

GE U111 HTT&TL, Lab 5

Ultrasound Imaging of a Hidden 3D Object

Vector & Matrix Analysis, Elaborate Looping & 3D Plotting in MATLAB

Contents

1	Overview: Programming & Experiment Goals	2
2	Homework Assignments	2
3	MATLAB Primer: 3D Plots	3
4	The Experiment	5
5	Challenge Extensions	7

1 Overview: Programming & Experiment Goals

This Lab concludes the first cycle of experiments and the MATLAB component of the course. The main engineering problem solving objective is to use the underwater ultrasound scanner to reconstruct and later plot a three dimensional (3D) picture of a distant, hidden object. This Lab builds on the preparatory work of Labs 3 and 4, including the program governing raster motion and ultrasound distance measurement in water. In relation to programming, the Lab's primary learning goal is to bring the ability to use concepts and tools learned in previous sessions at a higher level of complexity.

(MATLAB) programming concepts and continue reinforcing previously used tools, including

- Advanced looping
- Advanced logical branching
- user defined functions
- Analysis of matrix arrays
- 3d plots of functions of two variables

Basic (Engineering / Science) concepts, including

- 3D shape reconstruction
- Heuristic error correction in image processing

The Components & Equipment used in this experiment are as in Lab 4:

- The planar positioner.
- The glass aquarium.
- The underwater 1 MHz ultrasound transducer, serving as both an emitter and a receiver, attached to the positioner.
- A digital oscilloscope.

Experiments Outline & Goal. A target 3D object will be attached, parallel to the front wall of the aquarium, opposite the ultrasound transducer. The object is placed inside a box, covered with opaque, black gelatin. Our goal is to be able to recover and plot the 3D shape of this object. This will be done by using the ultrasound transducer to measure the distance from an 11×11 array of equally spaced points, covering a $10 \text{ cm} \times 10 \text{ cm}$ square in the 2D plane in which the transducer is moving, to a corresponding grid of points on the face of the shape, as schematically depicted in Figure 2. The maximal measured distance is the distance from the transducer to the wall. Subtracting all other distances from it, the distance of the various points on the face of the object, from the wall, will be computed. Since sound propagates in water and in gelatin in the same speed, distance measurement using ultrasound echoes will ignore the gelatine in the box. In other words, the gelatin is transparent to our ultrasound imaging device. The transducer will cover its domain in raster motion, using the program from Lab 3. At each grid point, the transducer will be used to measure the distance to the object, using the function `distmeasure` from Lab 4. The measured distance will be stored as a corresponding entry in an 11×11 matrix. A 3D plot of the object will be obtained, using a 3D plot command, and rotated to several view-points.

2 Homework Assignments

- Pre-Lab Homework
 - Read Chapter 9 in the text on 3D plotting
 - Read this handout carefully
 - The Pre-Lab tasks in this handout.
- Post-Lab report due one week after lab is completed.

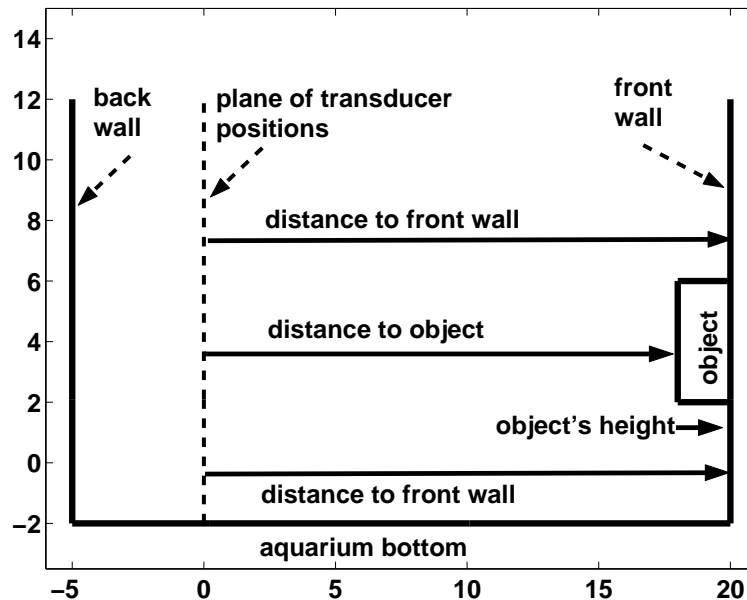


Figure 1: A Schematic depiction of the ultrasound imaging experiment. The transducer moves in a fixed plane. The distance to the remote obstacle facing it is measured from an array of transducer positions. Since (it is given that) the hidden object is flat, the measured distance may take one of two values: The distance to the wall or to the face of the object. The height of the object is the difference of the two.

