickTime)形式に動画に合成可能ですが, MOV型式の動画を再生できる環境は限られていますので注意が必要です (後述)。

【利用目的】

- 一般の方向けのソフトウェアではありません．主に研究者（学生含む）を対象としたソフトウェアです．
- モーションキャプチャーのシステムを持っている方には不要なソフトウェアです．
- カメラ撮影した静止画や，動画を何らかのソフトウェア（aviUtil, ffmpegなど）によって連続する静止画に分解したものの中の特徴点（事前に付けたマーカーなど）を次々とクリックして座標値をテキストファイルに記録するソフトウェアです．
たとえば，スポーツや日常生活におけるヒトの体や指の動きの計測，移動ロボットの軌跡や姿勢の変化，流体中の物体の動き，大きく変形する材料の歪み，植物の成長，マウスや昆虫などの移動など，さまざまな用途が考えられます．マウスをクリックした“回数”も記録されますので，画像中の何かの数を数える用途にも使えます．
- 同様の機能を持つソフトウェア（製品／フリーウェア，例えばImage PCなど）も存在しますが，「点の座標だけを計測」する機能に限定しているため，用途によっては効率的かつ使い勝手が良いでしょう．
- リアルタイムに，動画中のマーカーの動きを計測するソフトウェアを産業用カメラメーカーなどが販売しています．静止画に変換する必要がありませんし，自動的にマーカーを追跡（Tracking）してくれるので，本ソフトウェアよりも高速かつ高精度でしょう．本ソフトウェアは，あくまで**安価に手軽に画像計測を行いたい層**がターゲットです．

【できないこと】

このソフトウェアでは二次元の画像から“画像上のx, y座標”(pixel)およびその点の色情報 (R,G,B) を取得することしかできません。それを実世界の座標系および単位 ([m], [mm]など) に変換したり, カメラの歪みを校正したり, ステレオカメラ画像から三次元座標をDLT法 (Digital Linear Transformation) などで取得する作業は別のソフトウェアを用いて下さい。

- 動画を連続する静止画像に変換する機能はありません。別のソフトウェア (AviUtil, Ffmpegなど) を使用して下さい。
- 特徴点を自動的に検出する機能はありません。あなたの目と指と根気が必要です。
- **Visualize**では人の全身のような複雑なStick pictureを描く機能もありますが、各計測点間の関係を記述する便利なユーザーインターフェースはありません。線分定義ファイルをテキストエディターなどで作成して読み込んで下さい。
- 点と点の間を結んで線分とし、その線分の長さや線分間の角度を計測する機能もありません^(注)。これは各人、Excelやプログラミングで実現して下さい。あくまで各点の座標を計測する機能のみです (ただし、**measure**では計測点の色情報を記録できます。用途は想像できませんが)。

(注) Ver.3.2から搭載したQuick measureモードでは、2点間の距離と角度を出力可能です。

【measure, merge, visualize共通の説明】

- **measure**で読み込める静止画像はJPEG, PNG, GIF, TGA形式, **visualize**で出力可能なプロット画像はJPEG, PNG, TIFF, TGA形式です. **measure, merge**で読み込み可能なテキストファイルはタブ区切りCSV形式です. Excel等を用いた手作業での修正も容易です.
- **merge**で読み込む, 連続する静止画像のファイル名は「hoge hoge0000.jpg」のように拡張子の前に半角数字の付いた形式が好ましいです (勿論, 自由なファイル名でも問題はありません). **measure, merge**で複数のファイルを一度に読み込む際, ファイル名でソーティングしてからファイルを読み込みます.
- プログラムの終了は[ESC]キーです.

(技術的な補足)

measure, merge, visualizeは, フォルダー選択およびD&Dにおいて, ファイルの大雑把な指定 (無関係なファイルも含む選択対象の中から, 対応するファイルのみを抜き出して読み込む) が可能です. ただそのファイルが対応するファイルかどうかを調べるために, ファイル全体を読み込む必要があります. もしその候補のファイルが動画ファイルだった場合は, 数十MBのファイルを一旦読み込んでしまいます. この無駄を省くために, ファイルの拡張子から明らかに無関係だと判明するファイルは読み込むことなく候補から外します.

common.pde中の以下の二つの文字列配列で指定しています.

```
final String [] extensionListOfImage = {"jpg", "jpeg", "gif", "tga", "png"};
```

```
final String [] extensionListOfText   = {"txt"};
```

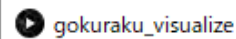
- **measure** : **extensionListOfImage**[]内の拡張子を持つファイルを静止画像として読み込む
- **merge** : **extensionListOfText**[]内の拡張子を持つファイルを座標データとして読み込む
- **visualize** : **extensionListOfText**[]内の拡張子を持つテキストを統合ファイルとして読み込む.

実際に読み込める正しいファイルかどうかは, 拡張子による判定後に改めて行われます (**validation**).

【measure, merge, visualize共通の説明】

- ソースリストは分割されています。common.pdeは3つのプログラム共通のコード（完全に同一内容）です。lib.pdeは各プログラム専用のコードです。lang.pdeは言語パック（後述）のデータ（コード）です。
- 各種設定の初期値はheader.pdeに記述してあります。変数名は大文字から始まります。
- measure, merge, visualizeには各種設定を変更可能なサブメニューが存在します。mergeは読み込んだ統合データのソーティングを行うサブメニューを持ちます。設定値は保存されませんので、起動するたびに設定する必要があります。
- いくつかのパラメーターはキーボード入力可能です。入力開始時には現在の値がプリセットされます。[BackSpace]キーにより1文字単位で消えますが、[SPACE]バーを押すと0に初期（文字列は全削除）化されますので、値を入力する前に[SPACE]バーを押すと変更が楽です。[Enter]キーを押すと確定します。[ESC]キーを押すと入力作業を中断できます。入力して変更した設定値はプログラムを終了すると消えますので、毎回、同じ値に設定したい場合はheader.pde内の初期値を書き換えるといいでしょう。

【注意】複数のmeasure, merge, visualizeを同時に起動していると、おかしい挙動をする場合があります。特に必要が無いならば、使用し終えたGokurakuツールは終了して下さい。



Gokuraku visualize (@tatsuva) Ver.3.0

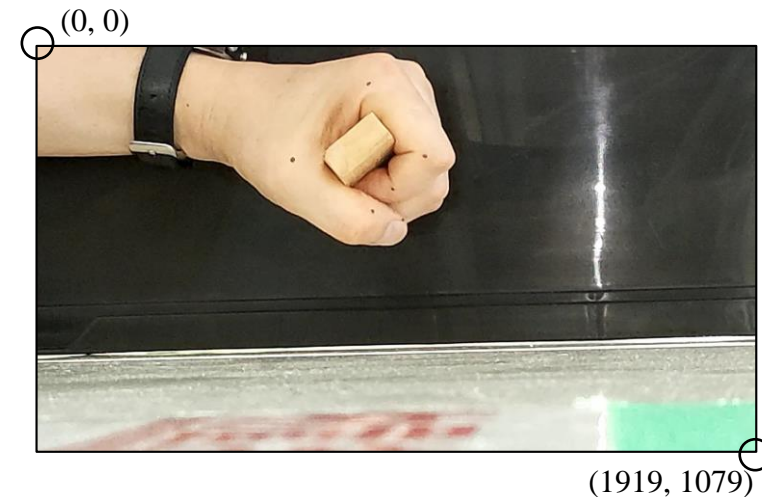
The data file should be sorted in advance using Excel or other tools.

Delay per image = 100

【measure, merge, visualize共通の説明】

(座標系)

- 座標系は静止画像の左上が(0, 0)です.
- 右の画像は1920×1080[pixel]の画像です (各ファイルのプロパティの“詳細”タブを確認下さい). この場合, 右下の座標は(1919, 1079)です.



(Y軸を反転したい) : (注) 手作業で変更してしまうとvisualizeできません

- 計測してmergeしたデータをExcel等で処理する際に, 元画像の縦のピクセル数-1 (例では1079) からY座標の値を引けば, 左下が(0, 0)の座標系 (単位はpixel) に変換できます.

(画像の傾きを補正したい) : (注) 変更するとvisualizeで静止画像と合成した際にズレます

- 回転行列は $\begin{bmatrix} P'_x \\ P'_y \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} P_x \\ P_y \end{bmatrix}$ です. 座標原点を中心として, 各座標を角度 θ 回転するには $P'_x = P_x \cos\theta - P_y \sin\theta$, $P'_y = P_x \sin\theta + P_y \cos\theta$ を計算して下さい (Excelの三角関数の引数は度ではなくラジアンですのでご注意ください).

(画像の比率を換算したい) : (注) 変更すると静止画像と合成した際にズレます.

- 画像中に既知の長さの物体 (方眼紙や定規) を入れて撮影し, その基準の長さを計測します. 例えば長さ10[mm]の物体が120pixelだったら, 10/120 [mm/pixel]です. この比率をx, y座標に掛ければ単位を[mm]に変換できます.
- カメラ撮影の特性上, 縦横比の微妙な違いや周辺部での歪みが発生する可能性があります.
- 奥行方向の歪みは遠方からズーム撮影することで, 影響を小さくすることができます.

【measure, merge, visualize共通の説明】

（校正係数を調べたい）

- 画面上の 1 pixelが何ミリ（あるいはメートルなど）に相当するか知りたい場合は、対象物を計測する際に同じカメラ位置で定規など長さの分かるものを同時に撮影して下さい。クイックメジャー機能でその長さをpixel単位で計測し、長さ/pixel数を計算した値が校正係数です。
- Gokurakuシリーズではデータ上では原則的に校正係数を適用しません（クイックメジャーモード時のみ活用）。これはオリジナルデータには極力、手を加えない設計思想に基づいています。したがってmeasureで計測する座標値も、静止画像を拡大縮小したとしても、その拡大縮小した画像上での座標値を記録します（倍率を掛けません）。倍率や校正係数を適用するのは、各人の後処理のソフトウェア（Excelなど）上で行って下さい。

（表示するウィンドウの初期位置）

- 初期値では画面左上を指定しています。
- デフォルトで表示する位置を各アプリケーションで変えたい場合、header.pde内の Init_PosX, Init_PosYの値を変更して下さい。

