A File Reader for the VisualSonics Vevo 770 Digital RF Data

Release 1.0.0

Matthew McCormick[1]

July 7, 2011

[1]University of Wisconsin-Madison, matt@mmmccormick.com

Abstract

A C++ file reader for the VisualSonics Vevo®770 high frequency ultrasound system is presented. The digital RF header file is read into an XML DOM representation, which can be serialized as XML, HTML, or accessed in C++ natively with data object binding. An InsightToolkit reader is also implemented.

Contents

1 Introduction

VisualSonics®(Toronto, Canada) manufactures high-frequency ultrasound imaging systems designed for pre-clinical research on small animal models. Anatomical and real-time physiological imaging of mice and rats is possible with resolutions up to 30 m and frame rates up to 240 frames per second (fps). Primary features of the system are intended to allow longitudinal, non-invasive monitoring of anatomical and hemodynamic features as well as for therapeutic intervention. Some of these features include 2D B-Mode, 3D B-Mode with a stepper motor system, M-Mode, pulse-wave Doppler, power Doppler, tissue Doppler, and contrast agent imaging. An ancillary feature of the system is the output of radio-frequency (RF) data

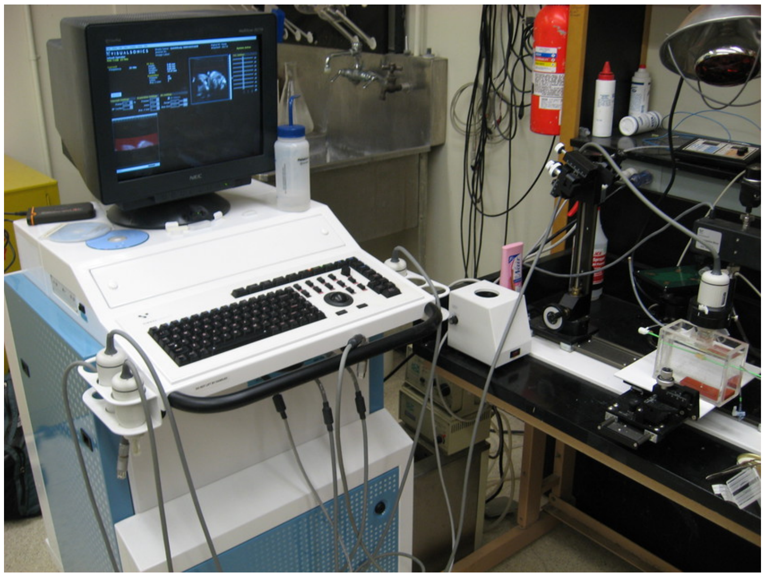


Figure 1: The Vevo®770 supports multiple transducers and has a standard console for interaction common to commercial imaging systems. The system also comes with a motion table designed to secure and position the transducer (right).

for analysis. In this article, methods are described to harness the RF output to perform advanced imaging research. This article is adapted from reference[1].

The Vevo 770 system is the last generation of Vevo®line systems that are designed around rotated single-element transducers; beginning with the Vevo 2100 and later systems, the transducers are a high frequency linear array design. The newer linear array transducer systems have better capabilities for pre-clinical imaging: a programmable transmit focus and dynamic receive focusing allow for a greater depth of field and better lateral resolution[2]. However, the single-element, high-frequency, wide bandwidth transducers of the Vevo 770 are desirable for the purpose of creating parametric ultrasound images. The simpler transducer geometry allows the system to be modeled during quantification of tissue-mimicking (TM) phantom acoustic properties. The same transducer can then be used in the collection of planar reflector, TM phantom, and echo-signals from tissue.

A photograph of the Vevo 770 imaging system is shown in Fig. 1 with a close-up of the transducer assembly in Fig. 2. The single element transducer rests at the end of a shaft whose pivot point is high within the body of the case assembly. The angular position is measured with a rotary encoder above the pivot point and scan conversion is necessary for proper display. In order to achieve high frame rates, the transducer is rotated quickly about the pivot point by a motor in the transducer housing. To facilitate good coupling between the transducer and *in vivo* animal models outside a water bath, the transducer element is encapsulated by a plastic basin and a replaceable thin film over the active element. The thin film must be placed on the transducer and the scanhead filled with water prior to each application.