3 MATLAB Primer: 3D Plots

Here you will learn how to plot the graphs of functions of two variables: $z=f(x,y)$, using the commands `mesh`, `surf` and `waterfall`. You will also learn and practice the use of the `view` command, to change the viewpoint of a displayed three dimensional plot. The immediate motivation for plotting such functions, in this session, is were the function $z=f(x,y)$ describes the height of the top face of an object, in the point placed directly above the (x,y) base coordinates, as depicted schematically in Figure 2.

1. The `mesh` command. The data needed to plot the graph of a function of two variables $z=f(x,y)$ consist of a finite set of values of the function, provided in terms of:
 - (a) Two one-dimensional arrays of samples of the x and y variables.
 - (b) A matrix of corresponding values of z , such that $z(m,k)=f(x(k),y(m))$.

The command `mesh(x,y,z)` plots the mesh defined by the points (vertices) whose coordinates are $(x(k),y(m),z(m,k))$. The plot is colored; the default color code is provided by the value of z , where red indicates the maximum value and blue indicates the minimum value. There are options to modify this color code, which are described in MATLAB reference books and using the MATLAB command `help colormap`.

2. The `surf` command. The purpose of the `surf` command and the rules of using it, are essentially the same as with the `mesh` command. The only difference is in the fact that the entire surface obtained as the graph is now colored, as opposed to only the mesh lines, in a mesh plot.
3. The `waterfall` command. It is the same as `mesh`, except that the column lines of the mesh are not drawn - thus producing a “waterfall” plot.
4. The `view` command. The `view(hangle,vangle)` command is used to set the angles of the viewpoint from which an observer sees the 3D plot. Here `hangle` is the horizontal rotation angle and `vangle` is the vertical

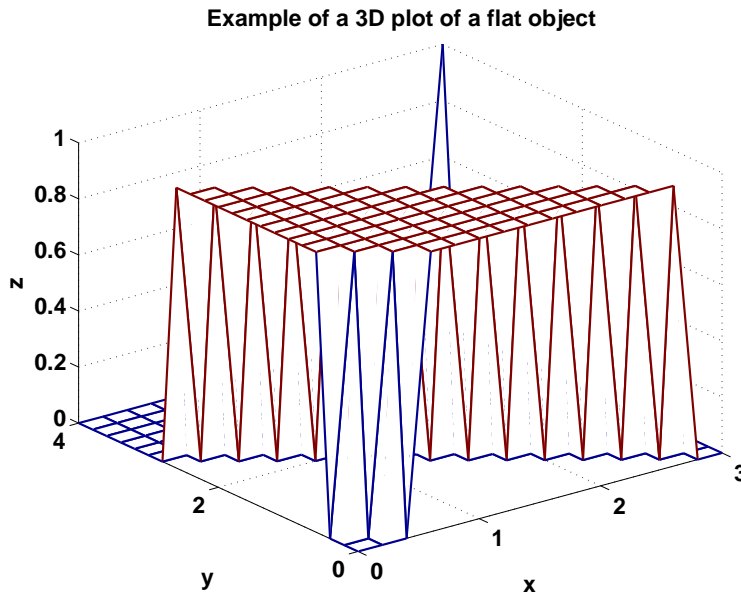


Figure 2: An example of a 3D plot Using the mesh command.

elevation angle (both in degrees). Thus `hangle` revolves about the z -axis, with positive values indicating counter-clockwise rotation of the viewpoint. Positive values of the elevation angle, `vangle`, correspond to moving above the object; and negative values move below. Here are some examples -

- (a) `hangle = -37.5, vangle = 30` is the default 3-D view.
- (b) `hangle = 0, vangle = 90` is directly overhead and the default 2D view.
- (c) `hangle = vangle = 0` is a horizontal view along the x axis.
- (d) `hangle = 180` provide a view behind the matrix.
- (e) `view(2)` sets the default 2D view, `hangle = 0, vangle = 90`.
- (f) `view(3)` sets the default 3D view, `hangle = -37.5, vangle = 30`.

The command `[hangle, vangle] = view` returns the values of `hangle` and of `vangle` for the current view point.

5. Manual Rotation. It is also possible to rotate the 3D plot using by clicking on the rotation icon, then bringing the pointer to a point on the plot, pressing and dragging the left mouse button. (Try!)
6. Labels & Title: The commands `xlabel`, `ylabel` and `title`, are used here in complete similarity to their use in two dimensional graphs. A label for the vertical axis, is provided by the command `zlabel('text')`.

Notice: It is important to pay attention to the order of the indices of entries of z (which may appear counterintuitive): the first index is the index of the y entry and the second is the index of the x entry. In particular, if `length(x)=K` and `length(y)=M`, then z is an $M \times K$ matrix.

