

(19) World Intellectual Property
Organization
International Bureau



(43) International Publication Date
21 May 2004 (21.05.2004)

PCT

(10) International Publication Number
WO 2004/041094 A2

(51) International Patent Classification⁷: **A61B 8/14**

(21) International Application Number:
PCT/US2003/035252

(22) International Filing Date:
5 November 2003 (05.11.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/423,881 5 November 2002 (05.11.2002) US
PCT/US03/24368 8 January 2003 (08.01.2003) US
60/470,525 12 May 2003 (12.05.2003) US
10/433,126 20 May 2003 (20.05.2003) US
10/633,186 31 July 2003 (31.07.2003) US
PCT/US03/14785 5 September 2003 (05.09.2003) US

(71) Applicant (for all designated States except US): **DIAGNOSTIC ULTRASOUND CORPORATION** [US/US];
21222-30th Drive SE, Suite 120, Bothell, WA 98021 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **CHALANA, Vikram** [US/US]; 15302 34th Dr. SE., Mill Creek, WA 98027 (US). **DUDYCHA, Stephen** [CA/US]; 8926 NE 178th St, Bothell, WA 98011 (US). **MCMORROW, Gerald, J.** [US/US]; 11810 NE 102nd Place, Kirkland, WA 98033 (US).

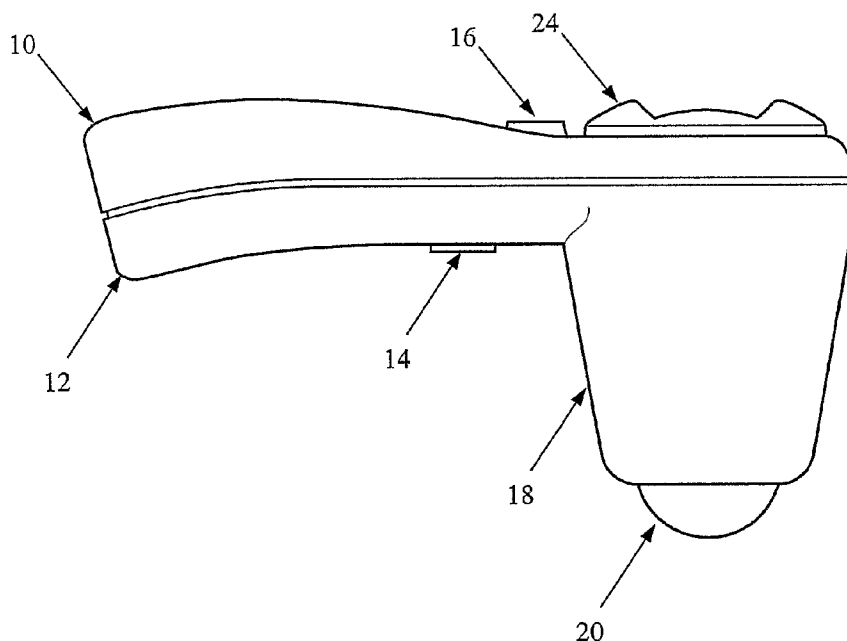
(74) Agent: **BYRNE, Mark, D.**; Black Lowe & Graham PLLc, 701 Fifth Avenue, Suite 4800, Seattle, WA 98104 (US).

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States (*regional*): ARIPO patent (BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE,

[Continued on next page]

(54) Title: 3D ULTRASOUND-BASED INSTRUMENT FOR NON-INVASIVE MEASUREMENT OF AMNIOTIC FLUID VOLUME



(57) Abstract: A hand-held 3D ultrasound instrument is disclosed which is used to non-invasively and automatically measure amniotic fluid volume in the uterus requiring a minimum of operator intervention. Using a 2D image-processing algorithm, the instrument gives automatic feedback to the user about where to acquire the 3D image set. The user acquires one or more 3D data sets covering all of the amniotic fluid in the uterus and this data is then processed using an optimized 3D algorithm to output the total amniotic fluid volume corrected for any fetal head brain volume contributions.

WO 2004/041094 A2



ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Published:

- *without international search report and to be republished upon receipt of that report*

5

10

**3D ULTRASOUND-BASED INSTRUMENT FOR NON-INVASIVE
MEASUREMENT OF AMNIOTIC FLUID VOLUME**

INVENTORS

Vikram Chalana

15

Stephen Dudycha

Gerald J. McMorro

20

PRIORITY CLAIM

This application is a continuation-in-part of and claims priority to PCT application serial number PCT/US03/24368 filed August 1, 2003, which claims priority to U.S. provisional patent application serial number 60/423,881 filed November 5, 2002 and U.S. provisional patent application serial number 60/400,624 filed August 2, 2002.

This application is also a continuation-in-part of and claims priority to PCT Application Serial No. PCT/US03/14785 filed May 9, 2003, which is a continuation of U.S. Patent application serial number 10/165,556 filed June 7, 2002.

This application is also a continuation-in-part of and claims priority to U.S. patent application serial number 10/633,186 filed July 7, 2003 which claims priority to U.S. provisional patent application serial number 60/423,881 filed November 5, 2002 and U.S. provisional patent application serial number 60/400,624 filed August 2, 2002, and to U.S. patent application serial number 10/443,126 filed May 20, 2003 which claims priority to U.S. provisional patent application serial number 60/423,881 filed November 5, 2002 and to U.S. provisional application 60/400,624 filed August 2, 2002.

This application also claims priority to U.S. provisional patent application serial number 60/470,525 filed May 12, 2003, and to U.S. patent application serial number 10/165,556 filed June 7, 2002. All of the above applications are herein incorporated by reference in their entirety as if fully set forth herein.

FIELD OF THE INVENTION

This invention pertains to the field of obstetrics, particularly to ultrasound-based non-invasive obstetric measurements.

BACKGROUND OF THE INVENTION

Measurement of the amount of Amniotic Fluid (AF) volume is critical for assessing the kidney and lung function of a fetus and also for assessing the placental function of the mother. Amniotic fluid volume is also a key measure to diagnose conditions such as polyhydramnios (too much AF) and oligohydramnios (too little AF). Polyhydramnios and oligohydramnios are diagnosed in about 7-8% of all pregnancies and these conditions are of concern because they may lead to birth defects or to delivery complications. The amniotic

fluid volume is also one of the important components of the fetal biophysical profile, a major indicator of fetal well-being.

The currently practiced and accepted method of quantitatively estimating the AF volume is from two-dimensional (2D) ultrasound images. The most commonly used measure is known as the use of the amniotic fluid index (AFI). AFI is the sum of vertical lengths of the largest AF pockets in each of the 4 quadrants. The four quadrants are defined by the umbilicus (the navel) and the linea nigra (the vertical mid-line of the abdomen). The transducer head is placed on the maternal abdomen along the longitudinal axis with the patient in the supine position. This measure was first proposed by Phelan et al (Phelan JP, Smith CV, Broussard P, Small M., "Amniotic fluid volume assessment with the four-quadrant technique at 36-42 weeks' gestation," J Reprod Med Jul; 32(7): 540-2, 1987) and then recorded for a large normal population over time by Moore and Cayle (Moore TR, Cayle JE. "The amniotic fluid index in normal human pregnancy," Am J Obstet Gynecol May; 162(5): 1168-73, 1990).

Even though the AFI measure is routinely used, studies have shown a very poor correlation of the AFI with the true AF volume (Sepulveda W, Flack NJ, Fisk NM., "Direct volume measurement at midtrimester amniocentesis in relation to ultrasonographic indexes of amniotic fluid volume," Am J Obstet Gynecol Apr; 170(4): 1160-3, 1994). The correlation coefficient was found to be as low as 0.55, even for experienced sonographers. The use of vertical diameter only and the use of only one pocket in each quadrant are two reasons why the AFI is not a very good measure of AF Volume (AFV).

Some of the other methods that have been used to estimate AF volume include:

Dye dilution technique. This is an invasive method where a dye is injected into the AF during amniocentesis and the final concentration of dye is measured from a sample of AF removed after several minutes. This technique is the accepted gold standard for AF volume measurement; however, it is an invasive and cumbersome method and is not routinely used.

5 **Subjective interpretation from ultrasound images.** This technique is obviously dependent on observer experience and has not been found to be very good or consistent at diagnosing oligo- or poly-hydramnios.

Vertical length of the largest single cord-free pocket. This is an earlier variation of the AFI where the diameter of only one pocket is measured to estimate the AF volume.

10 **Two-diameter areas of the largest AF pockets in the four quadrants.** This is similar to the AFI; however, in this case, two diameters are measured instead of only one for the largest pocket. This two diameter area has been recently shown to be better than AFI or the single pocket measurement in identifying oligohydramnios (Magann EF, Perry KG Jr, Chauhan SP, Anfanger PJ, Whitworth NS, Morrison JC., "The accuracy of ultrasound
15 evaluation of amniotic fluid volume in singleton pregnancies: the effect of operator experience and ultrasound interpretative technique," J Clin Ultrasound, Jun; 25(5):249-53, 1997).

The measurement of various anatomical structures using computational constructs are described, for example, in U.S patent 6,346,124 to Geiser, et al. (Autonomous Boundary
20 Detection System For Echocardiographic Images). Similarly, the measurement of bladder structures are covered in U.S. patent 6,213,949 to Ganguly, et al. (System For Estimating Bladder Volume) and U.S. patent to 5,235,985 to McMorro, et al., (Automatic Bladder

Scanning Apparatus). The measurement of fetal head structures is described in U.S. patent 5,605,155 to Chalana, et al., (Ultrasound System For Automatically Measuring Fetal Head Size). The measurement of fetal weight is described in U.S. patent 6,375,616 to Soferman, et al. (Automatic Fetal Weight Determination).

5 Pertaining to ultrasound-based determination of amniotic fluid volumes, Segiv et al. (in Segiv C, Akselrod S, Tepper R., "Application of a semiautomatic boundary detection algorithm for the assessment of amniotic fluid quantity from ultrasound images." *Ultrasound Med Biol*, May; 25(4): 515-26, 1999) describe a method for amniotic fluid segmentation from 2D images. However, the Segiv et al. method is interactive in nature and the identification of
10 amniotic fluid volume is very observer dependent. Moreover, the system described is not a dedicated device for amniotic fluid volume assessment.

 Grover et al. (Grover J, Mentakis EA, Ross MG, "Three-dimensional method for determination of amniotic fluid volume in intrauterine pockets." *Obstet Gynecol*, Dec; 90(6): 1007-10, 1997) describe the use of a urinary bladder volume instrument for amniotic fluid
15 volume measurement. The Grover et al. method makes use of the bladder volume instrument without any modifications and uses shape and other anatomical assumptions specific to the bladder that do not generalize to amniotic fluid pockets. Amniotic fluid pockets having shapes not consistent with the Grover et al. bladder model introduces analytical errors. Moreover, the bladder volume instrument does not allow for the possibility of more than one amniotic fluid
20 pocket in one image scan. Therefore, the amniotic fluid volume measurements made by the Grover et al. system may not be correct or accurate.

None of the currently used methods for AF volume estimation are ideal. Therefore, there is a need for better, non-invasive, and easier ways to accurately measure amniotic fluid volume.

SUMMARY OF THE INVENTION

5 The preferred form of the invention is a three dimensional (3D) ultrasound-based system and method using a hand-held 3D ultrasound device to acquire at least one 3D data set of a uterus and having a plurality of automated processes optimized to robustly locate and measure the volume of amniotic fluid in the uterus without resorting to pre-conceived models of the shapes of amniotic fluid pockets in ultrasound images. The automated process uses a
10 plurality of algorithms in a sequence that includes steps for image enhancement, segmentation, and polishing.

A hand-held 3D ultrasound device is used to image the uterus trans-abdominally. The user moves the device around on the maternal abdomen and, using 2D image processing to locate the amniotic fluid areas, the device gives feedback to the user about where to acquire
15 the 3D image data sets. The user acquires one or more 3D image data sets covering all of the amniotic fluid in the uterus and the data sets are then stored in the device or transferred to a host computer.

The 3D ultrasound device is configured to acquire the 3D image data sets in two formats. The first format is a collection of two-dimensional scanplanes, each scanplane
20 being separated from the other and representing a portion of the uterus being scanned. Each scanplane is formed from one-dimensional ultrasound A-lines confined within the limits of the 2D scanplane. The 3D data sets is then represented as a 3D array of 2D scanplanes. The

3D array of 2D scanplanes is an assembly of scanplanes, and may be assembled into a translational array, a wedge array, or a rotational array.

Alternatively, the 3D ultrasound device is configured to acquire the 3D image data sets from one-dimensional ultrasound A-lines distributed in 3D space of the uterus to form a
5 3D scancone of 3D-distributed scanline. The 3D scancone is not an assembly of 2D scanplanes.

The 3D image datasets, either as discrete scanplanes or 3D distributed scanlines, are then subjected to image enhancement and analysis processes. The processes are either implemented on the device itself or is implemented on the host computer. Alternatively, the
10 processes can also be implemented on a server or other computer to which the 3D ultrasound data sets are transferred.

In a preferred image enhancement process, each 2D image in the 3D dataset is first enhanced using non-linear filters by an image pre-filtering step. The image pre-filtering step includes an image-smoothing step to reduce image noise followed by an image-sharpening
15 step to obtain maximum contrast between organ wall boundaries.

A second process includes subjecting the resulting image of the first process to a location method to identify initial edge points between amniotic fluid and other fetal or maternal structures. The location method automatically determines the leading and trailing regions of wall locations along an A-mode one-dimensional scan line.

20 A third process includes subjecting the image of the first process to an intensity-based segmentation process where dark pixels (representing fluid) are automatically separated from bright pixels (representing tissue and other structures).

In a fourth process, the images resulting from the second and third step are combined to result in a single image representing likely amniotic fluid regions.

In a fifth process, the combined image is cleaned to make the output image smooth and to remove extraneous structures such as the fetal head and the fetal bladder.

5 In a sixth process, boundary line contours are placed on each 2D image. Thereafter, the method then calculates the total 3D volume of amniotic fluid in the uterus.

In cases in which uterus are too large to fit in a single 3D array of 2D scanplanes or a single 3D scancone of 3D distributed scanlines, especially as occurs during the second and third trimester of pregnancy, preferred alternate embodiments of the invention allow for
10 acquiring at least two 3D data sets, preferably four, each 3D data set having at least a partial ultrasonic view of the uterus, each partial view obtained from a different anatomical site of the patient.

In one embodiment a 3D array of 2D scanplanes is assembled such that the 3D array presents a composite image of the uterus that displays the amniotic fluid regions to provide
15 the basis for calculation of amniotic fluid volumes. In a preferred alternate embodiment, the user acquires the 3D data sets in quarter sections of the uterus when the patient is in a supine position. In this 4-quadrant supine procedure, four image cones of data are acquired near the midpoint of each uterine quadrant at substantially equally spaced intervals between quadrant centers. Image processing as outlined above is conducted for each quadrant image,
20 segmenting on the darker pixels or voxels associated with amniotic fluid. Correcting algorithms are applied to compensate for any quadrant-to-quadrant image cone overlap by registering and fixing one quadrant's image to another. The result is a fixed 3D mosaic

image of the uterus and the amniotic fluid volumes or regions in the uterus from the four separate image cones.

Similarly, in another preferred alternate embodiment, the user acquires one or more 3D image data sets of quarter sections of the uterus when the patient is in a lateral position.

5 In this multi-image cone lateral procedure, each image cones of data are acquired along a lateral line of substantially equally spaced intervals. Each image cone are subjected to the image processing as outlined above, with emphasis given to segmenting on the darker pixels or voxels associated with amniotic fluid. Scanplanes showing common pixel or voxel overlaps are registered into a common coordinate system along the lateral line. Correcting
10 algorithms are applied to compensate for any image cone overlap along the lateral line. The result is a fixed 3D mosaic image of the uterus and the amniotic fluid volumes or regions in the uterus from the four separate image cones.

In yet other preferred embodiments, at least two 3D scancone of 3D distributed scanlines are acquired at different anatomical sites, image processed, registered and fused
15 into a 3D mosaic image composite. Amniotic fluid volumes are then calculated.

The system and method further provides an automatic method to detect and correct for any contribution the fetal head provides to the amniotic fluid volume.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1 is a side view of a microprocessor-controlled, hand-held ultrasound
20 transceiver;

FIGURE 2A is a is depiction of the hand-held transceiver in use for scanning a patient;

FIGURE 2B is a perspective view of the hand-held transceiver device sitting in a communication cradle;

FIGURE 3 depicts a schematic view of a plurality of transceivers in connection with a server;

5 FIGURE 4 depicts a schematic view of a plurality of transceivers in connection with a server over a network;

FIGURE 5A a graphical representation of a plurality of scan lines forming a single scan plane;

10 FIGURE 5B is a graphical representation of a plurality of scanplanes forming a three-dimensional array having a substantially conic shape;

FIGURE 5C is a graphical representation of a plurality of 3D distributed scanlines emanating from the transceiver forming a scancone;

15 FIGURE 6 is a depiction of the hand-held transceiver placed laterally on a patient trans-abdominally to transmit ultrasound and receive ultrasound echoes for processing to determine amniotic fluid volumes;

FIGURE 7 shows a block diagram overview of the two-dimensional and three-dimensional Input, Image Enhancement, Intensity-Based Segmentation, Edge-Based Segmentation, Combine, Polish , Output, and Compute algorithms to visualize and determine the volume or area of amniotic fluid;

20 FIGURE 8A depicts the sub-algorithms of Image Enhancement;

FIGURE 8B depicts the sub-algorithms of Intensity-Based Segmentation;

FIGURE 8C depicts the sub-algorithms of Edge-Based Segmentation;

FIGURE 8D depicts the sub-algorithms of the Polish algorithm, including Close, Open, Remove Deep Regions, and Remove Fetal Head Regions;

FIGURE 8E depicts the sub-algorithms of the Remove Fetal Head Regions sub-
5 algorithm;

FIGURE 8F depicts the sub-algorithms of the Hough Transform sub-algorithm;

FIGURE 9 depicts the operation of a circular Hough transform algorithm;

FIGURE 10 shows results of sequentially applying the algorithm steps on a sample
image;

10 FIGURE 11 illustrates a set of intermediate images of the fetal head detection process;

FIGURE 12 presents a 4-panel series of sonographer amniotic fluid pocket outlines and
the algorithm output amniotic fluid pocket outlines;

FIGURE 13 illustrates a 4-quadrant supine procedure to acquire multiple image cones;

FIGURE 14 illustrates an in-line lateral line procedure to acquire multiple image cones;

15 FIGURE 15 is a block diagram overview of the rigid registration and correcting
algorithms used in processing multiple image cone data sets;

FIGURE 16 is a block diagram of the steps in the rigid registration algorithm;

FIGURE 17A is an example image showing a first view of a fixed scanplane;

FIGURE 17B is an example image showing a second view view of a moving scanplane
20 having some voxels in common with the first scanplane;

FIGURE 17C is a composite image of the first (fixed) and second (moving) images;

FIGURE 18A is an example image showing a first view of a fixed scanplane;

FIGURE 18B is an example image showing a second view of a moving scanplane having some voxels in common with the first view and a third view;

FIGURE 18C is a third view of a moving scanplane having some voxels in common
5 with the second view; and

FIGURE 18D is a composite image of the first (fixed), second (moving), and third (moving) views.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The preferred portable embodiment of the ultrasound transceiver of the amniotic fluid
10 volume measuring system are shown in FIGURES 1-4. The transceiver 10 includes a handle
12 having a trigger 14 and a top button 16, a transceiver housing 18 attached to the handle 12,
and a transceiver dome 20. A display 24 for user interaction is attached to the transceiver
housing 18 at an end opposite the transceiver dome 20. Housed within the transceiver 10 is a
single element transducer (not shown) that converts ultrasound waves to electrical signals. The
15 transceiver 10 is held in position against the body of a patient by a user for image acquisition
and signal processing. In operation, the transceiver 10 transmits a radio frequency ultrasound
signal at substantially 3.7 MHz to the body and then receives a returning echo signal. To
accommodate different patients having a variable range of obesity, the transceiver 10 can be
adjusted to transmit a range of probing ultrasound energy from approximately 2 MHz to
20 approximately 10 MHz radio frequencies.

The top button 16 selects for different acquisition volumes. The transceiver is
controlled by a microprocessor and software associated with the microprocessor and a digital

signal processor of a computer system. As used in this invention, the term "computer system" broadly comprises any microprocessor-based or other computer system capable of executing operating instructions and manipulating data, and is not limited to a traditional desktop or notebook computer. The display 24 presents alphanumeric or graphic data indicating the proper or optimal positioning of the transceiver 10 for initiating a series of scans. The transceiver 10 is configured to initiate the series of scans to obtain and present 3D images as either a 3D array of 2D scanplanes or as a single 3D scancone of 3D distributed scanlines. A suitable transceiver is the DCD372 made by Diagnostic Ultrasound. In alternate embodiments, the two- or three-dimensional image of a scan plane may be presented in the display 24.

Although the preferred ultrasound transceiver is described above, other transceivers may also be used. For example, the transceiver need not be battery-operated or otherwise portable, need not have a top-mounted display 24, and may include many other features or differences. The display 24 may be a liquid crystal display (LCD), a light emitting diode (LED), a cathode ray tube (CRT), or any suitable display capable of presenting alphanumeric data or graphic images.

FIGURE 2A is a photograph of the hand-held transceiver 10 for scanning a patient. The transceiver 10 is then positioned over the patient's abdomen by a user holding the handle 12 to place the transceiver housing 18 against the patient's abdomen. The top button 16 is centrally located on the handle 12. Once optimally positioned over the abdomen for scanning, the transceiver 10 transmits an ultrasound signal at substantially 3.7 MHz into the uterus. The transceiver 10 receives a return ultrasound echo signal emanating from the uterus and presents it on the display 24.

FIGURE 2B is a perspective view of the hand-held transceiver device sitting in a communication cradle. The transceiver 10 sits in a communication cradle 42 via the handle 12. This cradle can be connected to a standard USB port of any personal computer, enabling all the data on the device to be transferred to the computer and enabling new programs to be transferred into the device from the computer.

FIGURE 3 depicts a schematic view of a plurality of transceivers in connection with a server. FIGURE 3, by example, depicts each transceiver 10 being used to send probing ultrasound radiation to a uterus of a patient and to subsequently retrieve ultrasound echoes returning from the uterus, convert the ultrasound echoes into digital echo signals, store the digital echo signals, and process the digital echo signals by algorithms of the invention. A user holds the transceiver 10 by the handle 12 to send probing ultrasound signals and to receive incoming ultrasound echoes. The transceiver 10 is placed in the communication cradle 42 that is in signal communication with a computer 52, and operates as an amniotic fluid volume measuring system. Two amniotic fluid volume-measuring systems are depicted as representative though fewer or more systems may be used. As used in this invention, a “server” can be any computer software or hardware that responds to requests or issues commands to or from a client. Likewise, the server may be accessible by one or more client computers via the Internet, or may be in communication over a LAN or other network.

Each amniotic fluid volume measuring systems includes the transceiver 10 for acquiring data from a patient. The transceiver 10 is placed in the cradle 52 to establish signal communication with the computer 52. Signal communication as illustrated is by a wired connection from the cradle 42 to the computer 52. Signal communication between the transceiver 10 and the computer 52 may also be by wireless means, for example, infrared

signals or radio frequency signals. The wireless means of signal communication may occur between the cradle 42 and the computer 52, the transceiver 10 and the computer 52, or the transceiver 10 and the cradle 42.

A preferred first embodiment of the amniotic fluid volume measuring system includes
5 each transceiver 10 being separately used on a patient and sending signals proportionate to the received and acquired ultrasound echoes to the computer 52 for storage. Residing in each computer 52 are imaging programs having instructions to prepare and analyze a plurality of one dimensional (1D) images from the stored signals and transforms the plurality of 1D images into the plurality of 2D scanplanes. The imaging programs also present 3D renderings from the
10 plurality of 2D scanplanes. Also residing in each computer 52 are instructions to perform the additional ultrasound image enhancement procedures, including instructions to implement the image processing algorithms.

