**Making Camera Ready for Visual Perception**

**Introduction**

**What is Visual Perception?**

Finding an object in an image can be done using object detection but to get the location of the object in real world, we need to transform the image coordinates to 3d world coordinates. It can be done by various methods. In this blog I have discussed a method that requires no expensive hardware. In this method a reference pattern is used like a chessboard pattern. Using this method first we need to get camera parameters, both intrinsic matrix and later on the extrinsic matrix. With the help of these we achieve the coordinate transformation.

**Problem Statement**

* Misalignment—
  + The problem of manual transformation is that it is rigid and there is room for flexibility. As long as the object detection model is able to find the key or the button, slight misalignments and placements won’t impact the process.
* Time Consuming and manual Labor intensive —
  + Earlier, since the process was interface dependent, the robotic arm configuration needs to be changed manually. Now we just have to feed images and train the model with new interface and voila it is done.
* Distortions –
  + Camera calibration also reduces the lens distortions such as barrel and pincushion effect significantly which helps in better object detection and hence better accuracy.

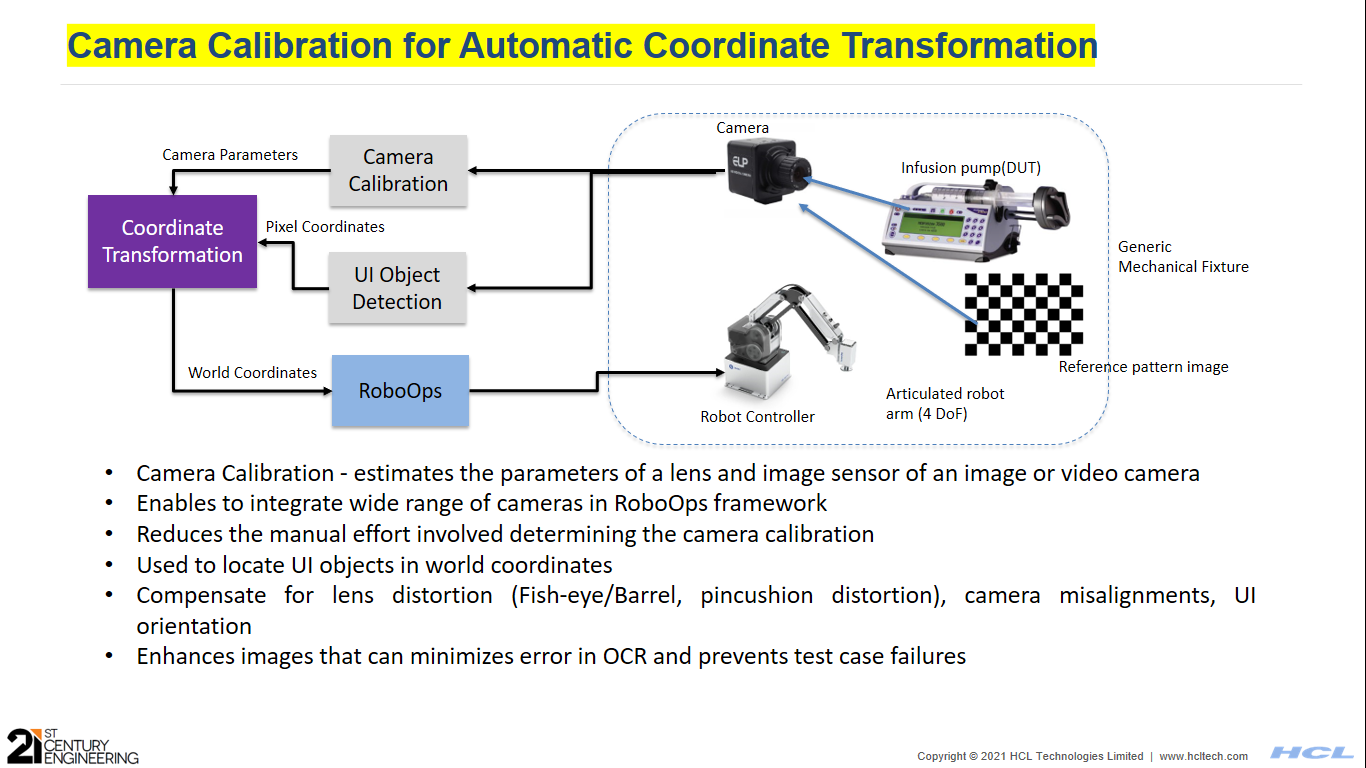


Figure Flow Diagram

**Solution**

Camera Intrinsic Parameters –

Camera intrinsic matrix is a transformation matrix that transforms camera coordinate system to pixel coordinate system. It depends on the internal factors such as focal length, field of view, aperture, etc.

Extrinsic Parameter –

Camera Extrinsic matrix is a transformation matrix that transforms world coordinate system to camera coordinate system. It is a combination of rotation matrix and translation matrix. It depends on the location and orientation of the camera.

