

Lab 2: Sensing & Perception -- Show Me The Money

Objective

Cynics claim there is no money in robotics. This laboratory aims to change that (with the help of a little pocket change).

The laboratory will investigate sensor interfacing and processing with the help of a RGB+D camera.

Now the money shot! In the environment is a money plate. Using the RGB+D camera (a Microsoft Kinect), this laboratory will:

- **calibrate** the RGB+D camera
- **segment** the valuable features (the money)
- **locate** the camera relative to a central frame
- **map** the location in the environment.

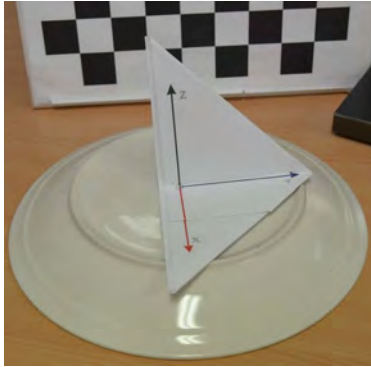
As money has different levels, so does the level of laboratory performance. To borrow the vernacular, there is poor performance (a 5¢ showing) to exceptional performance (“gold coin”). In general, exceptional solutions tend to be those that adeptly and automatically exploit structure in the scene

Thus, robotics not only accounts for the money, but shows us the money!

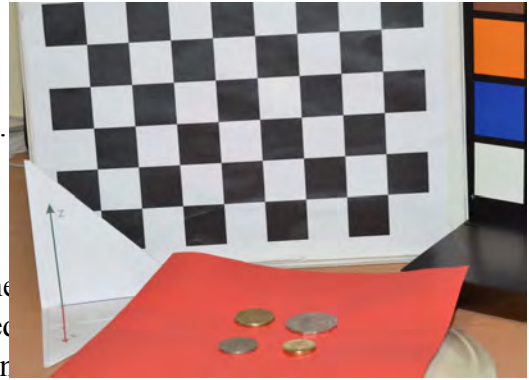


Workspace

The scene will consist of a flat plate and a central frame. The plate and/or central frame might not be on the table (i.e., they might be elevated and at orientation).



The plate's frame is defined as being right handed and located at the center point of the plate and oriented with the *z-axis* orthogonal and outward to the plane of the plate and with the *x-axis* defined as being oriented towards the camera (see illustration at left).



The workspace, especially for operation at more exceptional levels may include clutter in the form of foreign currency, keys, soda beverage cans, small chocolates, and other small random items that might be found in one's pocket (the catalyst for the laboratory is pocket change or rather to make sense from cents).

Scene Structure

The workspace will be defined with varying levels of structure and clutter, with lower performance standards having more structure. This is outlined as follows:

Item	Basic Level ("50¢")	Exceptional Level ("Gold coin")
Background	Monocolor non-white paper (e.g., red)	(Anything)
Plate	White	(Anything)
Plate Orientation	Level with table and frame ($\pm 5^\circ$)	(Anything)
Number of Coins	$1 \leq N \leq 4$	$0 \leq N \leq 32$
A\$ Notes	No	Maybe
Foreign Currency	No	Maybe
Clutter	No	Yes
Colour Calibration Target	Optional	No (except for task 1)
Calibration Pattern	Provided	Maybe

Calibration

In order to obtain a useful measurement, the Kinect needs to be calibrated first. Let us begin with the camera. As noted in class, the calibration parameters to connect raw (not to be confused with the DSLR image type of “RAW”) images to 3D measurements are:

- focal length at the center (f_c)
- principal point offsets from the center (c_c)
- lens skew and distortion (α_c)
- Orientation (R_c) and Position (p_c) of the camera

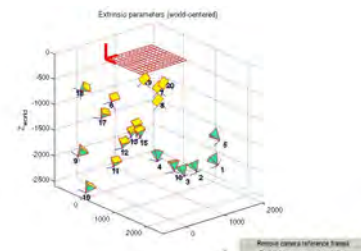
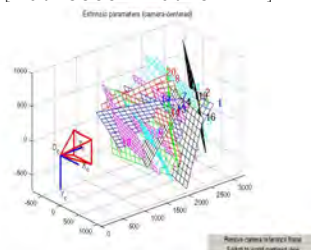
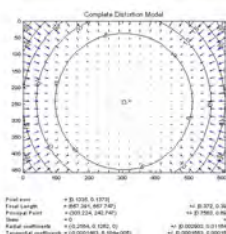


There may be a misalignment of the x -axis of the image coordinate system and the base line. However, this can be ignored if the depth coordinate is defined parallel with the image coordinate system instead of the baseline, but this makes later coordination of sensor data to motion complex.