A preferred second embodiment of the amniotic fluid volume measuring system is similar to the first embodiment, but the imaging programs and the instructions to perform the
15 additional ultrasound enhancement procedures are located on the server 56. Each computer 52 from each amniotic fluid volume measuring system receives the acquired signals from the transceiver 10 via the cradle 51 and stores the signals in the memory of the computer 52. The computer 52 subsequently retrieves the imaging programs and the instructions to perform the additional ultrasound enhancement procedures from the server 56. Thereafter, each computer
20 52 prepares the 1D images, 2D images, 3D renderings, and enhanced images from the retrieved imaging and ultrasound enhancement procedures. Results from the data analysis procedures are sent to the server 56 for storage.

A preferred third embodiment of the amniotic fluid volume measuring system is similar to the first and second embodiments, but the imaging programs and the instructions to perform the additional ultrasound enhancement procedures are located on the server 56 and executed on the server 56. Each computer 52 from each amniotic fluid volume measuring system receives
5 the acquired signals from the transceiver 10 and via the cradle 51 sends the acquired signals in the memory of the computer 52. The computer 52 subsequently sends the stored signals to the server 56. In the server 56, the imaging programs and the instructions to perform the additional ultrasound enhancement procedures are executed to prepare the 1D images, 2D images, 3D renderings, and enhanced images from the server 56 stored signals. Results from the data
10 analysis procedures are kept on the server 56, or alternatively, sent to the computer 52.

FIGURE 4 is a schematic view of a plurality of amniotic fluid measuring systems connected to a server over the Internet or other network 64. FIGURE 4 represents any of the first, second, or third embodiments of the invention advantageously deployed to other servers and computer systems through connections via the network.

15 FIGURE 5A a graphical representation of a plurality of scan lines forming a single scan plane. FIGURE 5A illustrates how ultrasound signals are used to make analyzable images, more specifically how a series of one-dimensional (1D) scanlines are used to produce a two-dimensional (2D) image. The 1D and 2D operational aspects of the single element transducer housed in the transceiver 10 is seen as it rotates mechanically about an angle ϕ . A scanline 214
20 of length r migrates between a first limiting position 218 and a second limiting position 222 as determined by the value of the angle ϕ , creating a fan-like 2D scanplane 210. In one preferred form, the transceiver 10 operates substantially at 3.7 MHz frequency and creates an approximately 18 cm deep scan line 214 and migrates within the angle ϕ having an angle of

approximately 0.027 radians. A first motor tilts the transducer approximately 60° clockwise and then counterclockwise forming the fan-like 2D scanplane presenting an approximate 120° 2D sector image. A plurality of scanlines, each scanline substantially equivalent to scanline 214 is recorded, between the first limiting position 218 and the second limiting position 222
5 formed by the unique tilt angle ϕ . The plurality of scanlines between the two extremes forms a scanplane 210. In the preferred embodiment, each scanplane contains 77 scan lines, although the number of lines can vary within the scope of this invention. The tilt angle ϕ sweeps through angles approximately between -60° and $+60^\circ$ for a total arc of approximately 120° .

FIGURE 5B is a graphical representation of a plurality of scanplanes forming a three-
10 dimensional array (3D) 240 having a substantially conic shape. FIGURE 5B illustrates how a 3D rendering is obtained from the plurality of 2D scanplanes. Within each scanplane 210 are the plurality of scanlines, each scanline equivalent to the scanline 214 and sharing a common rotational angle θ . In the preferred embodiment, each scanplane contains 77 scan lines, although the number of lines can vary within the scope of this invention. Each 2D sector
15 image scanplane 210 with tilt angle ϕ and range r (equivalent to the scanline 214) collectively forms a 3D conic array 240 with rotation angle θ . After gathering the 2D sector image, a second motor rotates the transducer between 3.75° or 7.5° to gather the next 120° sector image. This process is repeated until the transducer is rotated through 180° , resulting in the cone-shaped 3D conic array 240 data set with 24 planes rotationally assembled in the preferred
20 embodiment. The conic array could have fewer or more planes rotationally assembled. For example, preferred alternate embodiments of the conic array could include at least two scanplanes, or a range of scanplanes from 2 to 48 scanplanes. The upper range of the

scanplanes can be greater than 48 scanplanes. The tilt angle ϕ indicates the tilt of the scanline from the centerline in 2D sector image, and the rotation angle θ , identifies the particular rotation plane the sector image lies in. Therefore, any point in this 3D data set can be isolated using coordinates expressed as three parameters, $P(r, \phi, \theta)$.

5 As the scanlines are transmitted and received, the returning echoes are interpreted as analog electrical signals by a transducer, converted to digital signals by an analog-to-digital converter, and conveyed to the digital signal processor of the computer system for storage and analysis to determine the locations of the amniotic fluid walls. The computer system is representationally depicted in FIGURES 3 and 4 and includes a microprocessor, random access
10 memory (RAM), or other memory for storing processing instructions and data generated by the transceiver 10.

FIGURE 5C is a graphical representation of a plurality of 3D-distributed scanlines emanating from the transceiver 10 forming a scancone 300. The scancone 300 is formed by a plurality of 3D distributed scanlines that comprises a plurality of internal and peripheral
15 scanlines. The scanlines are one-dimensional ultrasound A-lines that emanate from the transceiver 10 at different coordinate directions, that taken as an aggregate, form a conic shape. The 3D-distributed A-lines (scanlines) are not necessarily confined within a scanplane, but instead are directed to sweep throughout the internal and along the periphery of the scancone 300. The 3D-distributed scanlines not only would occupy a given scanplane in a 3D array of
20 2D scanplanes, but also the inter-scanplane spaces, from the conic axis to and including the conic periphery. The transceiver 10 shows the same illustrated features from FIGURE 1, but is configured to distribute the ultrasound A-lines throughout 3D space in different coordinate directions to form the scancone 300.

The internal scanlines are represented by scanlines 312A-C. The number and location of the internal scanlines emanating from the transceiver 10 is the number of internal scanlines needed to be distributed within the scancone 300, at different positional coordinates, to sufficiently visualize structures or images within the scancone 300. The internal scanlines are not peripheral scanlines. The peripheral scanlines are represented by scanlines 314A-F and occupy the conic periphery, thus representing the peripheral limits of the scancone 300.

FIGURE 6 is a depiction of the hand-held transceiver placed on a patient trans-abdominally to transmit probing ultrasound and receive ultrasound echoes for processing to determine amniotic fluid volumes. The transceiver 10 is held by the handle 12 to position over a patient to measure the volume of amniotic fluid in an amniotic sac over a baby. A plurality of axes for describing the orientation of the baby, the amniotic sac, and mother is illustrated. The plurality of axes includes a vertical axis depicted on the line L(R) – L(L) for left and right orientations, a horizontal axis LI – LS for inferior and superior orientations, and a depth axis LA – LP for anterior and posterior orientations.

FIGURE 6 is representative of a preferred data acquisition protocol used for amniotic fluid volume determination. In this protocol, the transceiver 10 is the hand-held 3D ultrasound device (for example, model DCD372 from Diagnostic Ultrasound) and is used to image the uterus trans-abdominally. Initially during the targeting phase, the patient is in a supine position and the device is operated in a 2D continuous acquisition mode. A 2D continuous mode is where the data is continuously acquired in 2D and presented as a scanplane similar to the scanplane 210 on the display 24 while an operator physically moves the transceiver 10. An operator moves the transceiver 10 around on the maternal abdomen and the presses the trigger 14 of the transceiver 10 and continuously acquires real-time

feedback presented in 2D on the display 24. Amniotic fluid, where present, visually appears as dark regions along with an alphanumeric indication of amniotic fluid area (for example, in cm^2) on the display 24. Based on this real-time information in terms of the relative position of the transceiver 10 to the fetus, the operator decides which side of the uterus has more amniotic fluid by the presentation on the display 24. The side having more amniotic fluid presents as regions having larger darker regions on the display 24. Accordingly, the side displaying a large dark region registers greater alphanumeric area while the side with less fluid shows displays smaller dark regions and proportionately registers smaller alphanumeric area on the display 24. While amniotic fluid is present throughout the uterus, its distribution in the uterus depends upon where and how the fetus is positioned within the uterus. There is usually less amniotic fluid around the fetus's spine and back and more amniotic fluid in front of its abdomen and around the limbs.

Based on fetal position information acquired from data gathered under continuous acquisition mode, the patient is placed in a lateral recumbent position such that the fetus is displaced towards the ground creating a large pocket of amniotic fluid close to abdominal surface where the transceiver 10 can be placed as shown in FIGURE 6. For example, if large fluid pockets are found on the right side of the patient, the patient is asked to turn with the left side down and if large fluid pockets are found on the left side, the patient is asked to turn with the right side down.

After the patient has been placed in the desired position, the transceiver 10 is again operated in the 2D continuous acquisition mode and is moved around on the lateral surface of the patient's abdomen. The operator finds the location that shows the largest amniotic fluid area based on acquiring the largest dark region imaged and the largest alphanumeric value

displayed on the display 24. At the lateral abdominal location providing the largest dark region, the transceiver 10 is held in a fixed position, the trigger 14 is released to acquire a 3D image comprising a set of arrayed scanplanes. The 3D image presents a rotational array of the scanplanes 210 similar to the 3D array 240.

5 In a preferred alternate data acquisition protocol, the operator can reposition the transceiver 10 to different abdominal locations to acquire new 3D images comprised of different scanplane arrays similar to the 3D array 240. Multiple scan cones obtained from different positions provide the operator the ability to image the entire amniotic fluid region from different view points. In the case of a single image cone being too small to
10 accommodate a large AFV measurement, obtaining multiple 3D array 240 image cones ensures that the total volume of large AFV regions is determined. Multiple 3D images may also be acquired by pressing the top bottom 16 to select multiple conic arrays similar to the 3D array 240.

Depending on the position of the fetus relative to the location of the transceiver 10, a
15 single image scan may present an underestimated volume of AFV due to amniotic fluid pockets that remain hidden behind the limbs of the fetus. The hidden amniotic fluid pockets present as unquantifiable shadow-regions.

To guard against underestimating AFV, repeated positioning the transceiver 10 and rescanning can be done to obtain more than one ultrasound view to maximize detection of
20 amniotic fluid pockets. Repositioning and rescanning provides multiple views as a plurality of the 3D arrays 240 images cones. Acquiring multiple images cones improves the probability of obtaining initial estimates of AFV that otherwise could remain undetected and un-quantified in a single scan.

In an alternative scan protocol, the user determines and scans at only one location on the entire abdomen that shows the maximum amniotic fluid area while the patient is the supine position. As before, when the user presses the top button 16, 2D scanplane images equivalent to the scanplane 210 are continuously acquired and the amniotic fluid area on every image is automatically computed. The user selects one location that shows the maximum amniotic fluid area. At this location, as the user releases the scan button, a full 3D data cone is acquired and stored in the device's memory.

FIGURE 7 shows a block diagram overview the image enhancement, segmentation, and polishing algorithms of the amniotic fluid volume measuring system. The enhancement, segmentation, and polishing algorithms are applied to each scanplane 210 or to the entire scan cone 240 to automatically obtain amniotic fluid regions. For scanplanes substantially equivalent to scanplane 210, the algorithms are expressed in two-dimensional terms and use formulas to convert scanplane pixels (picture elements) into area units. For the scan cones substantially equivalent to the 3D conic array 240, the algorithms are expressed in three-dimensional terms and use formulas to convert voxels (volume elements) into volume units.

The algorithms expressed in 2D terms are used during the targeting phase where the operator trans-abdominally positions and repositions the transceiver 10 to obtain real-time feedback about the amniotic fluid area in each scanplane. The algorithms expressed in 3D terms are used to obtain the total amniotic fluid volume computed from the voxels contained within the calculated amniotic fluid regions in the 3D conic array 240.

FIGURE 7 represents an overview of a preferred method of the invention and includes a sequence of algorithms, many of which have sub-algorithms described in more specific detail in FIGURES 8A-F. FIGURE 7 begins with inputting data of an unprocessed image at step

410. After unprocessed image data 410 is entered (e.g., read from memory, scanned, or otherwise acquired), it is automatically subjected to an image enhancement algorithm 418 that reduces the noise in the data (including speckle noise) using one or more equations while preserving the salient edges on the image using one or more additional equations. Next, the enhanced images are segmented by two different methods whose results are eventually combined. A first segmentation method applies an intensity-based segmentation algorithm 422 that determines all pixels that are potentially fluid pixels based on their intensities. A second segmentation method applies an edge-based segmentation algorithm 438 that relies on detecting the fluid and tissue interfaces. The images obtained by the first segmentation algorithm 422 and the images obtained by the second segmentation algorithm 438 are brought together via a combination algorithm 442 to provide a substantially segmented image. The segmented image obtained from the combination algorithm 442 are then subjected to a polishing algorithm 464 in which the segmented image is cleaned-up by filling gaps with pixels and removing unlikely regions. The image obtained from the polishing algorithm 464 is outputted 480 for calculation of areas and volumes of segmented regions-of-interest. Finally the area or the volume of the segmented region-of-interest is computed 484 by multiplying pixels by a first resolution factor to obtain area, or voxels by a second resolution factor to obtain volume. For example, for pixels having a size of 0.8mm by 0.8 mm, the first resolution or conversion factor for pixel area is equivalent to 0.64 mm^2 , and the second resolution or conversion factor for voxel volume is equivalent to 0.512 mm^3 . Different unit lengths for pixels and voxels may be assigned, with a proportional change in pixel area and voxel volume conversion factors.

The enhancement, segmentation and polishing algorithms depicted in FIGURE 7 for measuring amniotic fluid areas or volumes are not limited to scanplanes assembled into rotational arrays equivalent to the 3D array 240. As additional examples, the enhancement, segmentation and polishing algorithms depicted in FIGURE 7 apply to translation arrays and wedge arrays. Translation arrays are substantially rectilinear image plane slices from incrementally repositioned ultrasound transceivers that are configured to acquire ultrasound rectilinear scanplanes separated by regular or irregular rectilinear spaces. The translation arrays can be made from transceivers configured to advance incrementally, or may be hand-positioned incrementally by an operator. The operator obtains a wedge array from ultrasound transceivers configured to acquire wedge-shaped scanplanes separated by regular or irregular angular spaces, and either mechanistically advanced or hand-tilted incrementally. Any number of scanplanes can be either translationally assembled or wedge-assembled ranges, but preferably in ranges greater than 2 scanplanes.

Other preferred embodiments of the enhancement, segmentation and polishing algorithms depicted in FIGURE 7 may be applied to images formed by line arrays, either spiral distributed or reconstructed random-lines. The line arrays are defined using points identified by the coordinates expressed by the three parameters, $P(r, \phi, \theta)$, where the values of r , ϕ , and θ can vary.

The enhancement, segmentation and polishing algorithms depicted in FIGURE 7 are not limited to ultrasound applications but may be employed in other imaging technologies utilizing scanplane arrays or individual scanplanes. For example, biological-based and non-biological-based images acquired using infrared, visible light, ultraviolet light, microwave, x-ray computed tomography, magnetic resonance, gamma rays, and positron emission are images

suitable for the algorithms depicted in FIGURE 7. Furthermore, the algorithms depicted in FIGURE 7 can be applied to facsimile transmitted images and documents.

FIGURES 8A-E depict expanded details of the preferred embodiments of enhancement, segmentation, and polishing algorithms described in Figure 7. Each of the following greater detailed algorithms are either implemented on the transceiver 10 itself or are implemented on the host computer 52 or on the server 56 computer to which the ultrasound data is transferred.

FIGURE 8A depicts the sub-algorithms of Image Enhancement. The sub-algorithms include a heat filter 514 to reduce noise and a shock filter 518 to sharpen edges. A combination of the heat and shock filters works very well at reducing noise and sharpening the data while preserving the significant discontinuities. First, the noisy signal is filtered using a 1D heat filter (Equation E1 below), which results in the reduction of noise and smoothing of edges. This step is followed by a shock-filtering step 518 (Equation E2 below), which results in the sharpening of the blurred signal. Noise reduction and edge sharpening is achieved by application of the following equations E1-E2. The algorithm of the heat filter 514 uses a heat equation E1. The heat equation E1 in partial differential equation (PDE) form for image processing is expressed as:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}, \quad \text{E 1}$$

where u is the image being processed. The image u is 2D, and is comprised of an array of pixels arranged in rows along the x-axis, and an array of pixels arranged in columns along the y-axis. The pixel intensity of each pixel in the image u has an initial input image pixel intensity (I) defined as $u_0 = I$. The value of I depends on the application, and commonly

occurs within ranges consistent with the application. For example, I can be as low as 0 to 1, or occupy middle ranges between 0 to 127 or 0 to 512. Similarly, I may have values occupying higher ranges of 0 to 1024 and 0 to 4096, or greater. The heat equation E1 results in a smoothing of the image and is equivalent to the Gaussian filtering of the image. The larger the number of iterations that it is applied for the more the input image is smoothed or blurred and the more the noise that is reduced.

The shock filter 518 is a PDE used to sharpen images as detailed below. The two dimensional shock filter E2 is expressed as:

$$\frac{\partial u}{\partial t} = -F(\ell(u))\|\nabla u\|, \quad \text{E 2}$$

where u is the image processed whose initial value is the input image pixel intensity (I): $u_0 = I$ where the $\ell(u)$ term is the Laplacian of the image u , F is a function of the Laplacian, and $\|\nabla u\|$ is the 2D gradient magnitude of image intensity defined by equation E3.

$$\|\nabla u\| = \sqrt{u_x^2 + u_y^2}, \quad \text{E 3}$$

where

u_x^2 = the square of the partial derivative of the pixel intensity (u) along the x-axis,

u_y^2 = the square of the partial derivative of the pixel intensity (u) along the y-axis,

the Laplacian $\ell(u)$ of the image, u , is expressed in equation E4 as

$$\ell(u) = u_{xx}u_x^2 + 2u_{xy}u_xu_y + u_{yy}u_y^2 \quad \text{E 4}$$

where equation E4 relates to equation E1 as follows:

u_x is the first partial derivative $\frac{\partial u}{\partial x}$ of u along the x-axis,

u_y is the first partial derivative $\frac{\partial u}{\partial y}$ of u along the y-axis,

u_x^2 is the square of the first partial derivative $\frac{\partial u}{\partial x}$ of u along the x-axis,

5 u_y^2 is the square of the first partial derivative $\frac{\partial u}{\partial y}$ of u along the y-axis,

u_{xx} is the second partial derivative $\frac{\partial^2 u}{\partial x^2}$ of u along the x-axis,

u_{yy} is the second partial derivative $\frac{\partial^2 u}{\partial y^2}$ of u along the y-axis,

u_{xy} is cross multiple first partial derivative $\frac{\partial u}{\partial x \partial y}$ of u along the x and y

axes, and

10 the sign of the function F modifies the Laplacian by the image gradient values selected to avoid placing spurious edges at points with small gradient values:

$$F(\ell(u)) = 1, \text{ if } \ell(u) > 0 \text{ and } \|\nabla u\| > t$$

$$= -1, \text{ if } \ell(u) < 0 \text{ and } \|\nabla u\| > t$$

$$= 0, \text{ otherwise}$$

15 where t is a threshold on the pixel gradient value $\|\nabla u\|$.

The combination of heat filtering and shock filtering produces an enhanced image ready to undergo the intensity-based and edge-based segmentation algorithms as discussed below.

FIGURE 8B depicts the sub-algorithms of Intensity-Based Segmentation (step 422 in FIGURE 7). The intensity-based segmentation step 422 uses a “k-means” intensity clustering technique where the enhanced image is subjected to a categorizing “k-means” clustering algorithm. The “k-means” algorithm categorizes pixel intensities into white, gray, and black pixel groups. Given the number of desired clusters or groups of intensities (k), the k-means algorithm is an iterative algorithm comprising four steps:

1. Initially determine or categorize cluster boundaries by defining a minimum and a maximum pixel intensity value for every white, gray, or black pixels into groups or k-clusters that are equally spaced in the entire intensity range.

2. Assign each pixel to one of the white, gray or black k-clusters based on the currently set cluster boundaries.

3. Calculate a mean intensity for each pixel intensity k-cluster or group based on the current assignment of pixels into the different k-clusters. The calculated mean intensity is defined as a cluster center. Thereafter, new cluster boundaries are determined as mid points between cluster centers.

4. Determine if the cluster boundaries significantly change locations from their previous values. Should the cluster boundaries change significantly from their previous values, iterate back to step 2, until the cluster centers do not change significantly between

iterations. Visually, the clustering process is manifest by the segmented image and repeated iterations continue until the segmented image does not change between the iterations.

The pixels in the cluster having the lowest intensity value – the darkest cluster – are defined as pixels associated with amniotic fluid. For the 2D algorithm, each image is clustered independently of the neighboring images. For the 3D algorithm, the entire volume is clustered together. To make this step faster, pixels are sampled at 2 or any multiple sampling rate factors before determining the cluster boundaries. The cluster boundaries determined from the down-sampled data are then applied to the entire data.

FIGURE 8C depicts the sub-algorithms of Edge-Based Segmentation (step 438 in FIGURE 7) and uses a sequence of four sub-algorithms. The sequence includes a spatial gradients algorithm, a hysteresis threshold algorithm, a Region-of-Interest (ROI) algorithm, and a matching edges filter algorithm.

The spatial gradient computes the x-directional and y-directional spatial gradients of the enhanced image. The Hysteresis threshold algorithm detects salient edges. Once the edges are detected, the regions defined by the edges are selected by a user employing the ROI algorithm to select regions-of-interest deemed relevant for analysis.

Since the enhanced image has very sharp transitions, the edge points can be easily determined by taking x- and y- derivatives using backward differences along x- and y- directions. The pixel gradient magnitude $\|\nabla I\|$ is then computed from the x- and y-derivative image in equation E5 as:

$$\|\nabla I\| = \sqrt{I_x^2 + I_y^2} \quad \text{E5}$$

Where I_x^2 = the square of x-derivative of intensity; and

I_y^2 = the square of y-derivative of intensity along the y-axis.

Significant edge points are then determined by thresholding the gradient magnitudes using a hysteresis thresholding operation. Other thresholding methods could also be used. In hysteresis thresholding 530, two threshold values, a lower threshold and a higher threshold, are used. First, the image is thresholded at the lower threshold value and a connected component labeling is carried out on the resulting image. Next, each connected edge component is preserved which has at least one edge pixel having a gradient magnitude greater than the upper threshold. This kind of thresholding scheme is good at retaining long connected edges that have one or more high gradient points.

In the preferred embodiment, the two thresholds are automatically estimated. The upper gradient threshold is estimated at a value such that at most 97% of the image pixels are marked as non-edges. The lower threshold is set at 50% of the value of the upper threshold. These percentages could be different in different implementations. Next, edge points that lie within a desired region-of-interest are selected 534. This region of interest selection 534 excludes points lying at the image boundaries and points lying too close to or too far from the transceiver 10. Finally, the matching edge filter 538 is applied to remove outlier edge points and fill in the area between the matching edge points.

The edge-matching algorithm 538 is applied to establish valid boundary edges and remove spurious edges while filling the regions between boundary edges. Edge points on an image have a directional component indicating the direction of the gradient. Pixels in scanlines crossing a boundary edge location will exhibit two gradient transitions depending on the pixel intensity directionality. Each gradient transition is given a positive or negative value depending on the pixel intensity directionality. For example, if the scanline approaches

an echo reflective bright wall from a darker region, then an ascending transition is established as the pixel intensity gradient increases to a maximum value, i.e., as the transition ascends from a dark region to a bright region. The ascending transition is given a positive numerical value. Similarly, as the scanline recedes from the echo reflective wall, a
5 descending transition is established as the pixel intensity gradient decreases to or approaches a minimum value. The descending transition is given a negative numerical value.

Valid boundary edges are those that exhibit ascending and descending pixel intensity gradients, or equivalently, exhibit paired or matched positive and negative numerical values. The valid boundary edges are retained in the image. Spurious or invalid boundary edges do
10 not exhibit paired ascending-descending pixel intensity gradients, i.e., do not exhibit paired or matched positive and negative numerical values. The spurious boundary edges are removed from the image.

For amniotic fluid volume related applications, most edge points for amniotic fluid surround a dark, closed region, with directions pointing inwards towards the center of the
15 region. Thus, for a convex-shaped region, the direction of a gradient for any edge point, the edge point having a gradient direction approximately opposite to the current point represents the matching edge point. Those edge points exhibiting an assigned positive and negative value are kept as valid edge points on the image because the negative value is paired with its positive value counterpart. Similarly, those edge point candidates having unmatched values,
20 i.e., those edge point candidates not having a negative-positive value pair, are deemed not to be true or valid edge points and are discarded from the image.

