

Vessel Motion Extraction from an Image Sequence

Esteban L. Castro-Feliciano
Naval Architecture and Marine Engineering Dept.
University of Michigan
Ann Arbor, Michigan
Email: eslucafe@umich.edu

Abstract—The abstract goes here.

I. INTRODUCTION

The use of motion data from a vehicle is common in all vehicle research fields. For marine applications, the motion of a vessel is recorded both in full and model scale tests. This motion information is used to measure the performance and safety of the vessel in a seaway. The method commonly used to record the vessel motion is to use accelerometers and gyros to measure the rotational and translational accelerations and speeds. Then, data post-processing is used to estimate the vessel's position [1]. However, if the actual displacements were not measured from a fixed frame of reference, it is very difficult – if not impossible – to validate the post-processing results.

With the wide availability of video cameras, it is an attractive idea to simply record the vessel tests, and extract the motion from the sequence of images. The attractiveness of this method is its simplicity of application, since the vessel would not need to have any equipment installed. Moreover, it is generally estimated that most of the rigid-body motions of the vessel occur at frequencies less than 15Hz, [2]; this means that even conventional video cameras recording at 24fps has the capabilities to extract motions up to 12Hz (Nyquist frequency) without any advanced signal processing such as in [3].

Consequently, the goal of this project is to extract motion histories of all six degrees of freedom (Fig. 1) of a vessel from a sequence of images. In real life applications, the sequence of images could come either from towing tank tests or open water testing. However, in this paper the images were created using the open-source ray-tracing program POV-Ray. Nevertheless, the motions of the virtual vessel are the motions recorded from a real open water test, therefore the vessels movement is actually representative of the real motions of a vessel in waves.

II. DATA USED

A. Image Sequence

The image seen in Fig. 2 is an example figure from the virtual environment that will be used to create the sequence of images. The vessel will follow a prescribed motion such as the one shown in Fig. 3. The program used for creating this image is POV-Ray, an open source ray-tracing software.

B. Prescribed Motion Histories

The motion data used in this paper comes from trials conducted by NSWC Carderock Detachment Norfolk, Combatant Craft Division. The vessel used was an 11.0m (36.1ft) long

vessel. Because the experiments recorded only the acceleration data, the velocity and displacement are estimated by offline processing.

III. METHODOLOGY

A. Vessel Motion DSP

One can see in the FFT of the signal in Fig. 4 that there is a lot of low frequency content, which causes non-realistic drift in the velocity and displacement estimations (these lower frequencies are magnified when integrating). The method used in this project to estimate the velocity and displacement values is to filter and integrate the acceleration data in frequency domain [4]. The filter used is a high-pass Sinc filter with a cut-off of 0.15Hz. This cut-off frequency appears to be a good compromise between reducing drift and preserving the rigid-body velocity and displacements especially at points where the craft is free-falling. Because the integration is performed in frequency domain, the Sinc filter was used only once at the acceleration. The reason for using a Sinc filter was because it completely removes the undesired frequencies, and therefore these frequencies do not grow in magnitude when performing the frequency domain integration. An example of the results can be seen in Fig. 3. The smoothing observed in the velocity and displacement time histories is due to the integration and not because of the use of a low-pass filter. The accuracy of this process cannot be quantitatively estimated since the actual vessel velocity and displacement were not measured.

B. Image DSP

The goal for the image DSP is to find three colored circles in the images (red, green and black). This falls inside the general computer vision problem of object recognition. This projects problem could be solved with DSP tools of “template matching” with an algebraic definition of the template (since a circle is a simple shape) [5]. The basic idea of this process is to use nonlinear optimizations so that the user defined template is translated and resized – and if required rotated in 3D – until the template minimized the error of the nearby image edges. However, since in this project we are just finding circles (since spheres look like circles in the images regardless of their orientation), a simpler approach was taken which uses some of the concepts of the more advanced template matching algorithms: object gradients and minimizing error between template and image edges. The used approach is the following:

- 1) Apply the Canny edge detection [6] on the blue channel (blue channel contains good edge information on all three colored circles)

- a) Filter image (noise reduction) by convolution with a Gaussian mask. A Gaussian mask is used since the gradients of the edges will be computed, and this noise reduction will reduce the emergence of false edges due to noise.
 - b) Find edge strength by convolution with Sobel operators. With the gradients in the x and y-direction, the magnitude and angle are found.
 - c) Eliminate non-maximums on edges to thin the line. This is achieved by using conditional statements and the edge angle information.
- 2) Find closed edges by running along the edges (using the edge angle information) until either the first pixel is reached (save this edge information) or the edge ends (information is discarded).
 - 3) Take the average of the RGB image values inside the closed edges, and decide if that closed edge is either red, green or black – these are user defined thresholds, and using $red_{avg} > 150$, $green_{avg} < 50$ and $blue_{avg} < 50$ for the red ball showed good results (and for the other colors accordingly).
 - 4) Perform a circle regression to the edge values for the found red, green and black objects.

A project note is that the this entire algorithm was programmed from scratch by the author, with the exception of the circle regression code.

