Localization of an AUV in an Absolute Reference Frame using a Small Scanning Sonar

ABSTRACT
This paper describes a method for implementing a localization system for the autonomous underwater vehicle SubjuGator, under development at the University of Florida's Machine Intelligence Laboratory, that will utilize a small scanning sonar augmented by SubjuGator's navigation system. Autonomous underwater vehicles (AUVs) typically will utilize relative positioning systems that are subject to significant errors over time [1] [3] [6]. Such errors can affect many AUV mission objectives such as mapping, searching in a predefined pattern, or any other mission where accurate absolute positions are required in subsea operations [6]. The objective of this system is to mitigate these errors by correcting the vehicle pose based upon the position of static objects in the operating environment. These objects will be observed using a miniature low cost scanning sonar. Other techniques at pose error correction almost always involve external assistance from either stationary undersea base stations or ships using GPS [7]. Results show that this sonar is adequate to correct long term position errors. Significant improvements could be realized with a larger test pool or environment as the one used created difficulty due to size, shape, and acoustic characteristics.

OBJECTIVES
The objective of this experiment was to determine the feasibility and methodology of using a small, low cost, scanning sonar to correct pose error over time in the Subjugators world model. The world model is the submarine's picture of the world around it and is essential for accurate navigation, obstacle avoidance, and mapping [1]. Many features of the software architecture that will not be discussed here are predicated upon the validity of this world model, and by association, the pose data on which it is constructed. Therefore, the validity of the world model over time is essential as is some predictability in the associated error variance.

The method and objectives discussed here have been changed slightly from those stated in the white paper. The white paper outlines a two part approach. First, apply the correct coordinate transformations, pre-algorithm signal conditioning, and projection onto a globally referenced tessellated grid space. This objective has not changed. The second half of the proposed work has been altered. A real time system for simultaneous localization and mapping by analyzing complete pictures generated by an entire scan was proposed. After experimentation, it was evident that waiting for complete scan information before generating new error information, was impractical, especially after utilizing constraints of the known environments in which the Subjugator operates. Information about target recognition and tracking was also omitted due to the volume of work it presented.

APPROACH
In order to post process the information, the sonar data must first be combined with the pose data which is captured separately for this experiment. Since the pose data is output at a rate of six to eight Hertz and the sonar outputs each return at a rate around 40-50 Hertz, an algorithm to combine the sonar data with the latest estimated pose information was developed. This algorithm essentially placed a zero order hold on the latest pose output and combined it with subsequent sonar returns until a new pose data set was available. As part of this process, several coordinate transformations are calculated and stored with each data record. These transformations are called the submarine aligned sonar head angle and the Earth aligned sonar head angle. The submarine aligned sonar head angle is the angle of the scanning beam in the submarines own coordinate frame [4]. The Earth aligned head angle is the scanning beam's

Localization of an AUV in an Absolute Reference Frame using a Small Scanning Sonar

angular relationship to the Earth's coordinate frame, which is referenced to magnetic north using the Subjugators onboard compass unit magnetometers [3]. The Earth aligned sonar head angle is used to place a fixed reference frame upon the sonar head's constantly translating frame (assuming the submarine is moving). It is essential to the algorithm that the compass heading is accurate and it is assumed that it will have a finite error that is not changing over time.

Once the data sets are fused, signal conditioning and then object detection are preformed. For this discussion the term "ray scan" will refer to one single return, which consists of 252 intensity values at each range bin for the 4.5 degree beam pointed at a certain Earth aligned sonar head angle. The term "scan" will refer to a complete 360-degree revolution of the sonar head which contains 80 "ray scans." First, as each ray scan is processed, a zero phase distortion digital FIR filter is used to remove high frequency noise. Zero phase distortion is achieved by processing the data forwards and then backwards through the same filter, resulting in a filter that has the square of the original magnitude response but virtually no phase distortion [5]. The fact that the filter exhibits zero phase delay is essential to preserving range information in each return waveform.

Once smoothed, the amplitude information can be examined for objects. However, before the waveform is suitable for lobe analysis, constraints of the Subjugator's known operating environments can be used to mitigate the chance for false positives due to reflections from pool walls and other acoustic phenomenon that arise when operating in man-made structures. These constraints must be determined on a case-by-case basis, but in general there are several parameters that must be determined empirically for each environment. First, the gain of the sonar must be tuned in order to achieve an optimal contrast without receiving too much return from either shallow operating areas or unnatural structure such as a corner of a pool (where acoustic ringing can cause higher amplitude returns). Second, range gates can be set up to remove high amplitude reflections caused by the large surface areas of pool walls, which can result in the appearance of echoes and interpreted as objects outside of the operating area. Finally, a correction for an anomaly in the sensor hardware is also applied in the range between zero to one meter. This anomaly was found to be a consistent amplitude time series signal that is present in all returns. It is removed by a simple subtraction of a stored time series from the ray scan. Once these corrections are applied, the probability of false object detection is greatly diminished.

