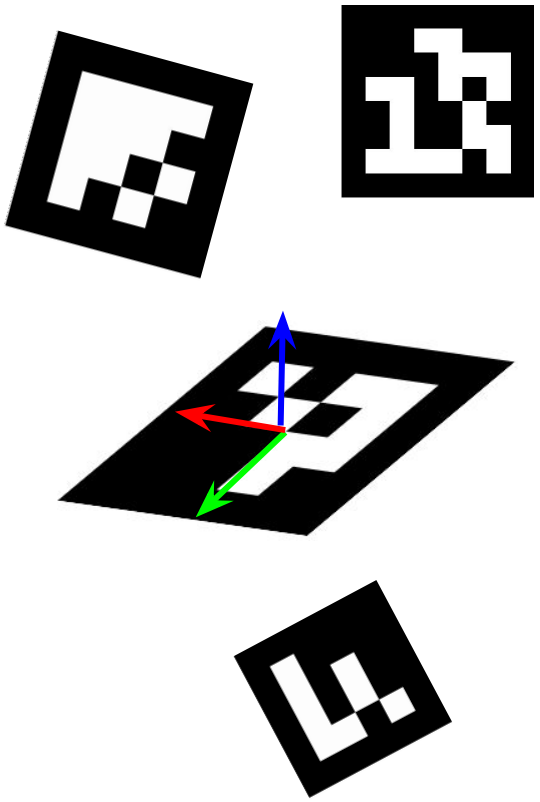
Computer Vision

Class 05



Raquel Frizera Vassallo

Class 04

01 Why use fiducial markers?

Applications

02 ArUco Markers

Markers
Dictionaries

03 ArUco Detection

04 Pose Estimation

01

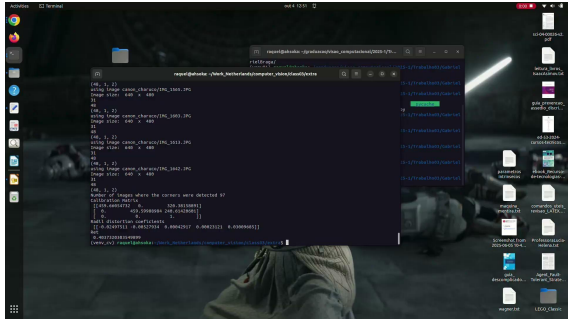
Why use fiducial markers?

Why use fiducial markers?

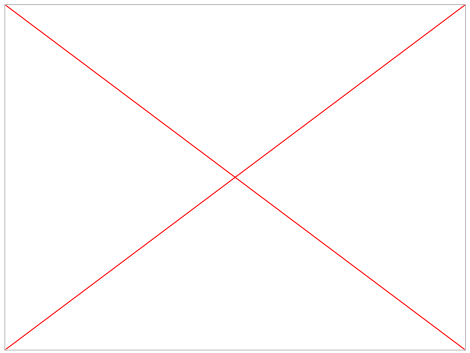
Finding correspondences between points in the real environment and their 2d image projection is usually a difficult step.

Fiducial markers (like ArUco, AprilTag, or ARToolKit markers) are used in many computer vision and robotics applications because they provide easy, reliable, and precisely measurable reference points in the real world.

The main benefit is that a single marker provides enough correspondences (its four corners) to obtain the camera pose.



Also, the inner binary codification makes them specially robust, allowing the possibility of applying error detection and correction techniques.



Applications

Application	Description	Example Marker / Use Case
Camera Calibration & Pose Estimation	Provides known geometric references to compute camera intrinsic/extrinsic parameters.	Using markers to estimate camera pose.
Augmented Reality (AR)	Acts as an anchor to overlay 3D virtual objects onto the real world.	Displaying a virtual model on top of a detected marker.
Robotics & Drone Navigation	Helps robots or UAVs localize and orient themselves relative to known positions.	A drone landing on a platform or a mobile robot correcting its pose by identifying markers in the workspace.
Industrial Automation	Assists in precise alignment, measurement, and quality control.	Robot arm locating a part using markers on the assembly line.
Multi-Camera Calibration	Aligns and synchronizes multiple cameras observing the same scene.	Calibrating a stereo or multi-camera motion-capture system.
Object Tracking & 3D Reconstruction	Serves as a fixed reference to track or reconstruct object motion.	Tracking rigid body motion using markers attached to objects.