Pre-Lab Task 0 Here you have to define and plot a grid of values of the function $z=10x\sin(2y)$ over the range of $0 \leq x \leq -1, 0 \leq y \leq 3$. You will use a relatively coarse grid in order to clearly see the mesh. Specifically, you have to do the following:

- (a) Define a vector consisting of 11, equally spaced values of $0 \leq x \leq -1$. (What should be the increment if `x(1)=-1` and `x(11)=0`?)
- (b) Define a vector consisting of 31 equally spaced values of $0 \leq y \leq 3$.

- (c) Define a 31×11 matrix z , with entries $z(m,k) = 10 * x(k) * \sin(2 * y(m))$.
- (d) Use the mesh command to plot the graph of z as a function of x and y .
- (e) Provide labels for the three axes and a title, in your previous plot, type your name on it, and include a printout in your pre-lab report. (Black and white prints are acceptable.)
- (f) To obtain a first-hand experience with the use of the view command, you will apply it to your existing plot.
 1. To visualize the effect of the parameter handle, apply each of the commands `view(0,30)`, `view(30,30)`, `view(60,30)`, `view(90,30)`, `view(120,30)`, `view(330,30)`. Observe the impact of each of these commands on the graph and briefly describe your overall observation in your lab report. Print out and include in the report prints of 3 of the 12 views (indicate the view angles on each print). Repeat the same type of experiment, now varying the value of `view` in increments of 30 degrees over the 0-330 degrees range, but keeping `handle=-37`.
- (g) Use the `surf` and the `waterfall` commands to plot the graph. Include axis labels, titles and your name and print a copy of each plot for your report. (Again, black and white prints are acceptable.)

4 The Experiment

Pre-Lab Task 1. Modify the raster motion program from Lab 3 as follows: The `fprintf` display command, in the original program, should be substituted by a set of two commands -

- (a) Call the `distmeasure` function, to measure the distance to the hidden body.
- (b) When the positioner is at the location $(x \text{ cm}, y \text{ cm})$, $0 \leq x, y \leq 10$, store the measured distance in the $(y+1, x+1)$ entry¹ of an 11×11 matrix named `dist`. That is, the entries in the matrix will be the distances to the object or the wall from the corresponding position of the transducer. (The reason we do not use `dist(y,x)` is that indices of entries in a matrix cannot take the value 0.)
- (c) Important: Accepting the old wisdom “Never despair of disaster”, following each update to the matrix `dist`, save the new value of the entire matrix in a data file named `dist.mat`, on your memory stick, using the command

```
save F:\dist.mat dist;
```

This way, even if the experiment stops midway, due to a hardware glitch, you will have the already collected data saved, and will be able to start the scan from the point of failure, and not from scratch. The significance of this is due to the fact that running this program will be VERY, lengthy due to the slow motion of the scanner, and the time needed to transmit data to the computer, in our lab setting. A commercial counterpart will operate on far shorter timescales, due to efficient configuration of the various system components.

Save your program as a script M-file named `scan3D.m`. Include a printout of the program in your Pre-Lab report and in the diskette you bring to the lab.

Lab Experiment 1. This is the main experiment.

- (a) Initiate the aquarium system (with the command `[dio,scope]=setup_tank`
- (b) Debug and execute the program `scan3D.m`.

Following the execution of the scan, we shall post-process the data and plot the reconstructed image.

Post-Lab Task 2. Preliminary post processing & plotting

¹The convention in matrix notation is to identify the first index of an entry with the row number and the second index with the column number. Viewing the matrix is part of the positive quadrant in the plane, the row number is a function of the y (vertical) coordinate and the column number is a function of the x (horizontal coordinate).

1. Load the data file `dist.mat`
2. To compute the distance from points on the face of the object to the wall, you have to subtract the distance from the transducer plane to the face of the object, from the distance from the transducer plane to the wall. These values will be stored in a new array named `height`

```
height=max(max(dist))-dist;
```

The entry `height(k,1)` is the measured height of the object from the wall, at the point whose coordinate values (in cm) are `x=l+1` and `y=k+1`.

Explain in your report why we need to take the `max` twice. Consult the book and / or the MATLAB help `max` if you need a refreshed definition of `max`.

3. Save the matrix `height` in a data file named `height.mat` in your disquette.
4. Plot the values of `height` versus the corresponding values of the `x` coordinate of the measured points:

```
mesh(0:1:10, 0:1:10,height)
xlabel('x coordinate of a point on the object face')
ylabel('y coordinate of a point on the object face')
zlabel('a raw plot of height of that point')
title('a two dimensional scan')
```

5. Use the `view` command (use the help manu if necessary) to observe the plotted object from several view points, and print those additional plots.
6. Type the names of your team members and include the printed plot in your lab report.

