

# Real-time envelope detection of ultrasound radiofrequency signals using OpenCV GPU framework

Zhuhuang Z<sup>1,2</sup>, Weiwei W<sup>2</sup>, Shuicai W<sup>1\*</sup>, Kebin J<sup>2</sup>, Chang KJ<sup>3</sup> and Tsui PH<sup>4,5,6\*</sup>

<sup>1</sup>College of Life Science and Bioengineering, Beijing University of Technology, Beijing, China

<sup>2</sup>Faculty of Information Technology, Beijing University of Technology, Beijing, China

<sup>3</sup>Department of Surgery, National Taiwan University Hospital, Taipei, Taiwan

<sup>4</sup>Department of Medical Imaging and Radiological Sciences, College of Medicine, Chang Gung University, Taoyuan, Taiwan

<sup>5</sup>Medical Imaging Research Center, Institute for Radiological Research, Chang Gung University and Chang Gung Memorial Hospital at Linkou, Taoyuan, Taiwan

<sup>6</sup>Department of Medical Imaging and Intervention, Chang Gung Memorial Hospital at Linkou, Taoyuan, Taiwan

## Abstract

Envelope detection of ultrasound backscattered signals is essential in envelope statistics based quantitative ultrasound tissue characterization. In this work, we proposed using the OpenCV graphics processing unit (GPU) framework for Hilbert transform based envelope detection of ultrasound radiofrequency (RF) signals. Experiments were conducted on 21 frames of ultrasound RF data (256 scan lines  $\times$  1552 sample points) using a computer with dual 2.60-GHz CPU, 64-GB RAM, and 2-GB GPU. Experimental results showed that OpenCV CPU and OpenCV GPU had an average running time of  $106.00 \pm 0.84$  ms and  $18.43 \pm 0.68$  ms, respectively. The proposed OpenCV GPU based envelope detection method may be used in ultrasonic applications demanding real-time computation of the envelope of ultrasound RF signals.

## Introduction

Ultrasound imaging has been widely used in routine diagnostics because of its low cost, nonionizing radiation, and real-time capability [1,2]. A transducer transmits ultrasound waves into a tissue, which interact with the tissue, and the transducer receives the echoes returned. Constructed from the amplitude of ultrasound echo signals, ultrasound B-mode images suffer from the drawbacks as being qualitative and operator-dependent [1]. To complement B-mode imaging, quantitative ultrasound (QUS) explores frequency, phase, or statistical information of backscattered ultrasound signals for tissue characterization [3–10]. In the context of QUS, parameters including backscatter coefficient, acoustic attenuation, speed of sound, envelope statistics, scatterer properties, and tissue elasticity can be quantified for tissue characterization [4,11,12].

When a tissue is modeled as a collection of scattering particles (i.e., scatterers), the interaction of incident ultrasound waves with tissue scatterers can be reflected in the unprocessed radiofrequency (RF) echo signals. Envelope statistics are directly related to tissue microstructures and are able to characterize structural alterations in tissues that are not apparent on ultrasound B-mode images [4–10]. Thus, they are useful for quantitative description of pathological state of tissues. Envelope detection of ultrasound RF data is essential for envelope statistics analysis [4–10]. In order to ensure real-time backscattered envelope analysis, the time efficiency of the envelope detection algorithm is demanded.

OpenCV is an open-source computer vision library written with C/C++ (<http://opencv.org/>) and has been widely used for image processing and computer vision applications [13–16]. Specifically, OpenCV has provided a graphics processing unit (GPU) framework

for real-time data processing [16]. Considering the open-source nature and GPU framework of OpenCV, we proposed using OpenCV GPU to detect ultrasound backscattered envelope in this work. Envelope detection experiments were conducted on 21 frames of ultrasound RF data. Experimental results demonstrated that real-time RF envelope detection could be realized with the OpenCV GPU framework.