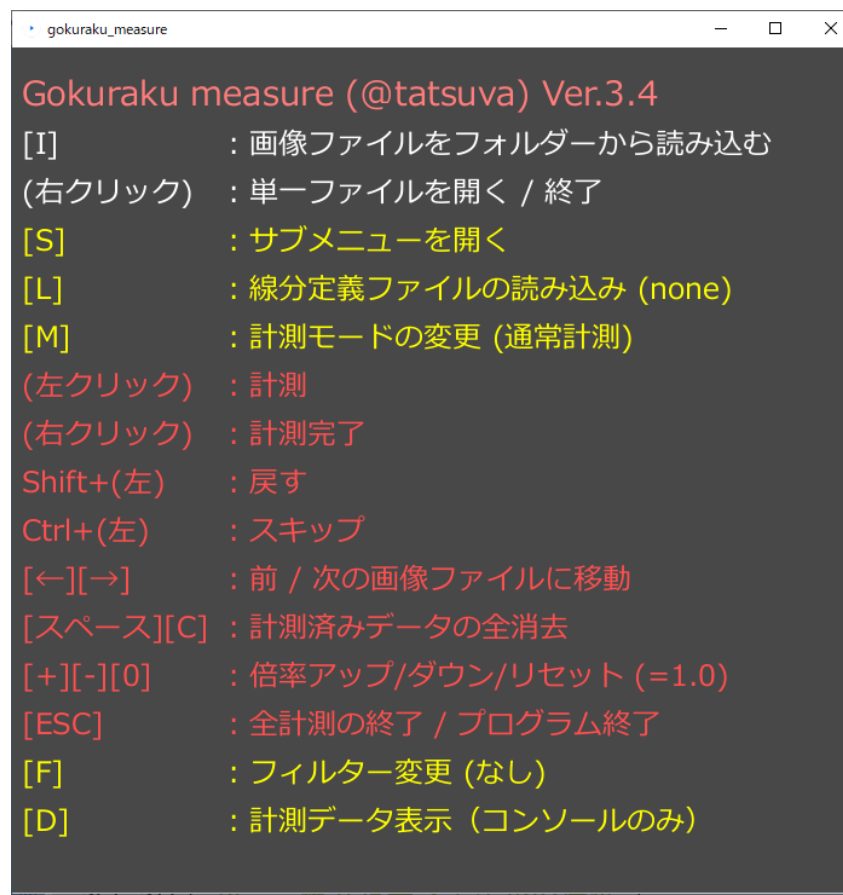
【Gokuraku measureの使い方】

大量の画像を計測する必要がある場合：一連の画像の中の何か所を計測するのかが事前に定まっている場合は、先に最大計測点数（**max measure number**）の設定と、線分定義ファイルの作成を行っておくと、作業が非常に効率的かつ安全に行えます。

最大計測点数を設定しておくと、指定した計測点を計測し終わると、自動的に計測データが出力され、次の画像が読み込まれます。この方法は非常に作業が素早く行える特徴がある反面、最後に計測した点が正しい位置をクリックできていたのかを目で確認する前に次の画像に作業が移ってしまう弱点があります。

線分定義ファイル（後述）を先に読み込んでおくと、計測済みの点と点を指定した色の線で結んでくれます。計測個所を1ヵ所飛ばしていないかを確認できますので、とても便利です。この線分定義ファイルは**visualize**と共通です。

measureは**visualize**のプレビュー同様に、カーソルキーで同時に読み込んだ静止画像を移動できます。計測済みデータは静止画像上にプロットされますので、計測終了後に再び静止画像群を読み込んで、正しく計測されたか目視確認すると良いでしょう。



【Gokuraku measureの使い方】

（静止画像の読み込み方）：お勧めは“フォルダーのD & D”

単一選択，フォルダー選択，D & Dの三種類の方法があります．

- **単一選択**：メニュー画面を右クリックするとファイル選択ダイアログボックスが表示されます．ファイルを1個だけ選択して計測する場合に適した方法です．フォルダーは選択できません．
- **フォルダー選択**：メニュー画面で「I」キーを押して表示されるフォルダー選択ダイアログボックスで選択したフォルダー内の画像ファイルが計測対象となります．ただし，選択したフォルダーの下階層のフォルダー内の画像は計測対象になりません．
- **D & D（ドラッグ&ドロップ）**：エクスプローラーなどで複数のファイルを選択してメニュー画面にドロップします．ドロップできるファイルの数には上限があります（未確認）．ドロップしたファイル中にフォルダーが含まれていても，そのフォルダー内の画像は計測対象になりません．ただし，特例として単一のフォルダーのみをD & Dした場合，そのフォルダーに含まれる画像ファイルはフォルダー選択時と同様に計測対象になります．

大量の画像を計測したい場合：画像（および線分定義ファイル）を一つのフォルダー内に保存し，そのフォルダーをD&Dするのが最も効率的な方法です．

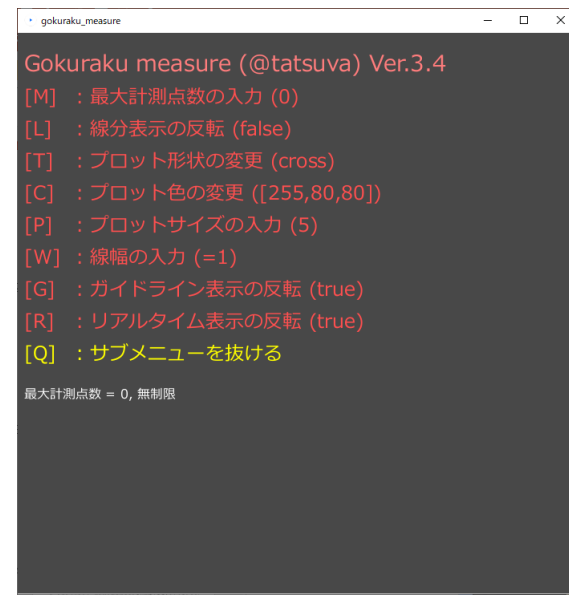
※ フォルダー選択，D&Dで複数ファイルを同時に選択した場合，ファイル名で静止画像ファイルをソートしてから順次読み込みます．選択されたファイルが画像データではない場合は無視されますので，あまり気にせず大雑把に放り込んで頂いてOKです．

【Gokuraku measureの使い方】

（各種設定の変更）

メインメニューで[S]キーを押すと設定変更のサブメニューに切り替わります。[Q]キーでメインメニューに戻ります。なお、計測中にもメインメニューやサブメニューの機能は利用できます（画像表示が優先されるので文字表示は見えませんが）。

- [M]キーを押すと最大計測点数（max measure number）を変更できます。初期値はheader.pdeのMax_measureです。ここで指定された数の計測が終了したら自動的に計測を終了して座標ファイルを出力します。数値に0を指定すればこの機能を無効化できます。計測対象の静止画像群の測定数が全ての画像で同じ場合は設定すると効率的に作業を進められます。
- [L]キーを押すと、プロット間を“線で結ぶ”／“結ばない”を切り替えられます。線分定義ファイルを用意している場合は、そちらを利用して下さい。

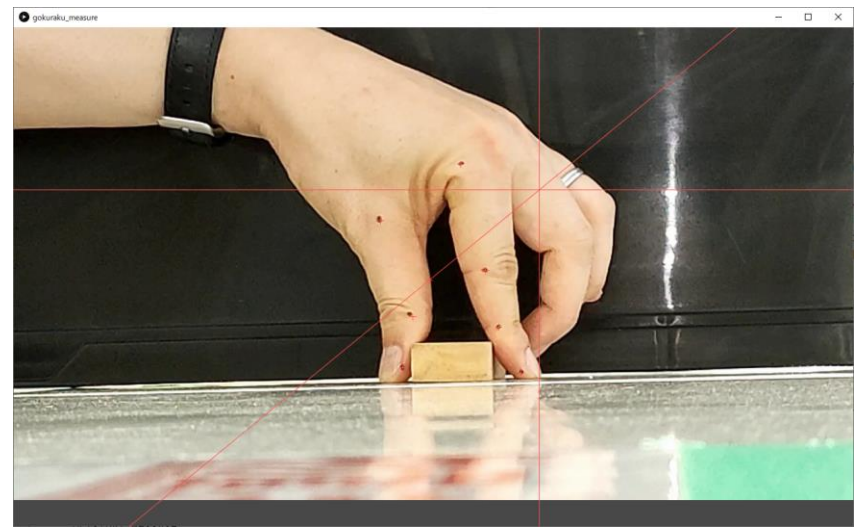
A screenshot of a terminal window titled "gokuraku_measure". The window displays the version "Gokuraku measure (@tatsuva) Ver.3.4" and a list of commands with their current values: [M] : 最大計測点数の入力 (0), [L] : 線分表示の反転 (false), [T] : プロット形状の変更 (cross), [C] : プロット色の変更 ([255,80,80]), [P] : プロットサイズの入力 (5), [W] : 線幅の入力 (=1), [G] : ガイドライン表示の反転 (true), [R] : リアルタイム表示の反転 (true), and [Q] : サブメニューを抜ける. At the bottom, it shows "最大計測点数 = 0, 無制限".

```
gokuraku_measure
Gokuraku measure (@tatsuva) Ver.3.4
[M] : 最大計測点数の入力 (0)
[L] : 線分表示の反転 (false)
[T] : プロット形状の変更 (cross)
[C] : プロット色の変更 ([255,80,80])
[P] : プロットサイズの入力 (5)
[W] : 線幅の入力 (=1)
[G] : ガイドライン表示の反転 (true)
[R] : リアルタイム表示の反転 (true)
[Q] : サブメニューを抜ける

最大計測点数 = 0, 無制限
```

【Gokuraku measureの使い方】

- [M]キーを押すと最大計測点数（max measure number）を変更できます．初期値はheader.pdeのMax_measureです．ここで指定された数の計測が終了したら自動的に計測を終了して座標ファイルを出力します．数値に0を指定すればこの機能を無効化できます．計測対象の静止画像群の測定数が全ての画像で同じ場合は，設定すると効率的に作業を進められます．
- [T], [C], [P], [W]キーを押すと，計測時のプロットの形状（noneはプロットなし），色，サイズ，線幅を変更できます．色はheader.pdeでプリセットした色の組み合わせから選択されます．これらの初期値を変更したい場合は修正して下さい．
- [G]キーを押すと，画面に表示するガイドラインを表示／非表示に切り替えられます．通常計測モードでは垂直線と水平線が表示されます．クイックメジャーモードでは，垂直線と水平線に加え，始点と現在のマウスポインターの位置の2点を通る直線が表示されます．（Ver.3.2以降）



【Gokuraku measureの使い方】

（計測方法）：メインメニュー

- [S]キーを押すと，設定画面に移動します.
- [L]キーを押すと，線分定義ファイルを読み込みます. 詳しくは【visualizeの使い方】を参照して下さい.
- [M]キーを押すたびに，計測モードが通常計測モード（Normal），クイックメジャーモード（Quick measure）の間で切り替わります.
- 読み込まれた計測対象の静止画像が既に計測済みだった場合，自動的に記録時のスケールで静止画像と座標データが読み込まれます. もしこの仕様が気に入らない場合は，header.pdeのForce_updateをtrueに変更して下さい（Ver. 2.1以前の仕様に戻る）.

