Project 1

Perception for Autonomous Robots-ENPM673

Report

By:

1. Sanket Acharya

2. Mithun Bharadwaj

3. Abhiram Dapke

INDEX

1. Definitions:

1.1 AR Tag (Fiducial Marker): A fiducial marker or fiducial is an object placed in the field of view of an imaging system which appears in the image produced, for use as a point of reference or a measure. It may be either something placed into or on the imaging subject, or a mark or set of marks in the reticle of an optical instrument.

1.2 Corner Detection: Corner detection is an approach used within computer vision systems to extract certain kinds of features and infer the contents of an image. Corner detection is frequently used in motion detection, image registration, video tracking, image mosaicing, panorama stitching, 3D modelling and object recognition. Corner detection overlaps with the topic of interest point detection.

1.3 Feature Generation: Feature generation is also known as feature construction, feature extraction or feature engineering. There are different interpretations of the terms feature generation, construction, extraction and engineering.

In machine learning, pattern recognition and in image processing, feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. Feature extraction is a dimensionality reduction process, where an initial set of raw variables is reduced to more manageable groups (features) for processing, while still accurately and completely describing the original data set.

Two goals of feature generation can be dimensionality reduction and accuracy improvement. When the goal of a feature generation method is dimensionality reduction, then the result will be a feature space which contains less features than the original feature space. However, when the goal is accuracy improvement, the resulting feature space will most likely contain more features than the original feature space.

1.3.1 SIFT: The scale-invariant feature transform (SIFT) is a feature detection algorithm in computer vision to detect and describe local features in images.

SIFT keypoints of objects are first extracted from a set of reference images and stored in a database. An object is recognized in a new image by individually comparing each feature from the new image to this database and finding candidate matching features based on Euclidean distance of their feature vectors. From the full set of matches, subsets of keypoints that agree on the object and its location, scale, and orientation in the new image are identified to filter out good matches.

1.3.2 SURF: In computer vision, speeded up robust features (SURF) is a patented local feature detector and descriptor. It can be used for tasks such as object recognition, image registration, classification or 3D reconstruction. It is partly inspired by the scale-invariant feature transform (SIFT) descriptor. The standard version of SURF is several times faster than SIFT and claimed by its authors to be more robust against different image transformations than SIFT.

To detect interest points, SURF uses an integer approximation of the determinant of Hessian blob detector, which can be computed with 3 integer operations using a precomputed integral image. Its feature descriptor is based on the sum of the Haar wavelet response around the point of interest. These can also be computed with the aid of the integral image.

SURF descriptors have been used to locate and recognize objects, people or faces, to reconstruct 3D scenes, to track objects and to extract points of interest.

1.3.3 ORB: Oriented FAST and rotated BRIEF (ORB) is a fast robust local feature detector that can be used in computer vision tasks like object recognition or 3D reconstruction. It is based on the FAST keypoint detector and the visual descriptor BRIEF (Binary Robust Independent Elementary Features). Its aim is to provide a fast and efficient alternative to SIFT.

1.4 Feature Matching: Feature matching means finding corresponding features from two similar datasets based on a search distance. One of the datasets is named source and the other target, especially when the feature matching is used to derive rubber-sheet links or to transfer attributes from source to target data. These datasets overlap each other but are not perfectly aligned due to inconsistent data collection, changes over time, or other reasons.

1.4.1 Brute-Force Search: A brute-force search or exhaustive search, also known as generate and test, is a very general problem-solving technique and algorithmic paradigm that consists of systematically enumerating all possible candidates for the solution and checking whether each candidate satisfies the problem's statement.

A brute-force algorithm to find the divisors of a natural number n would enumerate all integers from 1 to n and check whether each of them divides n without remainder. A brute-force approach for the eight queens puzzle would examine all possible arrangements of 8 pieces on the 64-square chessboard, and, for each arrangement, check whether each (queen) piece can attack any other.

While a brute-force search is simple to implement, and will always find a solution if it exists, its cost is proportional to the number of candidate solutions – which in many practical problems tends to grow very quickly as the size of the problem increases (combinatorial explosion). Therefore, brute-force search is typically used when the problem size is limited, or when there are problem-specific heuristics that can be used to reduce the set of candidate solutions to a manageable size. The method is also used when the simplicity of implementation is more important than speed.

1.5 Edge Detection: Edge detection includes a variety of mathematical methods that aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. The same problem of finding discontinuities in one-dimensional signals is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction.