The matching edge point algorithm 538 delineates edge points not lying on the boundary for removal from the desired dark regions. Thereafter, the region between any two

matching edge points is filled in with non-zero pixels to establish edge-based segmentation. In a preferred embodiment of the invention, only edge points whose directions are primarily oriented co-linearly with the scanline are sought to permit the detection of matching front wall and back wall pairs.

5 Returning to FIGURE 7, once Intensity-Based 422 and Edge-Based Segmentation 438 is completed, both segmentation methods use a combining step that combines the results of intensity-based segmentation 422 step and the edge-based segmentation 438 step using an AND Operator of Images 442. The AND Operator of Images 442 is achieved by a pixel-wise Boolean AND operator 442 step to produce a segmented image by computing the pixel
10 intersection of two images. The Boolean AND operation 442 represents the pixels as binary numbers and the corresponding assignment of an assigned intersection value as a binary number 1 or 0 by the combination of any two pixels. For example, consider any two pixels, say pixel_A and pixel_B, which can have a 1 or 0 as assigned values. If pixel_A's value is 1, and pixel_B's value is 1, the assigned intersection value of pixel_A and pixel_B is 1. If the binary
15 value of pixel_A and pixel_B are both 0, or if either pixel_A or pixel_B is 0, then the assigned intersection value of pixel_A and pixel_B is 0. The Boolean AND operation 542 takes the binary any two digital images as input, and outputs a third image with the pixel values made equivalent to the intersection of the two input images.

 Upon completion of the AND Operator of Images 442 algorithm, the polish 464
20 algorithm of FIGURE 7 is comprised of multiple sub-algorithms. FIGURE 8D depicts the sub-algorithms of the Polish 464 algorithm, including a Close 546 algorithm, an Open 550 algorithm, a Remove Deep Regions 554 algorithm, and a Remove Fetal Head Regions 560 algorithm.

Closing and opening algorithms are operations that process images based on the knowledge of the shape of objects contained on a black and white image, where white represents foreground regions and black represents background regions. Closing serves to remove background features on the image that are smaller than a specified size. Opening
5 serves to remove foreground features on the image that are smaller than a specified size. The size of the features to be removed is specified as an input to these operations. The opening algorithm 550 removes unlikely amniotic fluid regions from the segmented image based on *a-priori* knowledge of the size and location of amniotic fluid pockets.

Referring to FIGURE 8D, the closing 546 algorithm obtains the Apparent Amniotic
10 Fluid Area (AAFA) or Volume (AAFV) values. The AAFA and AAFV values are “Apparent” and maximal because these values may contain region areas or region volumes of non-amniotic origin unknowingly contributing to and obscuring what otherwise would be the true amniotic fluid volume. For example, the AAFA and AAFV values contain the true amniotic volumes, and possibly as well areas or volumes due to deep tissues and undetected
15 fetal head volumes. Thus the apparent area and volume values require correction or adjustments due to unknown contributions of deep tissue and of the fetal head in order to determine an Adjusted Amniotic Fluid Area (AdAFA) value or Volume (AdAVA) value 568.

The AdAFA and AdAVA values obtained by the Close 546 algorithm are reduced by the morphological opening algorithm 550. Thereafter, the AdAFA and AdAVA values are
20 further reduced by removing areas and volumes attributable to deep regions by using the Remove Deep Regions 554 algorithm. Thereafter, the polishing algorithm 464 continues by applying a fetal head region detection algorithm 560.

FIGURE 8E depicts the sub-algorithms of the Remove Fetal Head Regions sub-algorithm 560. The basic idea of the sub-algorithms of the fetal head detection algorithm 560 is that the edge points that potentially represent a fetal skull are detected. Thereafter, a circle finding algorithm to determine the best-fitting circle to these fetal skull edges is implemented. The radii of the circles that are searched are known a priori based on the fetus' gestational age. The best fitting circle whose fitting metric lies above a certain pre-specified threshold is marked as the fetal head and the region inside this circle is the fetal head region. The algorithms include a gestational Age 726 input, a determine head diameter factor 730 algorithm, a Head Edge Detection algorithm, 734, and a Hough transform procedure 736.

Fetal brain tissue has substantially similar ultrasound echo qualities as presented by amniotic fluid. If not detected and subtracted from amniotic fluid volumes, fetal brain tissue volumes will be measured as part of the total amniotic fluid volumes and lead to an overestimation and false diagnosis of oligo or poly-hyraminotic conditions. Thus detecting fetal head position, measuring fetal brain matter volumes, and deducting the fetal brain matter volumes from the amniotic fluid volumes to obtain a corrected amniotic fluid volume serves to establish accurately measure amniotic fluid volumes.

The gestational age input 726 begins the fetal head detection algorithm 560 and uses a head dimension table to obtain ranges of head bi-parietal diameters (BPD) to search for (e.g., 30 week gestational age corresponds to a 6 cm head diameter). The head diameter range is input to both the Head Edge Detection, 734, and the Hough Transform, 736., The head edge detection 734 algorithm seeks out the distinctively bright ultrasound echoes from the anterior and posterior walls of the fetal skull while the Hough Transform algorithm, 736, finds the

fetal head using circular shapes as models for the fetal head in the Cartesian image (pre-scan conversion to polar form).

Scanplanes processed by steps 522, 538, 530, are input to the head edge detection step 734. Applied as the first step in the fetal head detection algorithm 734 is the detection
5 of the potential head edges from among the edges found by the matching edge filter. The matching edge 538 filter outputs pairs of edge points potentially belonging to front walls or back walls. Not all of these walls correspond to fetal head locations. The edge points representing the fetal head are determined using the following heuristics:

- (1) Looking along a one dimensional A-mode scan line, fetal head locations present a
10 corresponding matching gradient in the opposing direction within a short distance approximately the same size as the thickness of the fetal skull. This distance is currently set to a value 1 cm.
- (2) The front wall and the back wall locations of the fetal head are within a range of diameters corresponding to the expected diameter 730 for the gestational age 726 of
15 the fetus. Walls that are too close or too far are not likely to be head locations.
- (3) A majority of the pixels between the front and back wall locations of the fetal head lie within the minimum intensity cluster as defined by the output of the clustering algorithm 422. The percentage of pixels that need to be dark is currently defined to be 80%.

20 The pixels found satisfying these features are then vertically dilated to produce a set of thick fetal head edges as the output of Head Edge Detection, 734.

FIGURE 8F depicts the sub-algorithms of the Hough transform procedure 736. The sub-algorithms include a Polar Hough Transform 738 algorithm, a find maximum Hough

value 742 algorithm 742, and a fill circle region 746. The Polar Hough Transform algorithm looks for fetal head structures in polar coordinate terms by converting from Cartesian coordinates using a plurality of equations. The fetal head, which appears like a circle in a 3D scan-converted Cartesian coordinate image, has a different shape in the pre-scan converted polar space. The fetal head shape is expressed in terms of polar coordinate terms explained as follows:

The coordinates of a circle in the Cartesian space (x,y) with center (x_0, y_0) and radius R are defined for an angle θ are derived and defined in equation E5 as:

$$\begin{aligned} x &= R \cos \theta + x_0 \\ y &= R \sin \theta + y_0 \\ \Rightarrow (x - x_0)^2 + (y - y_0)^2 &= R^2 \end{aligned} \quad \text{E5}$$

10 In polar space, the coordinates (r, ϕ) , with respect to the center (r_0, ϕ_0) , are derived and defined in equation E6 as:

$$\begin{aligned} r \sin \phi &= R \cos \theta + r_0 \sin \phi_0 \\ r \cos \phi &= R \sin \theta + r_0 \cos \phi_0 \\ \Rightarrow (r \sin \phi - r_0 \sin \phi_0)^2 + (r \cos \phi - r_0 \cos \phi_0)^2 &= R^2 \end{aligned} \quad \text{E6}$$

The Hough transform 736 algorithm using equations E5 and E6 attempts to find the best-fit circle to the edges of an image. A circle in the polar space is defined by a set of three parameters, (r_0, ϕ_0, R) representing the center and the radius of the circle.

The basic idea for the Hough transform 736 is as follows. Suppose a circle is sought having a fixed radius (say, $R1$) for which the best center of the circle is similarly sought. Now, every edge point on the input image lies on a potential circle whose center lays $R1$ pixels away from it. The set of potential centers themselves form a circle of radius $R1$ around each edge pixel. Now, drawing potential circles of radius $R1$ around each edge pixel, the point at which most circles intersect, a center of the circle that represents a best-fit circle to the given edge points is obtained. Therefore, each pixel in the Hough transform output contains a likelihood value that is simply the count of the number of circles passing through that point.

FIGURE 9 illustrates the Hough Transform 736 algorithm for a plurality of circles with a fixed radius in a Cartesian coordinate system. A portion of the plurality of circles is represented by a first circle 804a, a second circle 804b, and a third circle 804c. A plurality of edge pixels are represented as gray squares and an edge pixel 808 is shown. A circle is drawn around each edge pixel to distinguish a center location 812 of a best-fit circle 816 passing through each edge pixel point; the point of the center location through which most such circles pass (shown by a gray star 812) is the center of the best-fit circle 816 presented as a thick dark line. The circumference of the best fit circle 816 passes substantially through is central portion of each edge pixel, represented as a series of squares substantially equivalent to the edge pixel 808.

This search for best fitting circles can be easily extended to circles with varying radii by adding one more degree of freedom – however, a discrete set of radii around the mean radii for a given gestational age makes the search significantly faster, as it is not necessary to search all possible radii.

The next step in the head detection algorithm is selecting or rejecting best-fit circles based on its likelihood, in the find maximum Hough Value 742 algorithm. The greater the number of circles passing through a given point in the Hough-space, the more likely it is to be the center of a best-fit circle. A 2D metric as a maximum Hough value 742 of the Hough transform 736 output is defined for every image in a dataset. The 3D metric is defined as the maximum of the 2D metrics for the entire 3D dataset. A fetal head is selected on an image depending on whether its 3D metric value exceeds a preset 3D threshold and also whether the 2D metric exceeds a preset 2D threshold. The 3D threshold is currently set at 7 and the 2D threshold is currently set at 5. These thresholds have been determined by extensive training on images where the fetal head was known to be present or absent.

Thereafter, the fetal head detection algorithm concludes with a fill circle region 746 that incorporates pixels to the image within the detected circle. The fill circle region 746 algorithm fills the inside of the best fitting polar circle. Accordingly, the fill circle region 746 algorithm encloses and defines the area of the fetal brain tissue, permitting the area and volume to be calculated and deducted via algorithm 554 from the apparent amniotic fluid area and volume (AAFA or AAFV) to obtain a computation of the corrected amniotic fluid area or volume via algorithm 484.

FIGURE 10 shows the results of sequentially applying the algorithm steps of FIGURES 7 and 8A-D on an unprocessed sample image 820 presented within the confines of a scanplane substantially equivalent to the scanplane 210. The results of applying the heat filter 514 and shock filter 518 in enhancing the unprocessed sample is shown in enhanced image 840. The result of intensity-based segmentation algorithms 522 is shown in image 850. The results of edge-based segmentation 438 algorithm using sub-algorithms 526, 530, 534

and 538 of the enhanced image 840 is shown in segmented image 858. The result of the combination 442 utilizing the Boolean AND images 442 algorithm is shown in image 862 where white represents the amniotic fluid area. The result of applying the polishing 464 algorithm employing algorithms 542, 546, 550, 554, 560, and 564 is shown in image 864, which depicts the amniotic fluid area overlaid on the unprocessed sample image 810.

FIGURE 11 depicts a series of images showing the results of the above method to automatically detect, locate, and measure the area and volume of a fetal head using the algorithms outlined in FIGURES 7 and 8A-F. Beginning with an input image in polar coordinate form 920, the fetal head image is marked by distinctive bright echoes from the anterior and posterior walls of the fetal skull and a circular shape of the fetal head in the Cartesian image. The fetal head detection algorithm 734 operates on the polar coordinate data (i.e., pre-scan version, not yet converted to Cartesian coordinates).

An example output of applying the head edge detection 734 algorithm to detect potential head edges is shown in image 930. Occupying the space between the anterior and posterior walls are dilated black pixels 932 (stacks or short lines of black pixels representing thick edges). An example of the polar Hough transform 738 for one actual data sample for a specific radius is shown in polar coordinate image 940.

An example of the best-fit circle on real data polar data is shown in polar coordinate image 950 that has undergone the find maximum Hough value step 742. The polar coordinate image 950 is scan-converted to a Cartesian data in image 960 where the effects of finding maximum Hough value 742 algorithm are seen in Cartesian format.

FIGURE 12 presents a 4-panel series of sonographer amniotic fluid pocket outlines compared to the algorithm's output in a scanplane equivalent to scanplane 210. The top two

panels depict the sonographer's outlines of amniotic fluid pockets obtained by manual interactions with the display while the bottom two panels show the resulting amniotic fluid boundaries obtained from the instant invention's automatic application of 2D algorithms, 3D algorithms, combination heat and shock filter algorithms, and segmentation algorithms.

5 After the contours on all the images have been delineated, the volume of the segmented structure is computed. Two specific techniques for doing so are disclosed in detail in U.S. Pat. No. 5,235,985 to McMorro *et al*, herein incorporated by reference. This patent provides detailed explanations for non-invasively transmitting, receiving and processing ultrasound for calculating volumes of anatomical structures.

10 **Multiple Image Cone Acquisition and Image Processing Procedures:**

In some embodiments, multiple cones of data acquired at multiple anatomical sampling sites may be advantageous. For example, in some instances, the pregnant uterus may be too large to completely fit in one cone of data sampled from a single measurement or anatomical site of the patient (patient location). That is, the transceiver 10 is moved to different
15 anatomical locations of the patient to obtain different 3D views of the uterus from each measurement or transceiver location.

Obtaining multiple 3D views may be especially needed during the third trimester of pregnancy, or when twins or triplets are involved. In such cases, multiple data cones can be sampled from different anatomical sites at known intervals and then combined into a composite
20 image mosaic to present a large uterus in one, continuous image. In order to make a composite image mosaic that is anatomically accurate without duplicating the anatomical regions mutually viewed by adjacent data cones, ordinarily it is advantageous to obtain images from

adjacent data cones and then register and subsequently fuse them together. In a preferred embodiment, to acquire and process multiple 3D data sets or images cones, at least two 3D image cones are generally preferred, with one image cone defined as fixed, and the other image cone defined as moving.

5 The 3D image cones obtained from each anatomical site may be in the form of 3D arrays of 2D scanplanes, similar to the 3D array 240. Furthermore, the 3D image cone may be in the form of a wedge or a translational array of 2D scanplanes. Alternatively, the 3D image cone obtained from each anatomical site may be a 3D scancone of 3D-distributed scanlines, similar to the scancone 300.

10 The term “registration” with reference to digital images means the determination of a geometrical transformation or mapping that aligns viewpoint pixels or voxels from one data cone sample of the object (in this embodiment, the uterus) with viewpoint pixels or voxels from another data cone sampled at a different location from the object. That is, registration involves mathematically determining and converting the coordinates of common regions of an
15 object from one viewpoint to the coordinates of another viewpoint. After registration of at least two data cones to a common coordinate system, the registered data cone images are then fused together by combining the two registered data images by producing a reoriented version from the view of one of the registered data cones. That is, for example, a second data cone’s view is merged into a first data cone’s view by translating and rotating the pixels of the second
20 data cone’s pixels that are common with the pixels of the first data cone. Knowing how much to translate and rotate the second data cone’s common pixels or voxels allows the pixels or voxels in common between both data cones to be superimposed into approximately the same x, y, z, spatial coordinates so as to accurately portray the object being imaged. The more

precise and accurate the pixel or voxel rotation and translation, the more precise and accurate is the common pixel or voxel superimposition or overlap between adjacent image cones. The precise and accurate overlap between the images assures the construction of an anatomically correct composite image mosaic substantially devoid of duplicated anatomical regions.

5 To obtain the precise and accurate overlap of common pixels or voxels between the adjacent data cones, it is advantageous to utilize a geometrical transformation that substantially preserves most or all distances regarding line straightness, surface planarity, and angles between the lines as defined by the image pixels or voxels. That is, the preferred geometrical transformation that fosters obtaining an anatomically accurate mosaic image is a rigid
10 transformation that doesn't permit the distortion or deforming of the geometrical parameters or coordinates between the pixels or voxels common to both image cones.

The preferred rigid transformation first converts the polar coordinate scanplanes from adjacent image cones into in x, y, z Cartesian axes. After converting the scanplanes into the Cartesian system, a rigid transformation, T , is determined from the scanplanes of adjacent
15 image cones having pixels in common. The transformation T is a combination of a three-dimensional translation vector expressed in Cartesian as $t = (T_x, T_y, T_z)$, and a three-dimensional rotation R matrix expressed as a function of Euler angles $\theta_x, \theta_y, \theta_z$ around the x, y , and z axes. The transformation represents a shift and rotation conversion factor that aligns and overlaps common pixels from the scanplanes of the adjacent image cones.

20 In the preferred embodiment of the present invention, the common pixels used for the purposes of establishing registration of three-dimensional images are the boundaries of the amniotic fluid regions as determined by the amniotic fluid segmentation algorithm described above.

Several different protocols may be used to collect and process multiple cones of data from more than one measurement site are described in FIGURES 13-14.

FIGURE 13 illustrates a 4-quadrant supine procedure to acquire multiple image cones around the center point of uterine quadrants of a patient in a supine procedure. Here the patient lies supine (on her back) displacing most or all of the amniotic fluid towards the top. The uterus is divided into 4 quadrants defined by the umbilicus (the navel) and the linea-nigra (the vertical center line of the abdomen) and a single 3D scan is acquired at each quadrant. The 4- quadrant supine protocol acquires four different 3D scans in a two dimensional grid, each corner of the grid being a quadrant midpoint. Four cones of data are acquired by the transceiver 10 along the midpoints of quadrant 1, quadrant 2, quadrant 3, and quadrant 4. Thus, one 3D data cone per uterine quadrant midpoint is acquired such that each quadrant midpoint is mutually substantially equally spaced from each other in a four-corner grid array.

FIGURE 14 illustrates a multiple lateral line procedure to acquire multiple image cones in a linear array. Here the patient lies laterally (on her side), displacing most or all of the amniotic fluid towards the top. Four 3D images cones of data are acquired along a line of substantially equally space intervals. As illustrated, the transceiver 10 moves along the lateral line at position 1, position 2, position 3, and position 4. As illustrated in FIGURE 14, the inter-position distance or interval is approximately 6 cm.

The preferred embodiment for making a composite image mosaic involves obtaining four multiple image cones where the transceiver 10 is placed at four measurement sites over the patient in a supine or lateral position such that at least a portion of the uterus is ultrasonically viewable at each measurement site. The first measurement site is originally defined as fixed, and the second site is defined as moving and placed at a first known inter-site distance relative

to the first site. The second site images are registered and fused to the first site images. After fusing the second site images to the first site images, the third measurement site is defined as moving and placed at a second known inter-site distance relative to the fused second site now defined as fixed. The third site images are registered and fused to the second site images.

5 Similarly, after fusing the third site images to the second site images, the fourth measurement site is defined as moving and placed at a third known inter-site distance relative to the fused third site now defined as fixed. The fourth site images are registered and fused to the third site images.

The four measurement sites may be along a line or in an array. The array may include

10 rectangles, squares, diamond patterns, or other shapes. Preferably, the patient is positioned such that the baby moves downward with gravity in the uterus and displaces the amniotic fluid upwards toward the measuring positions of the transceiver 10.

The interval or distance between each measurement site is approximately equal, or may be unequal. For example in the lateral protocol, the second site is spaced approximately 6 cm

15 from the first site, the third site is spaced approximately 6 cm from the second site, and the fourth site is spaced approximately 6 cm from the third site. The spacing for unequal intervals could be, for example, the second site is spaced approximately 4 cm from the first site, the third site is spaced approximately 8 cm from the second site, and the third is spaced approximately 6 cm from the third site. The interval distance between measurement sites may be varied as long

20 as there are mutually viewable regions of portions of the uterus between adjacent measurement sites.

For uteruses not as large as requiring four measurement sites, two and three measurement sites may be sufficient for making a composite 3D image mosaic. For three

measurement sites, a triangular array is possible, with equal or unequal intervals. Furthermore, is the case when the second and third measurement sites have mutually viewable regions from the first measurement site, the second interval may be measured from the first measurement site instead of measuring from the second measurement site.

- 5 For very large uteruses not fully captured by four measurement or anatomical sites, greater than four measurement sites may be used to make a composite 3D image mosaic provided that each measurement site is ultrasonically viewable for at least a portion of the uterus. For five measurement sites, a pentagon array is possible, with equal or unequal intervals. Similarly, for six measurement sites, a hexagon array is possible, with equal or unequal intervals between
10 each measurement site. Other polygonal arrays are possible with increasing numbers of measurement sites.

The geometrical relationship between each image cone must be ascertained so that overlapping regions can be identified between any two image cones to permit the combining of adjacent neighboring cones so that a single 3D mosaic composite image is produced from the
15 4-quadrant or in-line laterally acquired images.

The translational and rotational adjustments of each moving cone to conform with the voxels common to the stationary image cone is guided by an inputted initial transform that has the expected translational and rotational values. The distance separating the transceiver
20 between image cone acquisitions predicts the expected translational and rotational values. For example, as shown in FIGURE 14, if 6 cm separates the image cones, then the expected translational and rotational values are proportionally estimated. For example, the (T_x, T_y, T_z) and $(\theta_x, \theta_y, \theta_z)$ Cartesian and Euler angle terms fixed images p voxel values are defined respectively as (6cm, 0cm, 0cm) and (0deg, 0deg, 0deg).

FIGURE 15 is a block diagram algorithm overview of the registration and correcting algorithms used in processing multiple image cone data sets. The algorithm overview 1000 shows how the entire amniotic fluid volume measurement process occurs from the multiply acquired image cones. First, each of the input cones 1004 is segmented 1008 to detect all amniotic fluid regions. The segmentation 1008 step is substantially similar to steps 418-480 of FIGURE 7. Next, these segmented regions are used to align (register) the different cones into one common coordinate system using a Rigid Registration 1012 algorithm. Next, the registered datasets from each image cone are fused with each other using a Fuse Data 1016 algorithm to produce a composite 3D mosaic image. Thereafter, the total amniotic fluid volume is computed 1020 from the fused or composite 3D mosaic image.

FIGURE 16 is a block diagram of the steps of the rigid registration algorithm 1012. The rigid algorithm 1012 is a 3D image registration algorithm and is a modification of the Iterated Closest Point (ICP) algorithm published by PJ Besl and ND McKay, in "A Method for Registration of 3-D Shapes," *IEEE Trans. Pattern Analysis & Machine Intelligence*, vol. 14, no. 2, Feb. 1992, pp. 239—256. The steps of the rigid registration algorithm 1012 serves to correct for overlap between adjacent 3D scan cones acquired in either the 4-quadrant supine grid procedure or lateral line multi data cone acquisition procedures. The rigid algorithm 1012 first processes the fixed image 1104 in polar coordinate terms to Cartesian coordinate terms using the 3D Scan Convert 1108 algorithm. Separately, the moving image 1124 is also converted to Cartesian coordinates using the 3D Scan Convert 1128 algorithm. Next, the edges of the amniotic fluid regions on the fixed and moving images are determined and converted into point sets p and q respectively by a 3D edge detection process 1112 and 1132. Also, the fixed image point set, p, undergoes a 3D distance transform process 1116 which maps every

voxel in a 3D image to a number representing the distance to the closest edge point in p . Pre-computing this distance transform makes subsequent distance calculations and closest point determinations very efficient.

Next, the known initial transform 1136, for example, $(6, 0, 0)$ for the Cartesian T_x , T_y ,
 5 T_z terms and $(0, 0, 0)$ for the θ_x , θ_y , θ_z Euler angle terms for an inter-transceiver interval of 6 cm, is subsequently applied to the moving image by the Apply Transform 1140 step. This transformed image is then compared to the fixed image to examine for the quantitative occurrence of overlapping voxels. If the overlap is less than 20%, there are not enough common voxels available for registration and the initial transform is considered sufficient for
 10 fusing at step 1016.

If the overlapping voxel sets by the initial transform exceed 20% of the fixed image p voxel sets, the q -voxels of the initial transform are subjected to an iterative sequence of rigid registration.

