

# CIS-UltraCal: An open-source ultrasound calibration toolkit

Ryan Kon and Joshua Leven, Kishore Kothapalli, Emad Boctor,  
Gabor Fichtinger, Greg Hager, Russell H. Taylor

Engineering Research Center for Computer Integrated Surgical  
Systems and Technology, Johns Hopkins University

## ABSTRACT

We present an open-source MATLAB toolkit for ultrasound calibration. It has a convenient graphical user interface which sits on top of an extensive API. Calibration using three different phantoms is explicitly supported: the cross-wire phantom,<sup>1</sup> the single-wall phantom,<sup>2</sup> and the Hopkins phantom.<sup>3</sup> Image processing of the Hopkins phantom is automated by making use of techniques from binary morphology, radon transform and RANSAC.<sup>4</sup> Numerous calibration and termination parameters are exposed. It is also modular, allowing one to apply the system to original phantoms by writing a minimum of new code.

**Keywords:** Ultrasound, Calibration, Hopkins, Cross-wire, Single-wall, Phantom, Image processing, Feature extraction

## 1. INTRODUCTION

Ultrasound is a widely required and vastly expanding medium for surgical research. As computers continue to become more powerful, ultrasound cannot only be received at higher speeds, but processed in real time, allowing for many possibilities for surgical augmentation. Such possibilities include ultrasound-guided biopsies,<sup>5,6</sup> ultrasound registration with pre-operative images,<sup>7</sup> and ultrasound-based augmented reality displays.<sup>8</sup>

However, a common prerequisite for all of these advances is an accurately calibrated ultrasound probe. That is, without knowing the position of each ultrasound image in physical space, no connection with the physical world can be made, and thus no connection with the surgical table.

Several different methods of ultrasound calibration have been presented over the years. Most methods commonly used today consist of a non-linear least squares problem resulting from the registration of the ultrasound image with one of a number of different phantoms.<sup>1-3,9-14</sup>

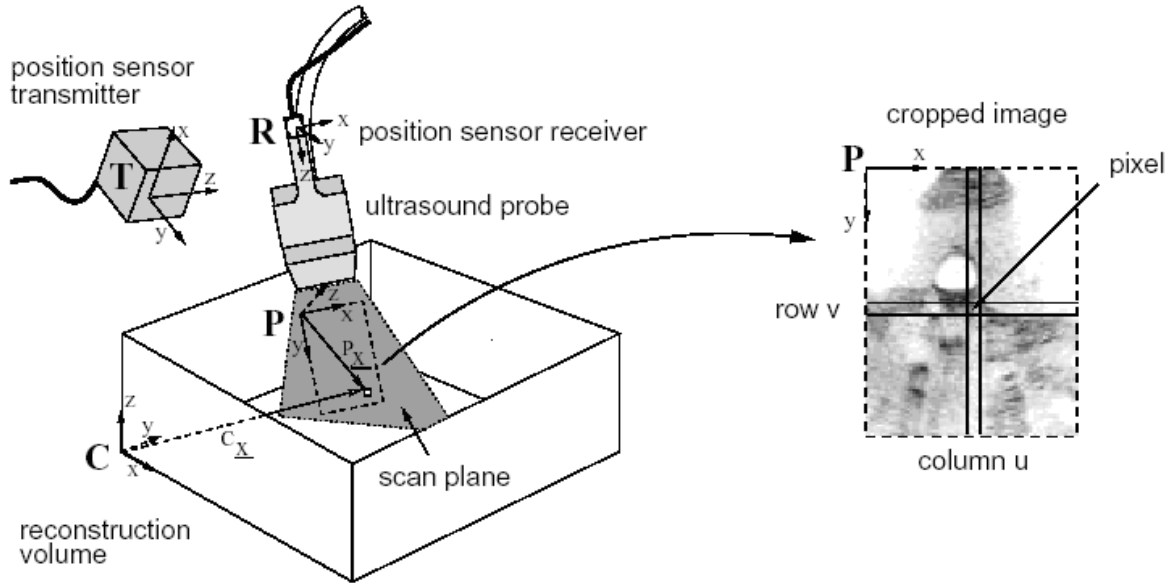
Though the methods of calibration are well understood, the implementation details are often left up to the individual researcher. For example, outliers are a persistent obstacle to an accurate ultrasound calibration, but defining which data points are outliers is not necessarily straight forward.