Two options exist on the system to collect RF data: a BNC output for triggered signal acquisition with an external oscilloscope or other analog-to-digital (A/D) device, or RF data can be collected with an on-board A/D board integrated with the *Digital-RF* software module if available. An advantage to the latter system is an integrated preview of the acquired data at the time of acquisition along with coordinated 3D acquisition via system software control of the stepper-motor.

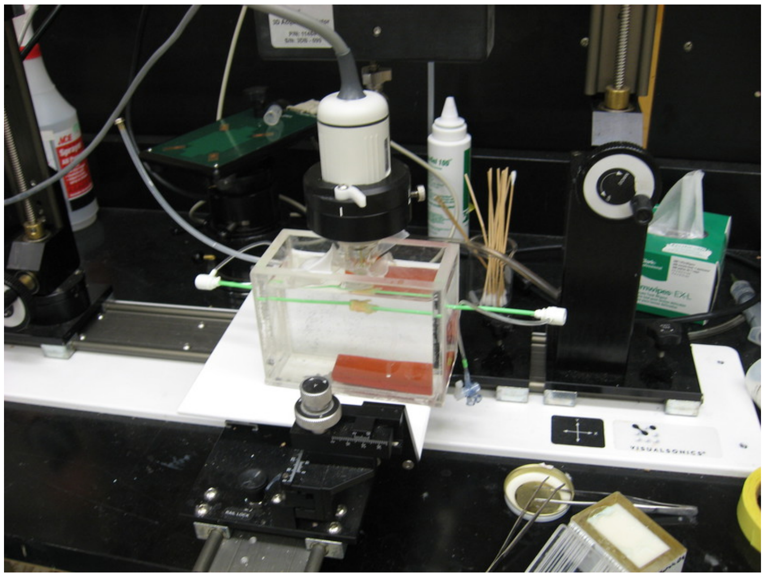


Figure 2: The Vevo 770 transducer (white) is held by a clamp connected to a precise linear stepper motor (top). The transducer element is suspended in a water filled capsule by a rod whose pivot point is high in the assembly housing above the clamp.

This article describes software to read files collected with the *Digital-RF* module. Software to create basic B-Mode images from the RF data can be found in another article [3].

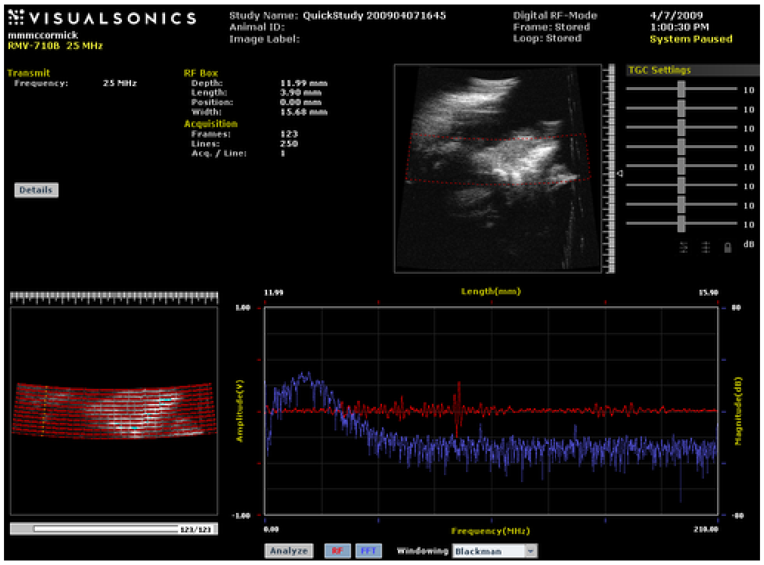
2 File Storage and Metadata Extraction

RF acquisition is performed in M-mode and is considerably slower than B-mode frame rates. RF acquisition can be collected as single 2D frames, but 3D data can be acquired with the optional high-precision stepper motor. Data is stored in a pair of non-standard plain text and binary files that contain system settings and raw data, respectively. A dataset contains a B-mode and saturation image of the region-of-interest (ROI) window for the first frame along with the RF data.

Data collection is well integrated into the user interface of the machine, but buffer limits on the A/D card limit the length of acquisition to a subset of the field of view, as illustrated in Fig. 3.