A transformation T serves to register a first voxel point set p from the first image cone
 15 by merging or overlapping a second voxel point set q from a second image cone that is common to p of the first image cone. A point in the first voxel point set p may be defined as $p_i = (x_i, y_i, z_i)$ and a point in the second voxel point set q may similarly be defined as $q_j = (x_j, y_j, z_j)$. If the first image cone is considered to be a fixed landmark, then the T factor is applied to align (translate and rotate) the moving voxel point set q onto the fixed voxel
 20 point set p .

The precision of T is often affected by noise in the images that accordingly affects the precision of t and R , and so the variability of each voxel point set will in turn affect the overall

variability of each matrix equation set for each point. The composite variability between the fixed voxel point set p and a corresponding moving voxel point set q is defined to have a cross-covariance matrix C_{pq} , more fully described in equation E8 as:

$$C_{pq} = \frac{1}{n} \sum_{i=1..n} (p_i - \bar{p})(q_i - \bar{q})^T \quad \text{E8}$$

5 where, n is the number of points in each point set and $\bar{p}$ and $\bar{q}$ are the central points in the two voxel point sets. How strong the correlation is between two sets data is determined by statistically analyzing the cross-covariance C_{pq} . The preferred embodiment uses a statistical process known as the Single Value Decomposition (SVD) originally developed by Eckart and Young (G. Eckart and G. Young, 1936, *The Approximation of One Matrix by Another of Lower*
 10 *Rank*, Psychometrika 1, 211-218). When numerical data is organized into matrix form, the SVD is applied to the matrix, and the resulting SVD values are determined to solve for the best fitting rotation transform R to be applied to the moving voxel point set q to align with the fixed voxel point set p to acquire optimum overlapping accuracy of the pixel or voxels common to the fixed and moving images.

15 Equation E9 gives the SVD value of the cross-covariance C_{pq} :

$$C_{pq} = UDV^T \quad \text{E9}$$

where D is a 3 x 3 diagonal matrix and U and V are orthogonal 3 x 3 matrices

Equation E10 further defines the rotational R description of the transformation T in terms of U and V orthogonal 3 x 3 matrices as:

$$20 \quad R = UV^T \quad \text{E10}$$

Equation E11 further defines the translation transform t description of the transformation T in terms of $\bar{p}$, $\bar{q}$ and R as:

$$t = \bar{p} - R\bar{q} \quad \text{E11}$$

Equations E8 through E11 present a method to determine the rigid transformation between two point sets p and q – this process corresponds to step 1152 in Figure 17.

The steps of the registration algorithm are applied iteratively until convergence. The iterative sequence includes a Find Closest Points on Fixed Image 1148 step, a Determine New Transform 1152 step, a Calculate Distances 1156 step, and Converged decision 1160 step.

In the Find Closest Points on Fixed Image 1148 step, corresponding q points are found for each point in the fixed set p . Correspondence is defined by determining the closest edge point on q to the edge point of p . The distance transform image helps locate these closest points. Once p and closest $-q$ pixels are identified, the Determine New Transform 1152 step calculates the rotation R via SVD analysis using equations E8-E10 and translation transform t via equation E11. If, at decision step 1160, the change in the average closest point distance between two iterations is less than 5%, then the *predicted-q* pixel candidates are considered converged and suitable for receiving the transforms R and t to rigidly register the moving image Transform 1136 onto the common voxels p of the 3D Scan Converted 1108 image. At this point, the rigid registration process is complete as closest proximity between voxel or pixel sets has occurred between the fixed and moving images, and the process continues with fusion at step 1016.

If, however, there is $> 5\%$ change between the *predicted-q* pixels and p pixels, another iteration cycle is applied via the Apply Transform 1140 to the Find Closest Points on Fixed

Image 1148 step, and is cycled through the converged 1160 decision block. Usually in 3 cycles, though as many as 20 iterative cycles, are engaged until the transformation T is considered converged.

5 A representative example for the application of the preferred embodiment for the registration and fusion of a moving image onto a fixed image is shown in FIGURES 17A-17C.

FIGURE 17A is a first measurement view of a fixed scanplane 1200A from a 3D data set measurement taken at a first site. A first pixel set p consistent for the dark pixels of AFV is shown in a region 1204A. The region 1204A has approximate x-y coordinates of (150, 120) that is closest to dark edge.

10 FIGURE 17B is a second measurement view of a moving scanplane 1200B from a 3D data set measurement taken at a second site. A second pixel set q consistent for the dark pixels of AFV is shown in a region 1204B. The region 1204B has approximate x-y coordinates of (50, 125) that is closest to dark edge.

FIGURE 17C is a composite image 1200C of the first (fixed) 1200A and second
15 (moving) 1200B images in which common pixels 1204B at approximate coordinates (50,125) is aligned or overlapped with the voxels 1204A at approximate coordinates (150, 120). That is, the region 1204B pixel set q is linearly and rotational transformed consistent with the closest edge selection methodology as shown in FIGURES 13A and 13B from employing the 3D Edge Detection 1112 step.. The composite image 1200C is a mosaic image from scanplanes having
20 approximately the same ϕ and rotation θ angles.

The registration and fusing of common pixel sets p and q from scanplanes having approximately the same ϕ and rotation θ angles can be repeated for other scanplanes in each

3D data set taken at the first (fixed) and second(moving) anatomical sites. For example, if the composite image 1200C above was for scanplane #1, then the process may be repeated for the remaining scanplanes #2-24 or #2-48 or greater as needed to capture a completed uterine mosaic image. Thus an array similar to the 3D array 240 from FIGURE 5B is assembled,
 5 except this time the scanplane array is made of composite images, each composited image belonging to a scanplane having approximately the same ϕ and rotation θ angles.

If a third and a fourth 3D data sets are taken, the respective registration, fusing, and assembling into scanplane arrays of composited images is undertaken with the same procedures. In this case, the scanplane composite array similar to the 3D array 240 is
 10 composed of a greater mosaic number of registered and fused scanplane images.

A representative example the fusing of two moving images onto a fixed image is shown in FIGURES 18A-18D.

FIGURE 18A is a first view of a fixed scanplane 1220A. Region 1224A is identified as p voxels approximately at the coordinates (150, 70).

15 FIGURE 18B is a second view of a first moving scanplane 1220B having some q voxels 1224B at x-y coordinates (300, 100) common with the first measurements p voxels at x-y coordinates (150, 70). Another set of voxels 1234A is shown roughly near the intersection of x-y coordinates (200, 125). As the transceiver 10 was moved only translationally, The scanplane 1220B from the second site has approximately the same tilt ϕ and rotation θ angles
 20 of the fixed scanplane 1220A taken from the first lateral in-line site.

FIGURE 18C is a third view of a moving scanplane 1220C. A region 1234B is identified as q voxels approximately at the x-y coordinates (250, 100) that are common with

the second views q voxels 1234A. The scanplane 1220c from the third lateral in-line site has approximately the same tilt ϕ and rotation θ angles of the fixed scanplane 1220A taken from the first lateral in-line site and the first moving scanplane 1220B taken from the second lateral in-line site.

5 FIGURE 18D is a composite mosaic image 1220D of the first (fixed) 1220A image , the second (moving) 1220B image, and the third (moving) 1220C image representing the sequential alignment and fusing of q voxel sets 1224B to 1224A, and 1234B with 1234A.

 A fourth image similarly could be made to bring about a 4-image mosaic from scanplanes from a fourth 3D data set acquired from the transceiver 10 taking measurements at
10 a fourth anatomical site where the fourth 3D data set is acquired with approximately the same tilt ϕ and rotation θ angles.

 The transceiver 10 is moved to different anatomical sites to collect 3D data sets by hand placement by an operator. Such hand placement could create the acquiring of 3D data sets under conditions in which the tilt ϕ and rotation θ angles are not approximately equal, but
15 differ enough to cause some measurement error requiring correction to use the rigid registration 1012 algorithm. In the event where the 3D data sets between anatomical sites, either between a moving supine site in relation to its beginning fixed site, or between a moving lateral site with its beginning fixed site, cannot be acquired with the tilt ϕ and rotation θ angles being approximately the same, then the built-in accelerometer measures the changes in tilt ϕ and
20 rotation θ angles and compensates accordingly so that acquired moving images are presented if though they were acquired under approximately equal tilt ϕ and rotation θ angle conditions.

Demonstrations of the algorithmic manipulation of pixels of the present invention are provided in *Appendix 1: Examples of Algorithmic Steps*. Source code of the algorithms of the present invention is provided in *Appendix 2: Matlab Source Code*.

While the preferred embodiment of the invention has been illustrated and described, as
5 noted above, many changes can be made without departing from the spirit and scope of the invention. For example, other uses of the invention include determining the areas and volumes of the prostate, heart, bladder, and other organs and body regions of clinical interest. Accordingly, the scope of the invention is not limited by the disclosure of the preferred embodiment.

APPENDIX 1: EXAMPLES OF ALGORITHMIC STEPS

EXAMPLE IMAGE

- 5 The Figure A below shows an example image on which some of the algorithmic steps are demonstrated. The size of this image is 20 rows by 20 columns.

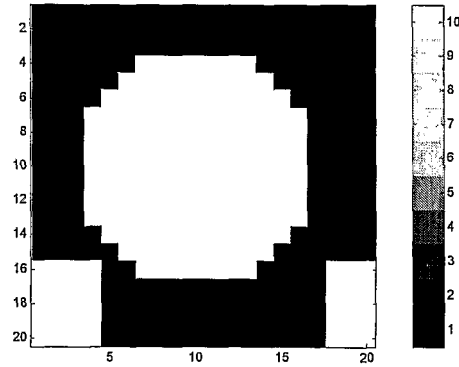


Figure A: Example input image with 20 rows and 20 columns.

10

The image shown above is represented as a matrix of numbers and can also be displayed in 20 x 20 matrix form where the black pixels on the image have a number zero and white pixels have a number 10 in this case:

15

```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 10 10 10 10 10 10 10 10 0 0 0 0 0 0
20 0 0 0 0 0 10 10 10 10 10 10 10 10 10 0 0 0 0 0
0 0 0 10 10 10 10 10 10 10 10 10 10 10 10 0 0 0 0
0 0 0 10 10 10 10 10 10 10 10 10 10 10 10 0 0 0 0
0 0 0 10 10 10 10 10 10 10 10 10 10 10 10 0 0 0 0
25 0 0 0 10 10 10 10 10 10 10 10 10 10 10 10 0 0 0 0
0 0 0 10 10 10 10 10 10 10 10 10 10 10 10 0 0 0 0
0 0 0 10 10 10 10 10 10 10 10 10 10 10 10 0 0 0 0
0 0 0 0 10 10 10 10 10 10 10 10 10 10 0 0 0 0 0
0 0 0 0 0 10 10 10 10 10 10 10 10 0 0 0 0 0 0
30 10 10 10 10 0 0 10 10 10 10 10 10 0 0 0 0 10 10 10
10 10 10 10 0 0 0 0 0 0 0 0 0 0 0 0 0 10 10 10
10 10 10 10 0 0 0 0 0 0 0 0 0 0 0 0 0 10 10 10
10 10 10 10 0 0 0 0 0 0 0 0 0 0 0 0 0 10 10 10
35 10 10 10 10 0 0 0 0 0 0 0 0 0 0 0 0 10 10 10

```

To illustrate some of our algorithms, we also use a noisy version of this example image. To create this image, we added random values to every pixel of the example image and this noisy image is shown in Figure B.

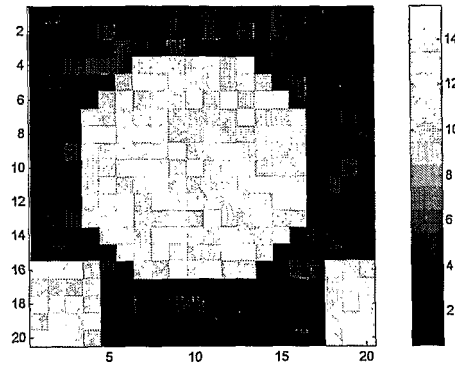


Figure B: Noisy image generated by adding random values to every pixel of the example image shown in Figure A.

As before, the noisy image is shown as a 20 x 20 matrix below:

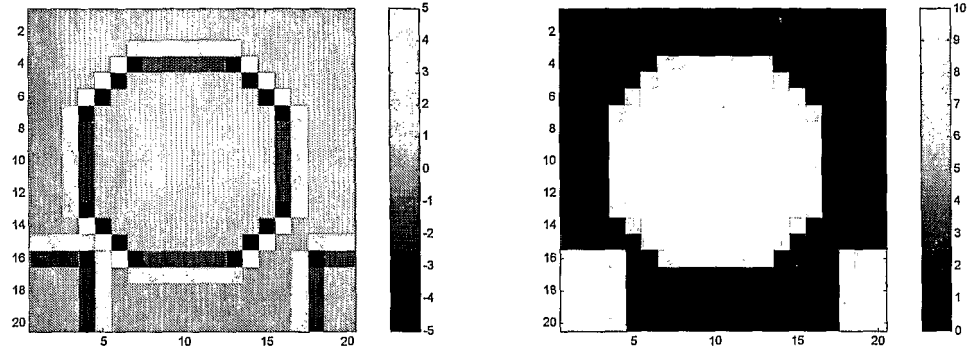
	1.4	4.1	1.1	1	3	0.074	2.7	0.37	3.8	0.91	4.5	0.67	3.3	0.52	4.2	2.6	4.6	2.8	0.86	1.7
10	2	0.83	3.5	2.3	4.7	1.4	3.1	0.97	3.3	2.5	4.7	0.11	4.3	0.79	4	3.6	2.8	3.8	0.65	2.8
	2.5	2	2.6	0.41	1.4	4.1	3.4	1.9	4.4	2.1	1.7	1.3	2.8	2	3.5	2.6	3.3	3.9	1.1	0.59
	3.6	2.6	4.7	4.3	4.4	4.9	13	11	11	13	12	11	15	2	2.3	3	3.9	2.4	0.53	0.85
	1.5	3.6	3.6	2.8	0.51	10	14	14	12	13	12	10	14	10	0.41	4.8	0.53	4	0.71	1.4
15	0.56	2.8	1.1	1.6	10	14	10	12	11	15	11	14	11	15	14	4.1	0.0054	2.4	2.3	2.8
	2.2	2.3	2.2	12	11	13	12	13	10	11	11	11	14	11	11	12	2.7	1	3.9	2.4
	2.3	2.2	0.86	14	15	13	14	15	10	11	13	10	11	12	12	13	0.034	2.9	1.4	4.8
	0.073	0.44	4.8	12	10	11	12	13	14	13	14	14	13	12	10	11	2.3	3.3	1.1	1.2
	3.3	2.2	1.8	10	11	14	15	15	12	15	12	13	13	11	12	11	0.98	3.4	4.5	2.4
20	3.6	1.8	0.25	11	15	11	13	14	12	10	12	11	14	14	14	14	3.9	4.7	0.037	2.6
	1.4	1.5	3.8	10	11	14	10	15	15	14	11	12	12	12	14	13	3.1	3.9	2.9	4
	1.3	4.3	4.5	13	12	13	10	11	11	10	14	10	12	14	12	10	0.078	3.7	2.7	0.97
	3.5	3.8	1.4	0.61	11	11	10	14	14	13	13	14	12	12	14	2.8	4.5	4.3	3.3	4.5
	3.9	4.7	1.3	2.6	3.4	13	11	15	13	11	11	14	13	12	2.4	2.3	3.8	5	1.6	4.6
25	15	13	15	11	4.8	3	13	11	14	14	15	11	11	2.5	2.8	4.5	4.5	13	11	10
	12	10	11	14	3.8	3.3	2.3	0.61	2.4	1.3	3.2	2.8	0.01	0.85	3.1	1.4	3.8	13	12	14
	15	13	15	12	3.3	0.92	3.5	3.8	5	4.7	1.1	0.8	4	2.6	3.3	0.33	1.9	14	11	15
	12	14	14	14	0.65	3.2	2.9	3.6	1.9	0.69	3.4	3	2.6	3.2	3.1	2.4	1.7	12	13	14
	14	15	14	10	0.48	0.85	2.5	3.3	2.7	2.6	3.3	1.7	1.1	0.081	3.4	4.9	2.5	13	15	15

HEAT FILTER – 514

The Laplacian (second derivative) of the example image is computed as the sum of the second partial x-derivative and second partial y-derivative of the image: $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$. This

Laplacian is the right-hand side of the heat equation E1.

The Laplacian of the example image is computed and this is shown in Figure C(1). Notice the negative values on the inside of the bright areas and positive values on the outside of the bright areas at the edges of the regions. The Laplacian is zero on pixels far from edges.



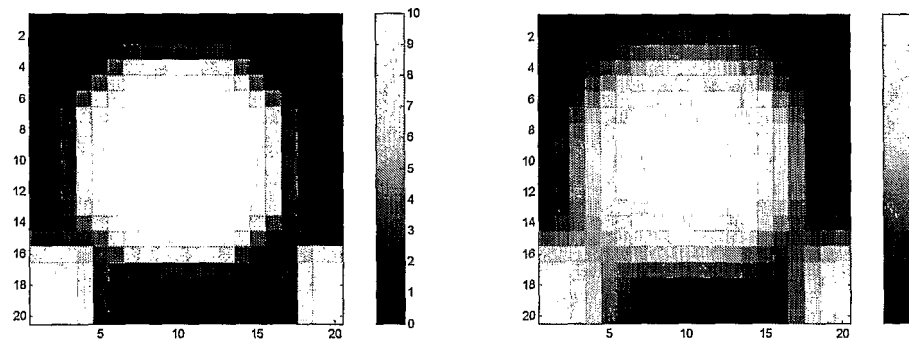
(1)

(2)

Figure C (1): The Laplacian of the example image shown in Figure A. (2) The Laplacian added back to the input image.

When this Laplacian is added back to the input image, after multiplying with a step size parameter, it results in blurring of the edges since the bright areas are reduced in brightness and the dark areas are increased in brightness – this output is shown in Figure C(2). This is the output of a single iteration of the heat filter. The heat filter outputs after 4 iterations and after 20 iterations are shown in (1) (2)

Figure D. Notice the progressive blurring of the image as you apply more iterations.



(1)

(2)

Figure D: Heat filter outputs after 4 iterations (1) and after 20 iterations (2) on the image shown in Figure A.

Applying 20 iterations of the heat filter to the noisy image from Figure B results in a reduction of noise and a blurring of the image. This output is shown in Figure E.

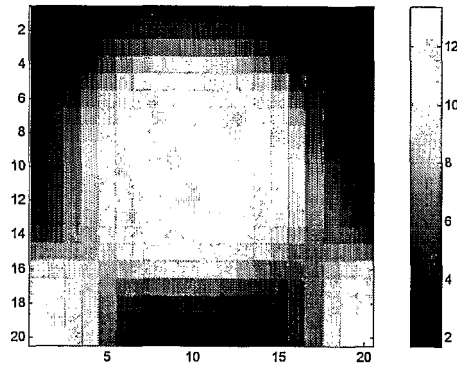


Figure E: The result of applying heat filter to a noisy image.

Note that in this image, the noise has been significantly removed, although the image is very blurred. This image looks very similar to the heat filtered version of the noiseless image shown in Figure D(2).

SHOCK FILTER – 518

The application of the shock filter to the blurred image is shown in Figure E to sharpen this image. For purposes of the shock filter, the Laplacian of an image (Equation E4) is computed as:

$$\ell(u) = u_{xx}u_x^2 + 2u_{xy}u_xu_y + u_{yy}u_y^2.$$

This is computed for the blurred image and is shown below in shown in Figure F(1). Again, notice the negative values on the inside of the bright areas and positive values on the outside of the bright areas at the edges of the regions.

The gradient magnitude of the blurred image (Equation E3) is:

$$\|\nabla u\| = \sqrt{u_x^2 + u_y^2}$$

This is computed for the blurred image and is shown in Figure F(2) below. Note that the gradient magnitude is highest at the edges and zero in smooth regions.

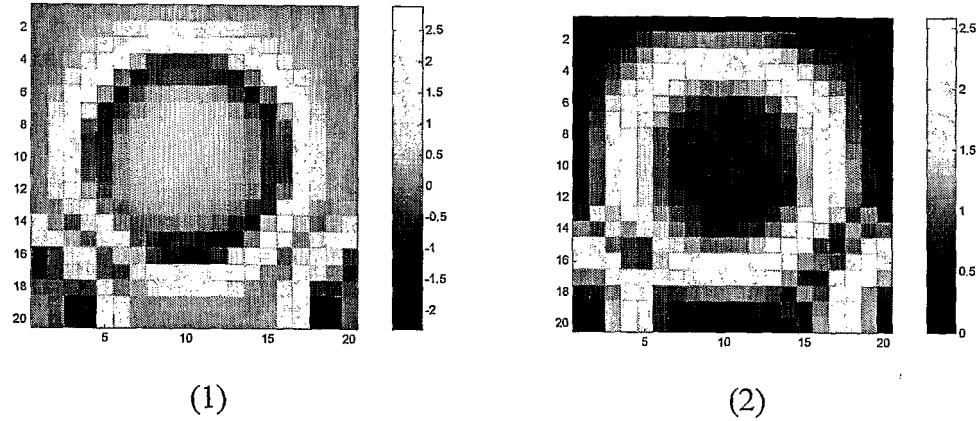


Figure F (1) The Laplacian of the blurred image. (2)
The gradient magnitude of the blurred image.

Using a threshold value, t , on pixel gradient to be 0.7 we can compute the F function, which is positive where the Laplacian is greater than zero and the gradient magnitude is greater than a threshold, negative where the Laplacian is less than zero and the gradient magnitude is greater than the threshold and zero elsewhere:

$$\begin{aligned}
 F(\ell(u)) &= 1, \text{ if } \ell(u) > 0 \text{ and } \|\nabla u\| > t \\
 &= -1, \text{ if } \ell(u) < 0 \text{ and } \|\nabla u\| > t \\
 &= 0, \text{ otherwise}
 \end{aligned}$$

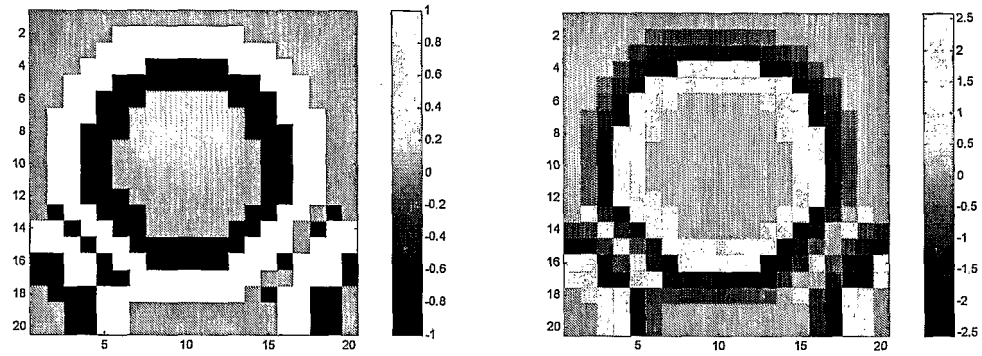
This F image is shown in Figure G(1) below. Note that as with the Laplacian image, everything outside the original white regions at the edge is 1, everything inside the original white regions at the edge is -1 . The boundaries of the original region can be seen as the transition between the negative and the positive F values (the zero crossings).

Now, the right hand side of the shock filter equation E2, is the negative of the product of the F function image Figure G(1) and the gradient magnitude image shown in Figure F(2):

$$-F(\ell(u))\|\nabla u\|$$

This right hand side of the shock filter equation is shown in Figure G(2). If we add this image to the original blurred image we can see that it has the opposite effect of the heat filter. The dark pixels in Figure G(2) are subtracted from the blurred image and the bright pixels are added to the blurred image basically, thereby making the blurred image more crisp.

The addition of this image to the input blurred image is basically what constitutes a single iteration of the shock filter – this addition is shown in Figure H(1). After 20 iterations, a crisper output as shown in Figure H(2) is obtained. While this image is not identical to the input noiseless image of Figure A, it is quite similar to it and much more noise free as compared to the noisy version shown in Figure B.



(1) (2)
Figure G: (1) The F function computed from the Laplacian and the gradient. (2) The negative of the product of the F function and the gradient.