Planar target camera calibration ([Zhang's method](#)) uses multiple viewpoints of a calibration target with calibration determined by correlating known points on the target with observed points by the camera. A common target (used by the calibration toolbox and OpenCV) is a black and white planar checkerboard as the corner points have high contrast allowing for (subpixel) precision even with noisy camera and target printing. In this laboratory, while camera placement is an option, the view is fixed. Thus, the camera has to be calibrated before; or, the target needs to have motion (e.g., be attached to a soft spring, such as a string holding the target, or even via a [mobile](#)); or, there has to be multiple viewpoints (or calibration targets) in the scene (e.g., a calibration cube).

Calibration will provide the intrinsic (perspective camera model parameters) and extrinsics (camera pose relative to the target). However, one might ask if this is necessary as one way to frame the problem is to put the camera well overhead, thus making the scene essentially para-perspective. While this is tempting and could even make calibration a direct linear (scale) operation, this comes at the cost of less occlusion robustness and no direct means for radial distortion correction. An example calibration (from the Bouguet Calibration Toolbox) is shown below:

```
Calibration results after optimization (with uncertainties):
Focal Length:      fc = [ 657.39071   657.74678 ] ± [ 0.37195   0.39793 ]
Principal point:    cc = [ 303.22367   242.74729 ] ± [ 0.75632   0.69189 ]
Skew:              alpha_c = [ 0.00000 ] ± [ 0.00000 ] ...
Distortion:         kc = [ -0.25541   0.12617   -0.00015   0.00006   0.00000 ] ±
[ 0.00290   0.01154   0.00016   0.00015   0.00000 ]
Pixel error:        err = [ 0.13355   0.13727 ]
```



The basic level of calibration would be to calibrate this in advance before the laboratory starts, whereas the exceptional level would be to calibrate camera online during the laboratory.

On The Money: Segmenting and Counting

It might sound ironic, but the process of determining the total value of funds on the money plate is potentially simpler than the process of (metrically) locating it. This is because it is possible to determine the total monetary value based on processing in pixel space, whereas locating the camera and mapping the coins is done in a metric space.

With this in mind, the approach then becomes one of segmenting the coins and then “recognizing” them using a prior model (i.e., predefined coin features for known denominations), thus giving the total value. The choice of segmentation approach and features and subsequent “classification” are all design decisions.

For example, the segmentation can be done by “dense” methods, such as color thresholds or brightness patches (i.e., sum of squared differences of windows of the intensity map). They can also be computed by “features” such as edges, “lines” (whereby a “line” is a geometric function, such as a conic section [e.g., circle]), [corner detection](#), etc. Do note that there are also image features that are computed from the aforementioned characteristics -- sometimes referred to as a descriptor -- [some of these include: SIFT](#), [SURF](#), and (more recently) [ORB](#). The process of determining which feature(s) to use to focus the search for money targets (coins and notes) is a design decision and may include multiple “features” or “descriptor” methods.

Of course, the process of determining the total value may be improved by finding the locations and size of the coins and notes. This is the focus of the following sections.

The basic level of operation would be to extra the coins and estimate the total value to within 25% of the total funds, whereas the exceptional level of operation would be to segment the coins and notes and estimate the value of each unit of coin/note (i.e., 5¢, 10¢, \$1, \$5, etc.) within 10%.

Localising the Camera

The next step to consider is to **fully** locate (i.e., position and orientation) the camera relative the central frame. As noted in the [Workspace section](#), the central frame is indicated by a METR4202 Frame and it will typically be located to the left of the money plate. Teams may augment the METR4202 Frame with an barcode or tag (e.g., a [caltag](#), [QR Code](#), etc.). Two sample frames (with [caltag](#) or [reacTIVision](#) fiducials) are on the class website. Teams may choose to build their own patterns.

Camera placement and calibration pattern(s) (if any) are design considerations for the team. Please make a considered choice and be able to justify this decision. A basic level of operation would be to position the camera to a team chosen frame with 10 cm of accuracy, whereas the exceptional team would be able to position and orient the camera using arbitrarily located frame.

Show Me The Money: Mapping

The final step is to locate the money in metric coordinates. For this portion of the laboratory, only the Australian coins in common circulation (i.e., 5¢, 10¢, 20¢, 50¢, \$1 and \$2) need to be identified (i.e., specialty coins, notes, foreign currency, and other “clutter” do not need to be mapped).