When data files are exported in *RAW* format, two files are saved for each acquisition. A file with the *.rdb* extension denotes a binary format file. This *.rdb* contains three images in sequence: two image of the ROI selected in the scout window followed by the RF data. Regardless of whether a 3D acquisition occurs, the ROI images are always 2D. These images contain the content found in the real-time preview of the ROI prior to scan conversion. The first image is a B-Mode image in two byte unsigned integer format written sequentially in an A-line format. All binary data is in Little Endian format significant byte (MSB). A saturation image with the same size as the B-Mode image follows. The saturation image is again in two-byte unsigned integer format, but the content is boolean; a non-zero sample indicates that the digitizer was saturated at that datum.

The ROI data is followed by RF data in the acquired volume of interest. Unlike the ROI images, the RF data is in a two-byte signed integer format. The RF data is written sequentially by samples within an A-line,


VISUALSONICS
mmmccormick
RMV-710B 25 MHz

Study Name: QuickStudy 200904071645
Animal ID:
Image Label:

Digital RF-Mode
Frame: Stored
Loop: Stored

4/7/2009
1:00:30 PM
System Paused

Transmit
Frequency: 25 MHz

RF Box
Depth: 11.99 mm
Length: 3.90 mm
Position: 0.00 mm
Width: 15.68 mm

Acquisition
Frames: 123
Lines: 250
Acq. / Line: 1

TGC Settings: 10, 10, 10, 10, 10, 10, 10, 10 dB

Windowing: Blackman

Figure 3: System B-Mode is shown in the upper right with a red overlay of the RF collection ROI. The lower right shows the ROI window B-Mode and saturation content, which is saved in the acquired file along with the RF data. The time and frequency content a selected A-line in the ROI window is shown in the lower right.

followed by A-lines within a frame, followed by the frame in the volume. There is more than one pulse-echo data segment saved for each A-line. To allow signal averaging with the transducer fixed in a given position, an average A-line signal is saved followed by the individual pulse-echo signals. Information on the number of A-lines, averaged number of signals, etc. that are required to read, analyze, and scan convert the binary data must be extracted from the metadata header file.

Each *.rdb* binary file has a *.rdi* metadata header file associated with it. This file has three sections, Image Info, Image Data, and Image Parameters. The Image Info section contains information related to the current acquisition such as an operator defined labels, the number of frames, or the acquisition time. The Image Data section contains information on byte offsets to A-line locations in the binary file for the ROI B-mode, ROI saturation, and the RF data. Finally, the Image Parameters section contains system settings such as the transmit pulse settings, time-gain compensation (TGC) settings, characteristics of the current transducer, ECG settings, or the stepper motor position. Example content from an *.rdi* is shown below.

```
"=== IMAGE INFO ==="
"Study Name","QuickStudy 201001201737"
"Image Id","54HTKMSSMJCKL2JSKMMF1TPCDW"
"Image Label",""
"Image Frames","136"
"Image Lines","250"
"Image Acquisition Per Line","1"
"Image Acquisition Size","4256","bytes"
...
"=== IMAGE DATA ==="
"ROI Data Offset - B-Mode","0","bytes"
"ROI Data Size - B-Mode","73472","bytes"
```

```
"ROI Data Offset - Saturation","73472","bytes"
"ROI Data Size - Saturation","73472","bytes"
"Image Data Offset - Frame 0 - Line 0 - Acq 0","146944","bytes"
"Image Data Offset - Frame 0 - Line 1 - Acq 0","151200","bytes"
....
"=== IMAGE PARAMETERS ==="
"RF-Mode/ActiveProbe/Notes","Rat Cardiology"
"RF-Mode/ActiveProbe/Sample-Time","154","s"
"RF-Mode/BModeSoft/V-Relative-Frame-Rate","4"
"RF-Mode/ActiveProbe/Focal-Length","15","mm"
```

Each parameter is described on a line with two to three fields delimited by quotations and commas. The first field is generally a key name. In the Image Parameters section, this can take a hierarchical form delimited by a forward slash. The second field is the value for the given key, which will contain an array of comma delimited numbers for an array of values. An optional third field contains the units for the value. The voluminous amount of Image Parameters results in a large file; typical size is 35,000 lines.