## Materials and methods

### Data acquisition

This study was approved by the Institutional Review Board of National Taiwan University Hospital and the patients signed informed consent forms. Breast ultrasound images were collected with a commercial portable ultrasound scanner (Model 3000, Terason, Burlington, MA, USA), with the raw RF data digitized at a sampling rate of 30 MHz. The transducer comprised a wideband linear array (Model 10L5, Terason) with a center frequency of 7.5 MHz. Twenty-one volunteer female patients with breast tumors were recruited. A sonographer performed the ultrasound scanning. Each image acquisition protocol involved a total of 256 beamformed scan lines

**Correspondence to:** RShuicai Wu, College of Life Science and Bioengineering, Beijing University of Technology, Beijing, China, E-mail: wushuicai@bjut.edu.cn

Po-Hsiang Tsui, Department of Medical Imaging and Radiological Sciences, College of Medicine, Chang Gung University, Taoyuan, Taiwan, E-mail: tsuiph@mail.cgu.edu.tw

**Key words:** ultrasound tissue characterization, envelope detection, Hilbert transform, graphics processing unit, OpenCV

**Received:** July 22, 2017; **Accepted:** August 14, 2017; **Published:** August 17, 2017

(1552 sample points per scan line) of the backscattered echoes. The interval between each scan line was 0.15 mm.

### Envelope detection

The instantaneous amplitude (A-line) of the ultrasound RF signal is obtained by envelope detection:

$$e(t) = |r(t) + j\hat{r}(t)| \quad (1)$$

where  $e(t)$  denotes the A-line,  $r(t)$  is the ultrasound RF signal,  $\hat{r}(t)$  denotes the Hilbert transform of  $r(t)$ , and  $j$  is the imaginary unit. The expression  $r(t) + j\hat{r}(t)$  is referred to as the analytic signal or complex RF echo. In previous studies, Hilbert transform has been frequently used for envelope detection of ultrasound RF signals [6–10]. Based on the property of Hilbert transform [17,18], the analytic signal  $r(t) + j\hat{r}(t)$  has no spectrum components at negative frequencies and its spectrum components at positive frequencies are doubled [19]. This property will be considered in the algorithm implementation of envelope detection based on Hilbert transform. Figure 1 shows the flow chart of the envelope detection algorithm.

### Algorithm implementation

Suppose that a frame of ultrasound RF data,  $r(t)$ , comprises  $M$  scan lines, and each scan line contains  $N$  sample points. For the  $i^{\text{th}}$  scan line of  $r(t)$ ,  $1 \leq i \leq M$ , the Hilbert transform is computed. Then, the envelope of the RF data is computed using Equation (1).

The Hilbert transform of the  $i^{\text{th}}$  scan line,  $s$ , is calculated using the discrete Fourier transform (DFT) [20] and involves a five-step algorithm:

- (i) Define a complex vector  $c$ , whose real part is  $s$  and imaginary part is zero vector.
- (ii) Conduct  $N$ -point forward DFT [20] to  $c$ , and a complex vector  $d$  is obtained.

For the  $k^{\text{th}}$  sample point,  $1 \leq k \leq N$ , if  $k \in [1, N/2]$ , let both the real part and imaginary part of  $d[k]$  be multiplied by 2; if  $k \in [(N/2)+1, N]$ , let both the real part and imaginary part of  $d[k]$  be zero.

Conduct  $N$ -point inverse DFT [20] to  $d$ , and a complex vector  $v$  is obtained.

The Hilbert transform of the  $i^{\text{th}}$  scan line  $s$  is computed as the imaginary part of  $v$  divided by  $N$ .