【Gokuraku measureの使い方】

（計測方法）

- 画像中で左クリックした位置の x, y 座標と色（R, G, B）が計測されます．間違った場所をクリックした場合は，[Shift]キーを押しながら左クリックすると取り消せます．
- 読み込んだ画像が小さ過ぎる／大き過ぎる場合は，[+]／[-]キーを押して拡大／縮小が可能です．[0]キーを押せば拡大率（scale）は1に戻ります．計測結果は拡大縮小された画像上の座標値です．計測時の表示の拡大率は座標ファイルに記録されます．この拡大率はvizualizeでも利用されます．ディスプレイのサイズよりも大きい高解像度な画像を計測する際に役立ちます（ただし，縮小しない方が計測の精度は高い．拡大すればさらに精度は高くなります：人間の視覚とマウス操作の精度）．
- 記録したい特徴点が隠れているなどして計測できない場合は，[CTRL]キーを押しながら左クリックすると計測をスキップできます．その際の座標値は-1, -1, 色情報は0, 0, 0が座標ファイルに保存されます．
- [ESC]キーを押すか，右クリックするとそこで計測終了です．1ヵ所でも計測済みの場合，計測が終了すると座標ファイルが自動的に出力されます（既に存在する場合は強制的に上書きされます）．max_measureを設定している場合は指定した数の計測が終了したら，自動的に次の画像ファイルが読み込まれます．

【Gokuraku measureの使い方】

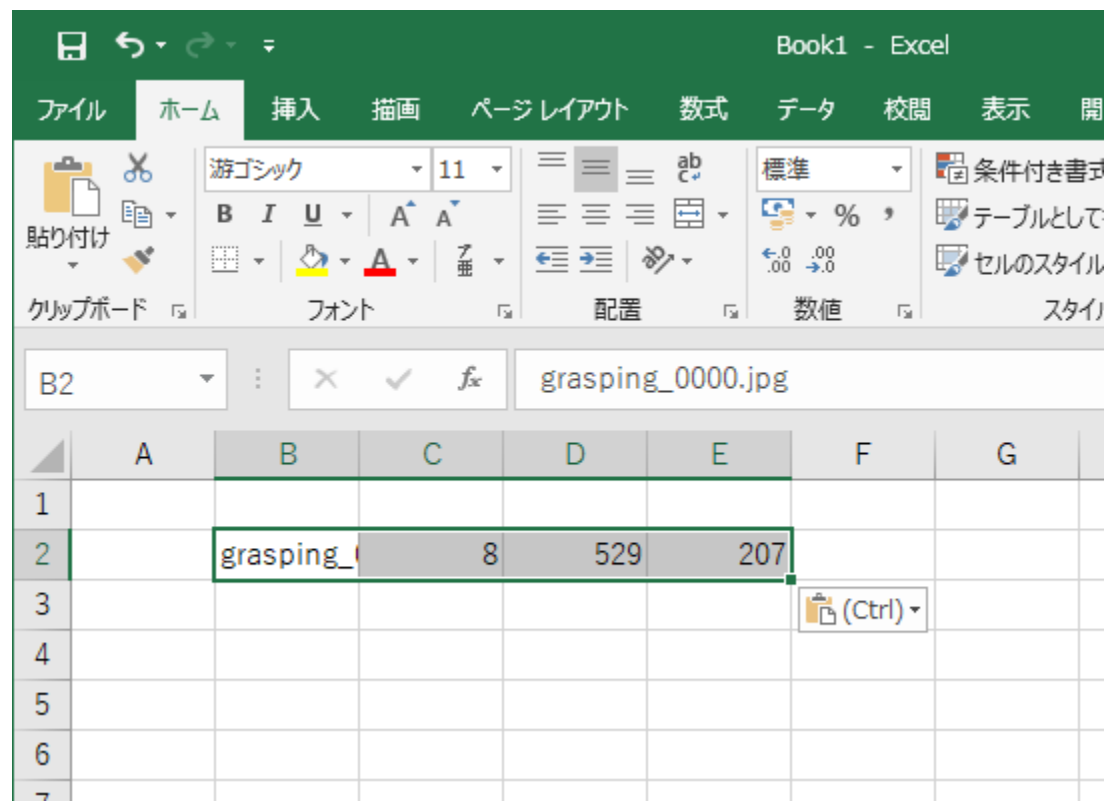
（計測方法）

- 計測中に[F]キーを押すと、読み込んだ静止画像の色をグレースケール化、色反転可能です（フィルター機能）。
- 計測中に[D]キーを押すと、Processingの統合開発環境では統合開発環境の画面下のコンソール領域に、現在の計測対象画像の計測済みのデータ（座標、色情報）が一覧表示されます。
- [SPACE]バーまたは[C]キーを押すと、記録された計測点が全てクリアされます。
- 計測した日時は座標ファイルにyyyymmddhhmmssの形式で保存されます。
- 複数の静止画像が同時に選択されている場合、静止画像が表示されている状態で[←]／[→]キーを押すと、一つ前（リトライ）／次の画像（スキップ）へ移動可能です。ただし、移動前に計測した計測点は破棄されます。
- リトライ／スキップで静止画像を移動した際に、既にその画像が計測済みであればプロットが表示されます。[Shirt]+左クリックで一つずつ取り消したり、[SPACE]バーか[C]キーを押して全ての座標データを削除して計測し直すことも可能です（header.pdeのForce_updateをtrueにすると本機能は無効化され、事前に計測した座標ファイルは読み込まれません）。

【Gokuraku measureの使い方】

(計測方法)

- 計測点を左クリックすると、クリップボードに画像ファイル名、計測点の番号、x座標値、y座標値がタブ区切りでコピーされます。Excelにペースト（たとえばCtrl+v）すると自動的にセルに分かれて貼り付けられます。座標ファイルに記録する以外に、このような方法でも計測値を外部アプリケーションに受け渡すことが可能です。



【Gokuraku measureの使い方】

（座標ファイルのフォーマット）

1 行目：拡大率，画像ファイルの幅，画像ファイルの高さ，画像ファイル名，計測日時，計測点数，スキップ数

2 行目以降：計測点，x座標，y座標，色情報（赤），色情報（緑），色情報（青）

- 計測点は1番目が0，2番目が1と，実際の計測順番よりも1小さな値です。
- 画像ファイルの高さと幅は拡大率と関係なく，オリジナルのサイズです（pixel）。
- x, y座標は画像左上隅を(0, 0)とし，右方向，下方向を正とする座標です（pixel）。座標値は画面に表示されていた静止画像の生のx,y座標ですので，一連の静止画像の中で拡大率を変更すると座標値の倍率が一部だけ変わってしまいますので注意して下さい。visualizeは拡大率を無視して拡大率1相当に補正する機能がありますので，可視化においては問題はありませんが，統合データをvisualize以外のソフトウェアで加工する場合は，拡大率でx, y座標値を割るなどして正規化して下さい。
- 計測をスキップした点の座標は(-1, -1)です。
- 色情報(RGB)は0～255の値で，値が大きいほどその色成分が明るいことを意味します。

（出力される座標ファイルについて）

- 出力される座標ファイルはテキストファイルですので，テキストエディターでの編集やExcelなどでの読み込みが可能です。データ間はタブ記号により区切られます。
- 出力される座標ファイルは画像データと同じフォルダーに保存されます。
- 座標ファイルのファイル名は画像ファイルのファイル名に".txt"を追加した名前になります（例：human01.jpg -> human01.jpg.txt）

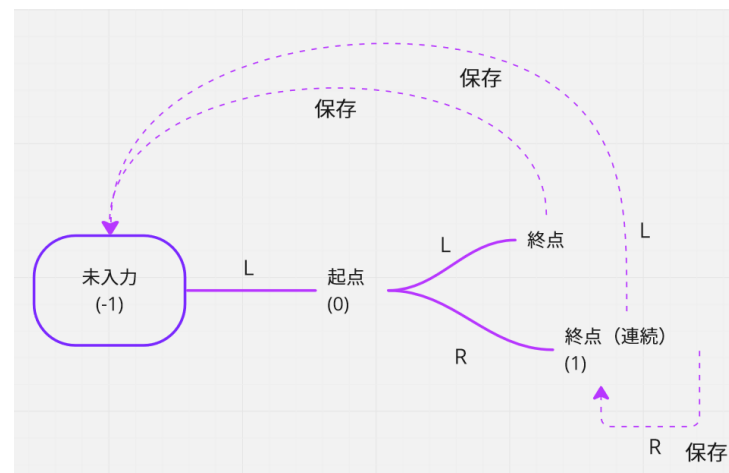
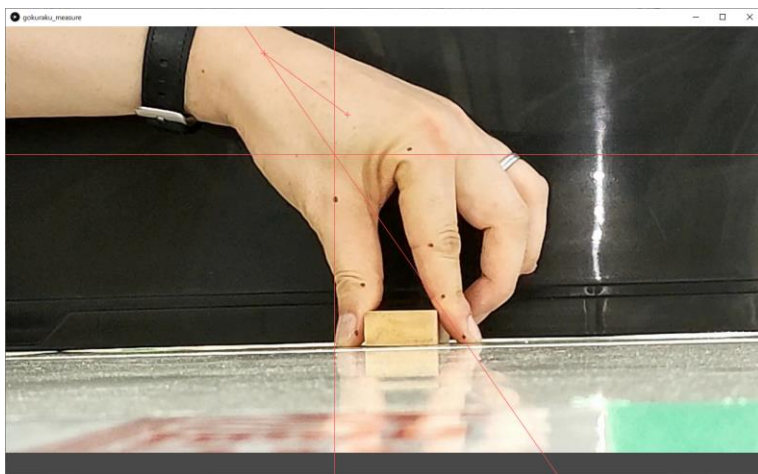
【Quick measureモード】：Ver.3.2以降

「とりあえず素早く測りたい」というニーズに答える新しい機能です。
画面上の2点（起点・終点）をクリックすると、

- 始点と終点のx座標値とy座標値を表示します。
- 2点間の距離（pixel）を表示します。
- 始点から終点に向かう線分のx軸に対する角度を表示します。
- ファイル出力は行いませんが、始点と終点を選択し終わると、画像ファイル名、始点と終点のx, y座標値、距離、角度がクリップボードにコピーされます。Excel等の外部アプリケーションに計測結果を容易に貼り付け可能です。
- ESCキーで終了しない限り、同じ画像の中で何度でも計測できます。
- 終点で右クリックすると、始点を維持したまま終点を選択し続けることができる。
- 通常計測モード同様に、複数の画像を選択して、次々とクイックメジャー可能です。

【Quick measureモード】

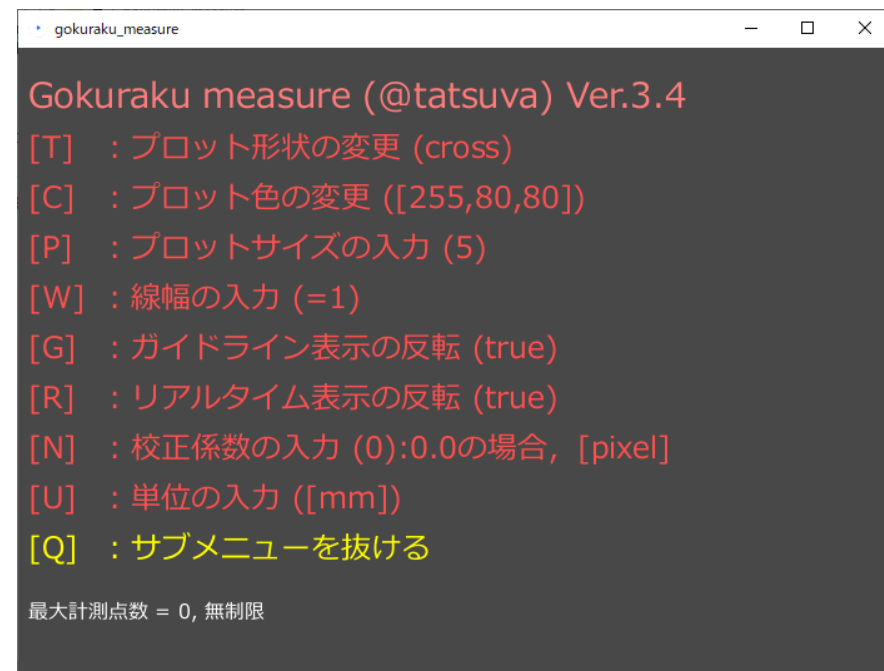
- 始点をクリック，終点をクリックを繰り返します.
- 始点は左クリックで指定します.
- 終点を左クリックすると計測終了です.
- 終点を右クリックしても計測は終了しますが，始点は計測されたままです．そのまま別の場所を右クリックすると，始点から新しい終点のデータが計測されます．この機能を使用すると，始点を固定したまま複数のポイントを計測可能です.
- 計測点のプロットの形状，大きさ，色，計測結果の線分とガイドラインの幅と色は設定画面で変更可能です（通常計測と共通）．



