Fiducial Markers

(Fixed basis of comparison)

⇒ It 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.

Square planar marker

Artag

These markers are composed of an external black border and an internal code (most often binary) to uniquely identify them.

AprilTag

→ Useful for a wide variety of tasks including augmented reality, robotics, and camera calibration.

→ Targets can be created from an ordinary printer, and the AprilTag detection software computes the precise 3D position, orientation, and identity of the tags relative to the camera.

→ AprilTags are conceptually similar to QR Codes, in that they are a type of two-dimensional bar code.

  → However, they are designed to encode far smaller data payloads (between 4 and 12 bits), allowing them to be detected more robustly and from longer ranges.

  → Further, they are designed for high localization accuracy— you can compute the precise 3D position of the AprilTag with respect to the camera.

ICRA 2011

We describe a new visual fiducial system that uses a 2D bar code style “tag”, allowing full 6 DOF localization of features from a single image.

★ Introduction

⇒ Visual fiducials are artificial landmarks designed to be easy to recognize and distinguish from one another.

QR Code

→ With a QR code, a human is typically involved in aligning the camera with the tag and photographs it at fairly high resolution obtaining hundreds of bytes, such as a web address.

⇒ In contrast, a visual fiducial has a small information payload (perhaps 12 bits), but is designed to be automatically detected and localized even when it is at very low resolution, unevenly lit, oddly rotated, or tucked away in the corner of an otherwise cluttered image.

⇒ Fiducial systems also are designed to detect multiple markers in a single image.

⇒ Visual fiducial systems are perhaps best known for their application to augmented reality, which spurred the development of several popular systems including ARToolkit and ARTag.

Planar tags have also been used to generate user interfaces that overlay robots' plans and task assignments onto a head-mounted display

The central contributions of this paper are:

- We describe a method for robustly detecting visual fiducials.
  - We propose a graph-based image segmentation algorithm based on local gradients that allows lines to be precisely estimated. We also describe a quad extraction method that can handle significant occlusions.
- We demonstrate that our detection system provides significantly better localization accuracy than previous systems.
- We describe a new coding system that address problems unique to 2D barcoding systems:
  - robustness to rotation and robustness to false positives arising from natural imagery.

    A result that indicates that a given condition is present when it actually is not present
- We specify and provide results on a set of benchmarks which will allow better comparisons of fiducial systems in the future.

Previous Work

{Example}

ARToolkit was among the first tag tracking systems, and was targeted at artificial reality applications.

- It differed, from AprilTag in that its payload was not directly encoded in binary: instead, it used symbols database such as the latin character 'A'.
- A major disadvantage of this approach is the computational cost associated with decoding tags, since each template required a separate, slow correlation operation.
- The tag detection scheme used by ARToolkit is based on a simple binarization of the input image based on a user-specified threshold.
- Not robust to changes in illumination.
- In general, ARToolkit's detections can not handle even modest occlusions of the tag's border.

ARTag provided improved detection and coding schemes.

- Robust to changes in lighting
- Details of the detector algorithm are not public
- Able to detect tags whose border is partially occluded
- ARTag also provided the first coding system based on forward error correction, which made tags easier to generate, faster to correlate, and provided greater orthogonality between tags.
- The performance of ARTag inspired several improvements to ARToolkit, which evolved into ARToolkitPlus and finally Studierstube Tracker
  - These versions introduced digitally-encoded payloads like those used in ARTag.
  - Despite being later work, our experiments show that these encoding systems do not perform as well as that used by ARTag

★ Detecton

⇒ Our system is composed of two major components:

- Tag detector
- Coding system

⇒ Loosely speaking, the detector attempts to find four-sided regions (“quads”) that have a darker interior than their exterior.

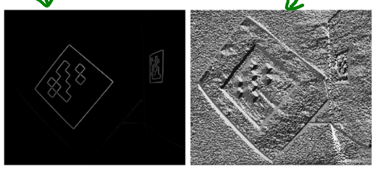
⇒ Quad detector is designed to have a very low false negative rate, and consequently has a high false positive rate.