Certain packages exist that provide an interface for ultrasound calibration, but these systems are not open to be modified or extended.<sup>11</sup> Thus, a researcher intending to perform a new calibration procedure would have to create a new software implementation, presenting a problem to those who are not inclined to do such programming.

To counter this, and to stimulate further ultrasound research, we have built CIS-UltraCal, a toolkit that incorporates many of the current ultrasound calibration methods. Ultrasound calibration typically requires a complicated software component that can be a hindrance to researchers. By providing an highly-modular open-source system, ultrasound calibration research can be more accessible, preventing researchers from doing any unnecessary programming.

---

Further author information: Send correspondence to {ryankon, josh.leven}@gmail.com



**Figure 1.** An overview of the calibration process. The ultrasound image frame  $P$ , the ultrasound probe frame  $R$ , the tracker frame  $T$ , and the reconstruction volume frame  $C$  are all apparent. (Courtesy of R. Prager)

## 2. BACKGROUND

### 2.1. Mathematical Background

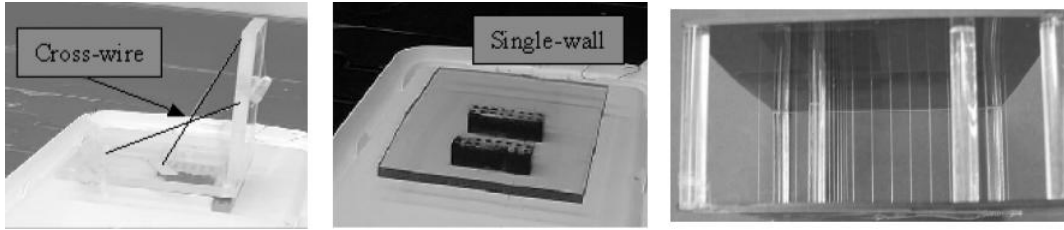
Ultrasound calibration, in broad terms (Figure 1), involves taking enough images of an object with known dimensions, in which a transformation from the image space to the object space is possible. How this is done depends, somewhat, on the method used. The most prevalent set-up consists of a tracking device, an ultrasound probe with a tracking probe attached, and a known object called a phantom. The resulting four reference frames are those of: the ultrasound image  $P$ , the ultrasound probe  $R$ , the tracking device  $T$ , and the phantom  $C$ . The goal is to produce the correct transformation from the ultrasound image to the phantom reference frame. The overall transformation can be expressed as a single equation of homogeneous transformations:

$$C_X = {}^C T_T {}^T T_R {}^R T_P \begin{pmatrix} s_x u \\ s_y v \\ 0 \\ 1 \end{pmatrix}$$

The equation adopts the convention that a transformation  ${}^L T_M$  is the *from* the frame  $M$  *to* the frame  $L$ . Further,  $s_x$  and  $s_y$  are the scale factors between image pixels and mm, and  $u$  and  $v$  are the  $x$  and  $y$  coordinates of the feature in question in the ultrasound image frame.

By using a phantom of known geometry, and features that are distinct within an ultrasound image,  $C_x$  can be known for specific pixels in the ultrasound image. Since  ${}^T T_R$  is known because of the tracking device, that leaves two transformations and two scaling factors or fourteen parameters that remain to be determined.

We decided to use the Levenberg-Marquardt algorithm to solve the resulting non-linear least squares problem. This method has been shown to provide accurate results with each of the phantoms we use in our system.<sup>1-3</sup> We refer you to the above reference for details of the algorithm with respect to the phantoms we have incorporated into CIS-UltraCal.



**Figure 2.** The Cross-wire (left), Single-wall (center), and Hopkins (right) phantoms. (Boctor 2003)

## 2.2. The Phantoms

CIS-UltraCal incorporates three distinct phantoms (Figure 2) explicitly into the system. Each of the cross-wire, single-wall, and Hopkins phantoms uses the Levenberg-Marquardt algorithm for its calibration.