【Quick measureモード】：校正値の入力（Ver.3.3以降）

- クイックメジャーモードで校正値および単位を入力可能になりました。
- 座標データは基本的にピクセル単位のままです。校正は各人，Excelなどで行って下さい。あくまでクイックメジャーモードでの表示用です。
- 測定モードをクイックメジャーモードに切り替えた状態で，[S]で設定画面に移動しますと，[N]と[U]で校正値と単位の文字列が入力可能になります。
- 校正値は単位ピクセル当たりの長さです。もし100ピクセルが5cmならば，[N]で0.05，[U]でcmを入力して下さい。校正値が0.0の場合は単位の文字列が入力されていても無視されます（[pixel]で表示）。

Ver.3.3以降でリアルタイム表示に対応しました。通常測定モードではマウスポインターの位置，クイックメジャーモードではマウスポインターの位置及び，起点決定後はマウスポインターの座標および線分の長さと角度がリアルタイム表示されます。この機能は設定の[R]キー，およびheader.pde内のMousepoint_realtimeをfalseに変更することで無効化できます。



```
gokuraku_measure

Gokuraku measure (@tatsuva) Ver.3.4
[T] : プロット形状の変更 (cross)
[C] : プロット色の変更 ([255,80,80])
[P] : プロットサイズの入力 (5)
[W] : 線幅の入力 (=1)
[G] : ガイドライン表示の反転 (true)
[R] : リアルタイム表示の反転 (true)
[N] : 校正係数の入力 (0):0.0の場合, [pixel]
[U] : 単位の入力 ([mm])
[Q] : サブメニューを抜ける

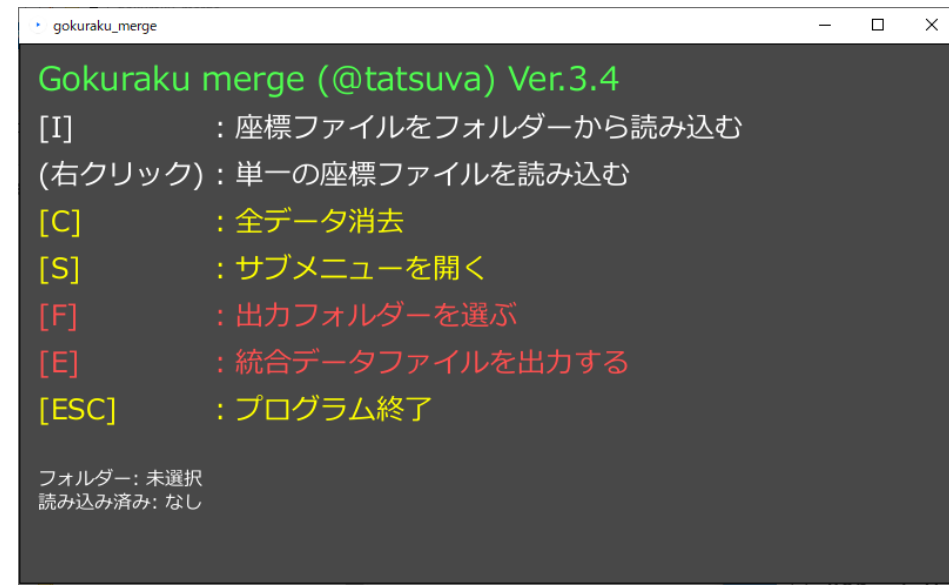
最大計測点数 = 0, 無制限
```

【Gokuraku mergeの使い方】

measure, visualizeに比べて、単純で機能の少ないプログラムです。

座標ファイルのデータを読み込んでvisualize用のデータ形式に変換すると同時に、複数の座標ファイルを一つの座標ファイルにまとめる作業のみを行います。

- メモリーに、読み込んだ座標ファイルのデータを次々と格納していきます。基本的に、後から読み込まれた座標ファイルのデータが後ろに追加されていきます。
- 通常は1つのフォルダー内にある座標ファイル群を一括で読み込ませれば、それで完了です。それ以外に、複数のフォルダーにある座標ファイル群を一つの統合ファイルにまとめることも可能です。ただし静止画像と合成したい場合は一つのフォルダーに座標ファイルと静止画ファイルをまとめておく必要があります（メモリー内に保存されるのは各座標ファイルのファイル名のみで、パスは記録されません。最後に読みだした座標ファイルのパスに基づいて静止画像を探す仕様です）。
- 複数の座標ファイルを読み込む際はファイル名でソート、読み込み終えたメモリー上のデータをサブメニュー内のコマンドでファイル名あるいは計測日時でソート可能です。ソート順は昇順のみです（要望があれば降順も実装しますが...）。



【Gokuraku mergeの使い方】

（座標ファイルの読み込み方）：お勧めは“フォルダーのD & D”

単一選択，フォルダー選択，D & Dの三種類の方法があります．

- **単一選択**：メニュー画面で右クリックするとファイル選択ダイアログボックスが表示されます．ファイルを1個ずつ選択する場合です．
- **フォルダー選択**：メニュー画面で「I」キーを押して表示されるフォルダー選択ダイアログボックスで選択したフォルダー内の座標ファイルが統合対象となります．ただし，選択したフォルダーの下階層のフォルダー内の計測データは統合対象になりません．
- **D & D（ドラッグ&ドロップ）**：エクスプローラーなどで複数のファイルを選択してメニュー画面にドロップします．ドロップできるファイルの数には上限があります（未確認）．ドロップしたファイル内にフォルダーが含まれていても，そのフォルダー内の座標ファイルは統合対象になりません．

座標ファイルを読み込むたびにデータが追加されていきます．[C]キーを押すと今までに読み込んだデータを全て破棄します．しくじったな，と思ったら[C]でクリアーして，やり直しましょう．

大量の画像を計測した場合：計測ファイルを一つのフォルダー内に保存し，そのフォルダーをD&Dするのが最も効率的な方法です．

※ フォルダー選択，D&Dで複数ファイルを同時に選択した場合，ファイル名で座標ファイルをソートしてから順次読み込みます．選択されたファイルが座標データではない場合は無視されますので，あまり気にせずに大雑把に放り込んで頂いてOKです．

【Gokuraku mergeの使い方】

（ソーティングサブメニュー）

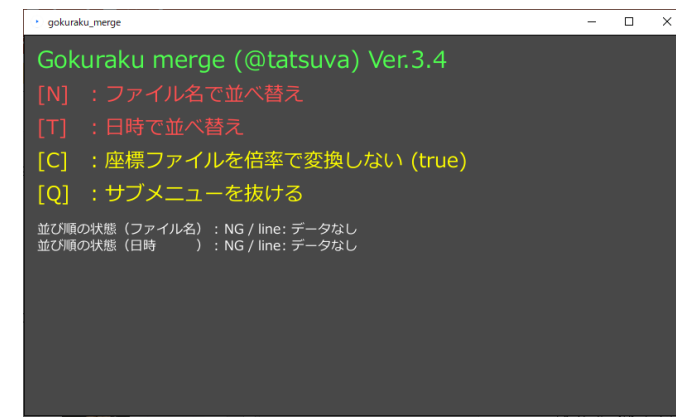
メインメニューで[S]キーを押すと、ソーティングサブメニューに切り替わります。

座標データは前ページの3種類の方法で読み込んだ順でメモリー内に蓄えられます。このメモリー内の座標データの並び順で統合ファイルは出力されます。ソーティングサブメニューでは、各座標データが静止画像ファイル名または計測日時の昇順で読み込まれているかをチェックする機能、静止画像ファイル名または計測日時で昇順にソーティングする機能を持ちます。

Order condition (by Filename), Order condition (by Time)がOKの場合は静止画像ファイル名、計測日時で昇順、NGの場合が昇順ではないことを表します。NGの場合のlineは問題の最初に発見された行番号（統合ファイル）です。OKの場合は読み込み済みの座標ファイル数です。

[F]キーを押すと静止画像のファイル名、[T]キーを押すと計測日時でメモリーに読み込まれた座標データをソーティングします。

※ 測定にミスがあることに後から気付いて特定のファイルを再測定した場合、他の座標ファイルに比べてこの座標ファイルの計測日時だけが大きく遅れているでしょう。ファイル名、日時でソーティングを繰り返しても、二つのOrder conditionが共にOKにはなりません（それで正しい）。どちらを優先するか、です（大半はファイル名）。



【Gokuraku mergeの使い方】

（do not convert input data by scaleの補足）：互換性のための機能

Ver.3より前のバージョンでは、座標ファイルを読み込んでマージする際に、各測定データの測定時の画像の倍率（scale）に基づいて座標値を変換していました。もし50%に画像を縮小して計測を行ったのであれば、x座標、y座標をそれぞれ2倍に拡大してmerged.txtに出力していました。この仕様ですと、mergeが「ただ単にマージするだけ」というコンセプトから外れていることに加え、元の座標ファイルの座標値をmerged.txtの座標値が異なるため、データが正しく読み込まれたかどうかの判断が難しくなる問題があります。Vizualizeすることなく、merged.txtのデータをExcel等で処理する方にとっては「余計なお節介」です。そこでVer.3以降は、mergeでは倍率に基づく座標値の変換を行わない仕様に変更しました。

Ver.3より前のバージョンのように、mergeで統合する際に、倍率1に変換した座標値を得られた方がExcel等での後処理が楽だというニーズもあるかも知れませんが、設定により変換機能をOn/Offできるようにしました。それがサブメニューの[C]です。デフォルトではtrueです。

※ 座標値の変換はファイル読み込み時に行われます。したがって do not convert input data by scale の設定の変更はファイルを読み込む前に行っておく必要があります。

【Gokuraku mergeの使い方】

（統合ファイルの出力）

- [F]キーを押すと統合ファイルを出力するフォルダーを選択可能です。なお、座標ファイルを読み込んだ際に、最後に読み込まれた座標ファイルのフォルダー名が自動的に設定されます。もし、どうしても別のフォルダーに出力したい場合のみ、エクスポート直前に変更して下さい。
- [E]キーを押すとメモリー中の統合された座標データを統合ファイルに出力します。ファイル名はheader.pde内で指定された“merged.txt”で固定です。変更したい場合はheader.pdeを修正して下さい。

※ 座標ファイル中の色情報（RGB）はデフォルトでは統合（出力）しません。もし色情報も統合したい場合はheader.pde内のMearge_color_info = falseをtrueに変更して下さい（使用頻度が低いのでメニューによる変更には対応しません）

【Gokuraku mergeの使い方】

（統合ファイルのフォーマット）

全行以下のフォーマットです.