Object detection is preformed through a simple thresholding process. First the amplitude time series is iterated until the threshold value is met. Once met, an object is said to be found and the time (range) value associated with the amplitude is stored along with the width of the amplitude lobe. The width of the lobe is defined as the distance between the first occurrence of the threshold point and the next. In this way the algorithm provides a level of protection from detecting one return as multiple objects clustered at close range. However, this technique is highly dependent upon proper smoothing of the waveform, as high frequency energy can significantly increase the probability of multiple threshold crossings

After each ray scan is processed for objects, this information can then be applied to a grid space such as the one in figure 1. The Subjugator's navigation system derives such a grid space from its pose over time, and is tessellated at a one by one decimeter resolution. For this discussion, the grid space will be finite in extent at 10 by 10 meters; this is arbitrary and is used here only to simplify the example. The grid is always referenced to the global coordinate frame, using the

Localization of an AUV in an Absolute Reference Frame using a Small Scanning Sonar

magnetic heading measurement, and is again predicated on an accurate measurement of current vehicle heading. The objects are placed into the grid based upon their absolute estimated position which is calculated using the perceived position of the vehicle, object range, and object bearing. The perceived x and y coordinates, of the object, relative to the vehicle are then corrected to be relative to the origin of the grid. These coordinates could be either arbitrary or possibly use a globally located position determined at the beginning of the mission using GPS on the surface. A complete 360-degree scan, once processed, will provide an accurate map of the operating area in all directions. Over time and the accumulation of scans, error due to the inherent capabilities of the navigation system will build up and shift the position of the x and y coordinate frame. However, since the magnetic compass heading contains only some small finite error, the shift will only occur linearly in the x and y directions. This ensures that there is no significant rotation between subsequent grids and will allow for a simpler and more robust error estimation technique than would have otherwise been required.

The method described to this point, although developed in post processing, could just as easily been implemented in its given form as a real-time processing engine on the vehicle without modification. To continue this discussion, an iterative process is needed to correct vehicle pose over time. First, upon the start of a mission, the Subjugator will acquire the surrounding areas static characteristics by station keeping at the (zero, zero) grid coordinate and completing multiple scans. These scans will then be compiled into a rasterized intensity image of the grid space, shown in figure 2, where each grid cell corresponds to the appropriate pixel. In this way several returns located in the same grid space will provide a larger intensity value in the rasterized image. This image will then serve as a baseline position reference for later comparison. Once the mission is underway, a sliding window of scans can then generate new images of the grid space and error in pose can be calculated using a two-dimensional cross-correlation to the baseline image. The two-dimensional cross-correlation can be calculated from (1), where f is the baseline image and t is the template, or in this implementation, the shifted image to be correlated.

$$c(u,v) = \sum_{x,y} f(x,y)t(x-u, y-v) \quad (1)$$

The two-dimensional cross-correlation is a standard technique used in many feature detection algorithms [8][9] and is ideal here because of the linear shift in x and y without any rotation in the image coordinate frame. The peak in correlation is used to calculate the error from the (zero, zero) point in the grid. A mesh plot of a correlation calculation is shown in figure 3. Since the error in the navigation system output over very short time periods and distances is negligible and considering the estimated ability of the sonar to accurately detect objects to within approximately 1 decimeter, it is reasonable to expect this system to correct vehicle pose to within 1 meter over long distances. Finally, in a real-time system an estimation technique such as a Kalman filter would be required to then integrate this error back into the navigation system to ensure correct positioning over time.

RESULTS

Post processing in the same manner that a real time system might operate was conducted in order assess the merits of the described mythology and algorithms. From this analysis, several results and discussions are warranted. First, it is apparent from experimentation that it will in fact be

possible, under constrained conditions such as the vehicles speed, magnetic heading accuracy, and environment, to use the small low cost sonar. This being said, the vehicle speed, and to a greater extent the magnetic heading accuracy, were found to induce significant error into the calculation. A plot of the Euclidean distance of the shift is shown in figure 4. Spikes in the error occur as seen in this plot. An example of a template image that generates such a spike is shown in figure 5. The error is generated when the tap delay of the FIR filter on the compass unit causes the magnetic heading feedback to exhibit delay, which in turn causes the Earth aligned sonar head angle to have delay as well during a high rate of turn. This phenomenon is apparent in the figure as a rotation of the pool walls and is essentially an incorrect mapping of the object(s) to the coordinate frame of the earth. This is a breakdown of the assumption made that the magnetic heading will always exhibit some finite constant error and is not necessarily a collapse of the algorithm as presented. Second, empirical evidence suggests that performance will be directly affected by the amount of structure in the range of the sonar. In situations where only parallel pool walls were in range, such as figure 6, the cross-correlation to the original baseline image generates large errors. Therefore, it will always be optimal if curvature, corners, or individual static objects are present in the baseline image and corresponding template image to be correlated. Finally, the accuracy of this algorithm is difficult to determine with any precision due to the inability to constrain the position of the vehicle to an absolute reference such as the GPS system. Underwater positioning systems accurate enough to provide a good comparison are complex and beyond the scope of the Subjugator's limited resources. Qualitatively, the algorithm appears adequate and to perform as expected. However, a quantitative analysis of the error data is not feasible with the current instrumentation.

