Kurz, Julian

Juliankurz@live.de

Exposee

Im Projekt „Lichtquellen Extrahierer“ wurde eine Software zur Extraktion und Verarbeitung von Lichtquellen entwickelt. Eine GUI ermöglicht das Laden, Analysieren und Exportieren von Dateien.

Lichtquellen Extrahierer

[Untertitel des Dokuments]

|  |  |
| --- | --- |
| Muss-Kriterien | Kann-Kriterien |
| * GUI für den Benutzer * Videodaten verarbeiten * Export als C-Code * Dateimanager (speichern und laden) | * Loop-Funktion * Plot der Lichtintensitäten * Direkte Kompilierung * Beispiel-Auswahl |

Es wurden fast alle Kriterien erfüllt. Außer die direkte Kompilierung. Dafür fehlte die Zeit und es hat sich als doch erheblich komplizierter herausgestellt als am Anfang erwartet.

# Anleitung

Um die Software nutzen zu können, benötigt man zunächst Matlab. Die Software wurde mit Matlab Version R2023b entwickelt, das für Studierende oder Universitätsangehörige in der Regel kostenlos verfügbar ist. Es sind keine speziellen Plugins erforderlich.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Im Lieferumfang befindet sich ein Datenpaket als ZIP-Datei, das alle Dokumente, Matlab-Dateien sowie Beispielvideos enthält. Sobald Matlab installiert und betriebsbereit ist, muss die ZIP-Datei entpackt und die Datei "GUI.m" gestartet werden. Der gesamte Code ist Open Source und kann vom Benutzer angepasst werden.

Ein Bild, das Screenshot, Text, Computer, Software enthält.

Automatisch generierte Beschreibung

Um einen Eindruck von den verarbeitbaren Datentypen zu erhalten, enthält die ZIP-Datei einen Ordner namens „sample\_videos“. Diese Videos können Sie sich gerne ansehen. Nach dem Start der Software finden Sie einen Button „Dateien Laden“, über den Sie Dateien aus dem Demo-Ordner auswählen können.

Ein Bild, das Text, Screenshot, Software, Computersymbol enthält.

Automatisch generierte Beschreibung

Während des Ladevorgangs werden in der Konsole alle aktuellen Rechenoperationen protokolliert. Solange die Software die Videodaten verarbeitet, bleibt die Benutzeroberfläche (GUI) gesperrt. Nach Abschluss der Berechnungen wird die GUI wieder freigegeben.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Im Tab "Plot" werden zunächst die mathematischen Mittelwerte pro Frame des gesamten Videos dargestellt. Es gibt auch einen „Loop“-Button, der den Schieberegler aktiviert. Wenn „Loop“ aktiviert ist, wird das Video geteilt und die Mittelpunkte nach außen verschoben, sodass Anfang und Ende des Videos identisch sind. Der Schieberegler bestimmt dabei, wie stark die Objekte ineinander verschoben werden.

Ein Bild, das Text, Screenshot, Reihe, Zahl enthält.

Automatisch generierte Beschreibung

Im Tab „Simulation“ können Sie eine Simulation erstellen. Zusätzlich wird das simulierte Video im Projektordner gespeichert. Dieser Vorgang erfolgt in Echtzeit und dauert daher genauso lange wie das Video. Der Button „Segmente Abspielen“ zeigt das verarbeitete Video in seinen Segmenten, was ebenfalls in Echtzeit abläuft, während die GUI nicht sichtbar ist.

Ein Bild, das Text, Screenshot, Software, Computer enthält.

Automatisch generierte Beschreibung

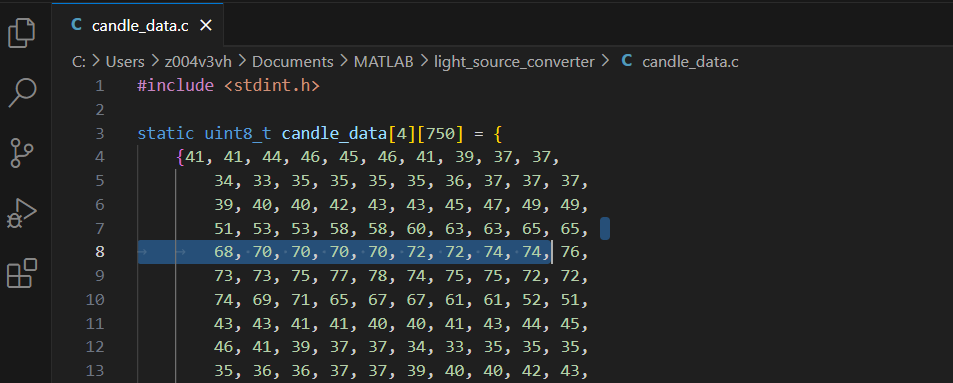
Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Ein Bild, das Flasche, rot, Design enthält.