In this study, the Hilbert transform based envelope detection algorithm was implemented using the C++ language with OpenCV 2.4.9 and Microsoft<sup>(R)</sup> Visual C++ 2010. Both OpenCV CPU and OpenCV GPU frameworks based envelope detection was performed, and C++ codes for the algorithm implementation were provided in Tables 1, 2 and 3, where “rf” denoted the ultrasound RF data matrix, and “envelope” was the envelope matrix of “rf”. Envelope detection with OpenCV CPU was based on the OpenCV matrix data structure, `cv::Mat`. Forward DFT and inverse DFT on the basis of the OpenCV CPU framework were computed using the OpenCV DFT function, `cv::dft()`. Envelope detection with OpenCV GPU was based on the OpenCV GPU matrix data structure, `cv::gpu::GpuMat`. Forward DFT and inverse DFT on the basis of the OpenCV GPU framework were conducted using the OpenCV GPU based DFT function, `cv::gpu::dft()`. The communication between CPU and GPU was performed using `GpuMat::upload()` and `GpuMat::download()` functions.

## Results

Envelope detection experiments were conducted on 21 frames of ultrasound RF data using a computer with dual 2.60-GHz Intel<sup>(R)</sup> Xeon<sup>(R)</sup> E5-2670 CPU, 64-GB RAM, and 2-GB NVIDIA<sup>(R)</sup> Quadro<sup>TM</sup> 4000 GPU. The number of scan lines and sample points are listed in Table 1. Experimental results showed that visually, the OpenCV GPU framework produced the same B-mode image as OpenCV CPU (Figure 2). For RF data of 256 scan lines  $\times$  1552 sample points ( $n = 21$ ), OpenCV CPU and OpenCV GPU had a running time of  $106.00 \pm 0.84$  ms and  $18.43 \pm 0.68$  ms, respectively (Table 4). The results demonstrated that the significantly improved computational efficiency of the OpenCV GPU method over the OpenCV CPU method.

**Table 1.** C++ code for envelope detection using the OpenCV CPU framework.

```
cv::Mat rfComplex(rf.rows, rf.cols, CV_32FC2);
std::vector<cv::Mat> combiner;
combiner.push_back(rf);
combiner.push_back(cv::Mat::zeros(rf.size(), V_32FC1));
cv::merge(combiner, rfComplex);
std::vector<cv::Mat> splitter;
splitter.push_back(cv::Mat(rfComplex.rows, rfComplex.cols, CV_32FC1));
splitter.push_back(cv::Mat(rfComplex.rows, rfComplex.cols, CV_32FC1));
cv::Mat rfSpectrum = cv::Mat(rfComplex.rows, rfComplex.cols, CV_32FC2);
// forward DFT
cv::dft(rfComplex, rfSpectrum, DFT_ROWS);
cv::multiply(rfSpectrum, factor, rfSpectrum, 1.0, CV_32FC2);
// inverse DFT
cv::dft(rfSpectrum, rfComplex, DFT_ROWS | DFT_INVERSE);
cv::split(rfComplex, splitter);
// get imaginary part
imag = splitter[1];
cv::multiply(imag, cv::Scalar(1.0/rf.cols), imag, 1.0, CV_32FC1);
cv::magnitude(rf, imag, envelope);
```