CONCLUSIONS

The methods presented in this paper are an excellent first step to providing a real-time pose error correction system for the Subjugator vehicle. The results, as originally hypothesized, indeed show that a significant error in pose is generated over time and distance traveled by the vehicle. Although the error correction algorithm is only valid under significantly constrained conditions, there is reason to believe that future integration of the IMU and magnetic heading compass unit could significantly alleviate the main source of this error.

A more significant evaluation of the algorithm is needed. A GPS unit could be installed into the Subjugator for absolute position references on the surface [7]. The experiment then needs to be performed in a large body of water where large distances can be covered and significant errors accumulated. GPS positions could be taken at both the start and end points of the mission, thereby providing a control signal with which to reference the error calculation. This experiment would be a good indication of long distance performance, but would still be only a loose metric of performance in real time.

This is a first attempt at basic SLAM (simultaneous localization and mapping) for the Subjugator vehicle. That stated, there will be significant future work necessary to make this feasible for real-time implementation. First, an error model must be developed that allows the fusion of calculated error with the navigation system's estimated pose. Second, two dimensional cross-correlation is very computationally intensive as an N^2 operation. An actual grid space could be on the order of hundreds of meters square and could become unrealistic to calculate in real time using the current implementation. Several other techniques such as gradient descent, the SSDA algorithm, and transform domain techniques might be used to mitigate this factor [9]. Finally, in

Localization of an AUV in an Absolute Reference Frame using a Small Scanning Sonar

addition to using the sonar for localization, a long term goal of the project is to use the sonar for object or target tracking.