画像ファイル名, 計測日時, 拡大率, 画像ファイルの幅, 画像ファイルの高さ, 計測点数, スキップ数⁽¹⁾,

1 個目の計測点のx座標, 1 個目の計測点のy座標, 2 個目の計測点の...

の繰り返し.

（座標データ読み込み時の注意点）

- mergeに読み込まれた順番で統合されます. フォルダ一選択やD & Dの場合, 期待した順番で読み込まれない場合があります. 期待した順番で読み込まれたかどうかはエクスポートした統合ファイルの1列目の画像ファイル名か, 2列目の計測日時で判断できます. 統合ファイルをExcelなどに読み込み（タブ区切りCSV）, 1列目または2列目で“並べ替え”すると良いでしょう.
- （上記補足）Ver.2.2からは, 選択された複数の座標ファイルを読み込む際に, ファイル名でソートिंगするようにしていますので, ほぼ（名前順による）期待通りの順番で読み込まれるでしょう. 不安でしたらソートングサブメニューで確認, 出力した統合ファイルで確認して下さい.

⁽¹⁾ スキップ数はVer.3以降で追加されたデータです. したがってVer.3より前のバージョンのmergeで作成したmerged.txtをVer.3以降のvisualizeに読み込ませると正常に動作しません. 座標ファイルが残っているならばmergeし直せば問題ありませんが, もし座標ファイルが存在せず, merged.txtしか手元にない場合は, 手作業（Excelなどを用いて）で値0の列を追加すればvisualizeでも読み込めるようになります.

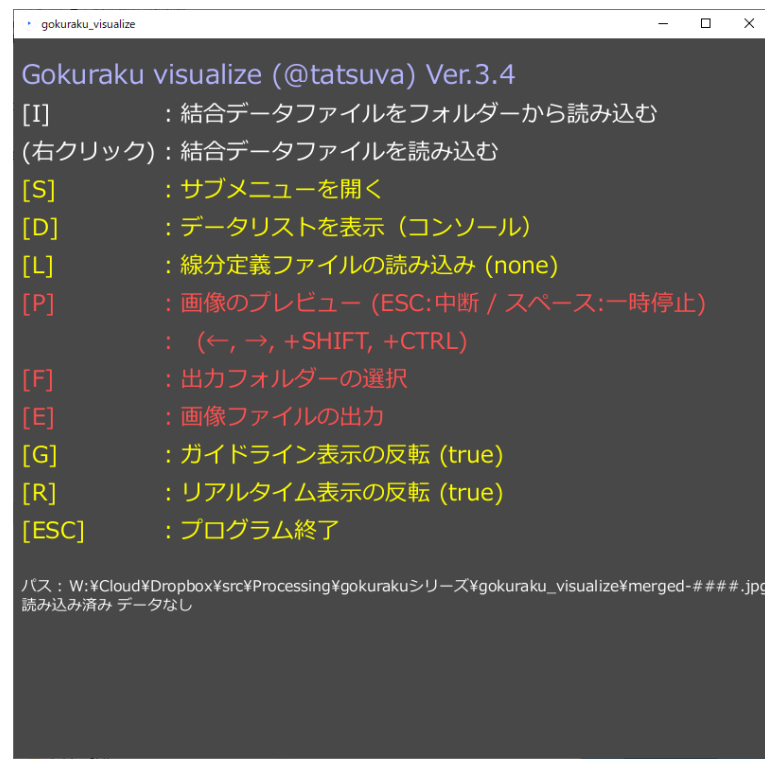
【Gokuraku visualizeの使い方】

統合ファイルを読み込んで、画面上で確認するためのビューワーです。

1. 画面上に各計測点を表示する（プロット画像）
2. 各計測点間を線で結ぶ（Stick picture）
3. 計測した静止画像にスーパーインポーズできる
4. 再生，コマ送り／巻き戻し，先頭／末尾に移動，などの操作が可能
5. 各プロット画像を画像ファイルとして出力可能（エクスポート）

本番はmergeで出力したmerged.txtの解析ですが，エクスポートした一連の静止画像から動画を作成してプレゼンテーションに用いても良いでしょう．

visualizeで一コマコマを確認し，正しくない測定点を発見したらmeasureで再計測し，mergeで統合し直してvisualizeで再び確認する．この手順を繰り返して測定したデータの精度を高めて下さい．



【Gokuraku visualizeの使い方】

（統合ファイルの読み込み方）

読み込むファイルは一つのみですが、フォルダー指定やD&Dで大雑把に指示することも可能としました。（大量の画像ファイルなどの中からmerged.txtを探すのは困難なため）。

- **単一選択**：メニュー画面で右クリックするとファイル選択ダイアログボックスが表示されます。統合ファイルを選んで下さい。
- **フォルダー選択**：メニュー画面で「I」キーを押して表示されるフォルダー選択ダイアログボックスで選択したフォルダー内の統合ファイルが読み込まれます。ただし、選択したフォルダーの下の階層のフォルダー内の計測データは統合対象になりません。
- **D & D（ドラッグ&ドロップ）**：エクスプローラーなどで統合ファイルを選択してメニュー画面にドロップします。フォルダーをドロップしても構いません。

※ 選択されたファイルが統合データではない場合は無視されますので、あまり気にせずに大雑把にフォルダー選択あるいはD&Dで放り込んで頂いてOKです。ファイル名もチェックしていませんので、merged.txt以外のファイル名でも問題ありませんが、同時に選択された候補の中に複数の統合データファイルが存在した場合、最初にフォーマットが一致したファイルのみを読み込みますので、その点だけは注意して下さい。

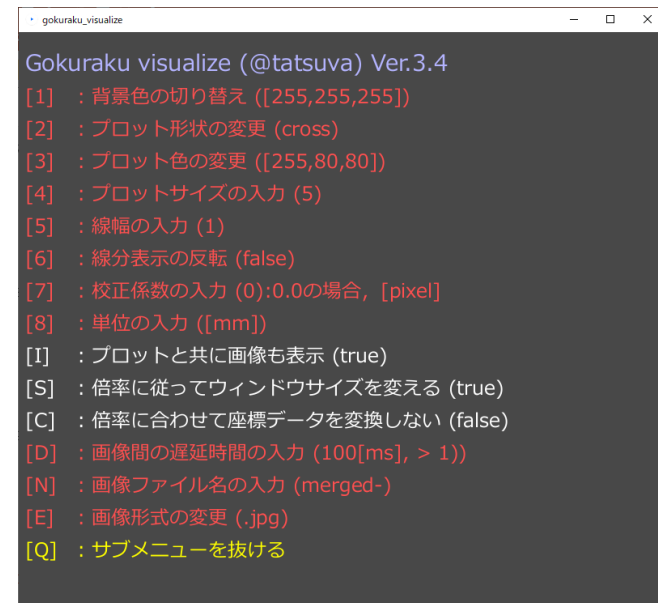
※ mergeと異なり、visualizeは統合データファイルを読み込む度にメモリークリアを行いますので、明示的にメモリークリアを行うコマンドはありません。もし複数の統合ファイルを同時に読み込みたい場合は、何らかの手段（テキストエディターなど）で統合ファイルを合体したファイルを新たに作成して読み込ませて下さい。

【Gokuraku visualizeの使い方】

※設定項目が多くなり過ぎてキー割り当てが困難になったため、Ver.3以前とキー割り当てが一部変更になっています。

（各種設定の変更）

- メインメニューで[S]キーを押すと設定変更のサブメニューに切り替わります。[Q]キーでメインメニューに戻ります。なお、計測中にもメインメニューやサブメニューの機能は利用できます。
- [1]キーを押すと、プロット画像の背景の色を切り替えられます。プリセットカラーは白、黒、灰色です。header.pdeを編集して変更できます。
 - [2], [3], [4], [5]キーを押すと、プロット時の計測点の形状（noneはプロットなし）、色、サイズ、線幅を変更できます。色はheader.pdeでプリセットした色の組み合わせから選択されます。これらの初期値を変更したい場合は修正して下さい。
 - [L]キーを押すと、計測点間を結ぶ線分（line segment, Stick picture）を”表示する／しない”を切り替えます。デフォルトでは全計測点間を線で結びますが、後述するようにheader.pde内でDefine_stick[][]を修正するか、線分定義ファイルを読み込むと、線分の組み合わせ、色、幅を細かく指定可能です。
 - [I]キーを押すと、静止画像を“読み込まない”／“読み込む”を切り替えます。
 - [S]キーを押すと、ウィンドウサイズを“計測時の拡大率で表示”／“元サイズで表示”するかを切り替えます。
 - [C]キーの倍率変換についてはmergeの説明を参照ください。互換性の為に残した設定ですので、基本的にデフォルト値（false）から変更の必要はありません。



```
gokuraku_visualize
Gokuraku visualize (@tatsuva) Ver.3.4
[1] : 背景色の切り替え ([255,255,255])
[2] : プロット形状の変更 (cross)
[3] : プロット色の変更 ([255,80,80])
[4] : プロットサイズの入力 (5)
[5] : 線幅の入力 (1)
[6] : 線分表示の反転 (false)
[7] : 校正係数の入力 (0):0.0の場合, [pixel]
[8] : 単位の入力 ([mm])
[I] : プロットと共に画像も表示 (true)
[S] : 倍率に従ってウィンドウサイズを変える (true)
[C] : 倍率に合わせて座標データを変換しない (false)
[D] : 画像間の遅延時間の入力 (100[ms], > 1))
[N] : 画像ファイル名の入力 (merged-)
[E] : 画像形式の変更 (.jpg)
[Q] : サブメニューを抜ける
```

【Gokuraku visualizeの使い方】

(各種設定の変更)

- [D]キーを押すと、プレビュー時の画像間の遅延時間を変更できます。単位は[ms]です。
- [N]キーを押すと、出力する静止画像のファイル名を変更できます。使用できる文字はA-Z, a-z, 0-9, -()のみです。デフォルトは"merged-"で、header.pde内のImage_filenameで設定されています。
- [E]キーを押すと、出力する静止画像のファイル形式を変更できます。デフォルトはJPEGで、PNG, TIF, TGAに変更可能です。