### 2.2.1. Cross-wire Phantom

The cross-wire phantom<sup>1</sup> consists of two thin intersecting wires that are fixed and taught. During calibration this intersection,  $C_X$ , is treated as the origin of the phantom coordinate system. This allows each processed image of cross-wire to produce three equations associated with the three zeros for the Levenberg-Marquardt algorithm:

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = {}^C T_T^T T_R^R T_P \begin{pmatrix} s_x u \\ s_y v \\ 0 \\ 1 \end{pmatrix}$$

Since the intersection point has no orientation, none of the three angles of  ${}^C T_T$  are identifiable. Thus the cross-wire phantom allows 11 parameters to be solved for.

### 2.2.2. Single-Wall Phantom

The single wall<sup>2</sup> is exactly that — one flat surface. It is visible as a line in an ultrasound image. The x and y coordinates of any image of the wall, and the rotation along the walls normal vector, are each unidentifiable. Setting the plane of the wall to be  $z = 0$ :

$$\begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix} = {}^C T_T^T T_R^R T_P \begin{pmatrix} s_x u \\ s_y v \\ 0 \\ 1 \end{pmatrix}$$

There is one equation per point, and 11 identifiable parameters.

### 2.2.3. Hopkins Phantom

The Hopkins phantom<sup>3</sup> consists of two parallel plastic plates with parallel nylon lines stretched between the plates, in a pattern to form two orthogonal planes. All parameters except the one along the nylon axes are identifiable, thus 13 parameters can be solved for. Furthermore, the regular shape of this phantom allows for automatic feature extraction.

It produces two functions per point, and has as many as 20 points per image. It is also possible to calculate a calibration accuracy with this method.

### 3. METHODS

The toolkit is written entirely in MATLAB. It provides both a graphical user interface as well as a convenient, highly-modular API. We have created interfaces for point and line image feature extraction, as well as a verification interface for the Hopkins Phantom automatic feature extraction. The Levenberg-Marquardt algorithm is run first on all the data, and then iterated as outliers are removed one by one. Termination is triggered by user-set threshold values of residual error, percentage of outliers, and (in the case of the Hopkins Phantom) reconstruction accuracy. The progress of this iteration is saved to a file which can be used to continue the process at a later time if it is halted.

The toolkit first allows a set of images and associated tracking data to be selected. Then, it provides an interface for selecting various features within these images. In the case of the Hopkins phantom, we introduce and implement an automatic feature extraction algorithm for the Hopkins phantom that allows for unattended processing of images. Finally, the Levenberg-Marquardt method is used for ultrasound calibration of each of the phantoms. The toolkit also allows the adjustment of several termination criteria.

#### 3.1. The Hopkins Phantom Image Feature Extraction Algorithm

The Hopkins phantom image feature extraction algorithm works as follows.

First, the initial image is segmented with a simple thresholding function. Each remaining contiguous region of the image is shrunk to its center of mass. Then radon transforms are used iteratively to determine the principle axis of the Hopkins Phantom in the segmented image. Then, the secondary axis is determined by the same method, using a perpendicular angle to the principle axis as an initial guess.

Next, the RANSAC algorithm is used to determine the position of the individual points in the image and determine two globally optimal axes. A random pair of points are chosen to define an axis; a score is calculated based on the intensity in the original ultrasound image at appropriate intervals along that line.

Once the two axes are determined, their intersection is found. Then, knowing the principle axis and the origin, we can find the whereabouts of the orienting post (an additional wire we typically place one unit from the seventh wire on the principle axis), and use that to determine, finally, which is the positive direction for each of the axes. At this point, we know both the position of each point in the image, and which is its associated wire in the Hopkins Phantom. Thus, the feature extraction is complete.

A summary of the Hopkins Phantom image feature extraction algorithm is shown below.