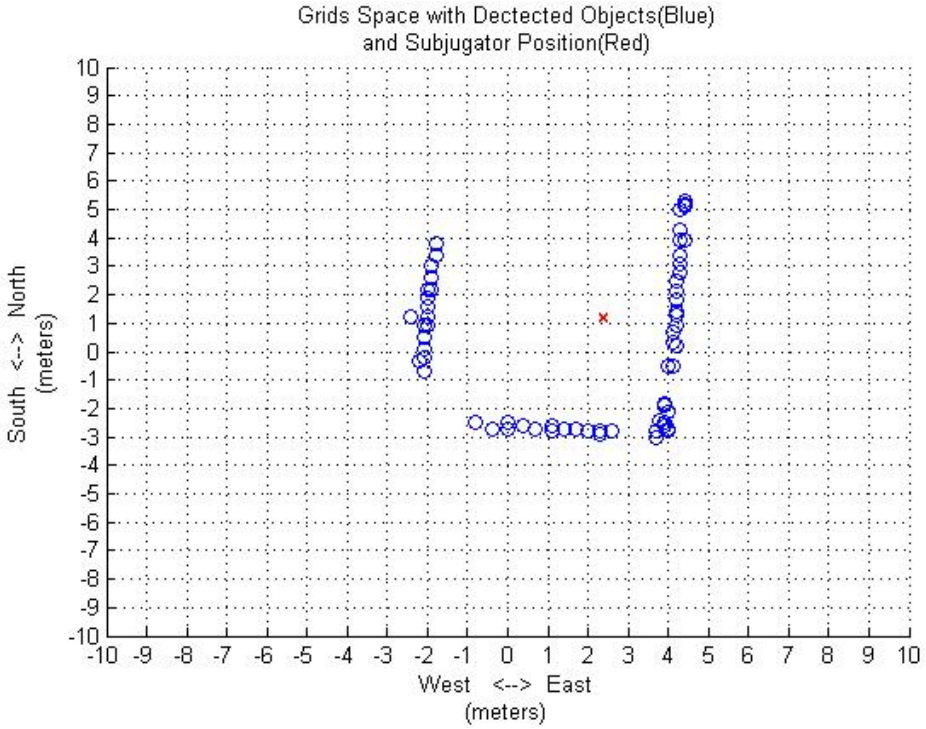

Figure 1

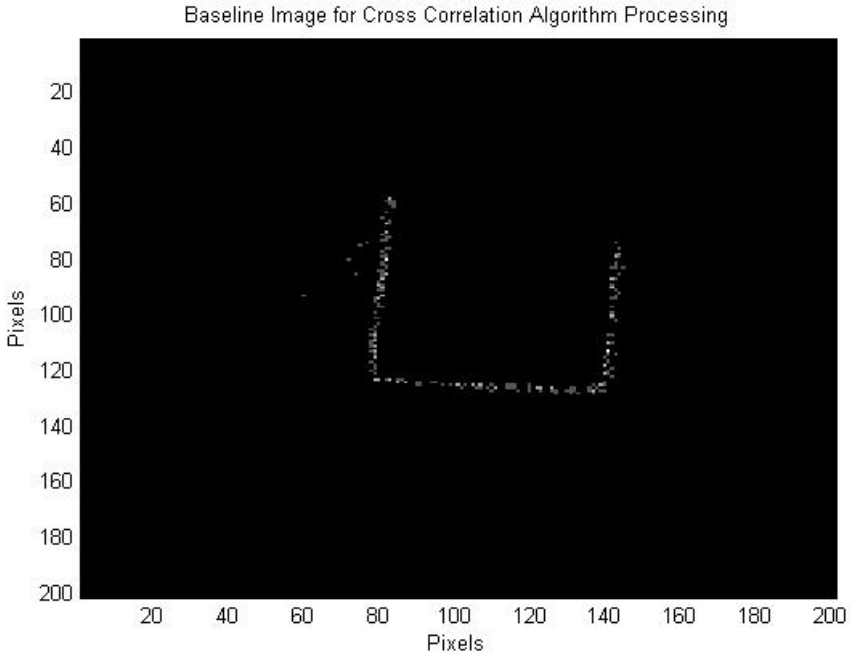

Figure 2

Localization of an AUV in an Absolute Reference Frame using a Small Scanning Sonar

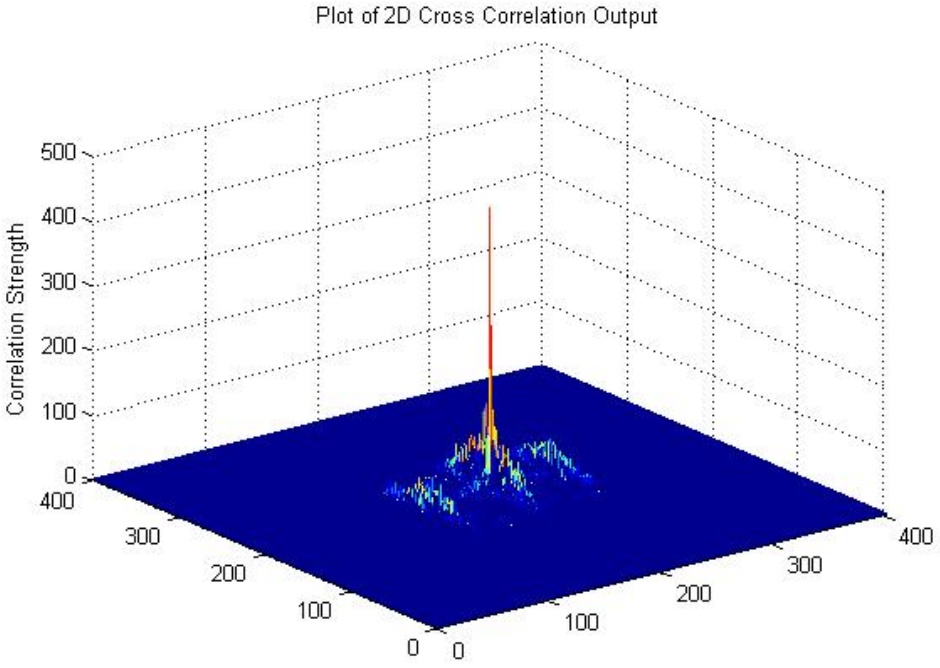

Figure 3

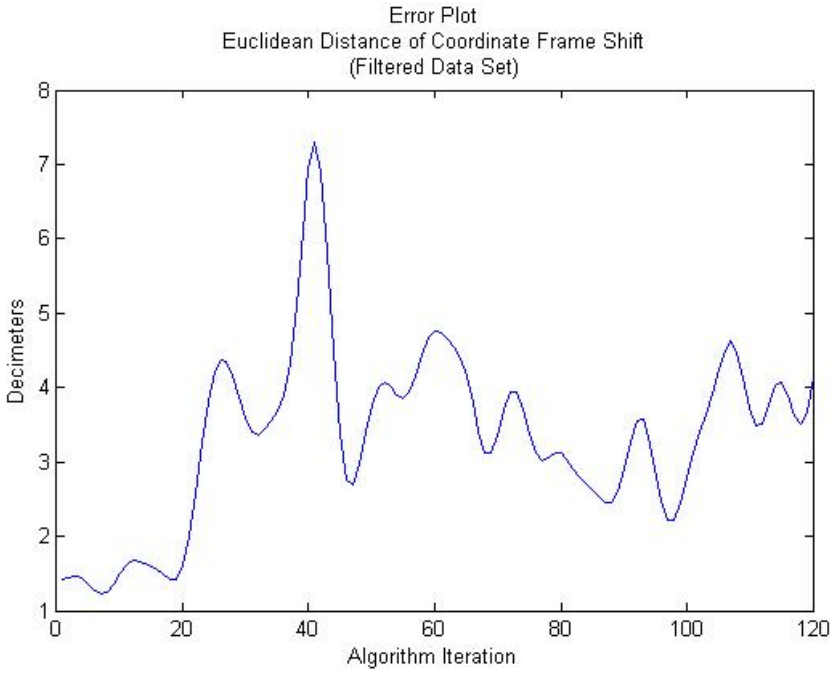

Figure 4

Localization of an AUV in an Absolute Reference Frame using a Small Scanning Sonar