Parameters for parsing the binary file can be found or derived from the Image Info section, which makes the Image Data section largely redundant. Parametric image formation and scan conversion relies on content dispersed throughout the Image Parameters section. To facilitate the extraction of values of a given key and conversion from plain text to the appropriate data type, a library was developed to parse the header content into an intermediate eXtensible Markup Language (XML) form [4]. The advantages of XML for this data set includes its broad support under diverse tools and programming languages as an open standard, a native text-based and hierarchical form, and some explicit specification of data types. The structure of the *.rdi* is transformed into an XML hierarchy by considering the main three sections as top level elements and division and sorting of the keys in the Image Parameters section into a hierarchy of child elements. This structure was determined by parsing an example header file instance with a Python script and defined using XMLSchema [5].

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<rdi xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="rdi.

<image_info>
  <Study_Name>QuickStudy 201001201737</Study_Name>
  <Image_Id>54HTKMSSMJCKL2JSKMMF1TPCDW</Image_Id>
  <Image_Label/>
  <Image_Frames>136</Image_Frames>
  <Image_Lines>250</Image_Lines>
  <Image_Acquisition_Per_Line>1</Image_Acquisition_Per_Line>
  <Image_Acquisition_Size>4256</Image_Acquisition_Size>
  <Animal_ID/>
  <Acquisition_Mode>Digital RF-Mode</Acquisition_Mode>
  <Acquisition_Date>1/20/2010</Acquisition_Date>
  <Acquisition_Time>5:42:14 PM</Acquisition_Time>
  <Acquisition_Operator>Default Operator</Acquisition_Operator>
</image_info>

<image_data/>
```

```
<image_parameters>
  <RF-Mode>
    <ActiveProbe>
      <Notes>Rat Cardiology</Notes>
      <Sample-Time units="s">154</Sample-Time>
      <Focal-Length units="mm">15</Focal-Length>
      <Acceleration-Limit-Slope>0</Acceleration-Limit-Slope>
```

The content is imported in C++ into a Xerces-C++ [6] object, from which it can be serialized into an XML file, above, to be easily processed by other programs. Alternatively, it can be transformed into a Hyper-Text Markup Language (HTML) to be easily examined in web browsers, as shown in Fig. 4. Transformation is specified through an EXtensible Stylesheet Language (XSLT) document and applied in memory with Xalan-C++ [7]. Most importantly, the parameters can be accessed in C++ as native data objects through the use of XML data binding with CodeSynthesis XSD [8] since an XMLSchema has been generated.

The transducer geometry, diagrammed in Fig. 5, requires scan conversion to be processed as a uniform grid. Header file keys that define the geometry include: *PE*, the pivot-to-encoder distance, *RF-Mode/ActiveProbe/Pivot-Encoder-Dist*, *SL*, the shaft-length, *RF-Mode/ActiveProbe/Pivot-Transducer-Fact-Dist*, *DL*, the delay length in the water path from the transducer to start of acquisition, *RF-Mode/RX/V-Delay-Length DD*, the digitizer depth, *RF-Mode/RX/V-Digi-Depth-Imaging* and *EP* the encoder position, *RF-Mode/RfModeSoft/V-Lines-Pos* Note that the last value is an array since it changes with every A-line.

This polar coordinate configuration is common in ultrasound imaging; it also occurs with a curvilinear array or phased array transducer, for example. The radius is given by $r = SL + DL + \frac{sc}{2f_s}$ where s is the sample number along the A-line, c is the assumed speed of sound (usually 1540 m/s), and f_s is the sampling frequency (*RF-Mode/RfModeSoft/SamplesPerSec*). The angle in radians is simply $\theta = EP/PE$. The Cartesian coordinates are then $x_1 = r\cos(\theta)$ and $x_2 = r\sin(\theta)$. For 3D imaging, the only other geometric parameter of importance is the frame spacing, which is found at *RF-Mode/3D/StepSize*.

3 Example

An Insight Toolkit (ITK) [9] `ImageIO` class was written for processing the data with ITK. The data is imported as an ”image”, i.e. geometry of uniform, anisotropic spacing in Cartesian format, with angle and radius information stored in the metadata dictionary for scan conversion after B-Mode or parametric image formation from the A-lines at their original sample locations. Code to perfrom scan conversion is described in a previous article[3].

A file reader is implemented in `itk::VisualSonicsImageIO`. As illustrated in Fig. 3, the depth collected in a single acquisition is limited by the A/D card. Multiple overlapping acquisitions can be combined with the `itk::VisualSonicsSeriesReader` using the metadata from the acquisition.

Example code follows,

