1 Specification of the problem

We want to use a range finder mounted on an IBM 7575 - SCARA robot to obtain a 3D model of a real object. The object is placed on a turntable and the robot arm is positioned so that the turntable is close enough within the range finder's field of view. Then a set of scans are taken at different points of view around the object and the scanned surfaces are registered and merged in a single 3D model.

One of the key points here is the registration of the multiple meshes obtained by scanning the object from different points of view. In order to do this, we have to transform all the data points in the multiple scans to be with respect to a single coordinate system.

Taking multiple view point scans is achieved, for robustness, by rotating the turntable instead of moving the robot arm around the object. The robot arm moves the range finder vertically from the top to the bottom of the object, then the turntable is rotated, exposing another part of the object's surface, and another scan is taken. This process is repeated until the entire surface of the object (except for the top and the bottom parts) is scanned.

Thus, the registration problem is to find the line, about which the rotation of the turntable takes place. The scanned data we obtain is with respect to the range finder's coordinate system. We have to find the axes of rotation with respect to the same coordinate system. This would allow us to rotate the scanned data back to its original position (the position at which the scanning process started) and thus obtain a registered set of scans.

We use 3 coordinate systems: one that is associated with the range finder and the robot arm, one that is associated with the turntable, and an additional world coordinate system (figure 1).

The range finder's coordinate system is only two dimensional: it has an x-axis, along which the laser beam scans, and an y-axis, which is depth, or distance to the object. Since we need a 3D coordinate system we introduce the z-axis to be paralel to robot arm that holds the range finder. The resulting coordinate system is not necessarily orthogonal, but the experiments so far show that the error is negligibly small.

The turntable's coordinate system has its origin at the center of the turntable. We assume that the range finder's scanning plane is parallel to the turntable's surface and introduce x and y axes to be parallel respectively to the x and y axes of the range finder's coordinate system. Since as the turntable rotates, the associated coordinated system rotates also with respect to the range finder, we need to specify that the above construction holds when the turntable is in its home position only (0 degrees of rotation).

Finally, we introduce a world coordinate system, which is the same as the turntable's coordinate system, when the turntable is in its home position. The difference here is that the world coordinate system is not associated with the turntable and remains the same even when the turntable rotates.

Having introduced the above coordinate systems, the calibration problem can be reformulated as finding the transformation between the turntable's coordinate system and the range finder's coordinate system.

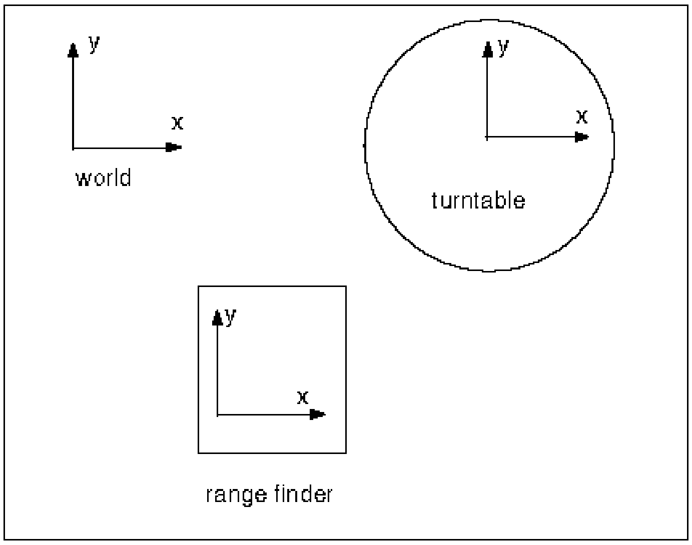

Figure 1: *Coordinate systems (z-axes point upwards)*

2 Solution

We use a calibration cube of known size c. The calibration data consists of only four scan lines of the cube, taken at 0, 90, 180, and 270 degrees of rotation of the turntable. A typical calibration data is shown on figure 2.