- Segment Image
  - Threshold / BW
  - Center of mass
- Search for X-axis
  - radon 180 degrees
  - progressively finer radons (10 degree at 0.5 degree intervals, etc.)
- Search for Y-axis
  - assume y-axis will be approx 90 degrees to x-axis
  - radon +/- 10 degrees of that
  - refine radon
- Find intersection of X- and Y-axes
- Use initial image to determine point positions
  - RANSAC best pair of pixels to determine line based on initial-image intensity

- Find “post”
  - Using final lines as guide, find suspected regions
  - Search areas for “post”
- Sort and order points based on post position
- Return X- and Y-axis points

## 4. THE GRAPHICAL USER INTERFACE

The graphical user interface for the CIS-UltraCal toolkit was designed with two main goals: modularity and ease-of-use. There are several different modules implemented within the system. Each phantom has its own associated image interface for correlating pixel coordinates with known features of the phantom. In addition, beneath each of these lies a main interface used to set many of the parameters for the calibration.

### 4.1. The Image Interfaces

An interactive movie player<sup>15</sup> was modified for use as the basis of the image interfaces for the cross-wire and single-wall phantoms. The user is able to display either a single image, or a sequence of images. The resulting “movie” can then be played forward, backward, and/or looped. The player can also be paused, as well as allowing stepping between images a single frame at a time.

The cross-wire version extends this player to allow a single pixel coordinate to be recorded from each of the images in the sequence. The user simply clicks the image at the pixel of the interesting feature. A red cross is then displayed at the clicked point, but only in the current frame of the movie. Multiple points are stored by clicking features within other frames of the movie sequence.

The single-wall version of the movie player allows the user to define a line of interest within any frames of value. By clicking at several different positions along the line in question, a linear regression line is calculated and displayed as well as a cross at each of the points. Multiple lines can be found, but only if each is in a unique frame.

The interface returns not the equation of the line, but the pixel coordinates of those points on the line closest to the pixels clicked. In this way, finite lines are never accidentally exceeded.

Though these two image interfaces are ideal for these two specific phantoms, they are built to be much more general. The cross-wire image interface can be considered an interface to locate a point feature in a single image or sequence of images — regardless of the phantom type. Likewise, the single-wall image interface can be reused to locate some linear feature in ultrasound images. This generality allows researchers to reuse these interfaces with other phantom types, as well as newly-created phantom types as well.

The Hopkins phantom image interface is significantly different. Because automatic feature extraction is implemented for the Hopkins phantom, CIS-UltraCal determines the pixel coordinates of the features in each image. The user may opt for a semi-automatic process. We have created a simple interface to allow the user to verify the correctness of the extraction. This interface, as well an example of the single-wall interface, is shown in Figure 3.

### 4.2. The Main Interfaces

In order to create an interface which requires many successive steps, but is still easy to use, we decided to keep parts of the interface hidden until they were needed, which helps to guide the user through the flow of an ultrasound calibration (Figure 4). In this section, we step through a cross-wire calibration. The progression of the interface is shown in Figure 5.

The interface is clearly divided into sections, and only the first section is presented upon starting CIS-UltraCal. This first section allows the user to begin a new calibration file or load a saved one. If a new calibration is desired, the user can then pick from one of the phantoms in the list. In this case, we use the cross-wire phantom. This section also includes a clear all button, that allows the user to reset the entire process at any time.