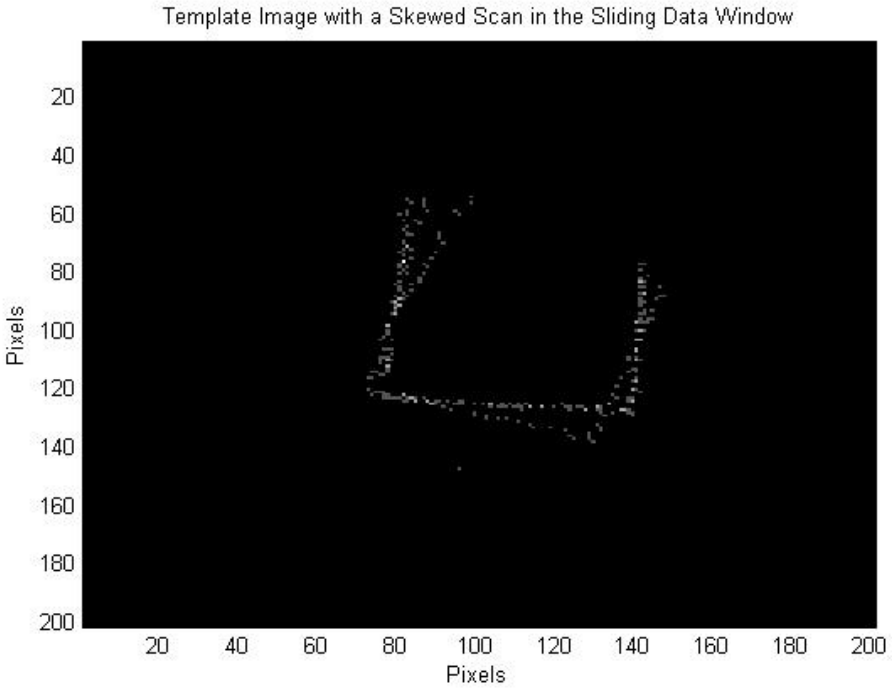

Figure 5

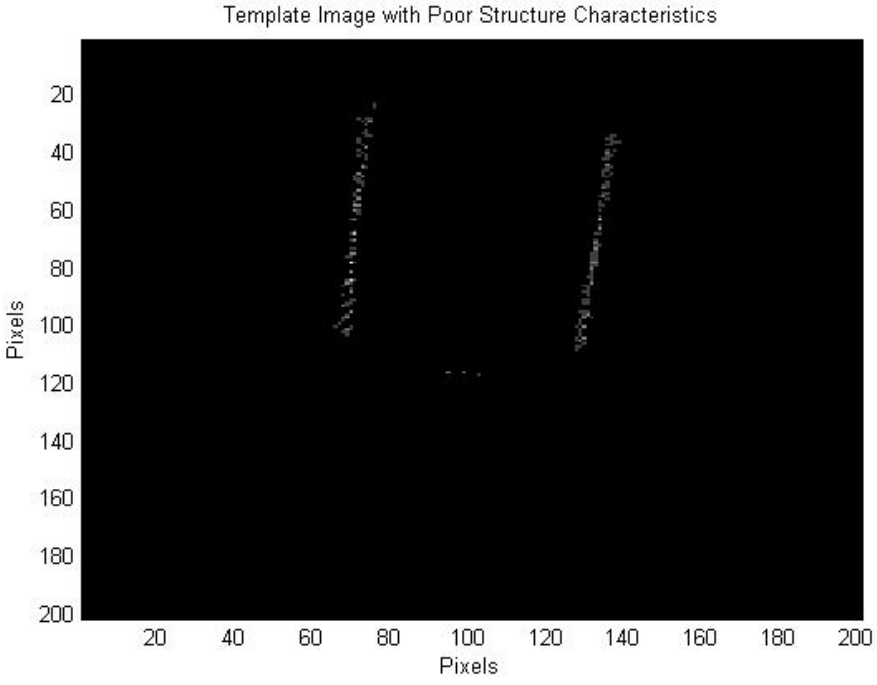

Figure 6

Localization of an AUV in an Absolute Reference Frame using a Small Scanning Sonar

REFERENCES

[1] Arom Hwang, Woojae Seong, Hang S Choi and Kyu-Yeul, “Concurrent Mapping and Localization Applied to SNUUV I” OCEANS '04. MTS/IEEE TECHNO-OCEAN '04

[2] Jenhwa Guo; Sheng-Wen Cheng; Te-Chih Liu, "AUV obstacle avoidance and navigation using image sequences of a sector scanning sonar," *Underwater Technology, 1998. Proceedings of the 1998 International Symposium on* , vol., no., pp.223-227, 15-17 Apr 1998

[3] Kevin Claycomb, et. al. , “SubjuGator 2007”, AUVSI and ONR's 10th International Autonomous Underwater Vehicle Competition, July 11-15, 2007.