The map should return the relative to the location and orientation of the coins relative to the origin frame (chosen in the step above). The default coin frame is such that it is located at the center of coin, with the z -axis is coming out of the “head” side coin and with the x -axis aligned the horizontal level implied by forward orientation of the head (this is approximately illustrated in the coin figure below). Note that Australian coins are of medallic orientation (like British coins) and thus rotate about the vertical (or y -axis) about the axis (unlike American coins which rotate about the horizontal (or x -axis)).



Teams may choose an alternative definition,, though they need to get approval from the teaching staff well in advance. A basic level of operation would be to be map the location of about 50% of the coin centers with 100 mm (10 cm) of error, whereas an exceptional mapping performance would be locate 90% of the coins with 10 mm (1 cm) of error.

Eureka: Golden Laboratory Tasks & Values

This laboratory consists of five focus areas in various aspects of control sensing and robotic perception. The focus areas tasks progress from basic to more challenging. Each task has various performance (and grading) levels varying from basic to advanced, with the latter receiving more points (or in the colloquialism of this laboratory more “golden coins”) as noted below:

1. Image Acquisition and Colour Calibration

Take an image of a [Gretag Macbeth Color checker](#) target find the returned value of the Neutral 6.5 (#a0a0a0 - chart #21) and the Orange Yellow (#e0a32e - chart #12) color targets in RGB, HSV and YCrCb color space (hint: see the `applycform` and `rgb2ycbcr` functions). These two colour values are those closest to the colour values of Australian coins.

- **Basic Level** (50¢ level): Identify the chart in a “[Basic Level](#)” environment and determine the RGB values for the chart values #12 and #21.
- **Intermediate Level** (\$1 level): Identify the chart in a “[Basic Level](#)” environment and determine the RGB, HSV, and YCrCb values for at least chart values #12 and #21.
- **Advanced Level** (\$2 level): Identify the chart in an “[Advanced Level](#)” environment and determine the RGB, HSV, and YCrCb values for at least chart values #12 and #21.

2. [Camera Calibration](#)

Teams should **fully** calibrate their camera (i.e., intrinsics and extrinsics). Teams are allowed to choose their pattern, but a A5 or larger **30 mm** square checkerboard pattern (the metric version of the checkerboard pattern provide in the Camera Calibration toolbox and the same format as in Tutorial 8) will be available.

- **Basic Level** (50¢ level): Calibrate the camera in a “[Basic Level](#)” environment. The teams may move the target themselves. While automatic operation is recommended, semi-manual operation is allowed.
- **Intermediate Level** (\$1 level): Fully automatic calibration of the camera in a “[Basic Level](#)” environment. Manual pattern motion (i.e., “calibration dancing”) is allowed.
- **Advanced Level** (\$2 level): Fully automatic calibration of the camera in an “[Advanced Level](#)” environment. Manual pattern motion is also allowed.
- **Super Advanced Level** (\$3 level): “Advanced Level Camera Calibration,” plus the calibration should be done without any manual intervention, thus the multiple calibration pattern views must be done automatically (through passive motion (e.g., spring, mobile, etc.) is allowed).

3. [Object Segmentation and Counting](#)

The system needs to be able to segment out the coins on the money plate so as to eventually determine the overall value of the funds in the scene. As noted, it is fine to use a dense (pixel based) solution or a more sophisticated (feature based) approach.

- **Basic Level** (75¢ level): For a “[Basic Level](#)” environment, the system will recognize (and physically enumerate or count) at least $\frac{1}{2}$ the coins (or the single coin, if there is only one) and will estimate the value of the coins to 25% of the total funds.
- **Intermediate Level** (\$1.50 level): In a “[Basic Level](#)” environment with some additional clutter added, the system is able to recognize (and count) all the coins present. It should estimate the total to within 10%.
- **Advanced Level** (\$3 level): In an “[Advanced Level](#)” environment, the system will recognize (and count) all the coins. If there are notes present, it should at least note their presence even if it can not identify and count the notes. It should estimate each of the coins to within 10% of their value.
- **Super Advanced Level** (\$4 level): Still in an “[Advanced Level](#)” environment, the system returns the exact value of coins and notes present.

4. [Locating the Camera](#)

The system should locate the camera (in metric coordinates) relative to the [central frame](#). Teams may provide a central frame target of their own choosing (i.e., using any given design or their own).