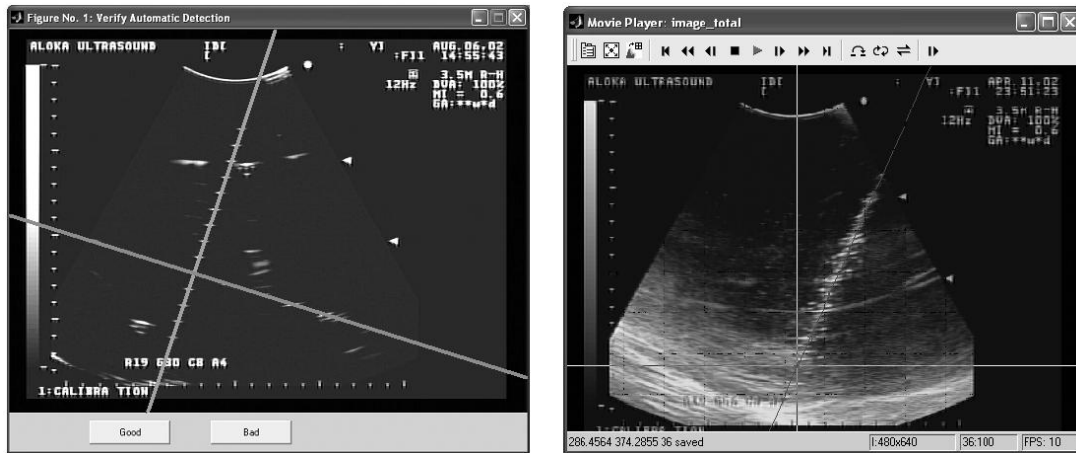


Figure 3. The Hopkins phantom automatic feature extraction results (left) and the Single-wall feature selection (right).

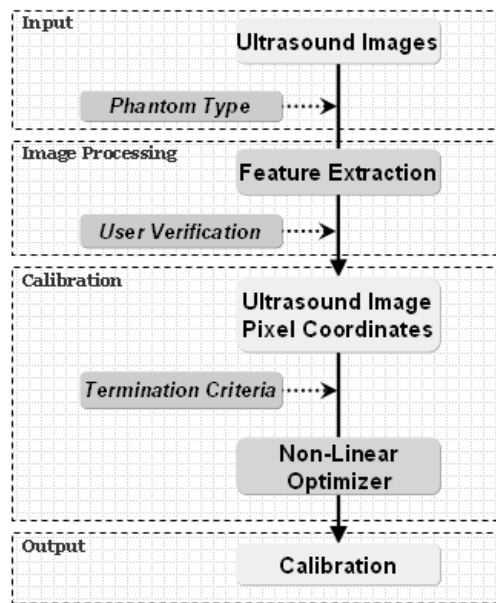


Figure 4. A flowchart of the calibration process



**Figure 5.** Cross-wire main interface

Next, the sections “Input Parameters” and “Image Processing” appear. The “Input Parameters” section allows the user to choose parent directory containing the data to be used. Additionally, the user must specify the tracking data filename and the image extension, which can be done by using the button immediately to the right, and selecting an example file. Once the parameters have been selected, the user can move to image processing. The user can then choose which directories to examine. After doing so, the “Examine First Image Set” button is activated. Upon pressing it, the appropriate image interface pops-up with the first set of images.

Sets of images are defined by their directory structure — all the images in the same directory are considered in the same image set.

When the user finishes examining the first image set, they are returned to the main interface from where they can press the same button, now labeled “Examine Next Image Set” to continue the calibration.

Once the user has completed the necessary image processing, they are presented with the final two sections of the interface. The user must choose between one of the three available termination criteria — residual, accuracy, or percentage outliers — and set an appropriate threshold.

The final decision left to the user is whether to calculate the Jacobian matrix explicitly, or to allow the built-in MATLAB function to estimate it. Once decided, the user can press “Calibrate” and the non-linear least square optimizer finds the calibration.

Because each phantom allows for a different set of parameters, each is associated with a slightly different interface for the user. Though functionally different, each interface is presented with the same general layout, providing a logical and familiar flow for the user regardless of which phantom is used.

## 5. RESULTS AND DISCUSSION

CIS-UltraCal is an open-source ultrasound calibration toolkit that provides its users with a convenient method of ultrasound calibration using current methods, and it provides a foundation for development of future ultrasound calibration techniques. This toolkit can be useful for ultrasound researchers of all types. Those whose research requires ultrasound calibration but does not focus on it will be able to take advantage of the graphical user interface to perform the calibration. Those researchers whose work focuses on ultrasound calibration will be able to use the underlying API, as well as integrate new ultrasound calibration techniques.

Testing of the segmentation procedure was done on synthetic images as well as actual images. On both classes of images with the Hopkins phantom the procedure yields a mean error of within 1 pixel. Previously, such procedures required manual segmentation. On a 3.0 GHz machine with 1 GB RAM, this algorithm runs to completion in approximately 15 seconds per image, which is a substantial improvement.