[4] Imagenex Technology Corporation, “Sonar Theory and Applications”, http://www.imagenex.com/sonar_theory.pdf

[5] Oppenheim, A.V., and R.W. Schafer, *Discrete-Time Signal Processing,* Prentice-Hall, 1989, pp.311-312

[6] George Kantor, Nathaniel Fairfield, Dominic Jonak and David Wettergreen, “Experiments in Navigation and Mapping with a Hovering AUV”

[7] Whitcomb, L.L., "Underwater robotics: out of the research laboratory and into the field," *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on* , vol.1, no., pp.709-716 vol.1, 2000

[8] John Richards, Xiuping Jia, *Remote Sensing Digital Image Analysis An Introduction*, Fourth Edition, Heidelberg: Springer.

[9] J. P. Lewis. *Fast normalized cross-correlation.* In Vision Interface, 1995. http://citeseer.ist.psu.edu/lewis95fast.html

Localization of an AUV in an Absolute Reference Frame using a Small Scanning Sonar

MATLAB CODE

```
%--parsesonar.m
clear;
clc;
close all;
scanmode = true;  %true for normal, false for fast
load('sonarcm2','sonarcm')
rangeGateEnable = false;
rangeGate = 15; %meters
count = 1;
load('baseline.mat');
N = 1;
dataimage = zeros(201,201);
save('dataimage.mat','dataimage');
data_file = fopen('forwardback.852','r');
n = 1;
load('ring.mat');
numScans = 1;
Num = fir1(21,.5);
prevMode = 3;
ringdata = 0;
sonarReturn = zeros(252,1);
Ranges = zeros(252,10);
Sizes = zeros(252,10);
load('ScansSet','Scans');
scancount = 2000;
sonarDataArray = zeros(5,252);
numerror = 1;
numimages = 1;

while(1)

data = fread(data_file,384);
Gain = data(39);

minute = data(21:34)-48;

min = minute(4)*10 + minute(5);

second = minute(7)*10 + minute(8) + minute(11)/10 + minute(12)/100;


HeadAngle = Scans(scancount).Headangle;
if(HeadAngle <= 0)

    alignedHeadAngle = abs(HeadAngle);

else

    alignedHeadAngle = (180-HeadAngle) + 180;

end

Scans(scancount).AlignedHeadAngle = alignedHeadAngle;
```

Localization of an AUV in an Absolute Reference Frame using a Small Scanning Sonar

```
Scans(scancount).EarthAlignedHeadAngle =
mod((Scans(scancount).AlignedHeadAngle + Scans(scancount).Pose.Heading),360);

Scans(scancount).EarthAlignedHeadAngle;

Range = Scans(scancount).RangeSetting;
Mode = bitand((uint8(data(38))),1);

if (prevMode ~= Mode)
    if(Mode == 0) %normal mode

            set(gcf,'colormap',sonarcm);
            C = ones(252,160);
            r = (0:251)'/251;
            theta = pi*(linspace(-79.5,79.5,160))/79.5;
            X = r*cos(theta);
            Y = r*sin(theta);
    else
        set(gcf,'colormap',sonarcm);
        C = ones(252,80);
        r = (0:251)'/251;
        theta = pi*(linspace(-39.5,39.5,80))/39.5;
        X = r*cos(theta);
        Y = r*sin(theta);
    end
end
prevMode = Mode;



sonarReturn = Scans(scancount).ReturnData;
sonarDataArray(numScans,:) = sonarReturn;



distanc = 5/252:(5/252):5;

[objectt Ranges Sizes] =
findobjects(filtfilt(fir1(21,.5),1,Scans(scancount).ReturnData),75);

Scans(scancount).EarthAlignedHeadAngle;
Scans(scancount).Ranges = Ranges;
updategrid(nav2cart(Scans(scancount).EarthAlignedHeadAngle)
,Ranges,Scans(scancount));




FSonar = filtfilt(Num,1,sonarReturn);
```

Localization of an AUV in an Absolute Reference Frame using a Small Scanning Sonar

```
if (Mode == 0) %normal


    if(HeadAngle > 0)

        if((HeadAngle/2.25) > 39)
            n = mod((HeadAngle/2.25 + 121),160);
        else
            n = (HeadAngle/2.25 + 121);
        end

    elseif (HeadAngle < 0)

     n = 121-(abs(HeadAngle)/2.25);

    else

        n = 121;

    end

    C(:,n) = FSonar(:,1);
    %figure(1);
    %subplot(1,2,1);

 %   surf(X,Y,C,'EdgeColor','none');
 %   caxis([0 175]);
 %   caxis('manual');
   % colorbar;
   % view(0,90);


        if count == 160

%           figure(2);
%           [x1,y1] = meshgrid(-1:.01:1, -1:.01:1);
%           z = griddata(X,Y,C, x1, y1,'linear');
%           img = uint8(imrotate(z,180));
%           imagesc(img);
%           %sfilt(img);
%           pause;
           count = 1;
        else

            count = count+1;
        end



else
```