Automatisch generierte Beschreibung

Das letzte Tab dient dazu, vier Vektoren in einer „unsigned char“-Matrix zu speichern.



# Erläuterung der Klassen

In meinem Code gibt es 6 Oberklassen. Im Folgende werden die Funktionsweisen und die Benutzung erklärt.

## BinaryVideoProcessing

|  |  |  |  |
| --- | --- | --- | --- |
| Method Name | Description | Input | Output |
| apply\_binary\_average | Applies a sliding window to retain white pixels if they remain white throughout the window. | threshold\_video (array of grayscale frames), frames (window size) | binary\_average\_video (array of binary grayscale frames) |
| apply\_shape | Retains only the contour of the first white area in a frame, setting the interior to black. | binary\_video (array of binary frames) | shapeed\_video (frames with contour of first white area) |
| count\_areas | Counts connected white areas in each frame of a binary video. | threshold (array of binary frames) | total\_areas (list of connected area counts) |
| count\_bottom\_white\_pixels | Counts white pixels at the bottom of the white area in each frame. | threshold\_video (array of binary frames) | white\_pixel\_counts (list of white pixel counts at bottom) |
| erase\_bottom\_line | Removes a specified number of bottom rows of white pixels in each frame. | threshold\_video (array of binary frames), lines (number of lines to remove) | bottom\_line\_video (frames with bottom rows removed) |
| extract\_area | Extracts the white area containing the brightest pixel in each frame. | threshold\_video (array of grayscale frames), brightest\_coords (coordinates of brightest pixel) | roi\_video (frames with only the brightest white area visible) |

## GrayVideoAnalyzer

|  |  |  |  |
| --- | --- | --- | --- |
| Method Name | Description | Input | Output |
| calculate\_optimal\_threshold | Calculates the optimal threshold value for the brightest pixels in a video. | video (struct of grayscale frames) | threshold\_level (value between 0 and 255) |
| calculate\_radius | Calculates the radius based on brightness and tolerance. | scaled\_video, brightest\_coords, video\_height, video\_width, radius\_tolerance\_percent | radius (calculated radius) |
| compute\_segment\_averages | Computes the average value of each segment for each frame in the video. | segmented\_video (struct of segmented frames) | avg\_values (average values of segments) |
| count\_areas | Counts the connected areas in each frame of a binarized video. | threshold (struct of binarized frames) | total\_areas (count of connected areas) |
| count\_pixels | Counts the pixels with a certain grayscale value or higher. | gray\_video (struct of grayscale frames), level (threshold level) | quantity (number of pixels matching the level) |
| get\_average | Calculates the average value of all pixels in a grayscale video. | gray\_video (struct of grayscale frames) | average (average pixel value) |
| get\_average\_coord | Computes the average coordinates of all pixels with the value 255 in each frame. | video (struct of binary frames) | average\_coords (average coordinates of bright pixels) |
| get\_brightest\_coord | Finds the brightest pixel in each frame of a video and checks for black pixels between bright points. | video (struct of grayscale frames) | brightest\_coords (coordinates of brightest pixels) |
| bresenham\_line | Uses Bresenham's algorithm to calculate all pixels on a line between two points. | x1, y1, x2, y2 (coordinates for line) | line\_pixels (list of pixels along the line) |
| get\_square\_average | Calculates the average pixel values in a square region around the brightest point in each frame. | gray\_video (struct of grayscale frames), brightest\_coords (coordinates), radius (square region size) | average (average value of square region) |
| optimize\_parameters | Optimizes window size and blur parameters for video processing. | scaled\_video (struct of scaled video) | optimal\_window, optimal\_blur (optimized parameters) |