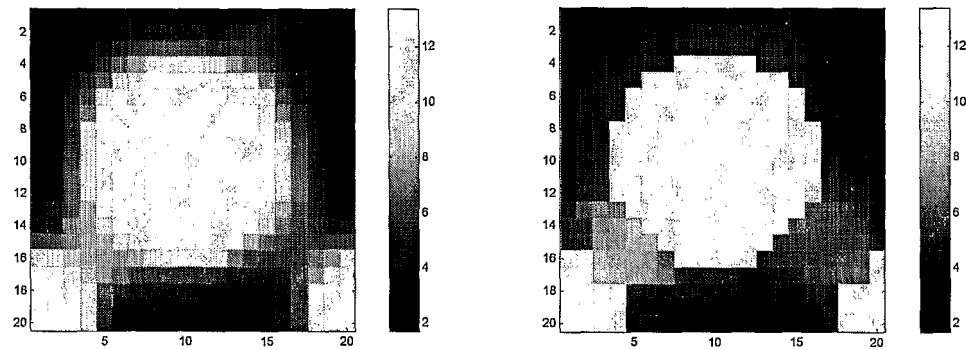


Figure H(1) The output of 1 iteration of the shock filter on the blurred image from Figure E. (2) The output of 20 iterations of the shock filter on the same blurred image.

INTENSITY CLUSTERING – 422

In intensity clustering, the enhanced image is subject to the k-means clustering algorithm. In this example, the enhanced image, shown in Figure H(2), is clustered into 4 clusters.

5 The minimum and maximum intensity values are 1.69 and 13.40. We divide that range into 4 partitions and get the following initial cluster boundaries.

1.19 4.12 7.04 9.98 13.90

10 With this partition, all pixels with values between 1.19 and 4.12 are assigned into cluster 1, all between 4.12 and 7.04 are assigned into cluster 2, etc. This initial assignment of every pixel to the 4 clusters is shown in Figure I(1). Based on this assignment new cluster centers are found and new partition boundaries found accordingly – after this first iteration, the cluster boundaries move slightly and they are now at:

15

1.19 4.08 6.30 9.57 13.90.

In the next iteration, the pixel assignment based on these partition boundaries is shown in Figure I(2). Finally, after 6 iterations, the partition boundaries change very little between iterations and are set to:

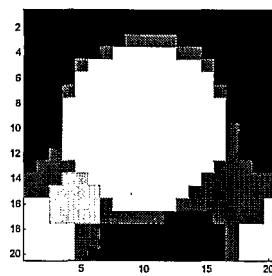
20

1.19 3.65 5.646 9.25 13.90

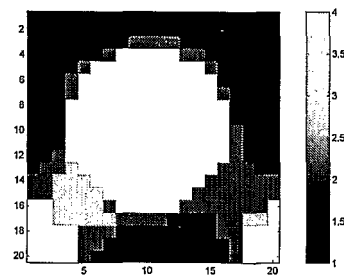
The final assignment of pixels to the 4 clusters is shown in Figure I(3). For this example, the output of the clustering step is the set of pixels assigned to the brightest cluster.

25

This output is shown in Figure I(4).



(1)



(2)

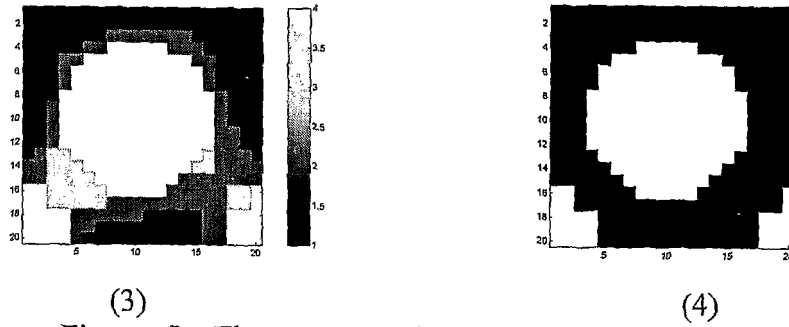


Figure I: The output of clustering at different iterations of the k-means clustering process. (1) Initial assignment (2) After the first iteration (3) Final assignment after 6 iterations. (4) The one brightest cluster representing the region of interest.

SPATIAL GRADIENT – 526

To compute the gradient magnitude of the enhanced image shown in Figure H(2), the x-derivative u_x and the y-derivative u_y of the image is calculated and then the sum of squares of the two images is calculated to get the gradient magnitude -- $\|\nabla u\| = \sqrt{u_x^2 + u_y^2}$. The x- and y-derivatives and the gradient magnitudes of the enhanced image are shown in Figure J.

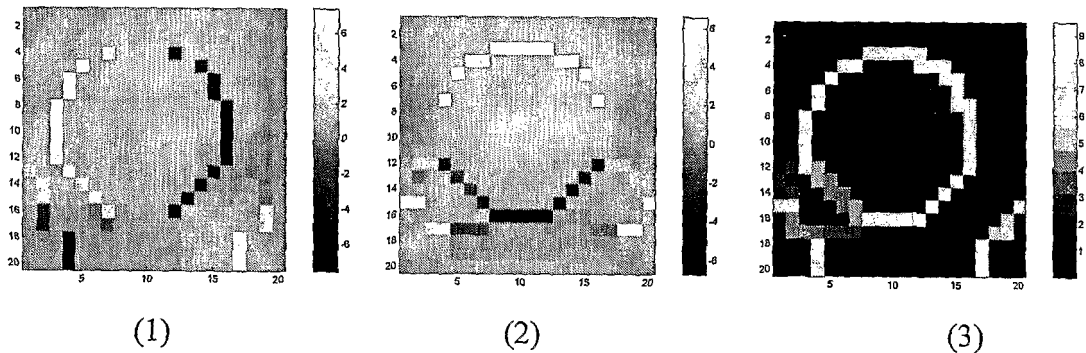


Figure J: Spatial gradients of the image shown in Figure H(2): (1) x-derivative of image (2) y-derivative of image (3) gradient magnitude of image.

HYSTERESIS THRESHOLD – 530

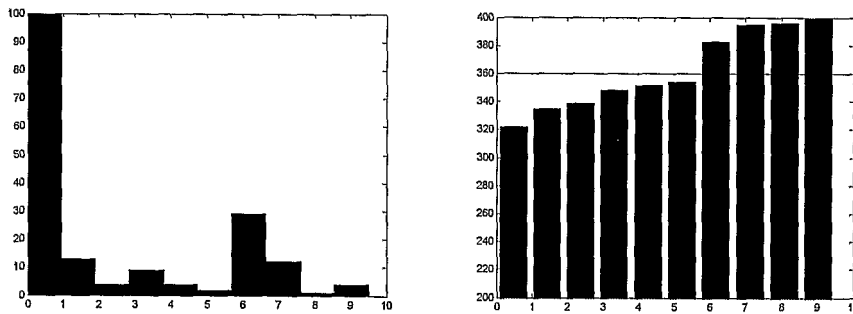
In the hysteresis threshold, two threshold values based on the percentage of non-edge pixels desired and the ratio of the two thresholds is determined. In this example, the percentage non-edge pixels desired is set to 90% and the lower to upper threshold ratio is set to 0.5.

5

The upper threshold is first computed by looking at the histogram and the cumulative histogram of the gradient magnitude image shown in Figure J(3). The histogram of the gradient magnitude image is shown in Figure K(1). The histogram shows the count of pixels of different gradient values on the image. The cumulative histogram calculated by cumulatively adding up the bins of the histogram is shown in Figure K(2). The cumulative histogram shows the count of all pixels less than a given value in the gradient magnitude image. The horizontal line shown in Figure K(2) corresponds to a y-value of 90% of the image pixels ($0.90 * 20 * 20 = 360$). The intersection of this line with the cumulative histogram gives the threshold at which 90% of the image pixels will be marked as non-edge and 10% will be marked as edge. From the graph, we can see that a threshold of 7 satisfies that condition. The lower threshold is set to 0.5 times the upper threshold of 7, which gives 3.5.

10

15



20

Figure K: Histogram and cumulative histogram of the gradient magnitude image.

25

For the hysteresis threshold, the threshold of the image at the two threshold values of 3.5 and at 7 is set and then all edge segments from the lower threshold image is selected that have at least one higher threshold pixel. The two thresholded images and the output of the Hysteresis threshold step are shown in Figure L. In this example, note that one edge segment – (row 15, column 2) has no pixel greater than the higher threshold and is therefore removed from the output. All other edge segments from Figure L(1) are retained in the output.

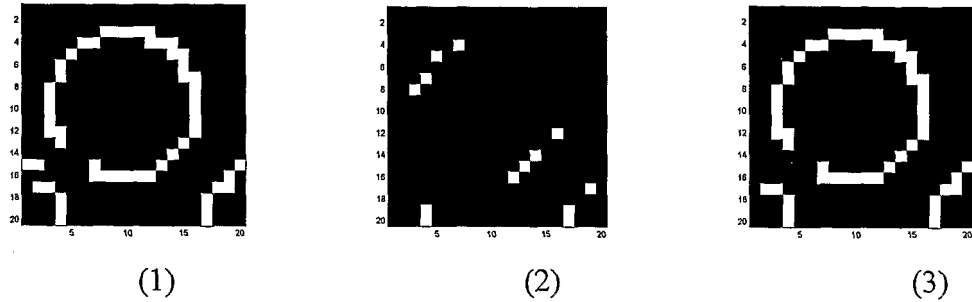


Figure L: (1) Gradient image thresholded at 3.5 – all pixels greater than 3.5 are shown as white (2) gradient image thresholded at 7, (3) Output of hysteresis threshold.

MATCHING EDGES FILTER – 538

The matching edges filter selects pairs of matching edges from among all the edges found by the hysteresis threshold step (Figure L(3)). The y-derivative image, shown in Figure J(2) is used in this step to help find the vertical matching edges. The matching edges found by this step of the algorithm are shown in Figure M(1) and the filled region between the leading the trailing edges are shown in Figure M(2). Note that this step gets rid of edges of regions on the periphery of the image since it cannot find matching trailing edges for those regions. Also, note that in the output filled image we have some unfilled lines where the gradient was primarily horizontal or where the matching edges did not exist.

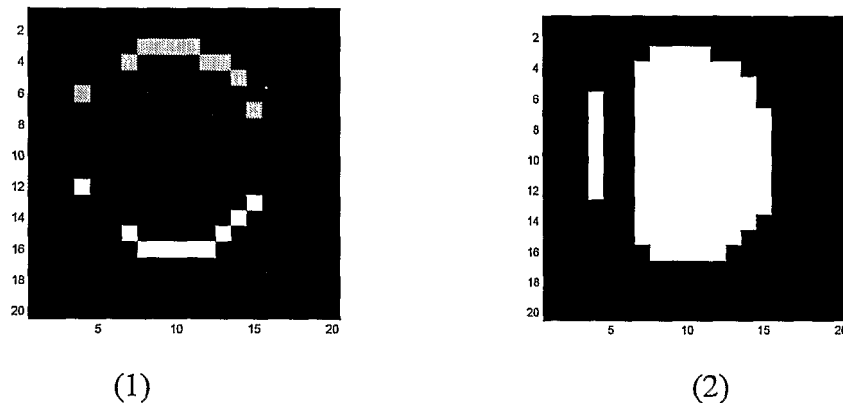


Figure M: (1) Matching edges found by the algorithm – leading edges shown in gray and lagging edges shown in white. (2) The region between the leading the lagging edges filled by white pixels.

AND IMAGES – 442

The two images shown in Figure I(4) and Figure M(2) represent the segmentation of the input noisy image shown in Figure B. Neither of them represents a perfect output. The region-based algorithm finds extraneous regions while the edge-based algorithm misses part of the desired region. The two images are combined using a Boolean AND operator and the output is shown in Figure N.

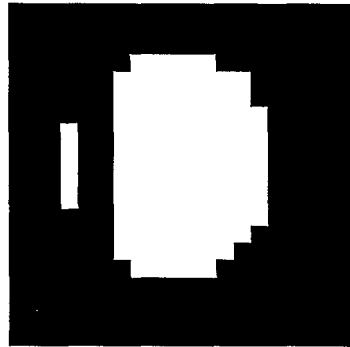


Figure N: Combining the region-based and edge-based segmentation.

CLOSE AND OPEN – 546 & 550

Using a structuring element shaped like a line of width 5 pixels, we close the output of AND images and this gives us a single region representing the structure of interest as shown in Figure O.

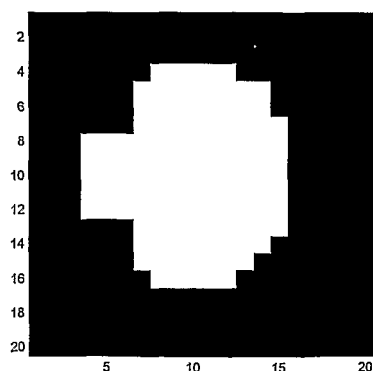
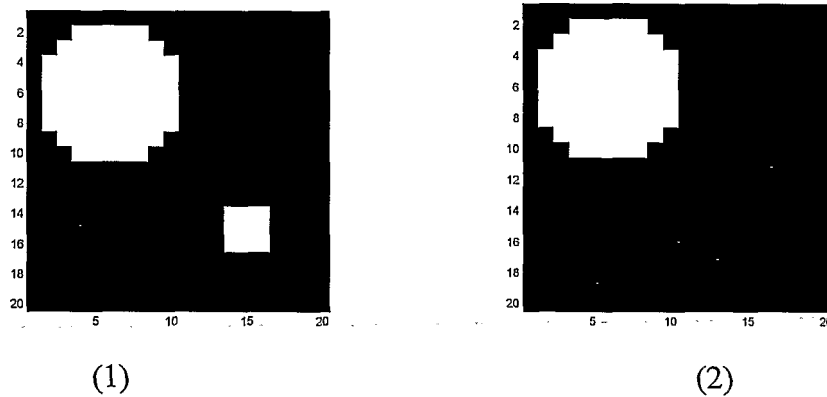


Figure O: Result of closing the image from Figure N.

Opening this image does not change the result – however, Figure P shows an example of opening on another image. The input image, Figure P(1), contains two regions, one large region and one small region. The large region is 10 pixels in width and height while the small region is 4 pixels in width and height. Using a structuring element shaped like a line of length 5 pixels and applying morphological opening to the input image results in an image shown in Figure P(2) where the smaller region whose width was smaller than the structuring element was removed from the image.



(1) (2)
Figure P: (1) Input image containing two foreground regions. (2) Output image after opening where the smaller region was removed because it was smaller than the structuring element.

FILTER DEEP REGIONS – 554

This filtering step is used to remove regions from a segmented image that start at a location deeper than a user specified depth threshold. This filtering is illustrated in Figure Q, where the input image contains two regions -- one region starts at row 2 and the other region starts at row 14. For each region on the input image, the starting row is determined after doing a connected component labeling – then using a depth threshold of 12, the regions where the starting row depth is greater than the depth threshold are removed from the input image.

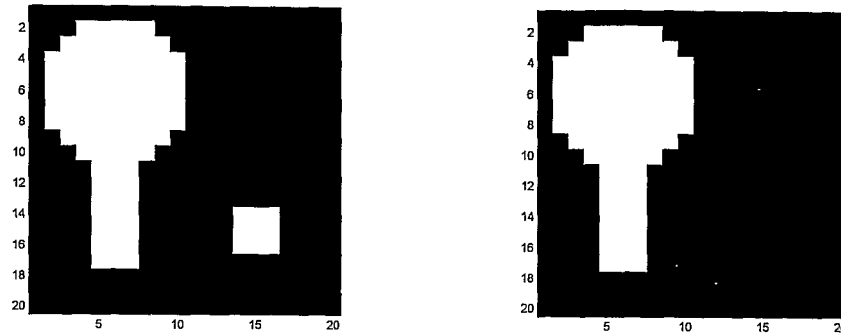


Figure Q: (1) Input image containing two foreground regions. (2) Output image after removing the region whose starting row is greater than the depth threshold of 12.

HEAD DIAMETER RANGE – 730

For a gestational age of 20 weeks, the bi-parietal diameter (BPD) is typically 46 mm. The calculated Head Diameter search range, which ranges from -30% to +30% of the typical BPD value, is found to be:

32.2 36.8 41.4 46.0 50.6 55.2 59.8

For a gestational age of 36 weeks, the bi-parietal diameter (BPD) is typically 88 mm. The calculated Head Diameter search range, which ranges from -30% to +30% of the typical BPD value, is found to be:

61.6 70.4 79.2 88.0 96.8 105.6 114.4

POLAR HOUGH TRANSFORM – 738

To illustrate the polar Hough transform, an exemplary image is generated of a circle in the Cartesian coordinates using the circle equations, E5. This circle image was then dilated to result in a disk like shape and then a few rows of data were set to zero. This input image is shown in Figure R(1). The goal of the circular Hough transform is to find the circle that best fits the white pixels in this input image.

Next, this Cartesian input image was converted to polar domain by inverse scan conversion and the resulting image is shown in Figure R(2). To this polar coordinate image,

the polar Hough transform for a circle was applied, where for each white pixel on the image a polar circle of a known radius was drawn. The output of the Hough transform is the sum of all such circles and this is shown in Figure R(3). The center of the best-fit circle has the most points passing through it. So, to determine the best-fit circle, we simply found the pixel on the Hough output with the maximum value. The point with the maximum value represents the center of the best-fit circle.

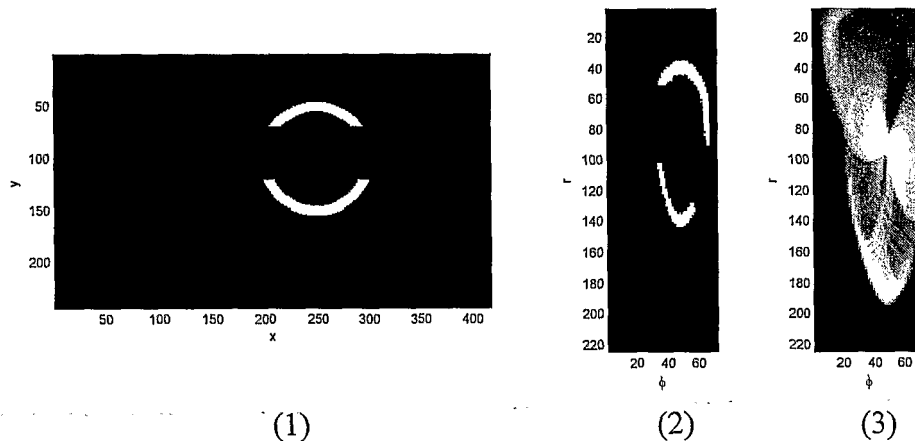


Figure R: (1) The input Cartesian coordinate partial circle (2) The partial circle converted to polar coordinates, (3) The Hough transform output showing the sum of polar circles drawn around each white pixel of the input image.

The best-fit circle determined by this Hough transform procedure is overlaid on the polar coordinate image and the Cartesian coordinate image and the two drawings are shown in Figure S.

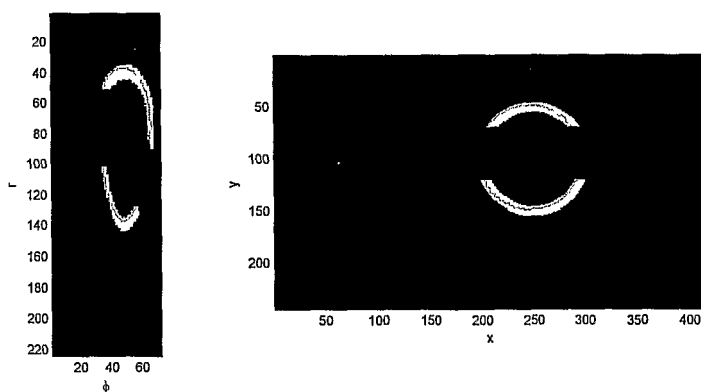


Figure S: The best fit circle found using a Polar coordinate Hough transform (1) overlaid on the polar

coordinate input data and (2) overlaid on the Cartesian coordinate input data.

FILL CIRCLE REGION – 746

A circle region in Cartesian space has a somewhat different shape in the Polar coordinate space. To illustrate this polar coordinate circle, we use both Cartesian coordinate and polar coordinate circles. First, a Cartesian coordinate circle of radius 50 pixels and a center (120,250) is drawn using the circle equation, E5, $(x - x_0)^2 + (y - y_0)^2 = R^2$ -- this circle is shown in Figure T(1).

Now, the Cartesian center coordinates (x_0, y_0) are converted from Cartesian to polar center (r_0, ϕ_0) using the scan conversion equations:

$$r_0^2 = (x_0^2 + y_0^2),$$

$$\phi_0 = \tan^{-1}(y_0 / x_0).$$

Next, the polar coordinate circle drawing and filling equation, E6, $(r \sin \phi - r_0 \sin \phi_0)^2 + (r \cos \phi - r_0 \cos \phi_0)^2 = R^2$,

is applied to draw and fill the same circle of radius $R=50$, in polar coordinates. This circle in the polar coordinates is shown in Figure T(2).

Finally, to verify that this polar coordinate circle is indeed a representation of the Cartesian coordinate circle, the polar circle image is scan-converted to generate the representative Cartesian coordinate circle image. This scan-converted circle is shown in Figure T(3). Comparing Figure T(1) and Figure T(3), it can be seen that while the two circles are not identical, they are fairly similar. The differences are due to the fact that the polar coordinate data cannot represent Cartesian data completely.

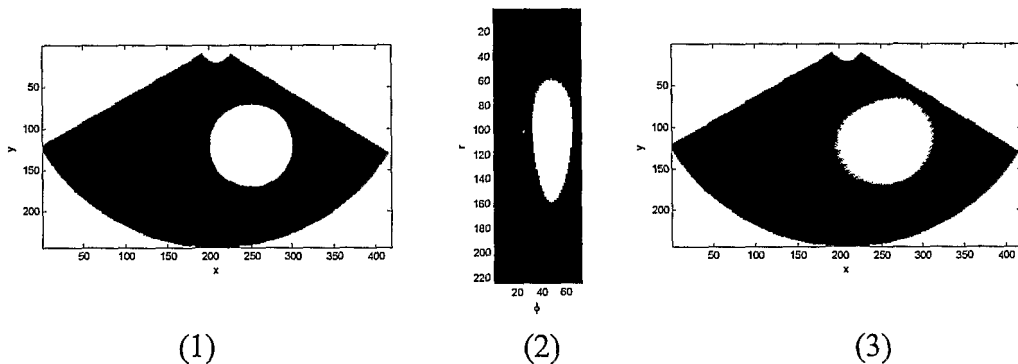


Figure T Polar coordinate circle filling. (1) A filled circle in a Cartesian coordinate image (2) The equivalent filled circle in polar coordinate image, (3) The scan converted polar coordinate filled circle image.