IV. CONCLUSION

With this algorithm, good results where obtained; some examples can be seen in Fig. 5-7. Due to time constraints, the trigonometry/perspective calculations to estimate the position of the vessel by using the circle radius and center could not be performed. However, we can see from the results that the found circles are very accurate, and the author suspects that good motion results could be obtained.

Moreover, the algorithm seemed robust against closed edges that where not the circles. An example of this can be seen in Fig. 8, where the reflection of the green ball created a strong closed edge that got up to step (2), but it did not match the color criteria from step (3).

V. SUGGESTED FUTURE WORK

The algorithm could be made more efficient if color information is used to find the first guesses of the circle edges. Moreover the algorithm could be improved if a regression fit criteria is set for the circles, so that any object that is closed and satisfy the color conditions but is not a perfect circle can be discarded.

VI. PROJECT SPECIFIC QUESTIONS

A. DSP Tools Used

- 1) DFT: The FFT function from Matlab was used to find the DFT's of the acceleration data
- 2) Filtering: High pass filtering was performed to the acceleration data, and Gaussian smoothing to the images

- 3) Convolution: Convolution was done on two occasions on the images, (1) to do the Gaussian smoothing and (2) find the edge strength.

B. Out-of-Class DSP Tool

While we had a homework where we performed filtering and edge detection, the “Canny Edge” detection was not used as a tool. Moreover, the using the gradients to identify an object is a tool not used in class.

APPENDIX A FIGURES

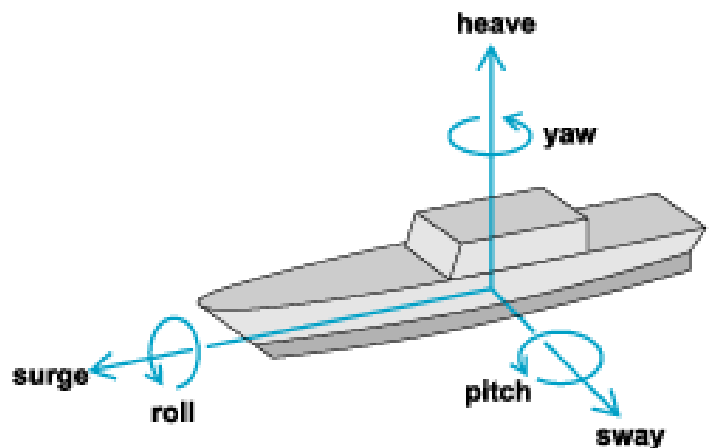


Fig. 1. Six degrees of freedom (Source: Answers.com).

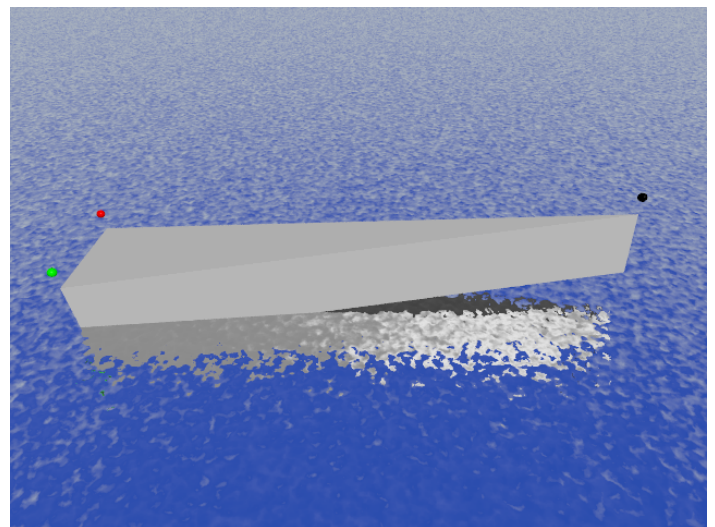


Fig. 2. POV-Ray image example.

