**ME 192 Lab 3**

**Robot Vision and Location Transformtion**

9/24/14

**OBJECTIVE**

Learn the basics of Adept Technology’s V+ robot vision including initialization of a vision reference frame, acquiring camera images, selecting the objects meeting the criteria, and performing linear and angular motion transformation for robot pick up operation. No actual robot pick up and drop off operation is performed in this lab.

**Part A – Vision program example**

Load the V+ program PID1S and look up the code for a quick start on V+ vision programming. This program takes a picture of laterally arranged 4-digit 1/4”x1/2” bar codes in the picture frame, reads the codes using scan lines, and outputs the value, 0 to 15. The code is used for pallet tracking on a conveyor.

**Part B – Vision tests for pick up locations**

*a. Download the Adept’s vision commands onto your PC/Mac for details in parameter setting.*

*b. Set up a vision inspection on either a SCARA or a six axis robot.*

Define a location **grip.loc** and move the gripper to the location. Place two objects (pegs) – one round and one square - under the camera such that they appear approximately in the middle of the frame screen. If a robot is not attached, move the objects under the fixed camera position.

Execute **CALL VSETUP (grip.loc, camref )** to (1) initiate the vision parameters, (2) retrieve camera offset data, (3) set the object size range, and (4) to establish a frame reference point.

**VSETUP** sets **camref = grip.loc:TRANS( , , -V[3], , , -V[4]):to.cam** where V[i]’s is a vector location obtained by **DECOMPOSE**ing the location **grip.loc,** **to.cam** is the camera offset data, V[3] = the gripper’s Z position, and V[4] = the relative roll angle at the gripper flange.

*Given θ1= Shoulder angle stored in V[1], θ2 = Elbow angle in V[2], θ3 = Roll angle in V[4], Gripper angle relative to the Base X axis is, g = θ1 + θ2 +(180 - θ3)= V[1]+V[2] + (180-V[4]).*

*b. Define a rectangular area of interest smaller than 50mm x 80 mm.*

 **VDEF.AOI 2000 = 3, \_\_, \_\_, \_\_, \_\_, 0** ; Shape 3 for the window: x0, y0, dx, dy & rotation angle

**VDISPLAY 1, 1** ; Display pictures in grey scale before binary conversion.

*c. Acquire a picture image and gather data*

**VPICTURE , 2 ;** Take a shot and retain two objects in the pixel range.

 **VWAIT** ; Wait ‘til image acquisition is completed.

**VWINDOW 2000 ;** Display and process only the area of interest to speed up.

*d. Perform two morphology operations - dilation once and erosion once - on the image to sharpen the object* boundaries. *The result is affected by the lighting condition and the camera iris setting. The ambient light is the best source of lighting as it does not cast shadow of an object. Diffused incandescent lighting or high frequency fluorescent lighting also work. Adjust the camera iris (aperture) and change the threshold value for converting grey scale pixel images into binary images.*

**VMORPH (, 2) = 2000**

**VMORPH (, 1) = 2000**

*d*. *Retrieve the location data of the objects (obj.loc) in the picture frame, one at a time. Use a loop.*

**VLOCATE ( ) obj.loc**

If there is no more left (**VFEATURE[1]= FALSE**), then **GOTO** *Step h* to finish.

If one is found, extract its location data.

Centroid: **x = DX(*obj.loc*), y = DY(obj.loc)**

Radius: **rad =** **VFEATURE(47)** ;The maximum edge distance from the centroid.

Angle : **rz =** **VFEATURE(45)** ;Angle of the maximum edge distance in camera frame.

Prior to data extraction, thresholding (grey to B&W conversion) and morphology operation (dilation and erosion) may be performed to enhance the object image. You may bypass this step on the first trial. You may also skip the Window function on the first trial.

*e. Perform a radial distance test on the object.*

Measure the distance to four edges using **VRULERI** at 90° increments starting at (**rz** + 45°).

Check if any of the radial distances (**ruler[2]**) is greater than 80% of **rad**.

If yes, the object is likely a circle. Go to *Step f* for a circularity test.

If not, go to *Step g* for a linearity test on a square. On a square, **ruler[2]** $≈ $**rad**/1.414.

*f. Perform a circularity test with* ***VFIND.ARC*** *- Binary correlation between a circular edge and a circle.*

Fit a circular edge within a ring around a guide circle at ± 25% of its radius.

If **ARC[0] = TRUE** and **FIT ARC[5] > 90**, then the peg hole is round with a 90+% fit.

**TYPE** the (x, y) location on the monitor and return to *Step d* to retrieve the next object data.

*g. Perform a linearity test with* ***VFIND.LINE*** *- Binary correlation between a straight edge and a line.*

Calculate the center point of one of the four edges that should exist if the object is a square.

Find an edge line along a guide line placed in the middle of a rectangular window. The guide line must incorporate the tilt angle of the square object in the camera frame.

If **LINE[0] = TRUE** and **FIT LINE[5]>90**, then the peg hole is square with a 90+% fit.