- **Eignen Level** (20¢ level): The central frame is placed at the camera origin.
- **Basic Level** (50¢ level): The central frame is placed in the workspace and if a target is used, then the target is visible by the camera. The estimated location is within 10 cm (total straight line distance) of the actual value (as surveyed by the tutors)
- **Intermediate Level** (\$1 level): The central frame is placed by the tutors to the left of the plate. The estimated location is within 10 cm of the actual value. Pose estimates are attempted.
- **Advanced Level** (\$2 level): Now in an “[Advanced Level](#)” environment, the system locate the camera automatically to an arbitrary location. The system also determines pose. The estimate is within 5 cm. The pose estimate is within 15°.

5. Mapping the Currency

The final (and arguably principal) focus area is to map (in metric coordinates) the locations of the Australian coins on the money plate relative to the central frame. In essence, this combines tasks 3 and 4 and is based on origin standards as with task 4. As noted, the default frame is at the coin center.

- **Basic Level** (75¢ level): In a “[Basic Level](#)” environment, the system will map the location of least ½ the coins with an error magnitude less than 100 mm.
- **Intermediate Level** (\$1.50 level): Still in a “[Basic Level](#)” environment, the system will map the location of 75% of the coins with an error magnitude of less than 25.4 mm.
- **Advanced Level** (\$3 level): In an “[Advanced Level](#)” environment, the system will map 85% of the coins with an error magnitude of less than 25.4 mm.
- **Super Advanced Level** (\$4 level): Again, in an “[Advanced Level](#)” environment, the system will map 90% of the coins with an error magnitude of less than 10 mm.

Custom Exceptionally Advanced Levels: For any of the tasks, it is possible to propose a custom task level. Please submit to metr4202@itee... by 29/9. Acceptance of the proposal (and an adjudicated value) will be emailed back.

Minting Grades: Assessment Criteria for Overall Lab Mark

While it is encouraged for teams to try all five focus areas, teams do **not** have to try all areas. Grades will be determined by the teaching staff based on the performance and explanation of team members as they perform the aforementioned focus area tasks. A general rule, but not absolute, mapping between points (or value) and the grades is:

Grade	Value Range (\$)	Comment
2	20¢-\$1.00	At least one task performed. For example, you are able to calibrate the Kinect in both color and metric space.
3	\$1.00-\$2.00	Very substandard performance, For example, you are able to detect the presence of target objects in one basic scene.
4	\$2.00-\$3.50	Basic level operation. For example, you are able to detect the location of at least one target object in some of the scenes with satisfactory accuracy.
5	\$3.50-\$6.00	Intermediate operation level. For example, you are able to detect the location of at least one target object in each scene with good accuracy.
6	\$6.00-\$10.00	Very good intermediate to Advance Level performance. For example, you are able to detect the location of some target objects in each scene with great accuracy.
7	\$10.00-\$15.00+	Excellent performance. Most of the tasks are attempted well. Teams are able to robustly detect the locations of objects in complex scenes with superb accuracy.

Teams and Groups

The project will be conducted in **teams of two** -- preferably taken from within your group from Laboratory 1. You may also choose an individual from another group (or laboratory session) as long as you understand that you may not be able to work together for the final project that will draw upon work completed in this one.

Other Programming Systems and Cameras

Teams may elect to use other programming languages and systems, such as Visual C or Python (i.e., the class is language / system neutral). In particular, teams may choose to use OpenCV.

Teams may choose to operate the Kinect's RGB camera in high resolution mode (i.e., they may use the 1280x1024 mode also provided by the MS Kinect SDK as compared to the 640x480 default mode provided by the OpenNI SDK). Similarly, teams may use another (web)camera on the proviso that the camera is autonomous (i.e., it can take pictures without manual intervention) and that the maximum resolution is set to (or automatically down-sampled to) 1280x1024. (n.b., the allure of high-resolution can be a trap in vision and signal processing applications as this comes with higher data processing requirements and often comes with more noise).

In both cases, however, the only supported programming system and hardware are MATLAB the the Kinect camera (using OpenNI/Primesense drivers).

External Sites/Programs

Some external programs and site that might help with the process are:

- [caltag - automatic marker detection for camera calibration](#)
- **Fiducial markers:** [reacTIVision](#), [ARToolKit](#), [AprilTags](#)
- [CLAMS -- Calibrating, localizing, and mapping, simultaneously](#)

Sample Images

A set of sample Kinect and camera images of the scenes are at:

<http://robotics.itee.uq.edu.au/~metr4202/tpl/SensingLab/sample/> .