Extrinsic parameters are the parameters that are dependent on external factors such as camera’s position and orientation.

World Coordinate

Camera Extrinsic Matrix

Camera Intrinsic Matrix

Image coordinate

Known values

Unknown values

With the help of object detection, we get the image coordinates of the object, i.e. T where u and v are center coordinates of the object in the image. It can be calculated using a reference image like a chess board pattern and taking its images.

After solving the above equation, we get the T . Calculated XYZ values are then sent to Inverse Kinetic function. This IK function takes XYZ as input and returns robotic arm motion parameters to place the stylus to its desired location.

OpenCV Camera Calibration –

This library helps to find the distortions, intrinsic and extrinsic parameter of a camera. Using these parameters, we can also undistort these images. Using multiple images increases the accuracy and it is better to keep the chess board pattern onto the same plane where our actual object is going to be.

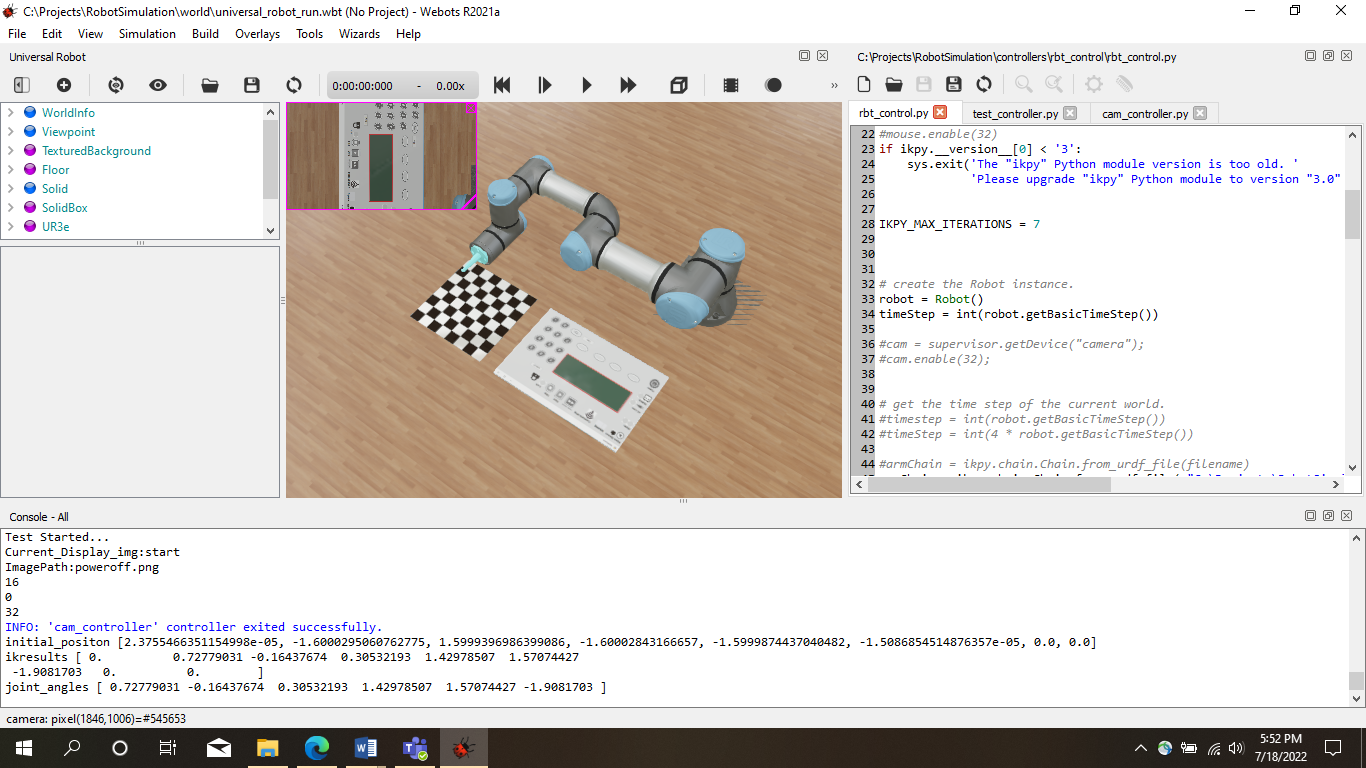
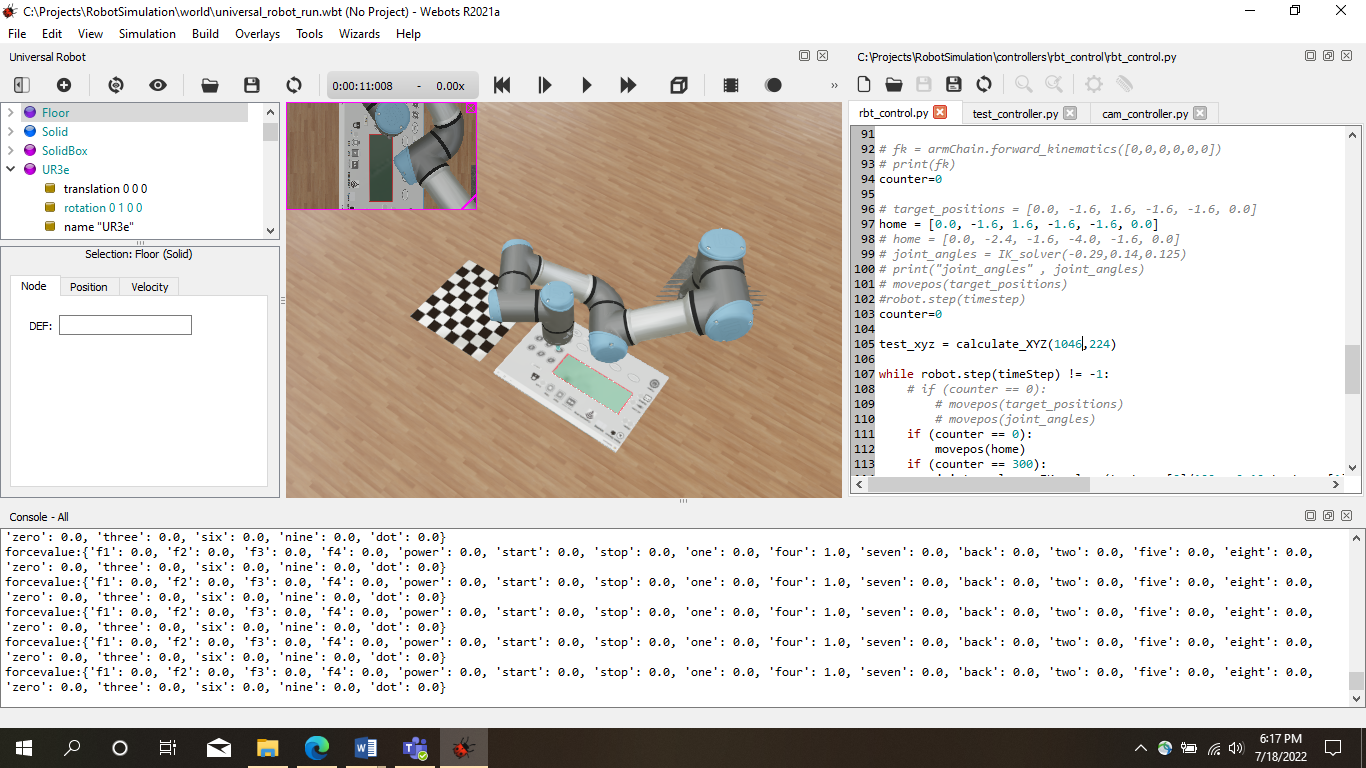
Onetime operation –