- ◆ TIFF (Tag Image file Format)は可逆圧縮形式ですので画質の劣化がありません。その反面、ファイルサイズは小さくなりません。古典的な画像ファイル形式のため、多くのソフトウェアで読み書きが可能です。
- ◆ TGA (Truevision Graphics Adapter, TARGA)も古典的な画像ファイル形式ですが、TIFFに比べると対応しているソフトウェアは少なく、あえて選択するメリットはありません。
- ◆ JPEG (Joint Photographic Experts Group)は広く用いられる非可逆圧縮の画像ファイル形式です。圧縮率を高くすると画質が劣化してしまいます。visualizeではsaveFrame()コマンドでスクリーンを画像ファイル化していますが、圧縮率を指定できませんので、実質、どの程度の劣化なのかは不明です。もし気になるならばTIFF形式かPNG形式を選ぶと良いでしょう。
- ◆ PNG (Portable Network Graphics)はインターネットの普及に伴い広まった比較的新しい画像ファイル形式です。GIF形式の後継として開発され、可逆圧縮である点、透過色に対応するなど、JPEGよりも優れた点が多く存在します。

JPEG or PNGは圧縮率の違い、TIFF or PNGは対応ソフトウェアの違いで選ぶと良いでしょう。

【Gokuraku visualizeの使い方】

（連続する静止画のプレビュー）

読み込んだ統合フォルダーと同じフォルダー内に座標を計測した際に用いた静止画像が存在する場合、画像ファイルを読み込んで、その上に計測点をプロットすることもできます（設定サブメニュー内の[I]コマンドで切り替え）。

- [P]キーを押すと先頭から順にプロット画像を生成して表示します。
- 連続再生中に[SHIFT]キーを押している間、再生速度は1/5に減速します。
- 連続再生中に[SPACE]バーを押すと一時停止、もう一度押すと連続再生に戻ります。
- 連続再生で最後のコマに到達したら一時停止に切り替わります。
- プレビュー中に[ESC]キーを押すとプレビューを中断します。

（一時停止中のキー操作）

- カーソルキーの[→]キーを押せば一コマ進み，[←]キーを押せば一コマ戻ります。
- [→][←]キーと同時に[SHIFT]キーを押すと，最終のコマ，最初のコマにジャンプします。
- [→][←]キーと同時に[CTRL]キーを押すと，+ 1 0 コマ，- 5 コマ先に移動します。

※ プレビュー時の画像間の遅延時間は設定サブメニュー内で変更可能ですが，初期値はheader.pdeの中のDelay_preview_per_imageで設定されています。速すぎたり遅すぎると感じる場合はこの数値を変更して下さい。

※ [CTRL]同時押し時のスキップ数はheader.pdeのSkip_frame_plus, Skip_frame_minusで変更可能

【Gokuraku visualizeの使い方】

（クイックメジャー機能の搭載）：Ver.3.4以降

プレビューを一時停止している間は、その画像に対してクイックメジャー機能が利用できます。measureのクイックメジャー機能と同一の機能を持っており、2点間の直線の長さ、角度、始点と終点のx, y座標が表示されます。マウスクリック時の計測結果はmeasure同様にクリップボードにタブ区切りでコピーされていますので、Excel等に容易に貼り付け可能です。マウスポインターのリアルタイム表示機能もmeasure同様に搭載しています。

※ガイドライン表示やリアルタイム表示機能はサブメニュー内の設定で無効に変更できます。

※ Ver.3.4.1から、画面下にリアルタイム表示される座標などがバタつく問題を解決しました。

（桁数を固定。距離はpixel時は小数点以下1桁、角度は小数点以下2桁。クリップボードには未加工の高い精度の計算結果がコピーされます）

【Gokuraku visualizeの使い方】

（連続する静止画のエクスポート）

デフォルトでは，連続する静止画像（プロット画像）ファイルは読み込んだ統合フォルダーと同じフォルダー内に保存されます．

- [F]キーを押すと，自動設定されたエクスポートフォルダーを変更できます．なお，統合ファイルを読み込むとそのフォルダーに変更されますので，必要な場合は都度変更して下さい．

- [E]キーを押すと，連続する静止画像をエクスポートフォルダーに出力します．

※この際の画面表示の速度はheader.pde内のDelay_export_per_image, Delay_export_afterで変更できます．Processingの仕様上，この数値を小さくし過ぎると正常な画像出力が行われない恐れがありますので注意して下さい．

※ 出力される連続静止画像ファイルのファイル名はmerged-????.jpgです．????の部分は0000, 0001, 0002,...の順です．9999以上の場合は，統合ファイルの読み込み時に自動的に5桁以上に設定されますのでご安心ください．"merged-"の部分はheader.pde内のImage_filenameを書き換えることで変更可能です．設定サブメニュー内でも一時的に変更可能です．

※ 技術的にはプレビューでスクリーンに描画した画像をsaveFrame()コマンドで画像に保存しているだけです．デフォルトではjpeg形式で保存しますが，tiff, tga, png形式でも保存可能です．保存形式は随時，設定サブメニューで変更可能です．header.pde内のImage_type_extの拡張子の順番を変更すれば，選択される優先順位を変更可能です．

【Gokuraku visualizeの使い方】

（線分描画／Stick pictureの指定方法①：header.pde内の初期値Define_stick）

この機能は②の線分定義ファイルで代替可能ですので、特別な理由がない場合は線分定義ファイルを活用して下さい。

形式： intの二次元配列です。

[Start, End, R, G, B, Width]

Start, End：線分を結ぶ計測点の番号です。0から始まります（表示上は1から）。

R, G, B：色指定（0～255）

width：線の太さ（1以上， pixel）

例）

Define_stick = {};（初期値）では全てのプロット間を計測順に線で結びます。

Define_stick = {{0, 1, 255, 0, 0, 1}, {1, 2, 0, 255, 0, 2}, {2, 3, 0, 0, 255, 3}}; とすると、

計測点 0と1を赤、太さ 1 で結ぶ。

計測点 1と2を緑、太さ 2 で結ぶ、

計測点 2と3を青、太さ 3 で結ぶ。

なお、計測時にスキップした計測点（[-1, -1]）は無視されます。

（この方法を使う特別な理由）もう、とにかく決まったデータを来る日も来る日も計測してプレビューする日々、といった感じに、本アプリケーションを特定の業務専用にする場合、いちいち線分定義ファイルを読み込む操作は面倒ですね。header.pdeを書き換えるのは、そんな特別な場合のみです。

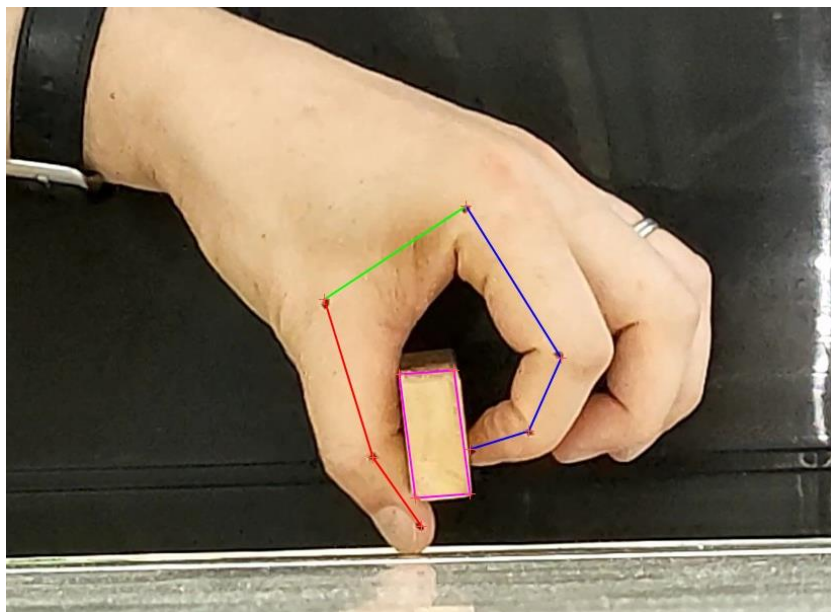
【Gokuraku visualizeの使い方】

（線分描画／Stick pictureの指定方法②：線分定義ファイルのインポート）

- メインメニューで[L]キーを押すと、線分定義ファイルのインポートが可能です。
- 読み込む線分定義ファイルのファイル名は自由です。ファイルはタブ区切りCSVファイルで、

Start End R(0-255) G(0-255) B(0-255) Width
の形式です.

※ 以下は、親指先端が計測点0, 時計回りに1, 2までが赤, 2と3の間が緑, 3から人差し指先端の6までが青, 四角柱断面の7～10までの4か所が紫, 線幅2で指定した例です。



File(F) Edit(E) View(V) Search(S) Window(W) Macro(M) Other(O)

define_stick.txt

0	1	255	0	0	2↓
1	2	255	0	0	2↓
2	3	0	255	0	2↓
3	4	0	0	255	2↓
4	5	0	0	255	2↓
5	6	0	0	255	2↓
7	8	255	0	255	2↓
8	9	255	0	255	2↓
9	10	255	0	255	2↓
10	7	255	0	255	2↓

[EOF]

【Gokurakuシリーズの実行環境構築】

（動作確認済み環境）

Windows10 + Processing 3.5.4 (64bit) です（Processing 4.2でも動作可能ですが不安定）。
GokurakuシリーズはProcessing上で動作するように作成されています。パス区切りに関しては、MacOS/Linuxにも配慮して、環境ごとに'¥'と'/'を使い分けるように作成しています。ただし、MacOS版/Linux版のProcessingでは動作確認を行っていません。

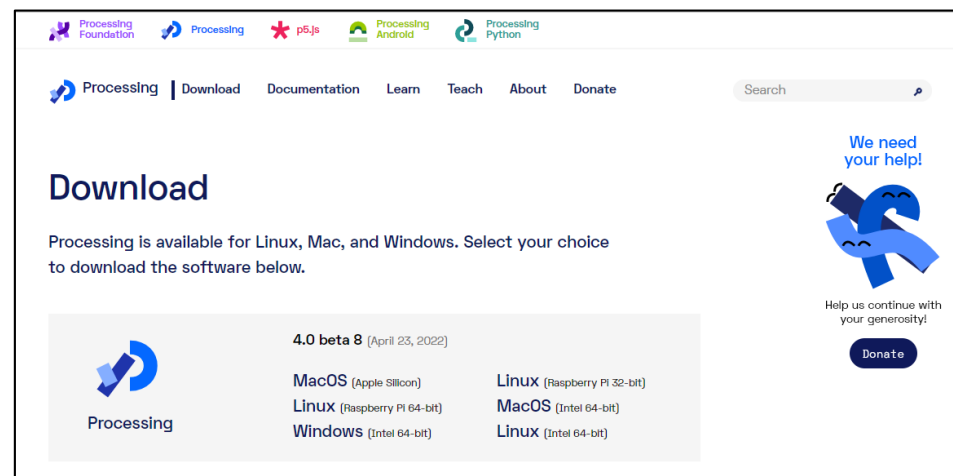
（動作環境の構築）

Processingの統合開発環境（IDE）は、以下のProcessing公式サイトダウンロードページからダウンロードできます。 <https://processing.org/download/>

特にインストール作業は必要ありません。たとえばダウンロードしたZIPファイルをUSBメモリー上に解凍して、その上でProcessingの統合開発環境を実行しても動作します。

※Ver.2.2から日本語フォントに対応しました（初期値はMeiryo）。環境によっては選択したフォントを利用できないかも知れません。その場合はheader.pde内のJapaneseFontを変更して下さい。