Figure 3 shows how a typical scan line looks like. As one can clearly see, the useful information in such a single scan line is in fact half of a square. This means that we can hope to recover accurately only the location of the two lines that are formed by intersecting the scanning plane of the range finder and the cube.

In general, this intersection should be an arbitrary triangle. Our experiments, however, show that the angle between the range finder's scanning plane and the turntable's plane is negligibly small. Thus we can assume that the data we get from a single scan line (after filtering out the backgroud) should be a rectangular triangle with its vertical lines being of size c.

The next step in the calibration process is to filter out the background data and split the image into two line segments. We apply a median filter to both line segments, in order to remove the outliers, and then fit lines to the resulting data. Having obtained the analytical representation of the two lines

$$\begin{aligned} L_{left}: &\quad a_l x + b_l y + c_l = 0 \\ L_{right}: &\quad a_r x + b_r y + c_r = 0 \end{aligned}$$

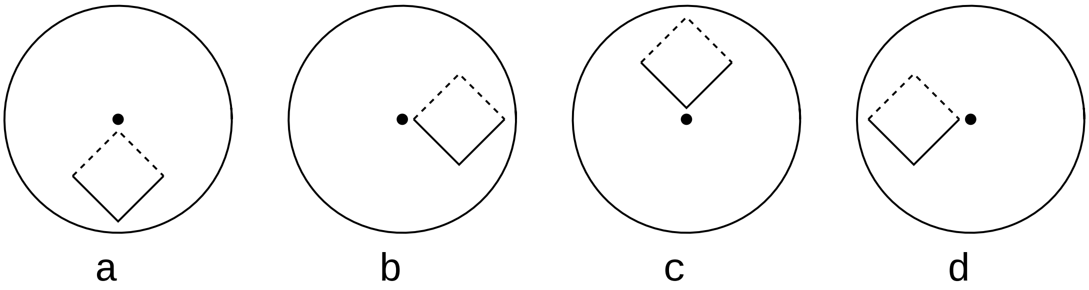

Figure 2: *The four calibration scans (the dashed lines are not visible to the range finder): a) at 0 degrees; b) at 90 degrees; c) at 180 degrees; d) at 270 degrees.*