5

10

APPENDIX 2: MATLAB SOURCE CODE

HEAT FILTER – 514

```

function out = heat_flow2D( I, num_iter, time_step )
5  % heat_flow2D: 2D heat flow filter same as Gaussian filtering.
  %
  % out = heat_flow2D( I, num_iter, time_step )
  % * I is the input image
  % * num_iter is the number of iterations for which to run the filter
10  % * time_step is the size of the time step between iterations.
  % * out is the output smoothed image
  %
  % The heat flow equation is  $I(t+1) = I(t) + \text{time\_step} * \text{Laplacian}(I)$ 
  % Gaussian standard deviation (sigma) =  $\sqrt{2 * \text{num\_iter} * \text{time\_step}}$ 
15  % or, if time_step = 0.25, num_iter =  $2 * \text{sigma}^2$ .

  [m n] = size(I);
  update = zeros(m, n);
  rows = 1:m;
20  cols = 1:n;

  prev_rows = rows - 1;
  prev_rows(1) = 1;
  next_rows = rows + 1;
25  next_rows(m) = m;

  prev_cols = cols - 1;
  prev_cols(1) = 1;
  next_cols = cols + 1;
30  next_cols(n) = n;

  for iter = 1:num_iter
    update = (I(rows, next_cols) + I(next_rows, cols) - 4*I(rows, cols) + I(prev_rows, cols) +
35    I(rows, prev_cols))/4;
    I = I + time_step * update;
  end

  out = I;

```

40

SHOCK FILTER - 518

```

function out = shock_filter_grad2D( I, num_iter, time_step, grad_thresh )
%-- 2D shock filter
%-- the equation is  $I(t+1) = I(t) - \text{sign}(I_{xx}) * |\text{grad}(I)|$ 

```

```

%-- sign(Ixx) is the laplace and grad(I) is the upwind derivative
% here instead of using Ixx for edges, we use magnitude of grad(I) as edge.

[m n] = size(I);
5  update = zeros(m, n);
   diff = zeros(m,n);
   gradmat = zeros(m, n);
   gradmag = zeros(m, n);
   laplace = zeros(m, n);
10  diffpos = zeros(m, n);
   diffneg = zeros(m, n);
   dplus = zeros( m, n);
   dminus = zeros( m, n);
   cond1 = zeros( m, n);
15  cond2 = zeros( m, n);

   rows = 1:m;
   cols = 1:n;

20  prev_rows = rows - 1;
   prev_rows(1) = 1;
   next_rows = rows + 1;
   next_rows(m) = m;

25  prev_cols = cols - 1;
   prev_cols(1) = 1;
   next_cols = cols + 1;
   next_cols(n) = n;

30  for iter = 1:num_iter
   %laplace = (I(rows, next_cols) + I(next_rows,cols) - 4*I(rows,cols) + I(prev_rows, cols) +
   I(rows, prev_cols))/4;
   laplace = laplace2( I );
35

   % instead of centered diff - lets use upwind
   %gradmat = 0.5*abs(I(next_rows, cols) - I(prev_rows, cols));

   %row derivative forward
40  diff = I(next_rows,cols) - I(rows,cols);
   cond1 = diff >= 0;
   cond2 = diff < 0;
   diffpos( cond1 ) = diff( cond1 );
   diffpos( cond2 ) = 0;
45  diffneg( cond2 ) = diff( cond2 );
   diffneg( cond1 ) = 0;

```

```

dplus = diffneg .* diffneg;
dminus = diffpos .* diffpos;

5  %row derivative backward
diff = I(rows,cols) - I(prev_rows,cols);
cond1 = diff >= 0;
cond2 = diff < 0;
diffpos( cond1 ) = diff( cond1 );
10 diffpos( cond2 ) = 0;
diffneg( cond2 ) = diff( cond2 );
diffneg( cond1 ) = 0;

dplus = dplus + diffpos .* diffpos;
15 dminus = dminus + diffneg .* diffneg;

%col derivative forward
diff = I(rows,next_cols) - I(rows,cols);
cond1 = diff >= 0;
20 cond2 = diff < 0;
diffpos( cond1 ) = diff( cond1 );
diffpos( cond2 ) = 0;
diffneg( cond2 ) = diff( cond2 );
diffneg( cond1 ) = 0;

25 dplus = dplus + diffneg .* diffneg;
dminus = dminus + diffpos .* diffpos;

%col derivative backward
30 diff = I(rows,cols) - I(rows,prev_cols);
cond1 = diff >= 0;
cond2 = diff < 0;
diffpos( cond1 ) = diff( cond1 );
diffpos( cond2 ) = 0;
35 diffneg( cond2 ) = diff( cond2 );
diffneg( cond1 ) = 0;

dplus = dplus + diffpos .* diffpos;
dminus = dminus + diffneg .* diffneg;

40 dplus = sqrt( dplus );
dminus = sqrt( dminus );

gradmat( laplace >= 0 ) = dplus( laplace >= 0 );
45 gradmat( laplace < 0 ) = dminus ( laplace < 0 );

```

```

gradmat(gradmat < grad_thresh) = 0;

laplace( laplace > 0 ) = 1;
laplace( laplace < 0 ) = -1;
5
    update = -gradmat .* laplace;
    I = I + time_step .* update;
end
10 out = I;

```

INTENSITY CLUSTERING – 422

```

function min_img = cluster_image( in_img, num_clusters, sample_factor, num_iterations )
% cluster_image: Cluster an intensity image using the k-means algorithm.
15 %
% out_img = cluster_image( in_img, num_clusters, sample_factor, num_iterations )
% * in_img is the input intensity image to be clustered
% * num_clusters is the number of output clusters desired
% * sample_factor is a factor by which to downsample the input data. If 1, then no
20 sampling, if 2, then reduces
% input data by half.
% * num_iterations is the maximum number of iterations that the algorithm should be run
if it does not converge.
% usually it converges in about 2-6 iterations - so you can set this to be about 20.
25 % * min_img is the output image consisting of pixels lying on the cluster with the
minimum mean.
%
% A simple k-means algorithm is used for this clustering.

30 convergence_threshold = 0.05;

in_data = downsample( in_img(:), sample_factor );

minval = min( in_data );
35 maxval = max( in_data );

incval = ( maxval - minval )/num_clusters;

%cluster boundaries
40 boundaries = zeros(1, num_clusters + 1 );
boundaries(1) = (minval-0.5);
boundaries(num_clusters+1) = (maxval+0.5);
for i = 2:num_clusters
    boundaries(i) = boundaries(i-1) + incval;

```

```

end

a = 1; b = [0.5 0.5];
centers = zeros( 1, num_clusters );
5 centers = filter( b, a, boundaries );
centers = centers( 2:end );
old_centers = centers;

10 for iter = 1:num_iterations

    %lets quantize the data according to the boundaries
    for j = 1:num_clusters
        cdata = in_data( in_data <= boundaries(j+1) & in_data > boundaries(j) );
        centers(j) = mean(cdata);
15    end

    %lets recompute the boundaries
    newb = filter( b, a, centers );
    newb( 1 ) = (minval-0.5);
20    boundaries = [newb, (maxval+0.5)];

    %centers
    diff = norm( ( centers - old_centers ) ./ old_centers );
    if ( diff <= convergence_threshold )
25        %disp(sprintf('converged in %d iterations', iter ));
        break;
    end

    old_centers = centers;
30 end

min_thresh1 = boundaries( 1 );
min_thresh2 = boundaries( 2 );
min_img = ( in_img <= min_thresh2 ) & ( in_img > min_thresh1 );
35 min_img = reshape( min_img, size( in_img ) );

```

SPATIAL GRADIENTS – 526

```

40 function [xgrad,ygrad,grad_mag] = spatial_gradients( in_img )
% spatial_gradients: Compute spatial gradients of the 2D input image
%
% [xgrad,ygrad,grad_mag] = spatial_gradients( in_img )
% * in_img is the input input image
% * xgrad is the x-direction gradient

```

```

% * ygrad is the y-direction gradient
% * grad_mag is the gradient magnitude
%
5 kernx = [-1 1];
  kerny = [-1; 1];

  xgrad = imfilter( in_img, kernx );
  ygrad = imfilter( in_img, kerny );
10 grad_mag = sqrt( xgrad.^2 + ygrad.^2 );

```

HYSTERESIS THRESHOLD – 530

```

function out = hysteresis_threshold( grad_mag, PercentNonEdge, FractionLow )
% hysteresis_threshold: Perform hysteresis thresholding by automatically computing two
15 thresholds
%
% out = hysteresis_threshold( grad_mag, PercentNonEdge, FractionLow )
% * grad_mag is gradient magnitude image to be thresholded
% * PercentNonEdge is the percentage of non-edge pixels desired (0.97 works good)
20 % * FractionLow is the ratio between the lower and the upper threshold (0.5 works good)

%lets try to figure out the thresholds
maxgrad = max( grad_mag(:) );
25 [c x] = hist( grad_mag(:), ceil( maxgrad ) );
[m n] = size( grad_mag );
high_thresh = min( find( cumsum( c ) > PercentNonEdge * m * n ) );

%lets threshold the gradients
30 out = hysteresis_threshold_core( grad_mag, FractionLow * high_thresh, high_thresh );

function out = hysteresis_threshold_core( img, t1, t2 )
%do a hysteresis thresholding of an image at a lower threshold of t1 and an upper threshold
35 of t2
%basically, we first do a thresholding at the lower value
%then we do a connected component
%then we pick those components which have at least one value above t2
%here, we are going to do it a little differently,
40 % 1. we will first threshold at t2
% 2. we will select one point each from t2
% 3. then we will threshold at t1
% 4. for each point from t2, we will use bwlabel/bwselect to find the component on the t1
image
45 %

```

```

%lets first threshold at higher threshold t2
t2_thresh = img > t2;

5  [r c] = find( t2_thresh );

%lets now threshold at lower thresh t1
t1_thresh = img > t1;

10 out = bwselect( t1_thresh, c, r, 8 );

```

MATCHING EDGES FILTER - 538

```

function [out_filled, out_edg2] = find_matching_vertical_edges( in_edg, ygrad )
% find_matching_vertical_edges: Find matatching edges in the vertical direction
15 %
% out_filled = find_matching_vertical_edges( in_edg, ygrad )
% * in_edg is the input edge image
% * ygrad is the y-direction gradient
% * out_filled is the output filled region between edges
20 %

out_ygrad = ygrad;
out_ygrad( in_edg == 0 ) = 0;

25 dist_thresh = 5;

%for every line, select the closest edge pairs
[npts nline] = size( in_edg );
out_edg2 = zeros( size( in_edg ) );
30 out_filled = out_edg2;

for line = 1:nline
    fw_pts = find( out_ygrad(:,line) < 0 );
    bw_pts = find( out_ygrad(:,line) > 0 );
35 * lfw = length( fw_pts );
    lbw = length( bw_pts );
    if ( lfw == 0 | lbw == 0 )
        continue;
    end
40 if ( ( lfw == 1 ) & ( lbw == 1 ) )
    if ( bw_pts(1) > fw_pts(1) )
        out_edg2( fw_pts(1), line ) = 1;
        out_edg2( bw_pts(1), line ) = 2;
        out_filled( fw_pts(1):bw_pts(1), line ) = 1;
    end
end

```

```

    end
    continue;
end

5. votes_fw = zeros( size( fw_pts ) );
   votes_bw = zeros( size( bw_pts ) );

   %for every front wall point find the closest backwall point
   for fw = 1:lfw
10      curr_fw = fw_pts( fw );
      min_bw = 0;
      min_dist = 1000000;
      for bw = 1:lbw
15         curr_dist = bw_pts( bw ) - curr_fw;
         if ( curr_dist > dist_thresh & curr_dist < min_dist )
            min_bw = bw;
            min_dist = curr_dist;
         end
      end
20      if ( min_bw > 0 )
         votes_fw( fw ) = votes_fw( fw ) + 1;
         votes_bw( min_bw ) = votes_bw( min_bw ) + 1;
      end
   end

25   %for every back wall point find the closest frontwall point
   for bw = 1:lbw
      curr_bw = bw_pts( bw );
      min_fw = 0;
30      min_dist = 1000000;
      for fw = 1:lfw
         curr_dist = curr_bw - fw_pts( fw );
         if ( curr_dist > dist_thresh & curr_dist < min_dist )
35            min_fw = fw;
            min_dist = curr_dist;
         end
      end
      if ( min_fw > 0 )
40         votes_fw( min_fw ) = votes_fw( min_fw ) + 1;
         votes_bw( bw ) = votes_bw( bw ) + 1;
      end
   end

45   %at this point, the fw and bw with votes greater than 1 should be matching
   good_fw_pts = fw_pts( votes_fw > 1 );
   good_bw_pts = bw_pts( votes_bw > 1 );

```

```

if ( length( good_fw_pts ) ~= length( good_bw_pts ) )
    disp('match not found' );
else
5   for i = 1:length( good_fw_pts )
        out_edg2( good_fw_pts(i), line ) = 1;
        out_edg2( good_bw_pts(i), line ) = 2;
        out_filled( good_fw_pts(i):good_bw_pts(i), line ) = 1;
    end
10  end
end

```

CLOSE & OPEN – 546 & 550

```

15 function out = close3D_brick( in, size1, size2, size3 )

    tmp = dilate3D_brick(in, size1, size2, size3 );
    out = erode3D_brick(in, size1, size2, size3 );

20 function out = open3D_brick( in, size1, size2, size3 )

    tmp = erode3D_brick(in, size1, size2, size3 );
    out = dilate3D_brick(in, size1, size2, size3 );

25 function out = erode3D_brick( in, size1, size2, size3 )
    % erode3D_brick: Erode a binary image with a brick shaped structuring element
    %
    % out_img = erode3D_brick( in, size1, size2, size3 )
    % * in_img is the input binary image to be dilated
30  % * size1 is the size of the brick along the row dimension
    % * size2 is the size of the brick along the column dimension
    % * size3 is the size of the brick along the slice dimension
    %
    % A seprable order-statistic filter with the ordinal as min is used.

35  [nr nc np] = size( in );
    out = in;

    for p = 1:np
40      %lets first do the dilation along rows
        out(:, :, p) = ordfilt2( out(:, :, p), 1, ones( size1, 1 ) );

        %lets first do the dilation along columns
        out(:, :, p) = ordfilt2( out(:, :, p), 1, ones( 1, size2 ) );
45  end

```

```

%now lets do it along the third dimension
%first permute dimension
tmp = permute( out, [3, 2, 1] );
5   for r = 1:nr
    tmp(:,:,r) = ordfilt2( tmp(:,:,r), 1, ones( size3, 1 ) );
end

    out = permute( tmp, [3, 2, 1] );
10
function out = dilate3D_brick( in, size1, size2, size3 )
% dilate3D_brick: Dilate a binary image with a brick shaped structuring element
%
% out_img = dilate3D_brick( in, size1, size2, size3 )
15 % * in_img is the input binary image to be dilated
% * size1 is the size of the brick along the row dimension
% * size2 is the size of the brick along the column dimension
% * size3 is the size of the brick along the slice dimension
%
20 % A seprable order-statistic filter with the ordinal as max is used.

[nr nc np] = size( in );
out = in;

25 for p = 1:np
    %lets first do the dilation along rows
    out(:,:,p) = ordfilt2( out(:,:,p), size1, ones( size1, 1 ) );

    %lets first do the dilation along columns
30 out(:,:,p) = ordfilt2( out(:,:,p), size2, ones( 1, size2 ) );
end

%now lets do it along the third dimension
%first permute dimension
35 tmp = permute( out, [3, 2, 1] );
for r = 1:nr
    tmp(:,:,r) = ordfilt2( tmp(:,:,r), size3, ones( size3, 1 ) );
end

40 out = permute( tmp, [3, 2, 1] );

```

FILTER DEEP REGIONS – 554

```

45 function out = filter_deep_regions( in, depthThreshold )
% filter_deep_regions: Filter out regions that start beyond the depth threshold

```

```

%
% out = filter_deep_regions( in, depthThreshold )
% * in is the input binary image
% * depthThreshold is the threshold beyond which regions are not acceptable
5 % * out is the output filtered binary image

[ccmp n] = bwlabel( in );
props = regionprops( ccmp, 'BoundingBox' );
out = zeros( size(in) );
10 j = 0;
for i = 1:n
    if ( props(i).BoundingBox(2) < depthThreshold )
        out = out + ( ccmp == i );
        j = j+1;
15 end
end
end

```

HEAD DIAMETER RANGE – 730

```

function rad_range = radrange_lookup( age, imgParams )
20 %lookup the radius range in pixels given the gestational age and the
%resolution

mean_bpd = bpd_lookup( age );
mean_rad = 0.5 * mean_bpd / imgParams.AxialResolution;
25 rad_range = [mean_rad - ( 0.3 * mean_rad )
    mean_rad - ( 0.2 * mean_rad ),
    mean_rad - ( 0.1 * mean_rad ),
    mean_rad,
    mean_rad + ( 0.1 * mean_rad ),
30 mean_rad + ( 0.2 * mean_rad )
    mean_rad + ( 0.3 * mean_rad )];

function bpd_out = bpd_lookup( age )
    %lookup the bpd in mm given the gestational age
35

    %BPD Table  wks; BPD in mm
    bpd_table = [14, 26;
        15, 30;
        16, 33.5;
40 17, 36.5;
        18, 40;
        19, 43;
        20, 46;
        21, 49.5;
45 22, 52.5;

```

```

    23, 55.5;
    24, 58;
    25, 61;
    26, 64;
5    27, 66.5;
    28, 69;
    29, 72;
    30, 74;
    31, 76.5;
10   32, 79;
    33, 81;
    34, 83.5;
    35, 86;
    36, 88;
15   37, 90;
    38, 92;
    39, 94;
    40, 96;
    ];
20
    bpd_out = 0;
    if ( age < 14 )
        return;
    end
25
    bpd_out = 100;
    if ( age > 40 )
        return;
    end
30
    index = round( age - 14 + 1 );
    bpd_out = bpd_table(index,2);

```

HEAD EDGE DETECTION – 734

```

35 function edg2 = find_head_regions( in_edg, ygrad, wall_edg, region_img, radrange )
    %use ygrads to find head regions

    IntenThresh = 15;
    DistThresh = 12;
40   ygrad( in_edg == 0 ) = 0;

    [npts nline] = size( ygrad );
45   edg2 = wall_edg;

```

```

%lets go over each line and find potential head locations
%potential head locations will have
% a. good front wall and back wall pairs identified by the wall_edg input
5 % b. a matching positive gradient before the front wall - within a short distance
% c. a matching negative gradient after the back wall - within a short distance
for line = 1:nline

    neg_grad_pts = find( ygrad(:,line) < 0 );
10 pos_grad_pts = find( ygrad(:,line) > 0 );
    ln = length( neg_grad_pts );
    lp = length( pos_grad_pts );

    fw = find( wall_edg(:,line) == 1 );
15 bw = find( wall_edg(:,line) == 2 );
    lfw = length(fw);
    lbw = length(bw);

    if ( lfw > 0 & lbw > 0 & lfw == lbw )
20     for i = 1:lfw
        dist = bw(i) - fw(i);
        %if distance is beyond the rad range, set to zero
        if ( dist < 2*radrange(1) | dist > 2*radrange(end) )
25             edg2(fw(i),line) = 0;
             edg2(bw(i),line) = 0;
             continue;
        end

        %if mean intensity is not at least 80% dark
30         if ( mean( region_img(fw(i):bw(i), line) ) < 0.8 )
             edg2(fw(i),line) = 0;
             edg2(bw(i),line) = 0;
             continue;
        end

35         %if within DistThresh of front wall is not a postive gradient point - else expand
        diffs = fw(i) - pos_grad_pts;
        if ( sum( diffs > 0 & diffs < DistThresh ) == 0 )
            edg2(fw(i),line) = 0;
40         else
            if ( fw(i) > 3 & fw(i) < npts - 3 )
                edg2( (fw(i)-3):(fw(i)+3), line ) = 1;
            end
        end
45     end

    %if within DistThresh of back wall is not a negative gradient point

```

```

diffs = neg_grad_pts - bw(i);
if ( sum( diffs > 0 & diffs < DistThresh ) == 0 )
    edg2(bw(i),line) = 0;
else
5     if ( bw(i) > 3 & bw(i) < npts - 3 )
        edg2( (bw(i)-3):(bw(i)+3), line) = 2;
    end
end
end
10 else
    edg2(:,line) = 0;
end
end

```

15 POLAR HOUGH TRANSFORM – 738

```

function out = hough_circle_polar( edg_img, radii, xducerOffset, phiAngles )
%compute a hough transform for circle finding on a pre-scan converted image
%the function goes through every edge pixel on the image and adds an accumulator
corresponding to every radii
20
nradii = length( radii );
[nrow ncol] = size( edg_img );

out = zeros( nrow, ncol, nradii );
25
angles = 0:18:360;
angles = pi*angles/180;
cosa = cos( angles );
sina = sin( angles );
30
prows = cosa;
pcols = cosa;

len = length( angles );
35 hlen = len / 2;

%go over every edge pixel
for row = 1:nrow
    for col = 1:ncol
40        curr_edg = edg_img( row, col );
        if ( curr_edg > 0 )
            for rad = 1:nradii
                [prows, pcols] = draw_polar_circle( row, col, radii(rad), cosa, sina, xducerOffset,
45                phiAngles, 0 );
                %xcoor = round(col + cosrad(rad,:));
            end
        end
    end
end

```

```

        %ycoor = round(row + sinrad(rad,:));

        for i = 1:len
            if ( prows(i) > 0 & prow(i) <= nrow & pcols(i) > 0 & pcols(i) <= ncol )
5              curr_out = out( prow(i), pcols(i), rad );
              %out( prow(i), pcols(i), rad ) = curr_out + 1;

              %if it is the front wall, then the center should be below it - on the lower half
10 of the circle
              if ( curr_edg == 1 & prow(i) > row )
                  out( prow(i), pcols(i), rad ) = curr_out + 1;
              end

              %if it is the back wall, then the center should be above it - on the upper half of
15 the circle
              if ( curr_edg == 2 & prow(i) < row )
                  out( prow(i), pcols(i), rad ) = curr_out + 1;
              end
            end %if
20        end % for i
        end % for rad
        end % if curr_edg
        end %for col
        end %for row
25

function [prows, pcols] = draw_polar_circle( pcent_row, pcent_col, rad, costheta, sintheta,
xducerOffset, phiAngles, draw )
    %draw a circle in polar coordiantes
30    % given a center row and column in polar coordinates, a radius in
    % cartesian, and cosine and sines of theta angles
    % yoff is the transducer offset
    % phiAngles is the list of phiAngles

35    if ( draw )
        x1 = rad * costheta + ( pcent_row + xducerOffset ) * sin( phiAngles( pcent_col ) );
        y1 = rad * sintheta + ( pcent_row + xducerOffset ) * cos( phiAngles( pcent_col ) );
    else
        x1 = -rad * costheta + ( pcent_row + xducerOffset ) * sin( phiAngles( pcent_col ) );
40        y1 = -rad * sintheta + ( pcent_row + xducerOffset ) * cos( phiAngles( pcent_col ) );
    end

    %now convert to polar
45    rads = sqrt( x1.^2 + y1.^2 );
    phis = atan2( x1, y1 );

```

```

prows = round(rads - xducerOffset);
pcols = interp1( phiAngles, 1:length(phiAngles), phis, 'nearest' );

```

5

FILL CIRCLE REGION - 746

```

function out_image = fill_polar_circle( pcent_row, pcent_col, rad, xducerOffset, phiAngles,
nrow, ncol )
10 % fill the inside of a circle in polar coordinates
% given a center row and column in polar coordinates, a radius in
% cartesian, and cosine and sines of theta angles
% yoff is the transducer offset
% phiAngles is the list of phiAngles
15
out_image = zeros( nrow, ncol );
sinc = sin( phiAngles( pcent_col ) );
cosc = cos( phiAngles( pcent_col ) );
rad2 = rad^2;
20 lhs = 0;

for x = 1:nrow
    for y = 1:(ncol-1)
        lhs = x^2 + pcent_row^2 - 2 * x * pcent_row * ( sin(phiAngles(y))*sinc +
25 cos(phiAngles(y))*cosc );
        if ( lhs <= rad2 )
            out_image( x, y ) = 1;
        end
    end
30 end

```

35

We claim:

1. A method to determine amniotic fluid volume in digital images, the method comprising:

positioning an ultrasound transceiver to send and receive echoes from a portion of

a uterus of a patient, the transceiver adapted to form a plurality of scanplanes;

enhancing the images of the amniotic fluid regions in the scanplanes with a

plurality of algorithms;

associating the scanplanes into an array, and

determining the amniotic fluid volume of the amniotic fluid regions within the

array.

2. The method of Claim 1, wherein plurality of scanplanes are acquired from a rotational array, a translational array, or a wedge array.

3. The method of Claim 1, wherein the plurality of algorithms includes algorithms for image enhancement, segmentation, and polishing.

4. The method of Claim 3, wherein segmentation further includes an intensity clustering step, a spatial gradients step, a hysteresis threshold step, a Region-of-Interest selection step, and a matching edges filter step.

5. The method of Claim 4, wherein the intensity clustering step is performed in a first parallel operation, and the spatial gradients, hysteresis threshold, Region-of-Interest selection, and matching edges filter steps are performed in a second parallel operation, and further wherein the results from the first parallel operation are combined with the results from the second parallel operation.

6. The method of Claim 3, wherein image enhancement further includes applying a heat filter and a shock filter to the digital images.

7. The method of Claim 6 wherein the heat filter is applied to the digital images followed by application of the shock filter to the digital images.

5 8. The method of Claim 1, wherein the amniotic fluid volume is adjusted for underestimation or overestimation.

9. The method of Claim 8, wherein the amniotic fluid volume is adjusted for underestimation by probing with adjustable ultrasound frequencies to penetrate deep tissues and to repositioning the transceiver to establish that deep tissues are exposed with
10 probing ultrasound of sufficient strength, to provide a reflecting ultrasound echo receivable by the transceiver, such that more than one rotational array to detect deep tissue and regions of the fetal head are obtained.