Post-Lab Task 3. It often is the case that your plot will include distortions. There may be several reasons for such distortions:

- (a) As you can see in the oscilloscope, the amplitude of the returned echo rise gradually (albeit quite rapidly) over its first several periods. A slightly conservative (i.e., high value of the threshold `amp_th`) may result with occasionally missing at least the first cycle or two of the returned echo. The duration of each period of the 1 MHz sound wave is 10^{-6} sec. Therefore the distance the sound traverses during each missed cycle is

$$\text{wave length} = \text{time period} \times \text{speed of sound} \approx 10^{-6} \text{ sec} \times 1482 \text{ m/sec} \approx 1.5 \text{ mm}$$
- (b) When the transducer faces a point near the edge of the object, part of the returned wave will be from the rear wall and part from the object face. We can therefore expect detection that is delayed relative to echoes from the object and advanced relative to an echo from the face of the object.
- (c) By design (in Lab 4), if no returned echo is detected, the function `distmeasure` returns the value of `distance=0`. In that case, the corresponding entry in `dist` will be zero and the value of the corresponding entry of `height` will be the distance from the transducer to the wall. Finding this value, which is not possible in the actual object, is an ambiguous indication that no real measurement was obtained at that point.
- (d) A (rare) burst of high amplitude noise, exceeding the value of `amp_th`, will cause a premature detection of a return echo. The corresponding entry of `dist` will be too small, and the corresponding entry of `height` will be too large than the actual value.

You therefore face two challenges:

- (i) The first challenge is to identify clearly erroneous values. A clue here is that, with a flat object, entries of height may take only one of two values: 0 and the actual height of the object. The maximal magnitude of small deviations due to small detection errors can be assessed by inspection of the entries of `height`. Using this [a priori](#) information, Entries of `height` whose values within are the admissible range can be safely be replaced by the mean value. Entries with larger deviations should be rejected.
- (ii) The second challenge is to find a method to replace rejected values by reasonable alternatives. Such alternatives may be based, e.g., on averaging, minimizing or maximizing the values of legitimate values in neighboring entries, continuing detected straight line edges, etc..

The last task is an “open ended one”, to write a post processing program that will create a corrected array, `c_height`, and plot it as a corrected picture of the hidden object.

5 Challenge Extensions

The ultrasound transducers as we have been using them produce an unfocused, parallel beam with a diameter equal to the 1.34 inch diameter of the transducer. This limits how accurately you can measure the shape of the block. You may have observed that if your transducer is right at edge of the target, you see two returned pulses, the first from the part of the ultrasound beam which hits the front of the target and the second returned pulse from the half of the beam which missed the target and reflects off the back of the tray. Can you think of a technique to use this double reflection to get a better image of the edges of the target?

If you finish imaging the hidden object and have time left, ask the instructor or TA for an acoustic lens to add to your transducer. A lens is any optical device which converts a parallel beam into a converging or diverging wave front. The acoustic lenses that are available are converging lenses with a focal length of 15.2 cm (=6 inches). This means that if your target is 15.2 cm in front of the transducer the parallel beam will converge to a point on the order of the size of the wavelength. (**bf Question:** ¿From the speed of sound in water that you have measured, calculate the wavelength of the 1 MHz ultrasound waves that are emitted from the transducer.)

The acoustic lenses need to be mounted on the transducer with no air layer in between the lens and the transducer to avoid greatly reducing the signal amplitude. You will be provided with some water-soluble coupling gel to use. Put a substantial amount of coupling gel on the transducer and then rotate the lens onto the transducer to try to minimize any air bubbles between the transducer and the lens.

You will also need to adjust the distance between the transducer and the front of the target so that the target is at the focal point of the acoustic lense. Ask for a screw driver to move the x-y positioner so that the transducer is 15.2 cm (6”) from the front of the target which is about 1/4” underneath the surface of the gelatin. You can adjust the distance to get the maximum reflected signal from the surface of the object – that will be when the ultrasound beam is focused at the object surface. Try to make sure that the positioner is square on the aquarium so that the front of the target will remain at the focal distance when the positioner moves.

With the lens you can do the following imaging tasks (or any other tasks you would like to try) as time allows:

1. Write a program to use the transducer with the lens to map out the edge of the target object to much better resolution than you were able to get with the unfocused transducer. You can discuss with your instructor or TA how you think you can do this. The key point is that the reflected beam should come back much later (and probably not as large) when you move the positioner so that the beam is not hitting the front of the target.
2. Some of the targets have a symbol milled 1/8” deep into the surface. The size of the symbols are about 1/2” square. (Calculate how much time delay you would expect to get if your beam reflected off of the bottom of a shape 1/8” deep.) Design and execute a program to image the symbol on the front of your target.