02

ArUco Markers

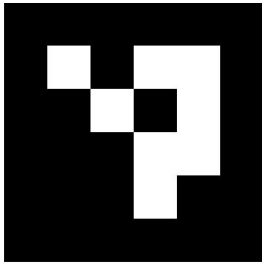
ArUco Markers

An ArUco marker is a synthetic square marker composed by:

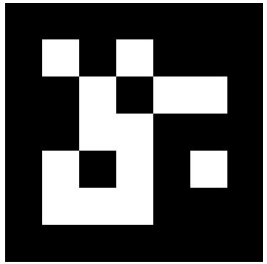
- a wide black border and
- an inner binary matrix which determines its identifier (id).

The black border facilitates its fast detection in the image and the binary codification allows its identification and the application of error detection and correction techniques.

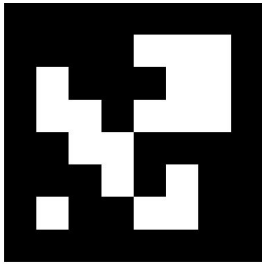
The marker size determines the size of the internal matrix.



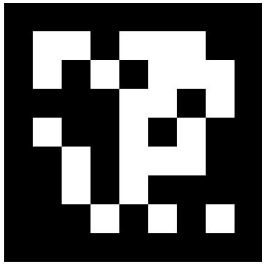
4x4



5x5



6x6



7x7

ArUco Dictionaries

A dictionary is a set of markers defined as the list of binary codifications of each of markers in the set.

- The dictionary defines the size of the markers (the number of bits/modules)
    - For instance the dictionary of markers 4x4 is composed by 16 bits.

- The dictionary size is the number of markers that compose the dictionary.

- The inter-marker distance is the minimum Hamming distance between dictionary markers that determines the dictionary's ability to detect and correct errors.
    - Smaller dictionary → larger marker sizes → better inter-marker distance
      That is good for detection and correction.
    - For instance, if you need only 10 markers in your application, it is better to use a dictionary composed only of those 10 markers than using a dictionary composed of 1000 markers. The dictionary composed of 10 markers will have a higher inter-marker distance and, thus, it will be more robust to errors.

03

ArUco Detection

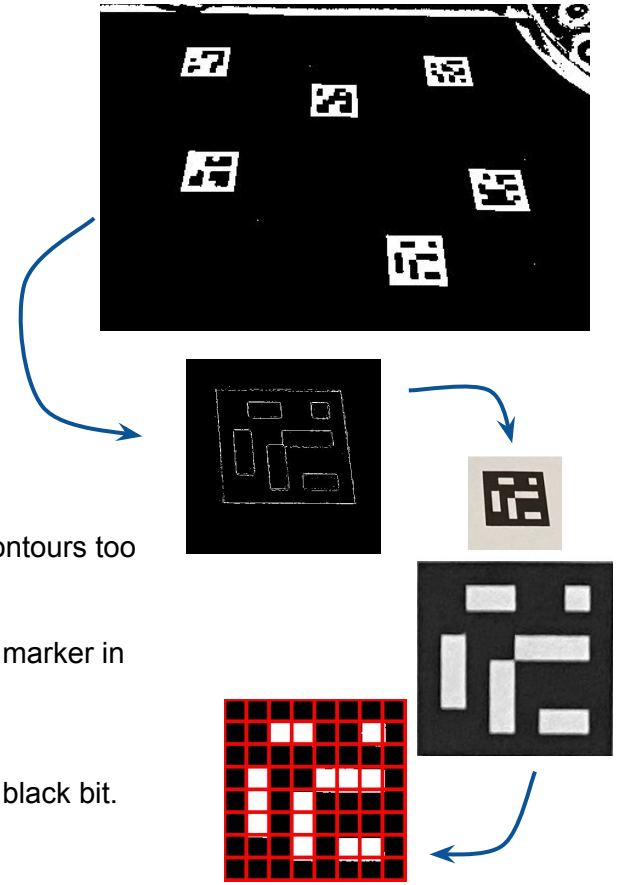
ArUco Detection

The detection returns a list of detected markers.
Each detected marker includes:

- The position of its four corners in the image (in their original order).
- The id of the marker.