10. The method of Claim 8, wherein amniotic fluid volume is adjusted for overestimation by automatically determining fetal head volume contribution to amniotic
15 fluid volume and deducting it from the amniotic fluid volume.

11. The method of Claim 10, wherein the steps to adjust for overestimated amniotic fluid volumes include a 2D clustering step, a matching edges step, an all edges step, a gestational age factor step, a head diameter step, an head edge detection step, and a Hough transform step.

20 12. The method of Claim 12, wherein the Hough transform step includes a polar Hough Transform step, a Find Maximum Hough value step, and a fill circle region step.

13. The method of Claim 12, wherein the polar Hough Transform step includes a first Hough transform to look for lines of a specified shape, and a second Hough transform to look for fetal head structures.

14. A method to determine the volume of structures in digital images, the method comprising:

positioning an ultrasound transceiver exterior to a patient at a plurality of patient locations orientated at approximately the same orientation such that at least a portion of the structure is within the range of the transceiver to send and receive echoes from the portion of a structure;

transmitting radio frequency ultrasound pulses distributed as 2D scanplanes, each scanplane having a plurality of scanlines, to, and receiving those pulses echoed back from, the internal and external boundaries of the structure; and, based on those pulses

a) enhancing the image of the structure in each scanplane;

b) forming a 3D array of scanplanes for each patient location;

c) aligning a common portion of images in similarly orientated scanplanes from adjacent 3D arrays;

d) fusing the images from similarly orientated scanplanes from adjacent 3D arrays and forming a mosaic 3D array, and

e) calculating the volume of the structure in the mosaic 3D array.

15. The method of Claim 14, wherein the structure is a uterus, the internal boundary is between the amniotic fluid and tissue, the plurality of patient locations is four such that a first 3D array is obtained from a first transceiver location, a second 3D array

is obtained from a second transceiver location, a third 3D array is obtained from a third transceiver location, and a fourth 3D array is obtained from a fourth transceiver location, each 3D array having the similarly orientated scanplanes of substantially the same ϕ and rotation θ angles.

- 5 16. The method of Claim 15, wherein the identification of the common portion of the amniotic fluid images is by selection of low intensity pixels at the internal boundary along the amniotic fluid in scanplanes having substantially the same ϕ and rotation θ angles from the first, second, third, and fourth 3D arrays.
- 10 17. The method of Claim 16, wherein the aligning and fusing of the common portion of images in scanplanes having substantially the same same ϕ and rotation θ angles from the first and second 3D arrays comprises:
- a) determining and applying a 3D distance transform that brings the low intensity pixels of a second image to closest proximity to the low intensity pixels of a first image that is common with the second image, and
 - 15 b) repeating step (a) as necessary to achieve closest proximity as shown to be a minimum change in location of the low intensity pixels, then fusing the second image to the first image.
- 20 18. The method of Claim 17, wherein the aligning and fusing of the common portion of images in scanplanes having substantially the same same ϕ and rotation θ angles from the second and third 3D arrays comprises:
- a) determining and applying a 3D distance transform that brings the low intensity pixels of a third image to closest proximity to the low intensity pixels of the second image that is common with the third image, and

b) repeating step (a) as necessary to achieve closest proximity as shown to be a minimum change in location of the low intensity pixels, then fusing the third image to the second image.

19. The method of Claim 19, wherein the aligning and fusing of common portion of images in scanplanes having substantially the same same ϕ and rotation θ angles from the third and fourth 3D arrays comprises:

a) determining and applying a 3D distance transform that brings the low intensity pixels of a fourth image to closest proximity to the low intensity pixels of the third image that is common with the fourth image, and

b) repeating step (a) as necessary to achieve closest proximity as shown to be a minimum change in location of the low intensity pixels, then fusing the fourth image to the third image.

20. The method of Claim 19, wherein the patient is in a supine position and the 3D arrays are obtained from the corners of a substantially rectangular grid, each corner being approximately the midpoint of a uterine quadrant.

21. The method of Claim 19, wherein the patient is in a lateral position and the 3D arrays are obtained along a line over the uterus, each 3D array being separated from the other by a measurable interval.

22. The method of Claim 14, wherein the transceiver is further adapted to measure the tilt ϕ and rotation θ angles differences and to further adjust the tilt ϕ and rotation θ angles between locations to be approximately the same.

23. The method of Claim 22, wherein the tilt ϕ and rotation θ angles are measured by an accelerometer.

24. A method to determine the volume of structures in digital images, the method comprising:

positioning an ultrasound transceiver exterior to a patient at a plurality of patient locations such that at least a portion of the structure is within the range of the transceiver to send and receive echoes from a portion of a structure;

transmitting radio frequency ultrasound pulses delivered as 3D-distributed scanlines to, and receiving those pulses echoed back from, the internal and external boundaries of the structure; and, based on those pulses

a) forming a three-dimensional scancone for each patient location;

b) enhancing the structural image in each scancone;

c) aligning a common portion of the structure between adjacent scancones;

d) fusing the aligned scancones to form a 3D mosaic image, and

e) calculating the volume of the structure in the 3D mosaic image.

25. The method of Claim 24, wherein the structure is a uterus, the internal boundary is between the amniotic fluid and tissue, the plurality of patient locations is four such that a first scancone is obtained from a first transceiver location, a second scancone is obtained from a second transceiver location, a third scancone is obtained from a third transceiver location, and a fourth scancone is obtained from a fourth transceiver location.

26. The method of Claim 25, wherein the identification of the common portion of the amniotic fluid images is by selection of low intensity pixels at the internal boundary along the amniotic fluid in the first, second, third, and fourth scancones.

27. The method of Claim 26, wherein the aligning and fusing of the common portion for the first and second scancones comprises:

a) determining and applying a 3D distance transform that brings the low intensity pixels of the second scancone to closest proximity to the low intensity pixels of the first scancone that is common with the second scancone, and

b) repeating step (a) as necessary to achieve closest proximity as shown to be a minimum change in location of the low intensity pixels, then fusing the second scancone to the first scancone.

28. The method of Claim 27, wherein the aligning and fusing of the common portion for the second and third scancones comprises:

a) determining and applying a 3D distance transform that brings the low intensity pixels of the third scancone to closest proximity to the low intensity pixels of the second scancone that is common with the third scancone, and

b) repeating step (a) as necessary to achieve closest proximity as shown to be a minimum change in location of the low intensity pixels, then fusing the third scancone to the second scancone.

29. The method of Claim 28, wherein the aligning and fusing of the common portion for the third and fourth scancones comprises:

a) determining and applying a 3D distance transform that brings the low intensity pixels of the fourth scancone to closest proximity to the low intensity pixels of the third scancone that is common with the fourth scancone, and

b) repeating step (a) as necessary to achieve closest proximity as shown to be a minimum change in location of the low intensity pixels, then fusing the fourth scancone to the third scancone.

30. The method of Claim 29, wherein the patient is in a supine position and the scancones are obtained from the corners of a substantially rectangular grid, each corner being approximately the midpoint of a uterine quadrant.

31. The method of Claim 29, wherein the patient is in a lateral position and the scancones are obtained along a line over the uterus, each scancone being separated from the other by a measurable interval.

32. A system for determining amniotic fluid volume, the system comprising:

a transceiver positioned on at least one location of a patient, the transceiver configured to deliver radio frequency ultrasound pulses to amniotic fluid regions of a patient, to receive echoes of the pulses reflected from the amniotic fluid regions, to convert the echoes to digital form, and to determine the tilt and rotational orientations of the ultrasound pulses;

a computer system in communication with the transceiver, the computer system having a microprocessor and a memory, the memory further containing stored programming instructions operable by the microprocessor to associate the plurality of scanplanes into a rotational array, and

the memory further containing instructions operable by the microprocessor to determine the presence of an amniotic fluid region in each scanplane and determine the amniotic fluid volume spanning between and through each scanplane of the rotational array.

33. The system of Claim 32, wherein tilt and rotational orientations of the ultrasound pulses is determined by an accelerometer in the transceiver, the accelerometer configured to measure differences in the angular positions of the transceiver.

5 34. The system of Claim 32, wherein each scanplane is arranged as a plurality of scanlines, each scanline of the plurality of scanlines being separated by approximately 1.5 degrees and having a length suitable for the dimension of amniotic fluid region.

35. The system of Claim 34, wherein scanplanes have similar orientations from a plurality of transceiver locations

10 36. The system of Claim 32, wherein the programming instructions include a plurality of algorithms for image enhancement, segmentation, and polishing.

37. The system of Claim 36, wherein the steps for image enhancement further include application of a heat filter followed by application of a shock filter.

38. The system of Claim 32, wherein the amniotic fluid volumes are adjusted for underestimation and overestimation.

15 39. The system of Claim 38, wherein the amniotic fluid volumes are adjusted for underestimation by probing with ultrasound frequencies having sufficient power and wavelength to penetrate through fatty tissue to reach amniotic fluid regions and to provide detectable echo signals receivable to the transceiver to reveal amniotic fluid regions.

20 40. The system of Claim 39, wherein the amniotic fluid volumes are further adjusted for underestimation by repositioning the transceiver to acquire more than one rotational array to detect deep tissue and regions of the fetal head.

41. The system of Claim 38, wherein the amniotic fluid volumes are adjusted for overestimation by detecting the location of a fetal head, determining the volume of the fetal head, and deducting the volume of the fetal head from the amniotic fluid volume spanning between and through each scanplane of the rotational array.

5 42. The system of Claim 32, wherein the computer system is configured for remote operation via an Internet web-based system, the internet web-based system having a plurality of programs that collect, analyze, and store amniotic fluid volume.