Localization of an AUV in an Absolute Reference Frame using a Small Scanning Sonar

```
    temphead = nav2cart(Scans(scancount).EarthAlignedHeadAngle);

    if((temphead >= 0) && (temphead < 180))

        temphead = temphead + 180;
        n = ceil((temphead/360)*79);
    else
         temphead = temphead - 180;
        n = ceil((temphead/360)*79);
    end


%     C(:,n) = FSonar(:,1);
%     figure(1);
%
%    surf(X,Y,C,'EdgeColor','none');
%    caxis([0 175]);
%     caxis('manual');
%     colorbar;
%     view(0,90);
    scancount = scancount + 1;

   % pause;


    if count == 80


        numScans = numScans + 1;
        if(numScans > 5)

            tempscanum =  (4)*80
            templateimage(numimages).img = createsonim(Scans((scancount -
(4)*80):scancount),tempscanum,201);
            figure(99);
            imagesc(templateimage(numimages).img);
           % numScans = 1;
            figure(100);
            clf;
            [xerr(numerror) yerr(numerror)] =  calcposeerror(baseline,
templateimage(numimages).img);
            edist(numimages) = sqrt(xerr(numerror)^2 + yerr(numerror)^2);
            figure(1);
            plot(edist);
            numerror = numerror + 1;
            numimages = numimages + 1;
        end
%       figure(2);
%       [x1,y1] = meshgrid(-1:.01:1, -1:.01:1);
%       z = griddata(X,Y,C, x1, y1,'linear');
%       img = uint8(imrotate(z,180));
%       imagesc(img);
%      % sfilt(img);
 %     pause;
```

Localization of an AUV in an Absolute Reference Frame using a Small Scanning Sonar

```
       count = 1;
    else

        count = count+1;
    end

end
end

function [xerr yerr] = calcposeerror(baseline,template)

    cordata = xcorr2(double(baseline),double(template));
    [rowpos,colpos] = max2dloc(cordata);
    xerr = round(colpos - (length(baseline)-1));
    yerr = round(rowpos - (length(baseline)-1));


end

function img = createsonim(Scan,numScans,imsize)


    img = zeros(imsize,imsize);
    for n = 1:1:numScans
        heading = nav2cart(Scan(n).EarthAlignedHeadAngle);
        range = Scan(n).Ranges;
        for i = 1:1:length(range)
            if(range ~= 0)
                [xpos ypos] = pol2cart(deg2rad(heading),(range(i)/252)*5);
                fypos = round((ypos+Scan(n).Pose.yPos/10)/.1)/10;
                fxpos = round((xpos+Scan(n).Pose.xPos/10)/.1)/10;

                [fxpos,fypos] = cart2im(fxpos*10,fypos*10,101);
                img(fypos,fxpos) = img(fypos,fxpos)+1;
            end
         end
    end
end

function [Y Yf width] = findobjects(X,thresh)

    Y = zeros(1,252);

    [Yf width nul] = findlocmax(X,thresh);
    if(~nul)


    for i = 1:1:length(Yf)

        Y(Yf(i):(Yf(i)+width(i))) = 200;
    end
    end

end
```

Localization of an AUV in an Absolute Reference Frame using a Small Scanning Sonar

```
%--parsedatafiles.m

clear;
clc;
close all;
go = false;
n = 0;
Scans.Headangle =0;
Scans.Time.Hour = 0;
Scans.Time.Min = 0;
Scans.Time.Sec = 0;
Scans.Time.mSec = 0;
Scans.Pose.xPos = 0;
Scans.Pose.yPose = 0;
Scans.Pose.Pitch = 0;
Scans.Pose.Roll = 0;
Scans.Pose.Heading = 0;
Scans.ReturnData = zeros(1,252);
Scans.Objects.Range = 0;
Scans.Objects.Width = 0;
Scans.Gain = 0;
Scans.RangeSetting = 0;
Scans.numObjects = 0;
load('frontback.mat');
dataSet = datasetnew;
data_file = fopen('forwardback.852','r');


for n = 1:1:10000;

    data = fread(data_file,384);

    Scans(n).Gain = data(39);
    minute = data(21:34)-48;

    min = minute(4)*10 + minute(5);
    Scans(n).Min = min;
    second = minute(7)*10 + minute(8);
    Scans(n).mSec = minute(11)*100 + minute(12)*10;
    Scans(n).Sec = second;
    HeadLo = uint16(data(106));
    HeadHigh = uint16(data(107));
    HeadLocation = double(((bitand(HeadHigh,62)/2)*256)+
bitor(bitand(HeadLo,127),(bitand(HeadHigh,1)*128)));
    HeadAngle = .15*(HeadLocation-1400);
    Scans(n).Headangle = HeadAngle;
    Scans(n).RangeSetting = data(108);
    Scans(n).ReturnData = data(113:113+251);


end

n = 1;
count = 1;
```