Figure 2 When Robot touches the button

Figure 1 Initial Position

Initially the robot remains at the initial position and camera is present at a fixed distance from the base. From there it takes several images of the reference pattern at different position and orientation. This helps in calculating the intrinsic parameters. We also get rvec (Rotation Vector) and tvec (Translation Vector) which will be used to generate extrinsic matrix. After this, we need to provide some real world coordinates with their corresponding image coordinates to calculate the scaling factor (S, see in the equation). We have used an OpenCV library, camera calibration, to find the required matrices. After we get all our required variables next step is to calculate the world coordinates. These are the onetime effort. After this all the steps, from object detection to pressing the button is automated.

**Reference Pattern 🡪 Camera Calibration🡪 Intrinsic & Extrinsic Matrix 🡪 XYZ Calculation 🡪 IK Function 🡪 Robotic arm movement**

**Key Benefits**

* Time is reduced significantly
* Less prone to human error
* Estimates the parameters of a lens and image sensor of an image or video camera
* Enables to integrate wide range of cameras in RoboOps framework
* Reduces the manual effort involved determining the camera calibration
* Used to locate UI objects in world coordinates
* Compensate for lens distortion (Fish-eye/Barrel, pincushion distortion), camera misalignments, UI orientation
* Enhances images that can minimizes error in OCR and prevents test case failures

**Overheads**

* Depth data is not very accurate
* Onetime effort to get all the matrices and images
* Captured image quality and training data might impact the accuracy
* Training for different UI will be needed, but this will be a onetime effort for each UI

**Conclusion**

This method of coordinate transformation helps to achieve good results without much expensive hardware and is also easy to set up. This method deals with the problems associated with the earlier method such as manually giving the robotic arm movement instructions. This method finds its use in device testing such as display units, smart phones and other electronic gadgets.

**Author Info**

**App – pick and place, engine inspection**

**Author: Sourabh Prakash <**[**mailto:sourabh.prakash@hcl.com**](mailto:sourabh.prakash@hcl.com)**>**