## GrayVideoConverter

|  |  |  |  |
| --- | --- | --- | --- |
| Method Name | Description | Input | Output |
| apply\_blur | Applies a blur filter to a video. | video (grayscale frames), blur\_radius | blurred\_video (blurred frames) |
| apply\_differential\_average | Calculates differential average with normalization based on brightness and movement. | threshold\_video (grayscale frames), window\_size | differential\_average\_video (differential average grayscale frames) |
| apply\_gray\_converter | Converts a video to grayscale. | video (VideoReader object) | gray\_video (grayscale frames) |
| apply\_moving\_average | Applies moving average to a binarized video. | threshold\_video (grayscale frames), frames (window size) | moving\_average\_video (grayscale frames with moving average) |
| apply\_threshold | Applies a threshold to create a black-and-white video. | video (grayscale frames), threshold\_level (optional) | threshold\_video (binary frames) |
| calculate\_optimal\_threshold | Calculates the optimal threshold using Otsu's algorithm. | video (grayscale frames) | threshold\_level (0-255) |
| apply\_video\_scaler | Scales the video based on a given factor and reduces frame intervals. | gray\_video (grayscale frames), scale\_factor, frame\_interval | scaled\_video (scaled and reduced frames) |
| crop\_video | Crops a specified region in the video. | gray\_video (grayscale frames), left, right, top, bottom | cut\_out\_video (cropped frames) |
| cut\_out | Cuts out a region of interest from the original grayscale video. | roi\_video (grayscale frames), gray\_video (grayscale frames) | cut\_out\_video (cropped frames with regions of interest) |
| cut\_out\_square | Cuts square regions from each frame based on coordinates. | gray\_video (grayscale frames), coord (coordinates), radius | square\_video (square cut-out frames) |
| cut\_out\_surface | Cuts out rectangular regions around a specified point in each frame. | gray\_video (grayscale frames), bottom\_coord (coordinates), width, height | cut\_out\_video (cut-out rectangular frames) |
| display\_video | Displays a binary video. | video (binary frames) | None |
| extract\_roi\_video | Extracts the region of interest from the video based on provided coordinates. | gray\_video (grayscale frames), coord\_bottom, coord\_side, max\_coord, FrameRate | roi\_video (extracted region frames) |
| rotate\_video | Rotates each frame of the video by a specified angle. | gray\_video (grayscale frames), angle\_rad (rotation angle in radians) | rotated\_video (rotated grayscale frames) |
| scaleGrayVideo | Optimizes scaling and frame rate based on movement and area calculations. | gray\_video (grayscale frames), FrameRate | scaled\_video (optimized frames) |

## GUI

|  |  |  |  |
| --- | --- | --- | --- |
| Method Name | Description | Input | Output |
| setWidgetsVisible | Activates or deactivates certain UI components. | app, state | None |
| setWidgetsEnabled | Enables or disables certain UI components. | app, state | None |
| closeRequest | Closes the GUI and terminates processes. | app, src, event | None |
| VideoladenButtonPushed | Loads a video when the 'Load Video' button is clicked. | app, event | None |
| processVideo | Processes the loaded video and analyzes its segments. | app, video\_path | None |
| updatePlots | Updates plots based on the current frame index. | app, frameIndex | None |
| SliderValueChanged | Updates plots when the slider value changes. | app, event | None |
| LoopCheckBoxValueChanged | Toggles loop functionality for the plots. | app, event | None |
| SimulationStartButtonPushed | Starts the simulation when the button is clicked. | app, event | None |
| SegmentsPlayButtonPushed | Plays segmented videos when the button is clicked. | app, event | None |
| CDateispeichernButtonPushed | Exports data as C-code when the button is clicked. | app, event | None |
| GUI (constructor) | Initializes the GUI and hides certain widgets on start. | None | GUI object |
| delete | Deletes the GUI when the app is closed. | app | None |

## LightSourceConverter

|  |  |  |  |
| --- | --- | --- | --- |
| Method Name | Description | Input | Output |
| export\_candle\_data | Exports data to a C file. | arrays | C file |
| extract\_light\_source | Extracts the light source from a video. | video | Extracted light source |
| get\_loop | Returns a looped version of the vector based on frames. | vector, frames | Looped vector |
| play\_segmented\_videos | Plays segmented videos. | None | None |
| play\_simulation\_video | Plays the simulation video. | None | None |
| plot\_segment\_averages\_color | Plots the segment averages using color. | None | Segment averages plot |
| segment\_cropped\_video | Segments the cropped video. | cropped\_video | Segmented video |

## LightSourceTracker

|  |  |  |  |
| --- | --- | --- | --- |
| Method Name | Description | Input | Output |
| apply\_tracker | Processes the binarized video and returns the coordinates of the bottom white pixels and maximum white area coordinates. | video | coord\_bottom, coord\_side |
| compute\_max\_coords | Computes the maximum coordinates based on a tracking point and side coordinates. | tracking\_point, coord\_side | max coordinates |
| find\_stable\_point | Finds a stable point in the video using contour detection and summing of frames. | contoured\_video, num\_frames\_to\_sum | stable point |
| get\_movement\_direction | Determines the movement direction based on average coordinates. | average\_slow\_coords, average\_middle\_coords | movement direction |
| interpolate\_coords | Interpolates the bottom white pixel coordinates over frames. | coord\_bottom | interpolated coordinates |
| process\_and\_rotate\_video | Processes and rotates the video at the specified frame rate. | filtered\_roi\_video, FrameRate | rotated video |