Due Date

The laboratory must be completed by **Thursday, October 10, 2013**. (This has been extended from the original date of September 26). The code should be submitted online via UQ Blackboard (or [Cloudstor](#)) by 11:59pm October 10, 2013. [A short individual report](#) (~1-2 pages) should be submitted by 11:59pm on October 11, 2013 via the Platypus submission system. Early submission is encouraged.

Demonstration

As with laboratory one, the system will need to be demonstrated. As with Laboratory 1, there will be signup times from October 7 to October 10 (i.e., during teaching week 11). During the demonstration period, teams may choose to demonstrate the focus area tasks in **any order** they choose. For each of the tasks, teams may repeat a task once if they choose; however, the team receives the value from either not both (i.e., repeat task demonstrations to not add).

Given the number of teams, the demonstration times (of 10 minutes total including setup, leaving 5 minutes for discussion) will be strictly enforced. It is recommended that teams come 15-30 minutes in advance of their demonstration appointment. It is also recommended that teams practise their demonstrations as time limits will be enforced even if teams have not been able to demonstrate their solutions to the five tasks (i.e., teams will receive grades not on the solutions they demonstrate not the solutions they might have, but did not deliver).

Deliverables & Submissions

It is also required that all code be submitted (via UQ Blackboard, Link TBA). As the laboratory is language and camera neutral, teams that do not use Matlab should provide very clear documentation and README files with their submissions.

For the following focus area tasks, please structure your functions around this structure:

1. **Colour Calibration:** given an RGB image of a Gretag Macbeth Color checker target, return the colour value of the square that is meant to be orange yellow (#e0a32e). The returned variables should be three vectors. One in RGB space, one in HSV and one in YCrCb:
 $rgb = [1 \times 3]$, $hsv = [1 \times 3]$, $lab = [1 \times 3]$
2. **Camera Calibration:** given a sequence of RGB+D data containing checkerboard patterns from an uncalibrated Kinect (or RGB for a camera), return the intrinsic and extrinsic constants of the Kinect. The returned intrinsic properties should be in a structure like following, the square brackets indicate the expected size of each property. Note that the extrinsics has three dimensions; the third dimension has a transformation matrix for each RGB calibration input image:
 $intrinsic.fc = [1 \times 2]$, $intrinsic.cc = [1 \times 2]$, $intrinsic.alpha_c = [1 \times 1]$,
 $intrinsic.kc = [1 \times 5]$, $intrinsic.err = [1 \times 2]$, $extrinsic.transformation_matrices = [4 \times 4 \times N]$
3. **Detection:** given an image from the Kinect/camera, it must return the bounding box of the coin(s) in pixels. There should be a row for each can detected. There should be four columns to represent the bounding box. The origin is at the bottom-left of the image, with the x-axis horizontal and y-axis vertical: [*minX*, *minY*, *width*, *height*]
4. **Localisation and Mapping:** given an image from the Kinect/camera, it must return the location of the camera relative and/or coin(s) (depending on the mode) relative to a central coordinate frame in metric coordinates (mm) with orientations in (degrees). There should be a row for each coin located. There should be 3-6 columns to represent the coin's pose: [*xPosition*, *yPosition*, *zPosition*, *Roll*, *Pitch*, *Yaw*]. Orientation may be in Quaternions, etc. if specified in the header.

Judges

The course coordinator, lecturers and tutors will act as judges. The course coordinator will act as chief judge. All decisions made by the judges will be final, and no correspondence will be entered into. Contestants may approach the organiser about possible designs that may be questionable under the rules listed above. Any queries will be treated with the utmost confidentiality and will not be divulged.

Caveats

Some general “reasonable person” rules apply to the code and its execution:

- Codes with fixed (predetermined) monetary estimates are not valid (even if the value is correct).
- Internet access may or may not be present -- the code should assume that it will not have Internet access during execution and thus operate in a self-contained manner. A “Mechanical Turk” or “phone home” solution is explicitly disallowed.
- Memory space may or may not be cleared between challenges and submissions -- The memory space might be cleared before each function. Thus, if your routines rely on parameters to be exchanged, it should do so by writing to a file. Similarly, if certain variables names (e.g, counters) are used between functions, then be sure to initialize them correctly.
- Each team’s submitted functions will be run in their own directory -- Reading other teams’ files or memory is disallowed.
- All source code(s) may be assessed -- Thus, it is requested that it is commented. If custom precompiled codes are used (e.g., mex files), the source code should also be submitted.
- Computational and memory resources -- the functions should be able to operate reasonably on a “standard” UQ EAIT Workstation. Judges may terminate execution after 2 minutes.

Go METR 4202
E Pluribus Unum!!!