**TYPE** the (x, y) location and the edge angle on screen and return to *Step d* for the next object.

*h. Display any other pertinent data for a hand-off to a robot picking operation.*

RULER(2)

VFEATURE(47)

***Radial Edge Detection***

 ***Circle vs. Square***

RULER(2)

VFEATURE(45)

**Circularity Test**

**Linearity Test**

Ring Radii = ± 25% of the circle radius

Window = 1 x ½ of the guide line length

**Part C – Repeat tests for drop-off locations**

Repeat Part B with a peg holder block for a drop off operation. The threshold value for binary pixel image conversion may need to be adjusted as the block is greyish metal in color. Make certain that the bottoms of the two recessed areas are lined with a black felt or paper and there are no deep shadows on the walls. A grey surface image is fine as it will be blackened by a proper selection of a binary threshold value.

**Part D – Location transformation analysis - Gripper, camera offset, object locations**

This part requires a robot activated along with its controller. The following is for the case in which the camera is attached to the Z axis on the robot arm. If the camera is mounted on a fixed stand, the transformation matrices will be different. Ask the instructor for help.

a) Using the ROTJOINT program, extract the transformation matrices for the following:

1. Gripper position (P0) = - At the reference location.
2. Object location (P1) = - In robot frame. Obtained by manually moving the arm to object.
3. Camera location (p2) = - With camera calibration data. Otherwise by actual measuring.
4. Object location (p3) = - In the camera frame. As found in the preceding exercises

b) Check the transformations to see they work out as expected.  or .

Explain any discrepancies found. Strip the Z values and/or roll angles if necessary using the DECOMPOSE command and redefining the positions.

**Report –** Due 10/8/14

Include a cover sheet, project description, team activity description, the program code, a screen shot of the output, transformation matrices and calculations, and a team activity log sheet.

**Appendix For Lab 3**

.PROGRAM pid1s()

 x0 = 61

 y0 = 23

 r0 = 27

 dx = 10.5

 dy = 13

;

; Define the area of interest - VDEF.A;rectangle, x1, y1, dx, dy, rotation

 VDEF.AOI 2000 = 3, 52, 2, 95, 40, 50

 VDISPLAY (1) 1, 1

 VPICTURE (1) 2, 1 ; For quick window frame grabbing. Find one object

 VWINDOW 2000

 ii = 16

 id1 = 0

;

 FOR i = 1 TO 4

 ii = ii/2 ; ID is a multiplier of 2

 VRULERI (1) ruler[] = 1, x0+(i-1)\*dx, y0+(i-1)\*dy, r0, 140

 TYPE "edge count ", i, ruler[0]

 IF (ruler[0] > 1) AND (ruler[0] < 5) THEN ; Edge count should be 4.

 id1 = id1+ii

 END

 END

 TYPE "pallet id from Camera 1 =", id1

 RETURN

.END

**Lab 4 Exercise for robot pick-up and drop-off operation**

*This part is for reference only. It will be used when actual pick up and drop off operation is done using a manipulator later in Lab 4.*

Proceed to have the robot pick up the pegs and place them in the matching holes in the peg holder block.

**SET P1 = camref: TRANS(x, y, z, , , g)** ; pick up location

where (**x, y**) is the centroid, **z** the pickup height, and **g** the gripper angle. **z** and **g** can be calculated or taught. To set the desired gripper height and the angle, use **PAUSE** or **PROMPT** command during the program execution. To set the gripper angle for the square peg, set **g** = **RZ**(**rz**) ± 45°. For the round peg, **g** may be any value. To accommodate the case in which the two objects are close together, pick up the round object first at an angle perpendicular to the line passing the centers of the two objects.

**Vision guided pick up routine**

**To strip the Z and the gripper angle from the offset gripper location**

MOVE grip.loc ; Move to the offset gripper location

BREAK

WAIT.EVENT , 1 ; Wait until the arm motion stops

HERE #grip.loc ; Precision location

DECOMPOSE jt[1] = #grip.loc ; Decompose into joint values

SET ref.loc = grip.loc : TRANS( , , -JT[3], , ,-JT[4]) ; Strip the Z and the angle value

**To incorporate the XY offset values and the roll angle into the gripper location:**

SET xy.loc = ref.loc : to.cam : obj.loc ; xy.loc = the absolute XY object coordinates.

; to.cam = XY camera offset

; obj.loc = XY coord. of the object on the frame

SET xyr.loc = xy.loc : RZ(rz ± 45) ; Append the desired grip angle in the robot frame.

 ; RZ(rz ± 45) = JT[1] + JT[2] + 180 –(rz ± 45)

; rz = Diagonal angle of the square (round) peg

**To incorporate the picking height into the pick location.**

TYPE “Set the desired grip height and enter PRO to continue”

PAUSE ; Set the grip height and type PRO as instructed.

HERE grip.Z ; To extract the gripper Z joint value.

DECOMPOSE jt[1] = grip.z ; Decompose into joint values

SET P1 = xyr.loc : TRANS( , , jt[3]) ; Incorporate the taught Z height into the pickup location.