Localization of an AUV in an Absolute Reference Frame using a Small Scanning Sonar

```
for i = 1:1:10000

    while(~go)
        go = false;

        minPoseSub = dataSet(count,5);
        minScan = Scans(i).Min;
        secPoseSub = dataSet(count,6);
        secScan = Scans(i).Sec;
        mSecPoseSub = dataSet(count,7);
        mSecScan = Scans(i).mSec;
        nextmSecPoseSub = dataSet((count+1),7);
        nextMinPoseSub = dataSet(count+1,5);
        nextSecPoseSub = dataSet(count+1,6);
        timePoseSub = (minPoseSub/60)*100 + secPoseSub/60 +
mSecPoseSub/100000;
        timeScan = (minScan/60)*100 + secScan/60 + mSecScan/100000;
        nextTimePoseSub = (nextMinPoseSub/60)*100 + nextSecPoseSub/60 +
nextmSecPoseSub/100000;
%
%         if(minScan == minPoseSub)
%
%         if(secScan == secPoseSub)
%
%             tmss = Scans(i).Sec;
%             tmmm = dataSet(count+1,6);
%
%             if((mSecScan >= mSecPoseSub) && ((tms < tmm) || (secScan <
tmmm)) )


    if((timeScan >= timePoseSub) && (timeScan < nextTimePoseSub))


        Scans(i).Pose.Heading = dataSet(count,8);
        Scans(i).Pose.Pitch = dataSet(count,9);
        Scans(i).Pose.Roll = dataSet(count,10);
        Scans(i).Pose.xPos = dataSet(count,11);
        Scans(i).Pose.yPos = dataSet(count,12);
        go =true;
    else

        count = count+1;
    end
%                 go = true;
%             else%if(Scans(i).mSec < dataSet((count+1),7))
%                 count = count +1;
%             end
%         elseif(Scans(i).Sec > dataSet(count,6))
%
%             count = count + 1;
%
%         end
%
%     elseif(Scans(i).Min > dataSet(Count,5)) %scan time is ahead of sub time
data
```

Localization of an AUV in an Absolute Reference Frame using a Small Scanning Sonar

```
%         count = count + 1;
%     end
%     end
%
%     go = false;
    end
go = false;
end
save('ScansSet','Scans');


function updategrid(heading,range,Scan)

    load('dataimage.mat')
    figure(100);
    hold on;


    for i = 1:1:length(range)
        if(range ~= 0)
            [xpos ypos] = pol2cart(deg2rad(heading),(range(i)/252)*5);
            fypos = round((ypos+Scan.Pose.yPos/10)/.1)/10;
            fxpos = round((xpos+Scan.Pose.xPos/10)/.1)/10;

            scatter(fxpos,fypos,'b');
            scatter(Scan.Pose.xPos/10,Scan.Pose.yPos/10,'x','r');
            ylabel('South  <-->  North');
            xlabel('West   <-->  East');
            [fxpos,fypos] = cart2im(fxpos*10,fypos*10,101);
            dataimage(fypos,fxpos) = dataimage(fypos,fxpos)+1;
        end
    end
    axis([-10 10 -10 10])
    set(gca,'XTick',[-10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10]);
    set(gca,'YTick',[-10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10]);
    grid on;
    save('dataimage.mat','dataimage');
end


function [Y width nullq] = findlocmax(X,thresh)

    count = 1;
    triggered = false;
    inlobe = false;
    firstpt = 0;
    secondpt = 0;
    maxes = 0;
    nullq = true;
    Y = 0;
    width = 0;
    for i = 1:1:length(X)

        if(X(i) >= thresh && (~triggered))
```

Localization of an AUV in an Absolute Reference Frame using a Small Scanning Sonar

```
            firstpt = i;
            triggered = true;

         elseif(triggered && (X(i) < thresh))

            width(count) = i - firstpt;
            Y(count) = firstpt;
            count = count+ 1;
            triggered = false;
            nullq = false;
        end



    end



end


function [xi,yi] = max2dloc(array)

    [x,ix] = max(array,[],1);
    [y,coli] = max(x);
    [y,rowi] = max(array(:,coli));

    xi = rowi;
    yi = coli;
end


function CartHeading = nav2cart(NavHeading)

    if((NavHeading >= 0) && (NavHeading <=90))

        CartHeading = abs(NavHeading - 90);

    elseif((NavHeading > 90) && (NavHeading <= 180))

        CartHeading = (360 - (NavHeading - 90));

    elseif((NavHeading > 180) && (NavHeading <= 270));
        CartHeading = (360 - NavHeading)+90;
    else
        CartHeading = (360 - NavHeading) + 90;
    end
end


function [x y] = cart2im(xc,yc,cent)

    if((xc < 0) && (yc < 0))
```

Localization of an AUV in an Absolute Reference Frame using a Small Scanning Sonar

```
        x = xc + cent;
        y = abs(yc) + cent;
    elseif((xc < 0) && (yc > 0))
        x = xc + cent;
        y = cent - yc;
    elseif((xc > 0) && (yc > 0))
        x = xc + cent;
        y = cent - yc;
    elseif((xc > 0) && (yc < 0))
        x = xc+ cent;
        y= abs(yc) + cent;
    elseif((xc == 0) && (yc < 0))
        x = cent;
        y = abs(yc) + cent;
    elseif((xc == 0) && (yc > 0))
        x = cent;
        y = cent - yc;
    elseif((xc < 0) && (yc == 0))
        x = xc + cent;
        y = cent;
    elseif((xc > 0) && (yc == 0))
        x = xc + cent;
        y = cent;
    else
        x = cent;
        y = cent;
    end

end
```