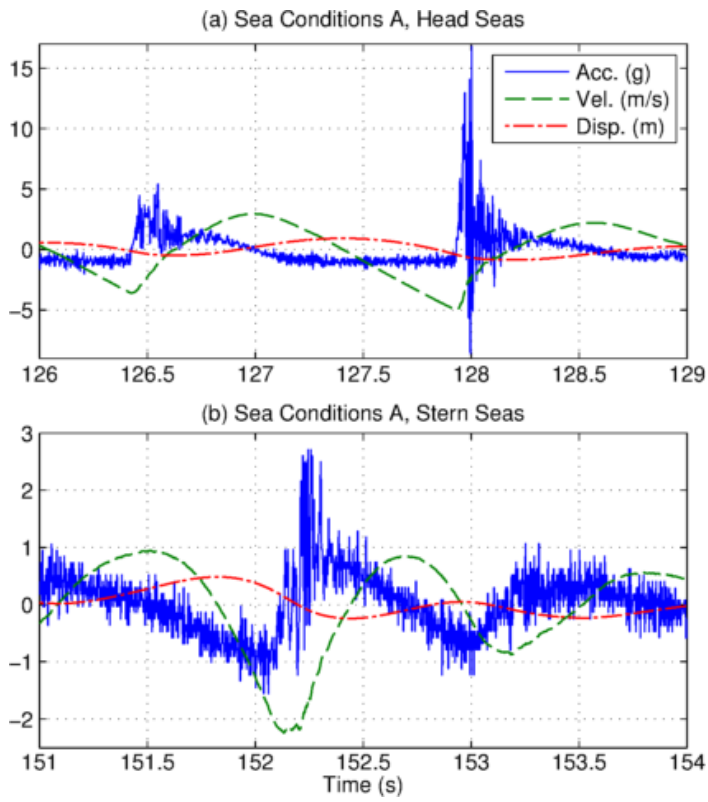


Fig. 3. Motion history example.

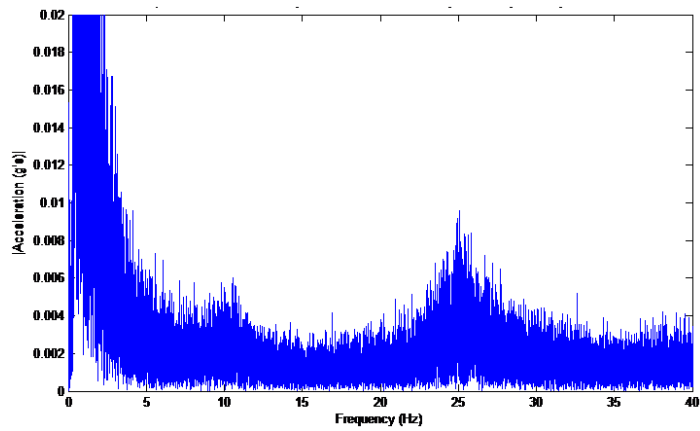


Fig. 4. Magnitude of FFT of acceleration history example.

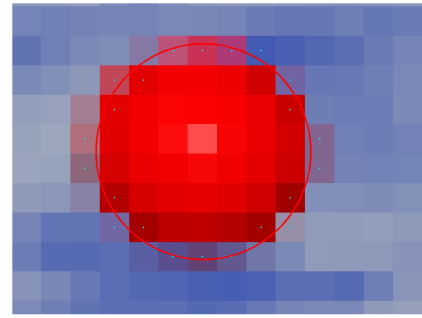


Fig. 5. Red ball result example (with edge pixels and regressed circle).

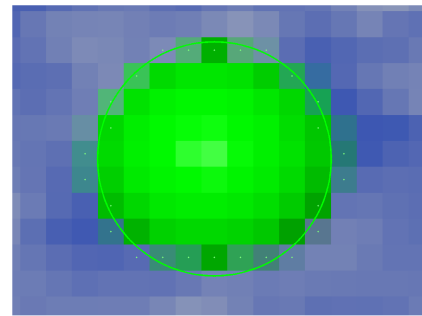


Fig. 6. Green ball result example .

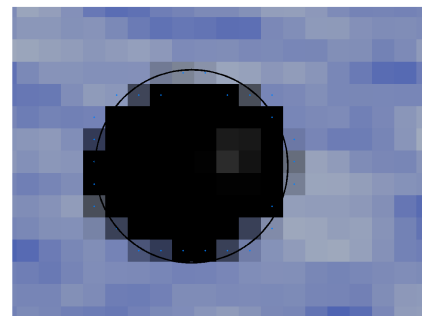


Fig. 7. Black ball result example.

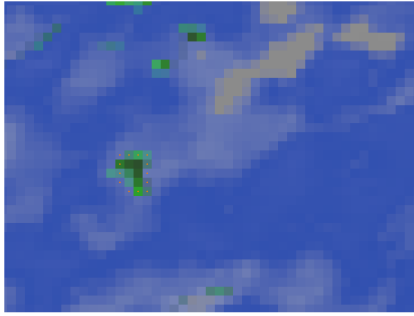


Fig. 8. Closed edge that did not match color criteria to be considered a circle.

REFERENCES

- [1] M. Arraigada and P. Manfred, "Calculation of displacements of measured accelerations, analysis of two accelerometers and application in road engineering," in *Sixth Swiss Transport Research Conference*, 2006.
- [2] M. R. Riley and T. W. Coats, "A simplified approach for analyzing accelerations induced by wave-impacts in high-speed planing craft," in *Chesapeake Power Boat Symposium*, 2012.
- [3] "Extracting audio from visual information: algorithm recovers speech vibrations of a potato-chip bag filmed through soundproof glass," 2014. [Online]. Available: <http://newsoffice.mit.edu/2014/algorithm-recovers-speech-from-vibrations-0804>
- [4] A. Brandt and R. Brincker, "Integrating time signals in frequency domain – comparison with time domain integration," *Measurement*, vol. 58, pp. 511–519, Dec. 2014.
- [5] J. L. Mundy, "Object recognition in the geometric era: a retrospective," *Toward Category-Level Object Recognition*, pp. 3–28, Dec. 2006.
- [6] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 679–698, Nov. 1986.