```
typedef signed short  PixelType;
typedef itk::Image< PixelType, 3 >ImageType;

typedef itk::ImageFileReader< ImageType > ReaderType;
typedef itk::ImageFileWriter< ImageType > WriterType;
```

```
ReaderType::Pointer reader = ReaderType::New();
WriterType::Pointer writer = WriterType::New();
reader->SetFileName( args.in_file.c_str() );
writer->SetFileName( args.out_file.c_str() );

typedef itk::VisualSonicsSeriesReader< ImageType > VisualSonicsReaderType;
VisualSonicsReaderType::Pointer vsReader = VisualSonicsReaderType::New();
bool isVisualSonicsFile = vsReader->GetImageIO()->CanReadFile( args.in_file.c_str() );
if( isVisualSonicsFile )
  {
  typedef itk::ArchetypeSeriesFileNames NameGeneratorType;
  NameGeneratorType::Pointer nameGenerator = NameGeneratorType::New();
  nameGenerator->SetArchetype( args.in_file.c_str() );
  vsReader->SetFileNames( nameGenerator->GetFileNames() );
  itk::VisualSonicsImageIO * vs_image_io = dynamic_cast< itk::VisualSonicsImageIO * >( vsReade
  if ( args.speed_of_sound > 0. )
    vs_image_io->SetSpeedOfSound( args.speed_of_sound );
  if ( args.start_of_aquisition_speed_of_sound > 0. )
    vs_image_io->SetStartOfAquisitionSpeedOfSound( args.start_of_aquisition_speed_of_sound );
  if ( args.acquisition_speed_of_sound > 0. )
    vs_image_io->SetAcquisitionSpeedOfSound( args.acquisition_speed_of_sound );
  writer->SetInput( vsReader->GetOutput() );
  vsReader->UpdateOutputInformation();
  writer->UseInputMetaDataDictionaryOff();
  itk::ImageIOBase::Pointer imageIO = itk::ImageIOFactory::CreateImageIO( args.out_file.c_str
    itk::ImageIOFactory::WriteMode );
  imageIO->SetMetaDataDictionary( vsReader->GetImageIO()->GetMetaDataDictionary() );

  writer->SetImageIO( imageIO );
  }
else
  {
  writer->SetInput( reader->GetOutput() );
  }

writer->Update();
```

The reader will store the distance from the pivot point to the start of acquisition (the ”Radius”) and the angle of the transducer arm (”Theta”) in the `MetaDataDictionary`.

Source code is distributed with the article along with tests and data for the tests. The code has been built and tested against ITK 3.20.0. Other dependecies are listed in the `README.rst` file distributed with the project. To run the tests without building the project on a Linux system, download the CDE package and run