※Processingで読み込めても実行できない場合は、Java Runtime Environmentのバージョンが古い可能性があります。JREを更新して下さい。

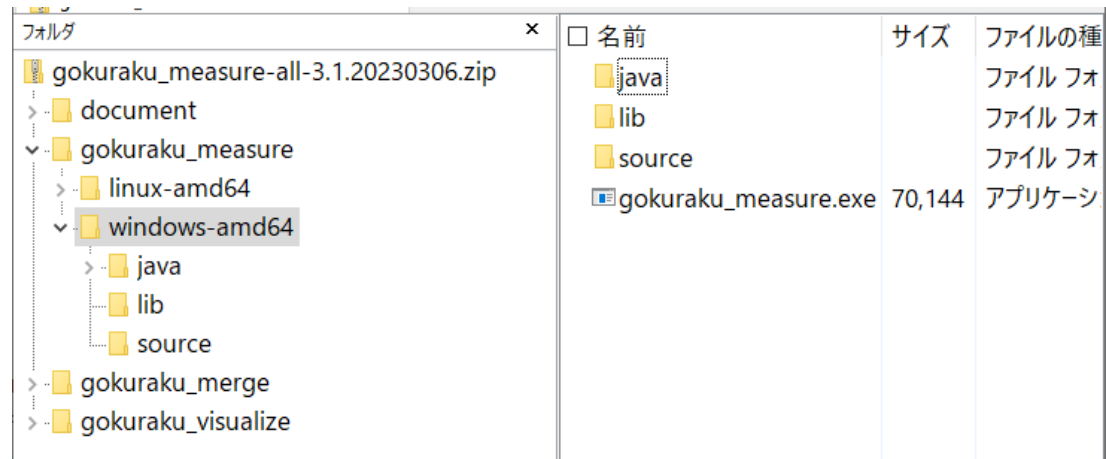


【Gokurakuシリーズの実行環境構築】

（Windows版）

Windows（64bit）用の実行形式も公開しています。Windows版を使用する場合、Processingの動作環境を用意する必要はありません、

gokuraku_measure, gokuraku_merge, gokuraku_visualizeのフォルダー内のwindows-amd64フォルダー内の各*.exeが実行形式です。このファイルをダブルクリックするとWindowsアプリケーションとして起動します。ただし、各フォルダー内のjava, lib, sourceのフォルダー内のファイルも必要ですので、*.exeファイルだけをコピーしても起動しません。全ファイル入りのアーカイブ（allをファイル名に含む）をダウンロード、展開後、各*.exeファイルのショートカットを作成して、そのショートカットをダブルクリックして起動すると良いでしょう。



【Gokurakuシリーズの実行環境構築】

(最新版のGokurakuシリーズのダウンロードと実行)

GokurakuシリーズのソースリストとWindows版実行形式を同梱したZIPファイルは以下のOSDNのサイトからダウンロードできます。

<https://osdn.net/projects/gokurakummeasure/releases/>

gokuraku_measure-all-?.?.*.zip に、measure, merge, visualize全てが含まれます。これらを好きなフォルダーに展開し、Processingから開けば実行可能です。

各フォルダー内にapplication.windows64というフォルダーがあります。その中のEXE形式のファイルを実行すれば、Processing無しでもWindows（64bit環境）上でGokurakuシリーズを実行できます。ただしEXEファイル単体では動作しません。その下にあるjava, lib, sourceのフォルダーのファイルも必要です。

Linux-amd64 フォルダーはLinux上でProcessing無しで実行する際に必要なプログラムです。Windows上で実行する場合は不要です。

当然ですが、Processing統合開発環境上で利用する場合はいずれも不要ですので、src版のみをダウンロードしてご利用ください。



【Gokurakuシリーズの活用例】 動画から連番静止画像を生成

(動画の処理)

- AviUtil (KENくん作) : <http://spring-fragrance.mints.ne.jp/aviutil/>
- + 拡張編集プラグイン (KENくん作) : 同上 (exedit92.zip)
- + MP4入力プラグイン (muken氏作) : https://pop.4-bit.jp/?page_id=7929
- + かんたんMP4出力 : <https://aoytsk.blog.jp/aviutil/34586383.html>
- + JPEG連番出力プラグイン (うえぽん氏作) : <http://auf.jpn.xxxxxxxx.jp/>

FFmpeg (<https://ffmpeg.org/>) を用いる方法もありますが、ここではAviUtil + プラグインを組み合わせての方法で説明します。まずAviUtilをダウンロードしてインストールします (AviUtil110.zipをダウンロードして適当なフォルダーに展開する)。aviutil.exeをダブルクリックして正常に起動することを確認後に、終了して下さい。

次に、aviutil.exeの存在するフォルダーの下に、Pluginsという名前のフォルダーを作成して、L-SMASH_Works_r940_plugins.zipの中から、lwcolor.auc, lwdumper.auf, lwinput.aui, lwmuxer.aufの4つのファイルをコピーします ([参考URL](#))。

AviUtilを起動して、ファイル→環境設定→入力プラグインの優先度の設定で、右の図のようにL-SMASH Works File Readerが存在すればOKです。これでMP4形式の動画を読み込みます。



【Gokurakuシリーズの活用例】 動画から連番静止画像を生成

- 続けて、連番JPEG画像出力のための「JPEG3点セット」をPluginsフォルダーに展開します。（jpeg_input.aui, jpeg_output.auo, jpeg_print.auf）
ファイル→プラグイン出力に“連番JPEG出力（YC48）”が出現すれば正常に読み込まれています。
- かんたんMP4出力は、easym4.auoをPluginsフォルダーにコピーします。
- 拡張編集プラグインは、Pluginsフォルダーではなく、aviutil.exeと同じフォルダー内にZIPファイル内の全ファイルを展開して下さい。



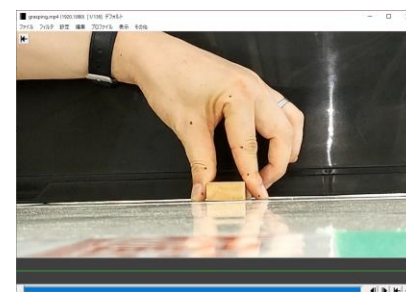
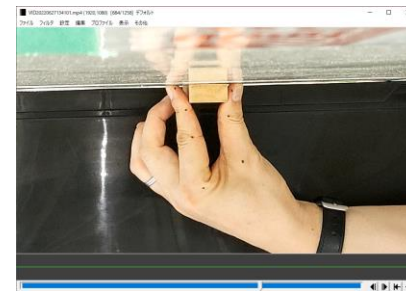
【Gokurakuシリーズの活用例】 動画から連番静止画像を生成

スマートフォンで動画を撮影しました。AviUtilで読み込むと、上下が逆さまでした。スマートフォンを逆さまにして撮影した影響かも知れません（Windowsのプレイヤーなどで表示すると正常ですが）。

VideoStudio2021で必要なシーンのみ切り取って再レンダリングして出力したところ、正しい向きに直りました。撮影時に気を付けるか、別のソフトウェアで修正が必要かも知れません。

正常に読み込めたら出力しましょう．ファイル→プラグイン出力から，“連番JPEG出力（YC48）”を選択します．ファイル選択ダイアログボックスが表示されますので，ファイル名に例えば”grasping”のようなファイル名を入力して保存を押します．入力したファイル名に”-0000”を追加した大量のJPEGファイルが指定したフォルダーに生成されます．あとは，この静止画のフォルダーを**measure**で選択すれば計測開始です．

今回の例では約 4 秒の動画ですので、秒間30コマで**136枚**の静止画が生成されました。



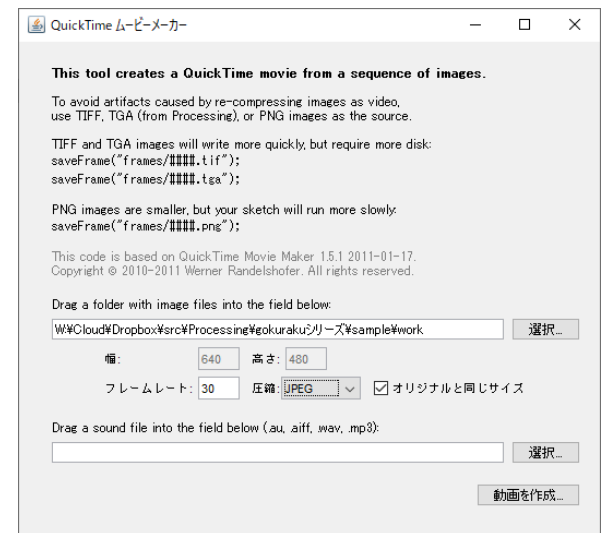
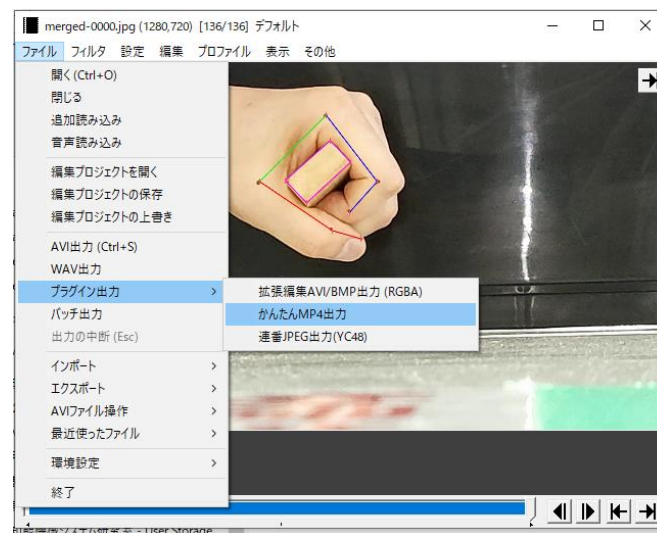
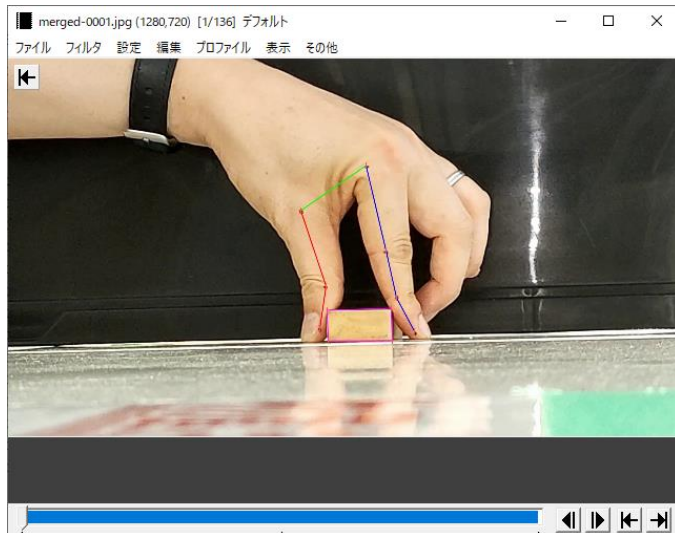
【Gokurakuシリーズの活用例】連番静止画像を読み込み，MP4形式の動画で出力

measure, merge, visualizeで処理を進めると，計測点，Stick picture，元画像を組み合わせた連番静止画像が得られます．これらをAviUtilに読み込みます．

visualizeはデフォルトでmerged-0000.jpgから静止画像を生成します．AviUtilでこの静止画像を読み込むと，自動的に残りの連番画像の番号順に読み込んでくれます．