- We rely on the coding system to reduce this false positive rate to useful levels.

{ A result that indicates that a given condition is not present when it actually is present }

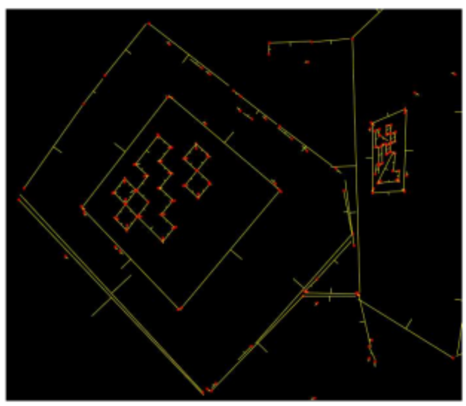
(A) Detecting line segment

1. Compute gradient at each pixel. Magnitude as well as Direction.



2. Using a graph based method, pixels with similar gradient direction and magnitude are clustered in components.
   - A graph is created in which each node represents a pixel.
   - Edges are added between adjacent pixels with an edge weight equal to the pixels’ difference in gradient direction.
   - These edges are then sorted and processed in terms of increasing edge weight.
   - For each edge, we test whether the connected components that the pixels belong to should be joined together.
   - This gradient-based clustering method is sensitive to noise in the image: leads to growth of the number of components.
     - The solution to this problem is to low-pass filter the image.

3. Using a weighted least square, a line segment is then fit to the pixels in each components.
   - The direction of the line segment is determined by the gradient direction.
     - Dark on left.
     - Light on right.
   - We adjust the line segment so that the above condition is met.



B Quad detection

- TASK: Find sequences of line segments that form a 4-sided shape, i.e., a quad.
- APPROACH: Depth-first search with a depth of four: each level of the search tree adds an edge to the quad.
  - At depth one, we consider all line segments.
  - At depths two through four, we consider all of the line segments that begin "close enough" to where the previous line segment ended and which obey a counter-clockwise winding order.
    - We choose large threshold (for close enough) which leads to a low false negative rate, but also results in a high false positive rate.
  - Once four lines have been found, a candidate quad detection is created.

C Homography Estimation

- We compute the 3×3 homography matrix that projects 2D points in homogeneous coordinates from the tag's coordinatesystem to the 2D image coordinate system.
  - [0 0 1]^T is at the center of the tag and the tag extends one unit in the x̂ and ŷ directions
  - The homography is computed using the Direct Linear Transform (DLT) algorithm.
  - For doing this we need the following:
    - Camera intrinsics
    - Camera extrinsics
    - Physical size of the tag

★ Payload Decoding

- We do this by computing the tag-relative coordinates of each bit field and then thresholding the resulting pixels.

★ Coding System

- Once the data payload is decoded from a quad, it is the job of the coding system to determine it is valid or not.
- The goals of a coding system are to:
  - Maximize the number of distinguishable codes.
  - Maximize the number of bit errors that can be detected or corrected.
  - Minimize the false positive/inter-tag confusion rate.
  - Minimize the total number of bits per tag (and thus the size of the tag).

⇒ We describe a new coding system based on modified lexicodes that provides significant advantages over previous methods.

Classical lexicode

- It is characterized using two parameters:
  - Number of bits n in each codeword.
  - Minimum Hamming distance between any two codeword d.
- Lexicodes can:
  - Correct: $\frac{d-1}{2}$ bit error.
  - Detect: $d/2$ bit error.

36h10

36 bit encoding

Hamming distance = 10

Why modified Lexicode

- The coding scheme must be robust to rotation. The standard lexicode generation algorithm does not guarantee this property.
- Lexicode generation algorithm should reject candidate codewords that result in simple geometric patterns (To reduce false positives).
- Rather than test codewords in order (0, 1, 2, 3, ...), we consider (b, b+1p, b+2p, b+3p, ...) where b is an arbitrary number, p is a large prime, and the lowest n bits are kept at each step.

⇒ Many useful code families are already computed and distributed with our software; most users will not need to generate their own code families.