**Table 2.** C++ code for envelope detection using the OpenCV GPU framework.

```
cv::gpu::Stream stream;
cv::gpu::GpuMat rfGPU = cv::gpu::GpuMat(rf.rows, rf.cols, CV_32FC1);
rfGPU.upload(rf);
cv::gpu::GpuMat rfComplexGPU = cv::gpu::GpuMat(rfGPU.rows, rfGPU.cols, CV_32FC2);
std::vector<cv::gpu::GpuMat> combiner;
combiner.push_back(rfGPU);
combiner.push_back( cv::gpu::GpuMat(rfGPU.rows, rfGPU.cols, CV_32FC1, cv::Scalar(0)) );
cv::gpu::merge(combiner, rfComplexGPU, stream);
cv::gpu::GpuMat factorGPU = cv::gpu::GpuMat(rfGPU.rows, rfGPU.cols, CV_32FC2);
factorGPU.upload(factor);
cv::gpu::GpuMat rfSpectrumGPU = cv::gpu::GpuMat(rfGPU.rows, rfGPU.cols, CV_32FC2);
std::vector<cv::gpu::GpuMat> splitter;
splitter.push_back(cv::gpu::GpuMat(rfGPU.rows, rfGPU.cols, CV_32FC1));
splitter.push_back(cv::gpu::GpuMat(rfGPU.rows, rfGPU.cols, CV_32FC1));
// forward DFT
cv::gpu::dft(rfComplexGPU, rfSpectrumGPU, rfComplexGPU.size(), DFT_ROWS, stream);
cv::gpu::multiply(rfSpectrumGPU, factorGPU, rfSpectrumGPU, 1.0, CV_32FC2, stream);
// inverse DFT
cv::gpu::dft(rfSpectrumGPU, rfComplexGPU, rfComplexGPU.size(), DFT_ROWS | DFT_INVERSE, stream);
cv::gpu::split(rfComplexGPU, splitter, stream);
// get imaginary part
cv::gpu::GpuMat imagGPU = splitter[1];
cv::gpu::multiply(imagGPU, cv::Scalar(1.0/rfGPU.cols), imagGPU, 1.0, CV_32FC1, stream);
cv::gpu::GpuMat magGPU = cv::gpu::GpuMat(rfGPU.rows, rfGPU.cols, CV_32FC1);
cv::gpu::magnitude(rfGPU, imagGPU, magGPU, stream);
magGPU.download(envelope);
stream.waitForCompletion();
```

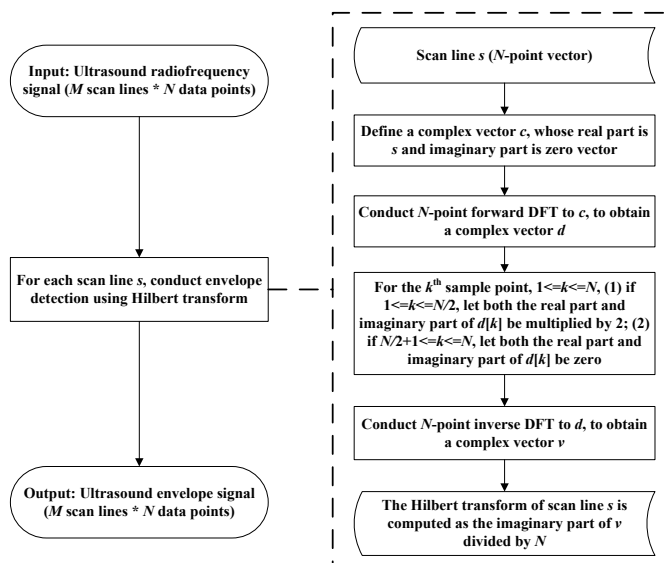


Figure 1. Flow chart of the ultrasound radiofrequency signal envelope detection algorithm.

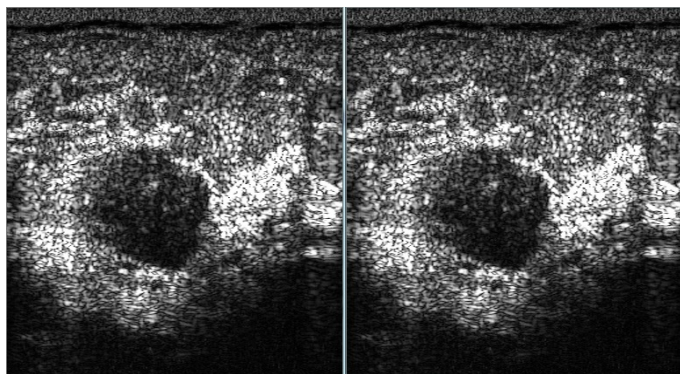


Figure 2. Visual comparison of breast ultrasound B-mode images computed from backscattered envelope detected using OpenCV CPU (left) and OpenCV GPU (right) frameworks