以上で画像の連結が完了です．ファイル→プラグイン出力から“かんたんMP4出力”を選択してファイルを保存すれば完成です．

※ AviUtilではなく，Processing IDEのツール→ムービーメーカーでも連番静止画像（tif, tga, jpeg, png）からも動画を生成可能です．ただし生成される動画はMOV(QuickTime)形式です．圧縮形式でAnimationを選ぶとWindowsでは再生できないMOVファイルが生成されますが，JPEGを選べば再生可能です．ただ，手軽さを考えるとムービーメーカーも良いのですが，流通性を考えるとAviUtilによるMP4形式がお勧めです．なお，画質の指定が可能である点から，精度を重視する場合はFfmpegも検討すべきでしょう（本ドキュメントでは未紹介）．



【Gokurakuシリーズの活用例】 サンプルデータで動作確認

配布しているサンプルデータをダウンロードして展開して下さい。この中には、

- grasping.mp4 : 撮影した動画ファイル
 - grasping-0000.jpg～grasping-0135.jpg : AviUtilで分解した静止画像ファイル
 - grasping-0000.jpg.txt～grasping-0135.jpg.txt : measureで計測した座標ファイル
 - merged.txt : 上記座標ファイルをmergeで結合した結合ファイル
 - define_stick.txt : 上記結合ファイル用の線分定義ファイル
 - merged-with-image.zip : visualizeで元画像ありで生成した画像ファイル
 - merged-without-image.zip : visualizeで元画像なしで生成した画像ファイル
-
- measureでの計測点は親指先のマークから人差し指，四角柱の四隅の計11点（max measure = 11），スケールは0.666...です（縮小1回）。
 - visualizeでの画像生成はscale有効，define_stickの読み込み（define_stick.txt）です。

まず，grasping-?????.jpg.txtとmerged.txtを削除して下さい。それからmeasureで拡大率を0.666...に1回縮小し，max_measureを11に設定，サンプルデータのフォルダーをmeasureのスクリーンにD&Dして計測開始。計測が終わったらmergeを起動し，同じくサンプルデータのフォルダーをD&Dしてエクスポート。最後はvisualizeでmerged.txtをインポートすればプレビュー可能になります。

【言語パックの修正・多言語の追加】：Ver. 3.4以降

Ver.3.4より多言語対応しました（英語＋日本語）． `measure`, `merge`, `visualize`の各ソースファイルのフォルダー内に`lang.pde`ファイルを追加しました．言語の選択は`header.pde`内の`Lang="ja"`を`"en"`に変更すればメニューなどが英語表示されます．

※ 表示されるメッセージを変更したい場合は，`lang.pde`内のメッセージを変更して下さい．ただしバージョンアップ時に`lang.pde`を上書きしてしまうと修正が消えますので気を付けて下さい．他の言語にも対応したい場合は，以下の`lang.pde`の記述ルールを理解して追記して下さい．

以下二つの関数を用いて記述します．

`set_key(String index, String description);` `index`は見出し語，`description`は英語の記述

`set_str(String lang, String description);` `lang`は言語の種類，`description`はその言語での記述

まず，`set_key()`である見出し語（`index`）の説明（`description`）を英語で記述します．それに引き続き`set_str()`で各言語毎の同じ`index`に対する記述を追加していきます．新たに`set_key()`で`index`が変更されるまでは同じ`index`のまま変化しません．

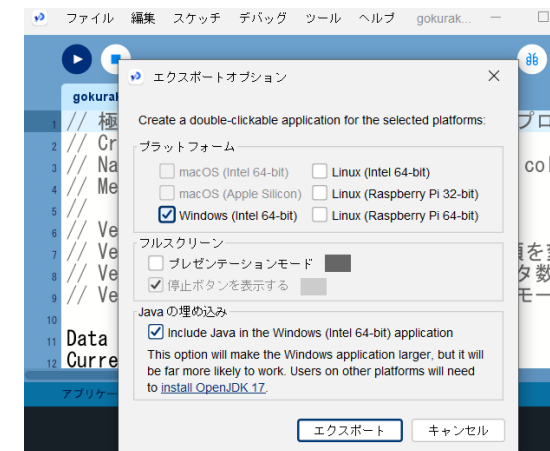
※厳密には，全ての`lang`の`description`が起動時に読み込まれるのではなく，`header.pde`内で宣言した`Lang`と一致する`lang`の`description`のみが読み込まれ，英語の`description`を上書きする仕組みです．

【初期設定値を変更したい：Windows/Linux版】

measure, merge, visualizeのいくつかの初期設定値（サブメニューで変更可能なパラメーターのいくつか）は、header.pdeを修正することで変更可能です。Processing統合開発環境上でGokurakuシリーズを動かすユーザーの場合はこのファイルを修正するだけで良いのですが、実行形式版（Windows/Linux）のユーザーはheader.pdeファイルを書き換えても初期設定値の変更は反映されません。

設定ファイルの書き出し、読み込み機能の搭載も考えたのですが...色々と面倒ですので、強い要望がない限りは機能拡張を行う予定はありません。

実行形式版を利用するユーザーで初期設定値を変更したい場合は、お手数ですが、Processingの統合開発環境をダウンロードして、measure , merge, visualizeを読み込み、それぞれ「ファイル」→「アプリケーションとしてエクスポート」で、変更したheader.pdeで実行形式をリビルドして頂けないでしょうか。



【Ver. 3.0以降の履歴】

Ver.3.4.1 (2023.03.24)

- クイックメジャーのショートメッセージ表示に Δx , Δy を追加
- 桁数を揃えて表示がバタつかないようにする.
- クリップボードにコピーされるファイル名が正しくないバグの修正 (**measure**)
- 依然として**measure** が **Processing4.2**で無反応になる問題の今度こそ本当の改良
- **measure**, **merge**, **visualize**でウィンドウサイズを手動で変更可能だった（実際にはフリーズ）のを無効化
- **visualize** : サブメニューの選択で、テンキーが対応していなかったのを修正

Ver.3.4 (2023.03.20) :

- **visualize** : クイックメジャーモードを搭載する
- 全て : 言語パック対応（日本語，英語）

【Ver. 3.0以降の履歴】

Ver.3.3(2023.03.17)

- **measure** : マウス位置や直線距離・角度（クイックメジャーモード）をリアルタイム表示可能にする
- **measure** : 校正值を入力可能とする、単位も入力可能とする（単なる文字列）
- **visuzlize** : Ctrl同時押しでコマを進める／戻すフレーム数をそれぞれ初期値で変更可能にする、

Ver.3.2 (2023.03.13)

- **measure** : Quick measureモードの追加
- **measure** : Max_measureの初期値を32から0（無制限）に変更
- **measure** : ガイドラインの表示機能の追加
- **全て** : 表示されるウィンドウの初期位置を画面に左上に固定

Ver.3.1 (2023.03.06)

- Processing4.2への対応（**resize**の障害を強引に回避）、ただし不安定なのでしばらくはProcessing 3.5.4を推奨
- **visualize** : 画像無しでプレビューする際に背景色でクリアされないバグ

Ver.3.0 (2023.03.03)

- ユーザーインタフェースの見直し：テキストカラー一色だったのを多色に変えることで、メニューのキー割り当てのグルーピングを行い、選択が容易になった、
- **measure** : 画像サイズが正しく記録されないバグを修正
- **measure** : 新しい画像を読み込む度に倍率（**scale**）を1に戻していたのを止める、
- **merge**: マージ出力時にスキップ数も出力するように仕様変更、これによりVer.2とはデータの互換性が無くなるが、一応、読み込みには問題がなかった、ただし、古いmerged.txtを**visualize**で読み込むと支障がある（Excel等でスキップ数の列を追加すれば読み込み可）、
- **merge**: データを読み込んだ際に座標値を**scale**で変換していたのを止める（オプションで互換性を確保）
- **visualize** : 画像読み込みをデフォルトに変更
- **visualize** : **merge**で座標値を変換するのではなく、**visualize**で変換するように変更（オプションで互換性を確保）

【既知の問題点】

(1) surface.setSize()の問題

Processing 4.2で動かした際に、measureとvisualizeで画像ファイルを読み込んでウィンドウサイズを変更した瞬間に無反応になってしまう問題が発生しました。以下のコード（setSize）の後に、setVisible(true)を追加することでProcessing 4.2でも動作するようになりましたが不安定です。時折、setVisual()追加前と同じように無反応になります。ただ完全にフリーズしている訳ではなく、統合開発環境で停止ボタンを押せば正常にプログラムは停止します。

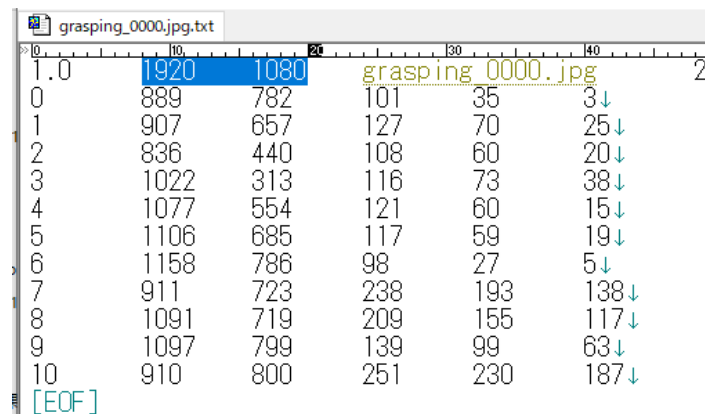
```
surface.setSize(img.width, img.height + 40);  
surface.setVisible(true);
```

※ 最後のコード改善後は発生していないので根絶できた可能性が高い。measureだけフリーズが時折発生していたが、これは改善したものと思われる（surface.setSize()を1スキャン内に複数回実施していたのを修正）。

(2) 画像ファイルの幅と高さの値が座標ファイルに記録されない問題

再現性が低く問題解決が難しい問題です。時折、発生します。記録した座標ファイルの1行目、右下図の青くハイライトされた二つの値が共に0になってしまう時があります。この値がゼロの場合、mergeでその座標ファイルは結合されません。validation()のチェックを一時的に緩めますが、もし異常が発見された場合は手作業で正しい値に変更して下さい。

暫定的に、merge時にこの問題の発生しているファイルを発見した場合、ショートメッセージエリアにワーニングを表示するようにしました。発見時は手作業で直して下さい。



	1.0	1920	1080	grasping_0000.jpg	21
0		889	782	101 35	3↓
1		907	657	127 70	25↓
2		836	440	108 60	20↓
3		1022	313	116 73	38↓
4		1077	554	121 60	15↓
5		1106	685	117 59	19↓
6		1158	786	98 27	5↓
7		911	723	238 193	138↓
8		1091	719	209 155	117↓
9		1097	799	139 99	63↓
10		910	800	251 230	187↓
				[EOF]	