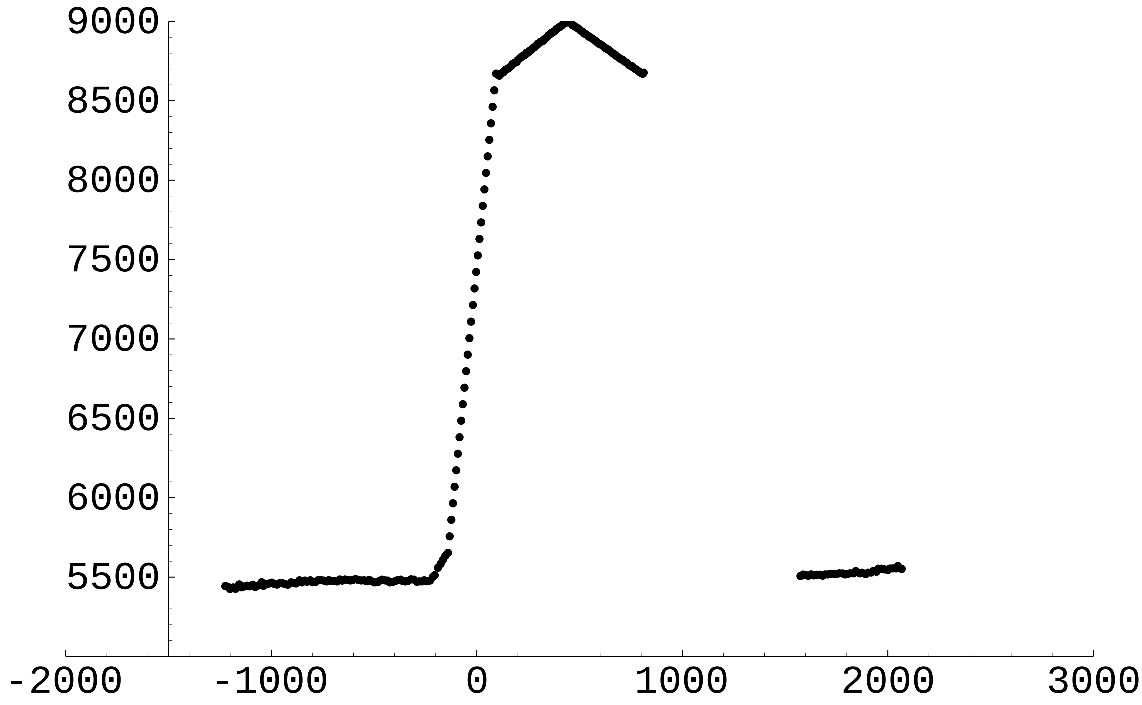

Figure 3: *A typical scan line of a cube.*

we solve the linear system

$$\left| \begin{array}{l} a_l x + b_l y + c_l = 0 \\ a_r x + b_r y + c_r = 0 \end{array} \right.$$

for x and y and obtain the coordinates of the intersection point of these lines. This is in fact the intersection point of the visible vertical edge of the cube and the scanning plane.

We apply the method above to find the edges P_0, P_{90}, P_{180}, and P_{270} of all 4 scans. In addition to that, we calculate the normal vectors along the two lines in the first scan (the one at 0 deg):

$$\begin{aligned} \vec{v}_{left} = & \quad (-b_l, a_l, 0) \\ \vec{v}_{right} = & \quad (b_r, -a_r, 0) \end{aligned}$$

Knowing the two normal vectors above, the coordinates of P_0 and the size of the cube, we can restore the whole intersection of the cube and the scanning plane for this scan:

$$\begin{aligned} P_{left} &= P_0 + c\,\vec{v}_{left} \\ P_{right} &= P_0 + c\,\vec{v}_{right} \\ P_{back} &= P_{left} + c\,\vec{v}_{right} \end{aligned}$$

Now we can rotate the intersection points P_{90}, P_{180}, and P_{270} along the center (x_c, y_c) of the turntable back to their home positions. Ideally they should coincide their calculated origins:

$$\begin{aligned} P_{left} &\equiv R(x_c, y_c, -90)P_{90} \\ P_{back} &\equiv R(x_c, y_c, -180)P_{180} \\ P_{right} &\equiv R(x_c, y_c, -270)P_{270} \end{aligned} \quad ,$$

where

$$R(x_c, y_c, \phi) = \begin{bmatrix} cos\,\phi & -sin\,\phi & x_c\,(1 - cos\,\phi) + y_c\,sin\,\phi \\ sin\,\phi & cos\,\phi & y_c\,(1 - cos\,\phi) - x_c\,sin\,\phi \\ 0 & 0 & 1 \end{bmatrix}.$$

Here, $R(x_c,\ y_c,\ \phi)$ is a matrix that performs rotation by ϕ degrees about $(x_c,\ y_c)$. Its dimensions are 3x3 because we use homogenious coordinates. It can be derived as follows:

$$\left| \begin{array}{l} x\prime = (x - x_c)\,cos\,\phi - (y - y_c)\,sin\,\phi + x_c \\ y\prime = (x - x_c)\,sin\,\phi + (y - y_c)\,cos\,\phi + y_c \end{array} \right.$$

is the transformation that rotates a point $(x,\ y)$ by an angle ϕ about the point (x_c, y_c). Here, the expressions $(x - x_c)$ and $(y - y_c)$ translate the center of the coordinate system to $(x_c,\ y_c)$, the sum/difference of the products with $sin\,\phi$ and $cos\,\phi$ performs the rotation, and the addition of x_c and y_c brings us back to the original coordinate system. This transformation can be rewritten as:

$$\left| \begin{array}{l} x\prime = x\,cos\,\phi - y\,sin\,\phi + x_c - x_c\,cos\,\phi + y_c\,sin\,\phi \\ y\prime = x\,sin\,\phi + y\,cos\,\phi + y_c - x_c\,sin\,\phi - y_c\,cos\,\phi \end{array} \right. .$$