<u>1.6 Homography Estimation:</u> In the field of computer vision, any two images of the same planar surface in space are related by a homography (assuming a pinhole camera model). This has many practical applications, such as image rectification, image registration, or computation of camera motion—rotation and translation—between two images. Once camera rotation and translation have been extracted from an estimated homography matrix, this information may be used for navigation, or to insert models of 3D objects into an image or video, so that they are rendered with the correct perspective and appear to have been part of the original scene camera motion—rotation and translation—between two images. Once camera rotation and translation have been extracted from an estimated homography matrix, this information may be used for navigation, or to insert models of 3D objects into an image or video, so that they are rendered with the correct perspective and appear to have been part of the original scene.

2. PROCEDURE:

The pipeline to detect AR Tags is explained in the following steps:

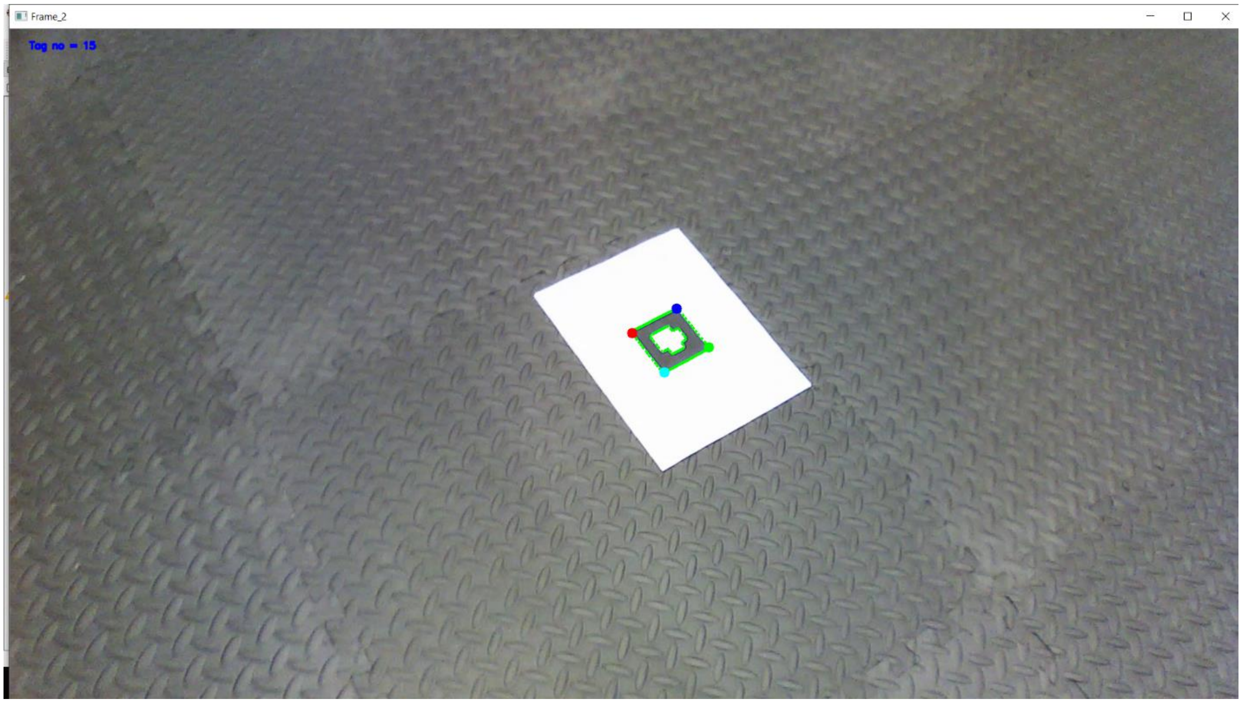
- Load the video using cv2.VideoCapture(filename).
- Read the video frame-wise for ease of manipulation.
- Convert each frame to grayscale image.
- Smooth the image by applying a Gaussian filter to it. This removes noise to make the detection of the tag easier and avoids false detections.
- Threshold is applied using cv2.threshold.
- The next step is to detect the edges in the image. We found contours using cv2.findContours.
- Choose the region of interest from multiple contours either using the hierarchy return parameter which is slightly unreliable or area thresholding.
- Compute the extreme points of the contour as a method for corner detection.
- The function cv2.getPerspectiveTransform is used to get the transformation matrix between the region of interest and the reference tag.
- Then, the function cv2.warpPerspective is used to transform the region of interest.