Table 3. C++ code for preprocessing.

```
rf.convertTo(rf, CV_32FC1);
cv::Mat imag(rf.rows, rf.cols, CV_32FC1); // imaginary part
cv::Mat factor = cv::Mat::ones( rf.rows, rf.cols, CV_32FC2 );
for (int i=0; i<rf.rows; ++i)
{
    for (int j=1; j<rf.cols/2; ++j)
    {
        factor.at<Vec2f>(i, j)[0] = 2;
        factor.at<Vec2f>(i, j)[1] = 2;
        factor.at<Vec2f>(i, rf.cols-j)[0] = 0;
        factor.at<Vec2f>(i, rf.cols-j)[1] = 0;
    }
}
cv::Mat envelope(rf.rows, rf.cols, CV_32FC1);
```

## Discussion and conclusions

Envelope detection of ultrasound backscattered signals is an essential step in many ultrasonic applications, such as envelope statistics based tissue characterization [4–10]. Real-time envelope detection is desired for fast analysis of ultrasound RF signals. However, envelope detection on CPU can be a time-consuming process. To accelerate the computation, parallel GPUs, especially the GPGPU for general purpose computing on GPU, have been used in many fields including machine learning and medical image processing [21,22].

Table 4. Running time for envelope detection of radiofrequency data (256 scan lines  $\times$  1552 sample points) using OpenCV CPU and OpenCV GPU frameworks ( $n = 21$ ).

No.	Running time for OpenCV CPU (ms)	Running time for OpenCV GPU (ms)
	107	18
	106	19
	105	19
	106	19
	105	19
	107	19
	106	19
	105	18
	106	18
	105	19
	106	19
	107	19
	107	19
	107	18
	107	19
	107	18
	105	17
	106	18
	106	18
	105	17
	105	18
Average	106.00 $\pm$ 0.84	18.43 $\pm$ 0.68

The CUDA (Compute Unified Device Architecture) and OpenCL platforms are the two most widely adopted programming models for GPU computing. OpenCV is an open-source computer vision library with a GPU framework. The OpenCV GPU framework is implemented based on the CUDA platform and is easy and flexible to use. OpenCV has been used in various ultrasound applications including breast ultrasound image segmentation [14] and quasi-real-time ultrasound backscattered statistical parameter imaging (i.e., ultrasound Nakagami imaging) [15]. However, in our previous work, the envelope detection of RF signals was realized on CPU [15], limiting the frame rate of ultrasound Nakagami imaging (1–3 frames per second). In applications such as real-time ultrasonic characterization of cardiac tissues, the computation speed is highly demanded.

To date, quantitative ultrasound research using OpenCV GPU has not been conducted. To our knowledge, this work is the first to use OpenCV GPU framework for backscattered envelope detection. In this work, we proposed using OpenCV GPU framework for envelope detection of ultrasound RF signals. Experimental results demonstrated that the OpenCV GPU framework is capable for real-time envelope detection. The proposed OpenCV GPU based envelope detection method may be used in various ultrasound tissue characterization applications demanding fast computation of the envelope of backscattered RF signals.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos. 71661167001 and 71781260096) and by the International Scientific Research Collaboration Seed Fund of Beijing University of Technology.

## Conflict of interest

The authors declared that they have no competing interest.