```
tar xvjf rdi-reader_cde.tar.bz2
cd cde-package/cde-root/tmp/rdib
./ctest.cde
```

References

[1] McCormick, Matthew. *Carotid Plaque Characterization with Medical Ultrasound*. Ph.D. thesis, University of Wisconsin-Madison, 2011. 1

[2] Madsen, Ernest, Frank, Gary, McCormick, Matthew, and Deaner, Meagan. Anechoic Sphere Phantom for Estimating 3-D Resolution of Very High Frequency Ultrasound Scanners. *IEEE Transactions on Ultrasonics Ferroelectrics and Frequency Control*, 57(10):2284–2292, 2010. 1

[3] McCormick, Matthew. Fast Ultrasound B-Mode Implementation for Commodity Hardware. *Insight Journal*, January-Ju, 2010. URL `http://hdl.handle.net/10380/3159`. 1, 3

[4] Bray, Tim, Paoli, Jean, Sperberg-McQueen, C. M., Maler, Eve, and Yergeau, Francois. Extensible Markup Language (XML) 1.0, 2008. URL `http://www.w3.org/TR/REC-xml/`. 2

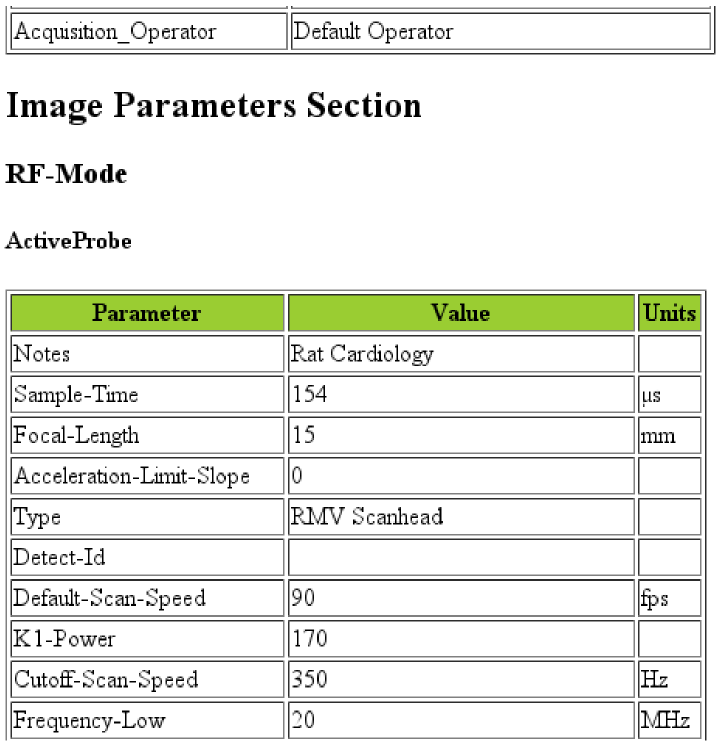
[5] Fallside, David and Walmsley, Priscilla. XML Schema Part 0: Primer Second Edition, 2004. URL `http://www.w3.org/TR/xmlschema-0`. 2

[6] Xerces-C++ XML Parser, 2011. URL `http://xerces.apache.org/xerces-c/`. 2

[7] Xalan-C++ XSLT Processor, 2010. URL `http://xml.apache.org/xalan-c/`. 2

[8] XSD: XML Data Binding for C++, 2010. URL `http://www.codesynthesis.com/products/xsd/`. 2

[9] Yoo, T.S., Ackerman, M.J., Lorensen, W.E., Schroeder, W., Chalana, V., Aylward, S., Metaxes, D., and Whitaker, R. Engineering and Algorithm Design for an Image Processing API: A Technical Report on ITK - The Insight Toolkit. In J. Westwood, editor, *Medicine Meets Virtual Reality*, pages 586–592. IOS Press, Amsterdam, 2002. 3



Acquisition_Operator	Default Operator

Image Parameters Section

RF-Mode

ActiveProbe

Parameter	Value	Units
Notes	Rat Cardiology	
Sample-Time	154	µs
Focal-Length	15	mm
Acceleration-Limit-Slope	0	
Type	RMV Scanhead	
Detect-Id		
Default-Scan-Speed	90	fps
K1-Power	170	
Cutoff-Scan-Speed	350	Hz
Frequency-Low	20	MHz

Figure 4: Rendering of the header file contents after transformation from XML to HTML.

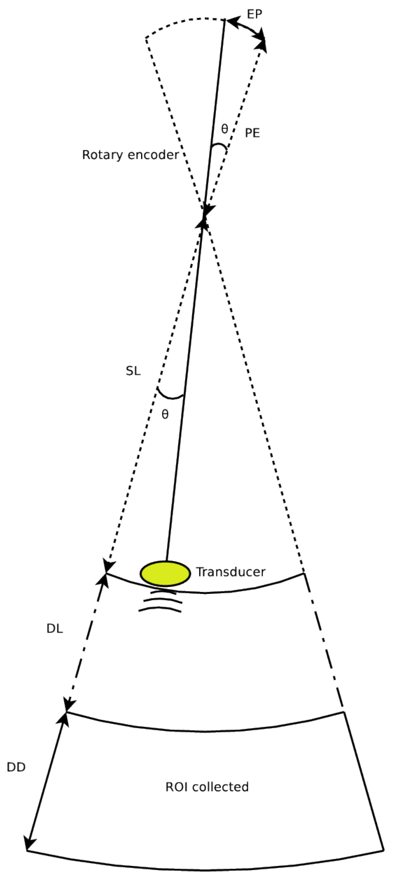


Figure 5: Diagram of the Vevo 770 geometric parameters used in field of view calculations. The transducer sits at the end of a shaft, and the angle of rotation is recorded by a rotary encoder attached to an extension of the shaft across the pivot point. Parameters stored in the metadata file include *PE*, the pivot-to-encoder distance, *SL*, the shaft length, *DL*, the delay length in the water path from the transducer to start of acquisition, *DD*, the digitizer depth, and *EP*, the encoder position.