Additionally, the toolkit provides substantial time savings. Because the cross-wire and single-wall phantoms have their own clicking interfaces that automatically record coordinates for the calibration, users still using these phantoms should experience further increases in efficiency, in addition to the time saved from not having to implement the software package.

## 6. CONCLUSION

CIS-UltraCal is an enabling technology. We hope for it to be adopted into the ultrasound community as the Camera Calibration Toolbox for MATLAB<sup>16</sup> has been adopted by the computer vision community. Our system is both a convenient toolkit for the current state of ultrasound calibration as well as a platform on which to base the development of future calibration methods.

## REFERENCES

1. P. R. Detmer, G. Bashein, T. Hodges, K. W. Beach, E. P. Filer, D. H. Burns, , and D. S. Jr., "3d ultrasonic image feature localization based on magnetic scanhead tracking: in vitro calibration and validation," *Ultrasound in Medicine and Biology* **20(9)**, pp. 923–936, 1994.
2. R. W. Prager, R. N. Rohling, A. H. Gee, and L. Berman, "Rapid calibration for 3-d freehand ultrasound," *Ultrasound in Med. Biol.* **24(6)**, pp. 855–869, 1998.
3. E. M. Boctor, A. Jain, M. A. Choti, R. H. Taylor, and G. Fichtinger, "A rapid calibration method for registration and 3d tracking of ultrasound images using spatial localizer," in *Medical Imaging 2003: Ultrasonic Imaging and Signal Processing*, M. H. Loew, ed., *Proc. SPIE* **5035**, pp. 521–532, February 15, 2003.
4. M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Comm. of the ACM* **24**, pp. 381–395, 1981.
5. B. L. D.G. Gobbi and T. Peters, "Real-time 3d ultrasound for intra-operative surgical guidance," in *Medical Imaging 2001*, M. H. Loew, ed., *Proc. SPIE* **4319**, pp. 264–271, 2001.
6. A. Fenster, K. Surry, G. Mills, and D. Downey, "3d ultrasound guided breast biopsy system," *Ultrasonics* **42(1-9)**, pp. 769–774, 2004.
7. S. W. Julien Jomier and S. R. Aylward, "Vascular image registration for intra-operative 3d ultrasound annotation," 2004.
8. H. Fuchs, M. A. Livingston, R. Raskar, D. Colucci, K. Keller, A. State, J. R. Crawford, P. Rademacher, S. H. Drake, and A. A. Meyer, "Augmented reality visualization for laparoscopic surgery," in *Medical Imaging 2001, Proceedings of First International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 11–13, October 1998.
9. J. Carr, *Surface Reconstruction in 3D Medical Imaging*. PhD thesis, University of Canterbury, Christchurch, New Zealand, 1996.
10. A. Fenster, D. B. Downey, and N. H. Cardinal, "Rapid calibration for 3-d freehand ultrasound," *Physics in medicine and biology* **46**, pp. 67–99, 2001.
11. R. W. Prager, A. H. Gee, and L. Berman, "Stradx: real-time acquisition and visualisation of freehand three-dimensional ultrasound," *Medical Image Analysis* **3(2)**, pp. 129–140, 1999.
12. N. Pagoulatos, D. R. Haynor, and Y. Kim, "A fast calibration method for 3-d tracking of ultrasound images using a spatial localizer," *Ultrasound in Med. Biol* **27(9)**, pp. 1219–1229, 2001.
13. P. H. F. Rousseau and C. Barillot, "A fully automatic calibration procedure for freehand 3d ultrasound," 2002.
14. D. M. Muratore and R. L. G. Jr., "Beam calibration without a phantom for creating a 3d free-hand ultrasound system," *Ultrasound in Med. Biol* **27(11)**, pp. 1557–1566, 2001.
15. D. Orofino, "Interactive matlab movie player." MATLAB Central File Exchange, 2003.
16. J.-Y. Bouguet, "Camera calibration toolbox for matlab." Computer Vision Research Group, Dept. of Electrical Engineering, California Institute of Technology.