## References

- Shung KK (2015) Diagnostic Ultrasound: Imaging and Blood Flow Measurements, 2nd Edition. Boca Raton: CRC Press.
- Szabo TL (2014) Diagnostic Ultrasound Imaging: Inside Out, 2nd Edition. New York: Academic Press.
- Insana MF, Hall TJ (1990) Parametric ultrasound imaging from backscatter coefficient measurements: image formation and interpretation. *Ultrason. Imaging* 12: 245-267. [[Crossref](#)]
- Mamou J, Oelze ML (2013) Quantitative Ultrasound in Soft Tissues. Heidelberg: Springer.
- Oelze ML, Mamou J (2016) Review of quantitative ultrasound: envelope statistics and backscatter coefficient imaging and contributions to diagnostic ultrasound. *IEEE Trans Ultrason Ferroelectr Freq Control* 63: 336-351. [[Crossref](#)]
- Tsui PH, Ma HY, Zhou Z, Ho MC, Lee YH (2014) Window-modulated compounding Nakagami imaging for ultrasound tissue characterization. *Ultrasonics* 54: 1448-1459. [[Crossref](#)]
- Tsui PH, Tsai YW (2015) Artifact reduction of ultrasound Nakagami imaging by combining multifocus image reconstruction and the noise-assisted correlation algorithm. *Ultrason Imaging* 37: 53-69. [[Crossref](#)]
- Tsui PH, Wan YL, Tai DI, Shu YC (2015) Effects of estimators on ultrasound Nakagami imaging in visualizing the change in the backscattered statistics from a Rayleigh distribution to a Pre-Rayleigh distribution. *Ultrasound Med Biol* 41: 2240-2251. [[Crossref](#)]
- Al-Kadi OS, Chung DY, Coussios CC, Noble JA (2016) Heterogeneous tissue characterization using ultrasound: a comparison of fractal analysis backscatter models on liver tumors. *Ultrasound Med Biol* 42: 1612-1626. [[Crossref](#)]
- Destempes F, Franceschini E, Yu FT, Cloutier G (2016) Unifying concepts of statistical and spectral quantitative ultrasound techniques. *IEEE Trans Med Imaging* 35: 488-500. [[Crossref](#)]
- Parker KJ, Dooley MM, Rubens DJ (2011) Imaging the elastic properties of tissue: the 20 year perspective. *Phys Med Biol* 56: R1-R29. [[Crossref](#)]
- Sarvazyan A, Hall TJ, Urban MW, Fatemi M, Aglyamov SR, et al. (2011) An Overview of Elastography - An Emerging Branch of Medical Imaging. *Curr Med Imaging Rev* 7: 255-282. [[Crossref](#)]
- Bradski G, Kaehler A (2008) Learning OpenCV: Computer Vision with the OpenCV Library. Sebastopol, CA: O'reilly Media.
- Zhou Z, Wu W, Wu S, Tsui PH, Lin CC, et al. (2014) Semi-automatic breast ultrasound image segmentation based on mean shift and graph cuts. *Ultrason Imaging* 36: 256-276. [[Crossref](#)]
- Zhou Z, Wu S, Wang CY, Ma HY, Lin CC, et al. (2015) Monitoring radiofrequency ablation using real-time ultrasound Nakagami imaging combined with frequency and temporal compounding techniques. *PLoS ONE* 10: e0118030. [[Crossref](#)]
- Pulli, K, Baksheev A, Korniyakov K, Eruhimov V (2012) Real-time computer vision with OpenCV. *Commun ACM* 55: 61-69.
- Hahn SL (1996) Hilbert Transforms in Signal Processing. Boston: Artech House, Inc.
- Potamianos A, Maragos P (1994) A comparison of the energy operator and the Hilbert transform approach to signal and speech demodulation. *Signal Process* 37: 95-120.
- Yu C, Zhang C, Xie L (2012) An envelope signal based deconvolution algorithm for ultrasound imaging. *Signal Process* 92: 793-800.
- Oppenheim A, Schaffer R (2010) Discrete-Time Signal Processing, Third Ed. Upper Saddle River, NJ: Prentice Hall
- Eklund A, Dufort P, Forsberg D, LaConte SM (2013) Medical image processing on the GPU - past, present and future. *Med Image Anal* 17: 1073-1094. [[Crossref](#)]
- Smistad E, Falch TL, Bozorgi M, Elster AC, Lindseth F (2015) Medical image segmentation on GPUs--a comprehensive review. *Med Image Anal* 20: 1-18. [[Crossref](#)]