The detection process includes two main steps:

- Detection of marker candidates.
  - Find square shapes that are candidates.
  - Adaptive thresholding to segment the markers.
  - Extract contours.
  - Those not convex or not similar to a square shape are discarded.
  - Extra filtering (removing contours that are too small or too big, removing contours too close to each other, etc).
- Determine if they are actually markers by analyzing their inner codification.
  - Extract the marker bits by using a perspective transformation to obtain the marker in its canonical form.
  - Threshold the canonical image to separate white and black bits.
  - Divide into different cells according to the marker size and the border size.
  - Count the black or white pixels in each cell to determine if it is a white or a black bit.
  - Determine if the marker belongs to the specific dictionary.
  - Apply correction techniques when necessary.



04

Pose Estimation

Pose Estimation

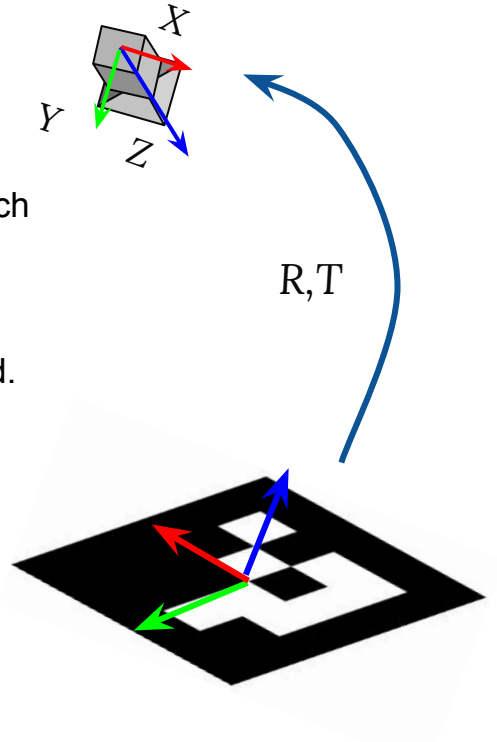
To perform camera pose estimation, you need to know:

- Camera matrix (intrinsic parameters);
- Distortion coefficients.

When you estimate the pose with ArUco markers, you can estimate the pose of each marker individually or you can use ArUco Boards if you want to estimate one pose from a set of markers.

Using ArUco boards instead of single markers allows some markers to be occluded.

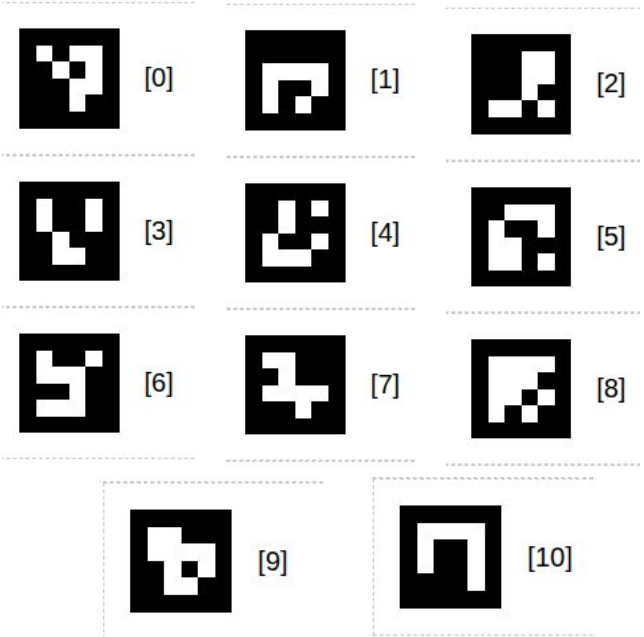
The camera pose relative to the marker is a 3D transformation from the marker coordinate system to the camera coordinate system. It is specified by Rotation and Translation vectors



Online ArUco markers generator

https://chev.me/arucogen/

https://fodi.github.io/arucosheetgen/



So now...
Let’s play with ArUco Markers

Credits

Open CV site:
https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html