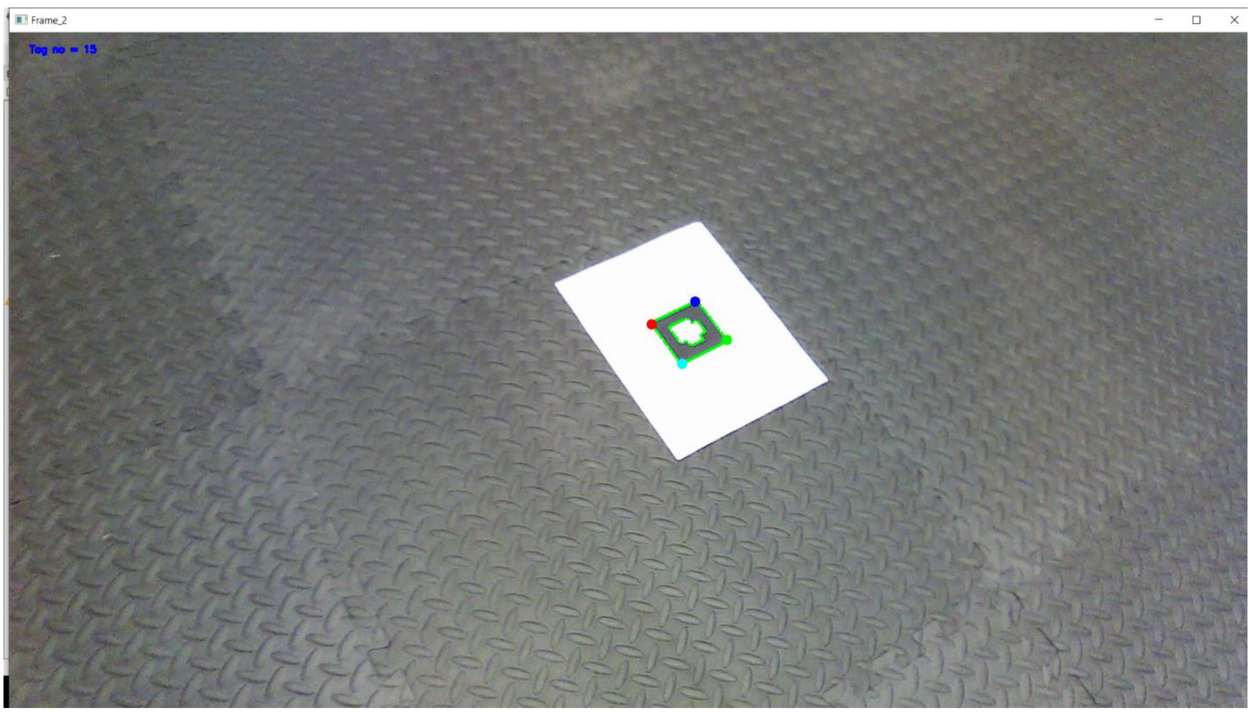
- The transformed image is of the same dimensions as the reference dag making it easier to decode.
- Using the orientation block and the encoding scheme gives, a custom function Tag_no was created.

The procedure for estimating camera's pose and the projection model consists of the following steps:

1. Detect four corners of the marker in the input image captured by camera (the true corners for the upright orientation of marker).
2. Determine the world coordinates of the corners, and compute the homography between the detected corner points in the image (pixel coordinates) and the corresponding points in the reference (world) coordinate system.
3. Determine the camera's pose {the rotation matrix R and translation vector t.}
4. Construct the projection matrix.
5. Now, project any point from the world coordinate system to the image plane.

SCREENSHOTS OF OUTPUTS:





3. LIMITATIONS AND SCOPE FOR IMPROVEMENT

- Pipeline for implementation can be optimized further by splitting it into constituent methods that robustly handles adverse scenarios.
- Better tracking can be implanted to handle cases where the tag goes out of frame.
- Corner point detection can be improved as the extreme points calculated in the implementation have a slight tendency to drift affecting the decoding of the tag number.
- Implement the cube placement on the AR tag using homography and camera calibration matrix.

4. REFERENCES:

1. Wikipedia
2. ENPM 673 project 1 pdf
3. Supplementary homography pdf
4. https://www.cs.ubc.ca/grads/resources/thesis/May09/Dubrofsky_Elan.pdf
5. https://www.uio.no/studier/emner/matnat/its/UNIK4690/v16/forelesninger/lecture_4_3-estimating-homographies-from-feature-correspondences.pdf