Institute for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

ArUco Library

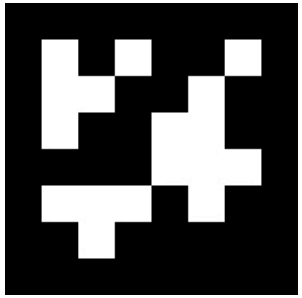
This library is used for Augmented Reality applications and can detect the position and orientation of markers in an image, relative to the camera frame.

The library uses a set of predefined markers. You can print them by executing the following in a terminal:

```
aruco_create_marker <makerid(0:1023)> outfile.jpg sizeInPixels
```

Example:

```
aruco_create_marker 4 outfile.jpg 216
```



Weblink to ArUco:

http://www.uco.es/investiga/grupos/ava/node/26

Or just use this convenient website to print markers:

https://tn1ck.github.io/aruco-print/

Institute for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Detecting Markers in OpenCV Images

```
MarkerDetector mDetector;
vector<Marker> markers;
mDetector.detect(input_image,markers,camera_parameters);
```

Each marker has a unique ID: `Marker.id`

To get the position you need the real size of the marker.

```
Marker.calculateExtrinsics(0.018,camera_parameters,true);
```

Where 0.018 is the size in meters.

Institute for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Detecting Markers in OpenCV Images

Accessing marker position	`Marker.Tvec.at<float>(coordinate_number);` `//translation matrix`
Accessing marker orientation	`Marker.Rvec.at<float>(coordinate_number);` `//rotation matrix`
Drawing the marker on an image	`Marker.draw(input_image,Scalar(0,0,255),2);` `//draws in input_image with border color Scalar(B,G,R) and line thickness 2.`
Showing the resulting image	`Imshow("markers",input_image);` `waitKey(10); //as before`

Institute for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Marker Detection Example