Thus

$$\begin{bmatrix} x\prime \\ y\prime \end{bmatrix} = \begin{bmatrix} cos\,\phi & -sin\,\phi \\ sin\,\phi & cos\,\phi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} x_c(1 - cos\,\phi) + y_c sin\,\phi \\ y_c(1 - cos\,\phi) - x_c sin\,\phi \end{bmatrix},$$

which, translated to homogenious coordinates, looks like:

$$\begin{bmatrix} x\prime \\ y\prime \\ 1 \end{bmatrix} = \begin{bmatrix} cos\,\phi & -sin\,\phi & x_c(1 - cos\,\phi) + y_c sin\,\phi \\ sin\,\phi & cos\,\phi & y_c(1 - cos\,\phi) - x_c sin\,\phi \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

In real world, however, no measuring device is ideal. The measured data we have is not exact. Thus, because of the error introduced by the range finder, when we rotate the points P_{90}, P_{180}, and P_{270} back, they will be close to, but not coincident with P_{left}, P_{back}, and P_{right}.

The typical way out of this situation is to construct and minimize an energy function. This function in our case is the sum of the squared distances:

$$\begin{aligned} E(x_c, y_c) = \quad & dist(R(x_c, y_c, -90), P_{left})^2 + \\ & dist(R(x_c, y_c, -180), P_{back})^2 + \\ & dist(R(x_c, y_c, -270), P_{right})^2 \end{aligned}$$

where

$$dist(X, Y) = \|\overrightarrow{XY}\|$$

Minimazing this function, we obtain a close approximation of the coordinates of the center of the turntable.

As we mentioned earlier, experimental results show that the range finder's scanning plane is almost parallel to the turntables surface. Put in another words, the angle between the two planes in negligibly small and the z-axeses of all coordinate systems are parallel. Thus, the resulting transformation matrix between the turntable's coordinate system and the range finder's coordinate system is:

$$T_L^{TT} = \begin{bmatrix} 1 & 0 & 0 & -x_c \\ 0 & 1 & 0 & -y_c \\ 0 & 0 & 1 & 150 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Note: The offset 150 for the z-coordinate is peculiar for the range finder. It is used to move the scanned data closer to the center of the coordinate system.

Now, having a data point $P_L = [x_L, y_L, z_L, 1]^T$ taken by the range finder, we can find its coordinates $P_{TT} = [x_{TT}, y_{TT}, z_{TT}, 1]^T$ with respect to the turntable's coordinate system:

$$P_{TT} = T_L^{TT}\, P_L.$$

Knowing the rotational angles ψ_i of the turntable, at which each scan of a set of scans was taken, we can rotate the scans back and thus obtain a registered set of scans:

$$P_W = R(-\psi_i)\, P_{TT} = R(-\psi_i)\, T_L^{TT}\, P_L,$$

where P_W are the coordinates of the point P_L with respect to the world coordinate system, and

$$R(\psi_i) = \begin{bmatrix} cos\,\psi_i & -sin\,\psi_i & 0 & 0 \\ sin\,\psi_i & cos\,\psi_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

3 The program

The calibration process involves running two programs: *calscan* and *Calibration* or, alternatively, *calscan2* and *Calibration2*, which are the second enhanced version of the calibration procedure.

The first one, *calscan*, is used to acquire the data necessary for the calibration. It is a Unix shell script, requires no input, and produces raw data scan files.

The second program is a Mathematica script to compute the transform that is needed. On completion, it will save in the current directory a file *turntbl.trn* which contains the calculated transformation matrix.

Note: For security and statistical reasons, the previous transformation matrices are not deleted. They are kept at the end of the file. Each time a new matrix is calculated, the program includes it at the begining of the file and puts the date and the time of the calibration immediately behind it. Thus, at any given time, the matrix occupying the first 4 lines is the most recent one.

The complete detailed instructions how to run the program are available in *html* format and can be found in *http://www.cs.columbia.edu/ atanas/Calibration/instructions.html.*