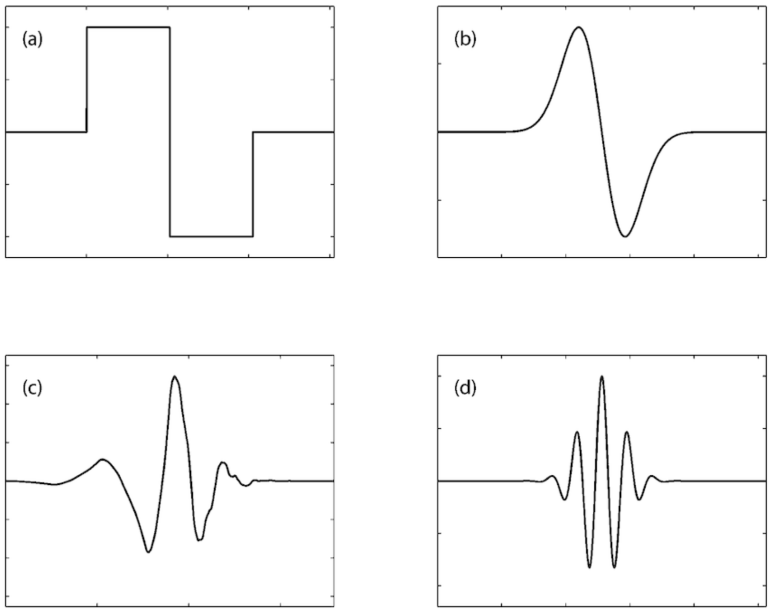
Wavelet Processing

Functions: discrete wavelet transform: matrix math, linear algebra, convolution, decomposition
Inputs: images, artificial matrix data
Metrics: processing time

1. Overview

Wavelets are brief, oscillating signals that begin from and return to a zero value within a finite duration. That characteristic, in conjunction with “scaling” and “shifting,” makes them very useful in signal processing. Scaling a wavelet – stretching it out or compressing it in time – can allow it to represent either the long-term movement of a signal or a shorter-term fluctuation, respectively. Shifting wavelets – aligning their centers to different points with respect to an original signal – can permit them to represent distinct events at varying points in that signal. Thus, it is possible to approximate a signal as a sum of shifted and scaled wavelets, via a “wavelet transform.” This is similar to the intuition behind a Fourier transform, which represents a signal as a sum of infinite-duration sine waves, but a wavelet transform can more effectively deal with signals that have abrupt changes or discontinuities. Wavelet transforms are useful in various applications including image processing, signal denoising, and compression (to name a few). For further information, see https://www.eecis.udel.edu/~amer/CISC651/IEEEwavelet.pdf or https://www.mathworks.com/videos/understanding-wavelets-part-1-what-are-wavelets-121279.html.



Common wavelets: (a) Haar, (b) Gaussian (order 1), (c) Daubechies (order 4), and (d) Morlet. Source: http://dx.doi.org/10.1785/0120060255

MilSpec’s wavelet processing benchmark applies the Haar wavelet (a square-shaped pulse, as above) to denoise sample images or random data (either one- or two-dimensional). Specifically, the test performs a discrete wavelet transform with a configurable number of decompositions, based on open-source implementations for 1D[1] and 2D[2] decomposition (the 2D implementation requires OpenCV, as noted below). The MilSpec benchmark integrates these two implementations with a common testharness, performance measurement tools, and assorted code cleaning/interface modification. Future versions of this benchmark will add additional wavelet choices and/or input data options.

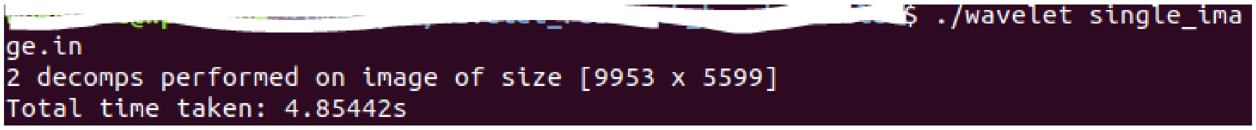
2. System requirements

Platform: Ubuntu 18.04 LTS with g++ 7.4.0. Code may build and run successfully on other versions/platforms, but has not been tested with them.
Storage: ~30MB for code and sample inputs. RAM requirements vary significantly by testcase, from a few KB for small input matrices to ~4GB for very large randomly-generated inputs.
Dependencies: OpenCV 3.2.0 (may work with later versions, but not tested).

3. Build and run

To benchmark:

- Install OpenCV: `sudo apt-get update` and `sudo apt-get install libopencv-dev`
- Verify compatible OpenCV version: `pkg-config --modversion opencv`
- Download and extract the zipfile from www.adacenter.org/milspec
- From the `wavelet/` directory, `make clean && make`
- Choose a testcase: `1d-array.in`, `1d-random.in`, `batch.in` (process multiple input images from a given directory), `random.in`, or `single_image.in`.
- From the `wavelet/` directory, `./wavelet [testcase]`
- Results are displayed in the terminal, as below:



```
[redacted]$ ./wavelet single_ima
ge.in
2 decomps performed on image of size [9953 x 5599]
Total time taken: 4.85442s
```

4. Code structure

Coming soon!

5. MilSpec development notes, errata, changelog

v0.9:

- Changes from baseline wavelet code for MilSpec:
  - Removed unused/unwanted code from baseline implementations (e.g. GUI configuration)
  - Built wrapper for 1D/2D baseline implementations

[1] https://github.com/matteotiziano/simple-wavelet
[2] https://stackoverflow.com/questions/20071854/wavelet-transform-in-opencv/20072775#20072775

- Created makefile
- Added basic configuration options and file-based testcase input
- Created basic testcases
- Added performance metrics

6. Acknowledgements

MilSpec is a project under development at the University of Michigan – Ann Arbor by Pete Ehrett, Bing Schaefer, Adrian Berding, Nathan Block, John Paul Koenig, Pranav Srinivasan, Todd Austin, and Valeria Bertacco. This project is supported by the Applications Driving Architectures Center, one of six centers of JUMP, a Semiconductor Research Corporation program co-sponsored by DARPA.

The original wavelet processing implementations upon which the MilSpec versions are based are available at https://github.com/matteotiziano/simple-wavelet (1D) and https://stackoverflow.com/questions/20071854/wavelet-transform-in-opencv/20072775#20072775 (2D).