1/22

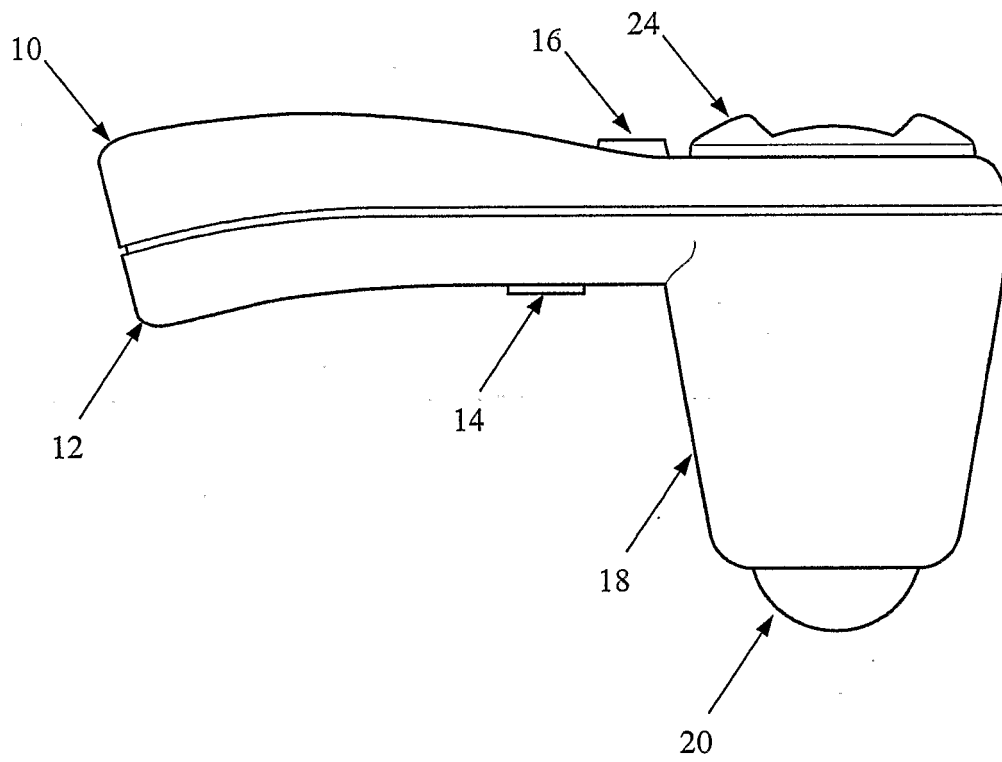


FIG. 1

2/22

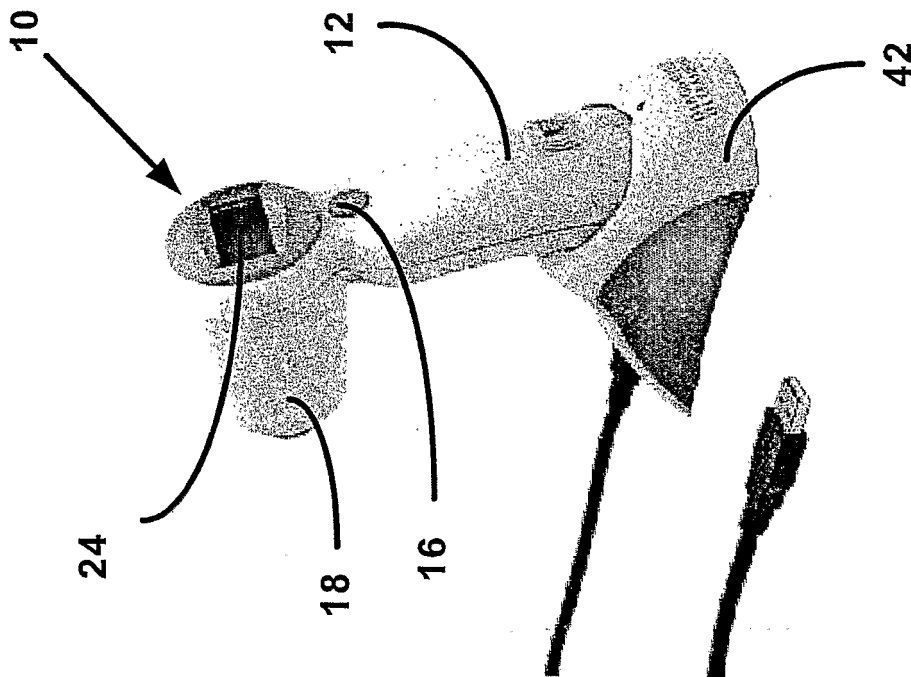


FIG. 2B

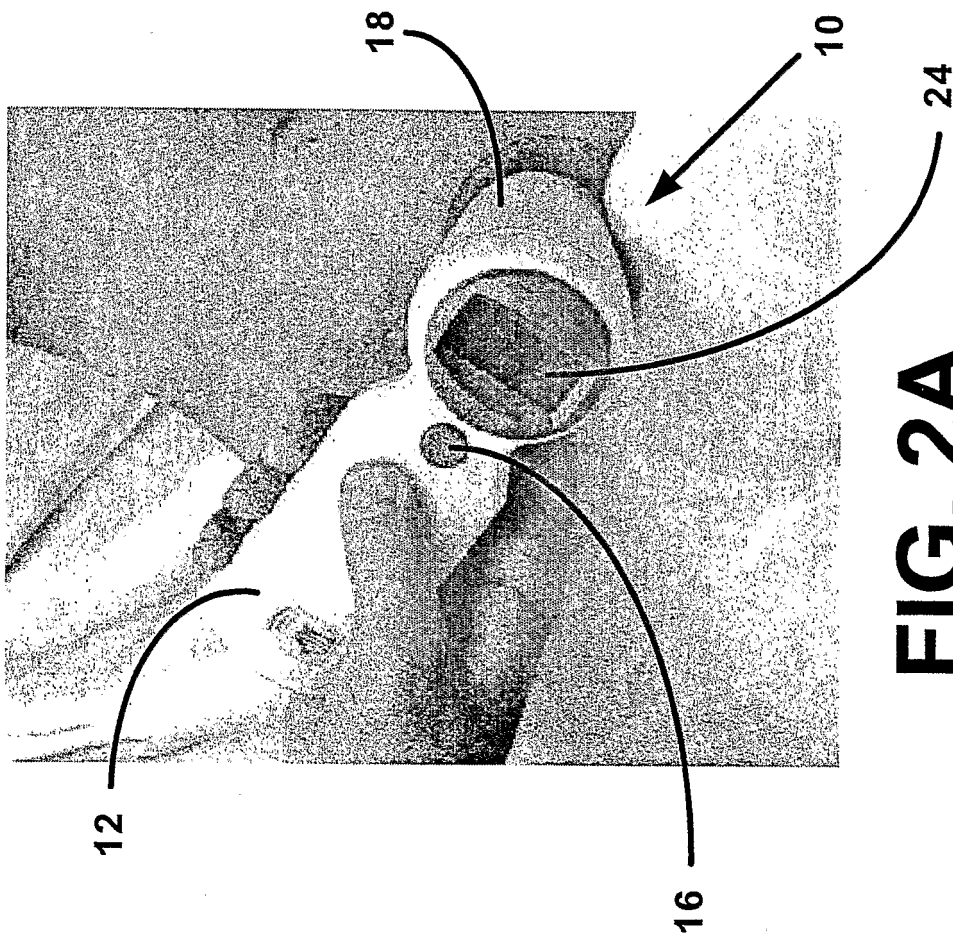


FIG. 2A

3/22

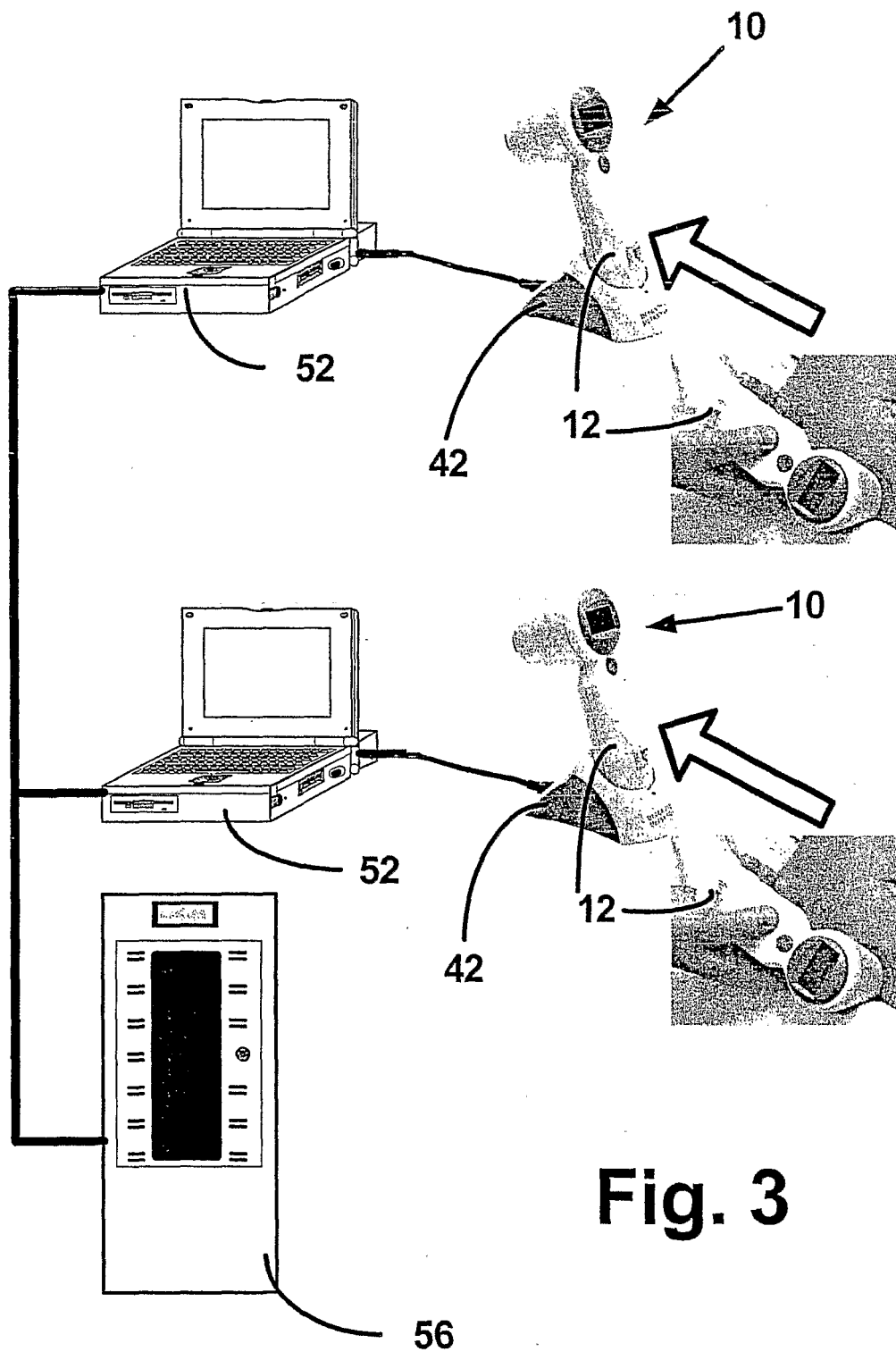


Fig. 3

4/22

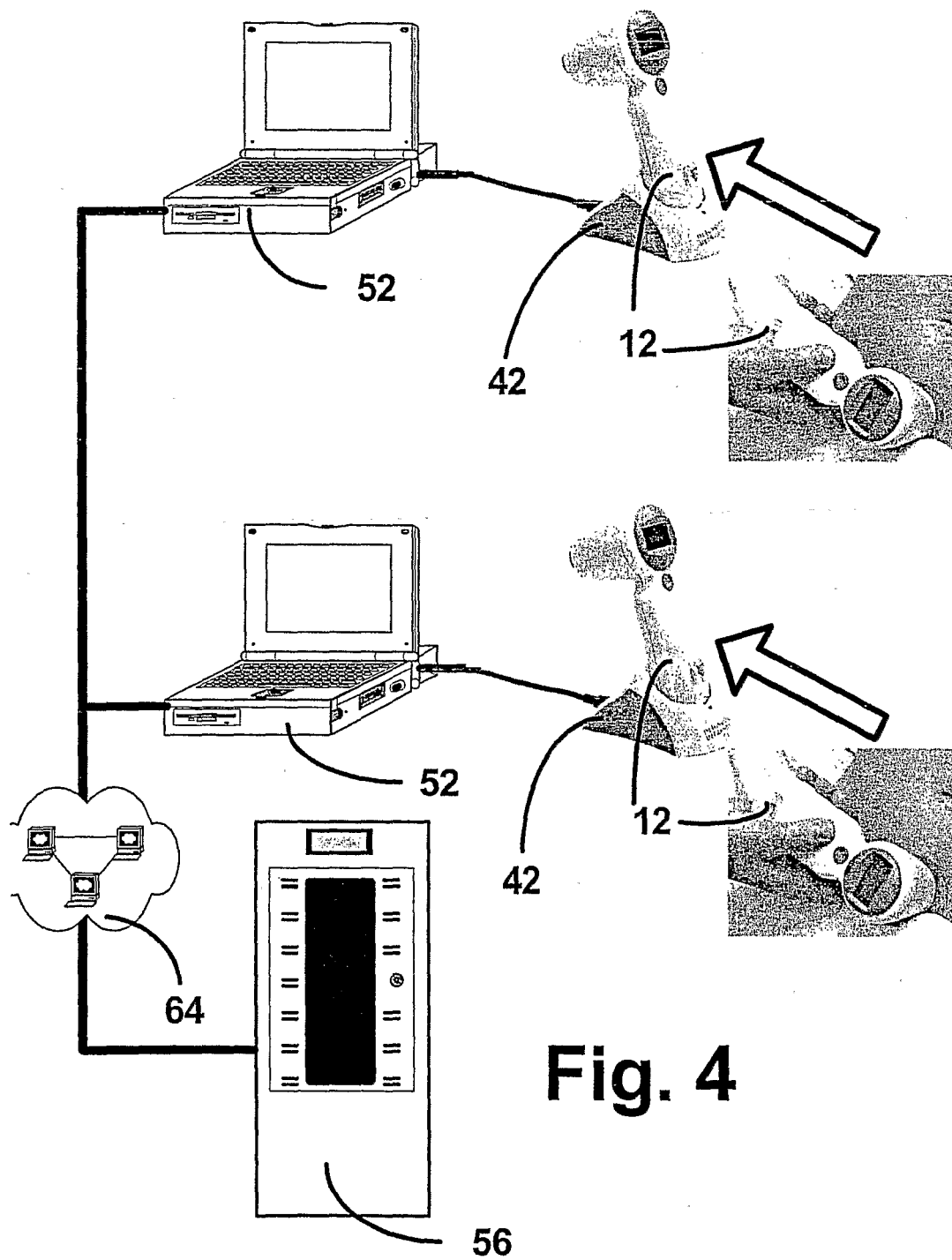
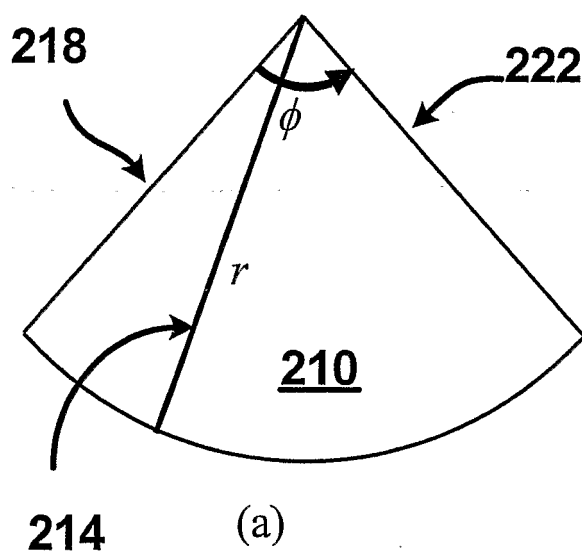
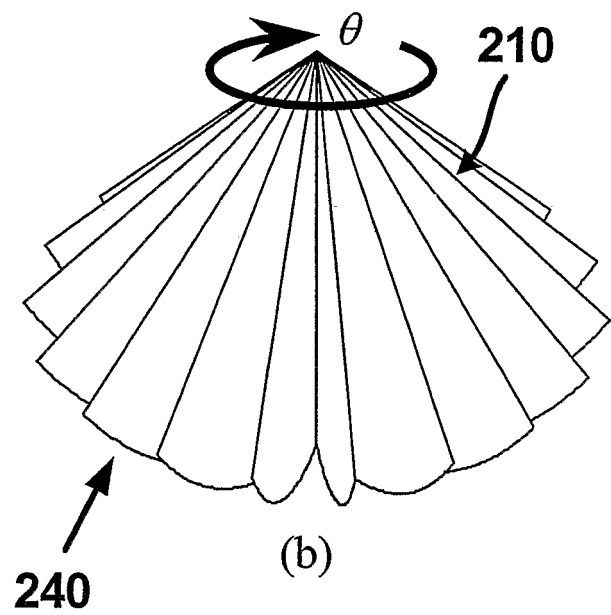
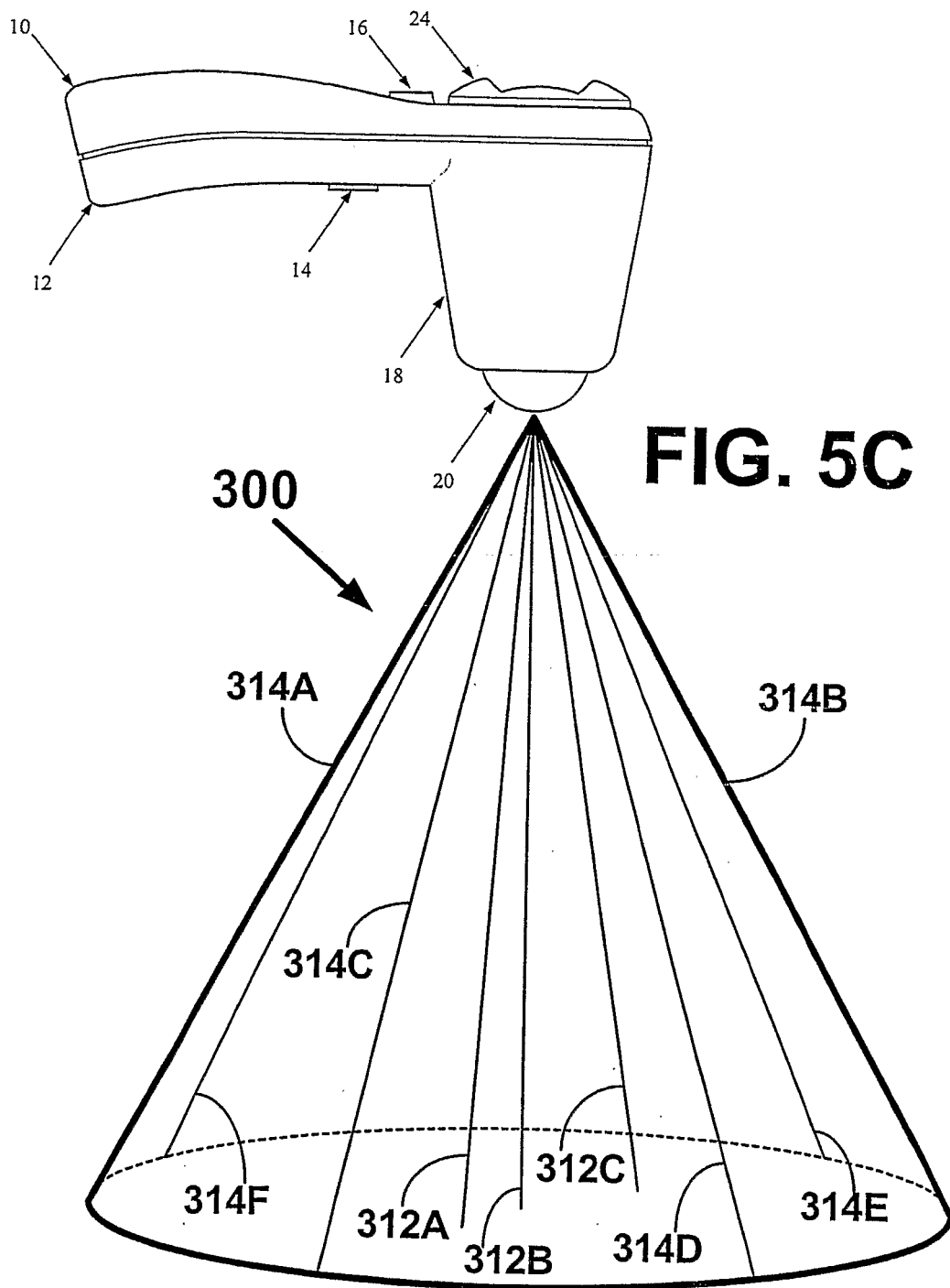


Fig. 4

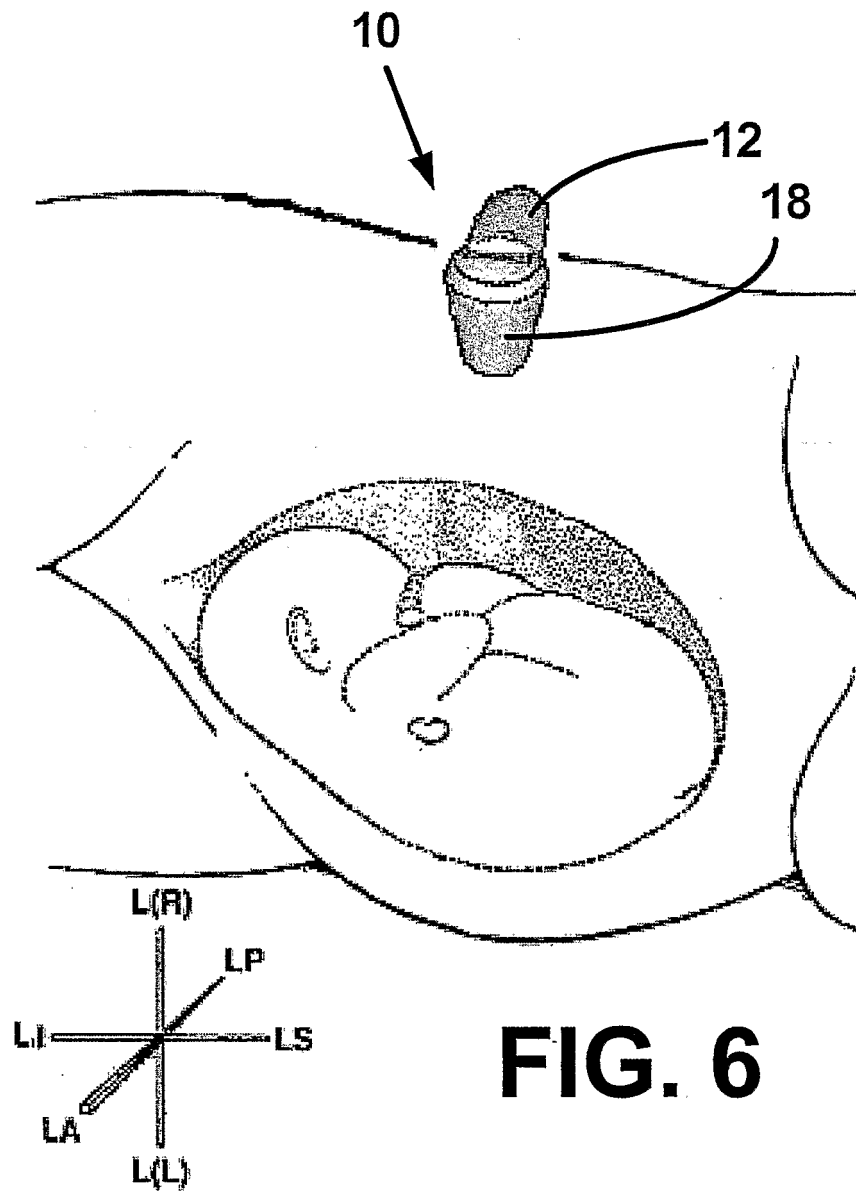
5/22

**FIG. 5A****FIG. 5B**

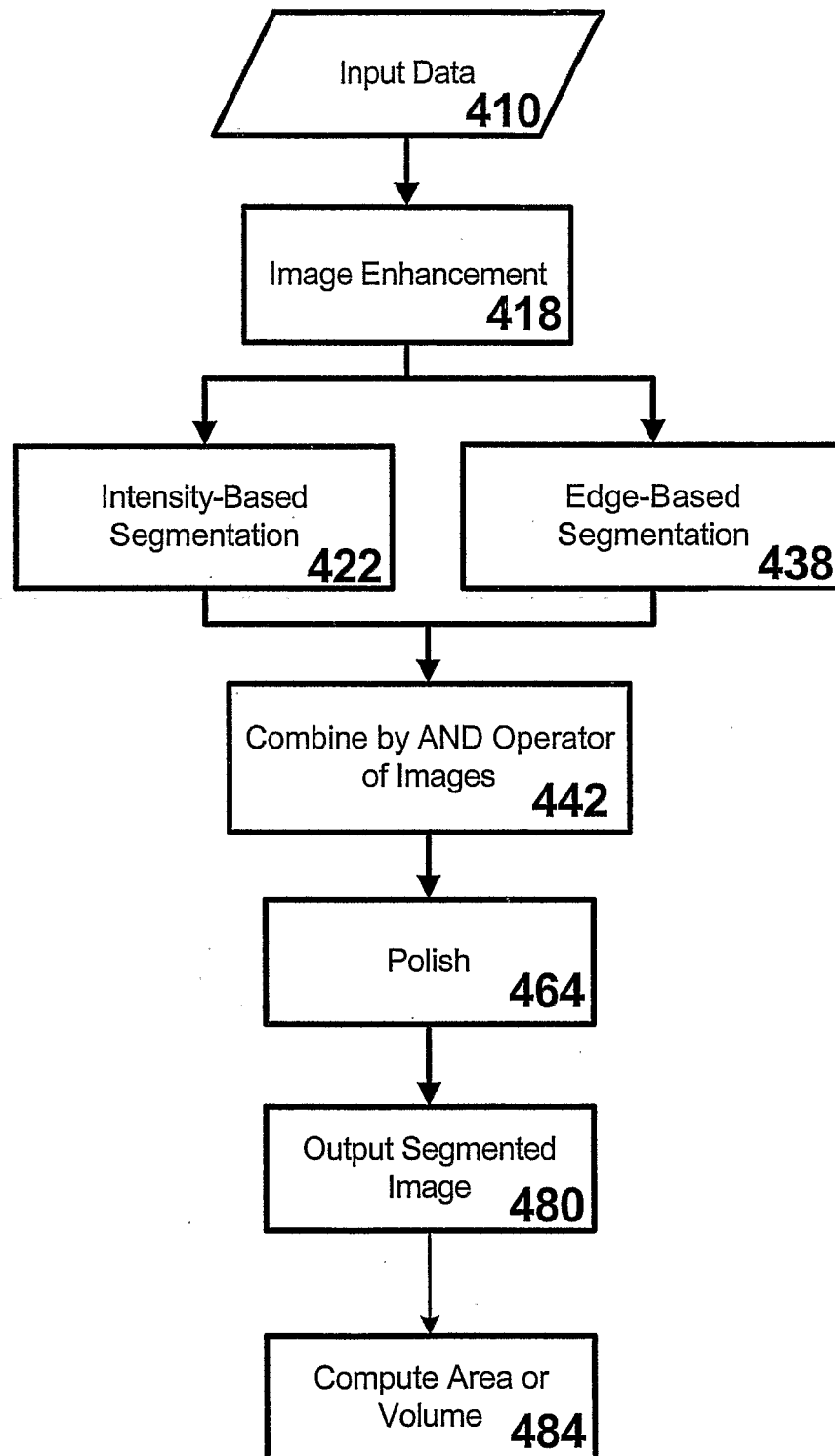
6/22



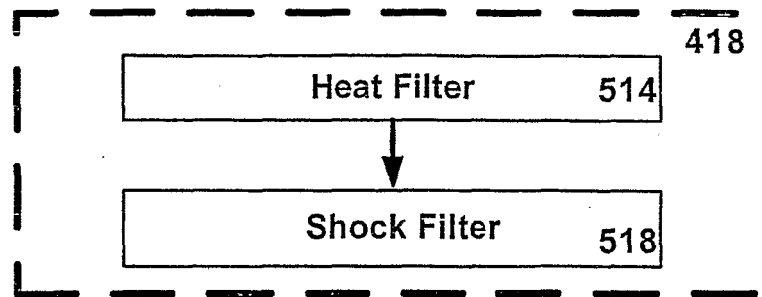
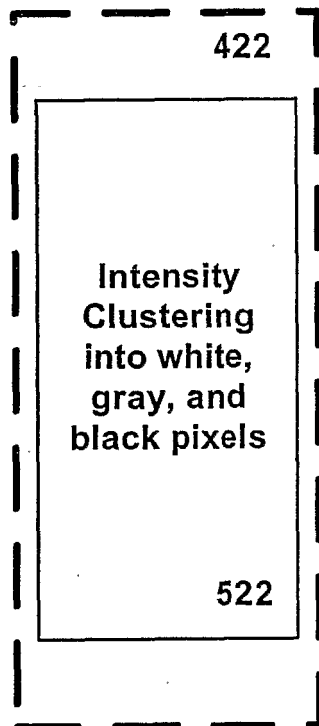
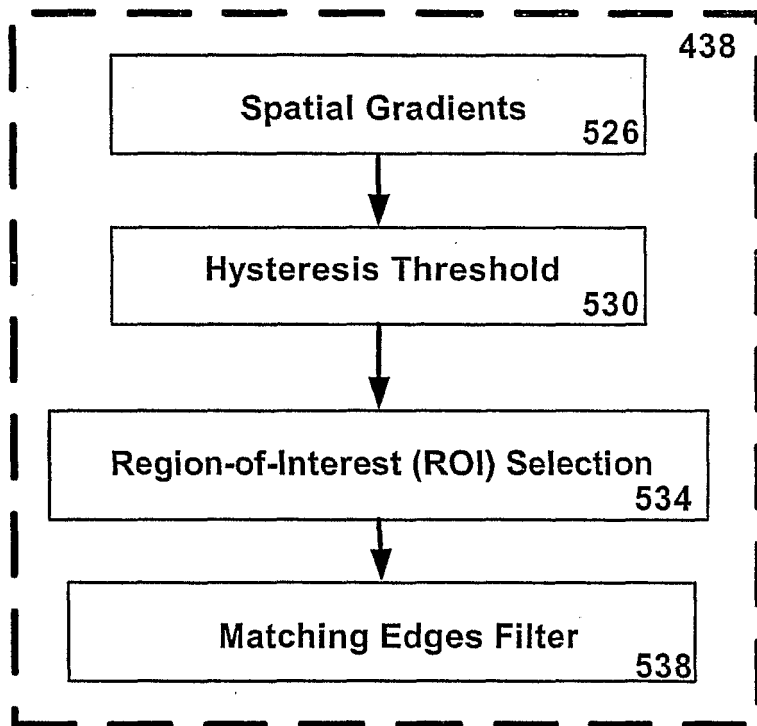
7/22



8/22

**FIG. 7**

9/22

**FIG. 8A****FIG. 8B****FIG. 8C**

10/22

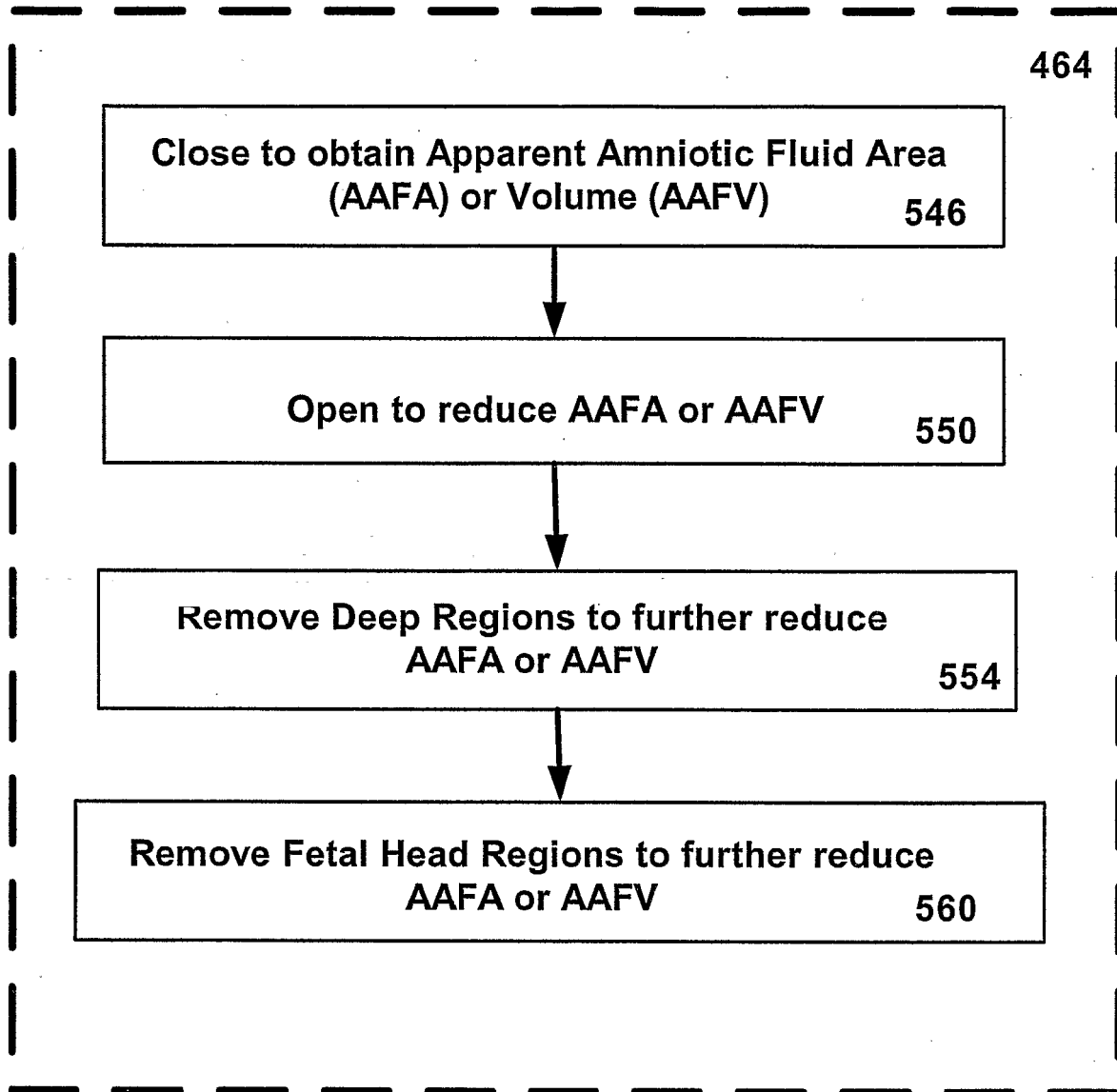
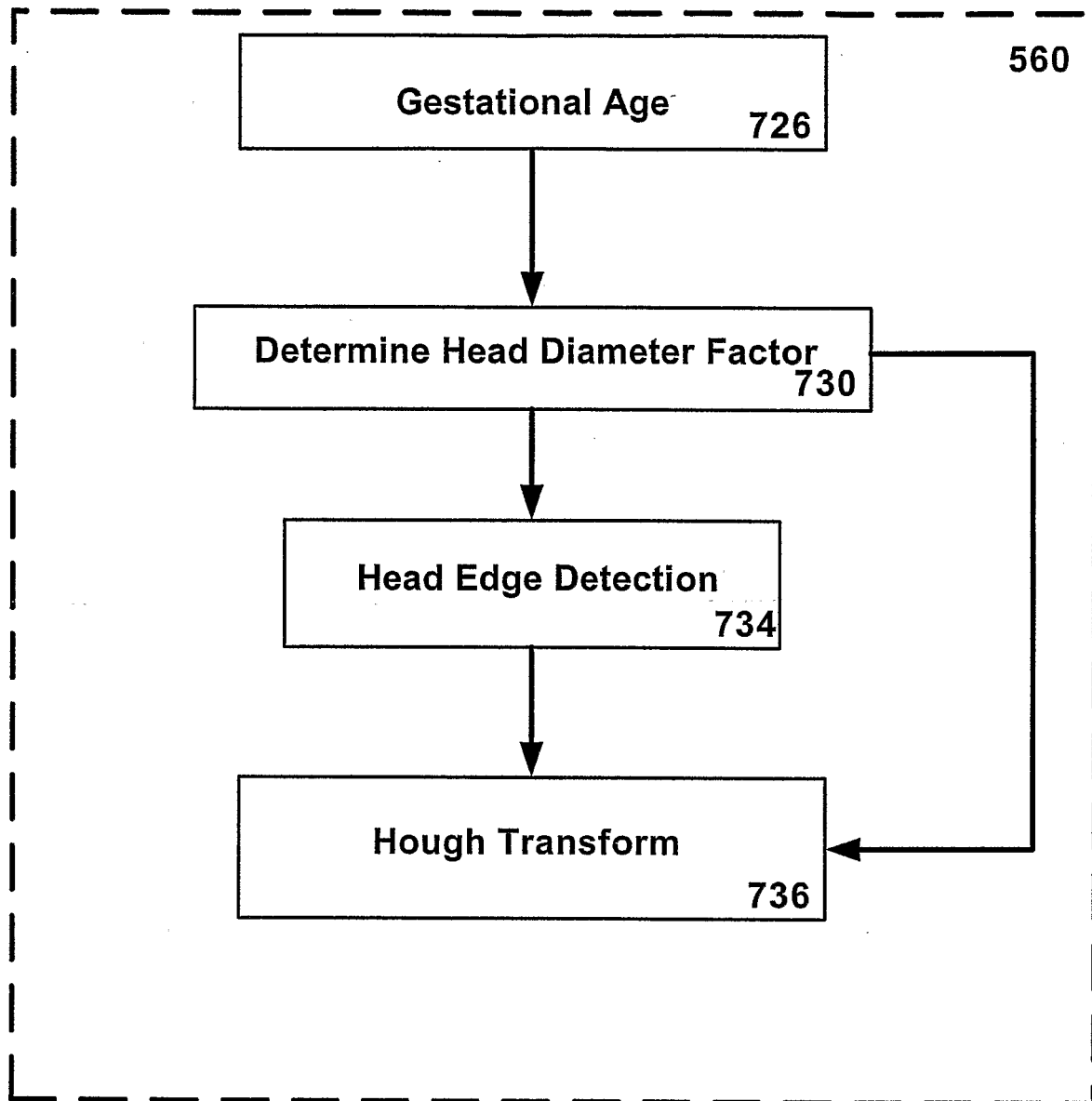
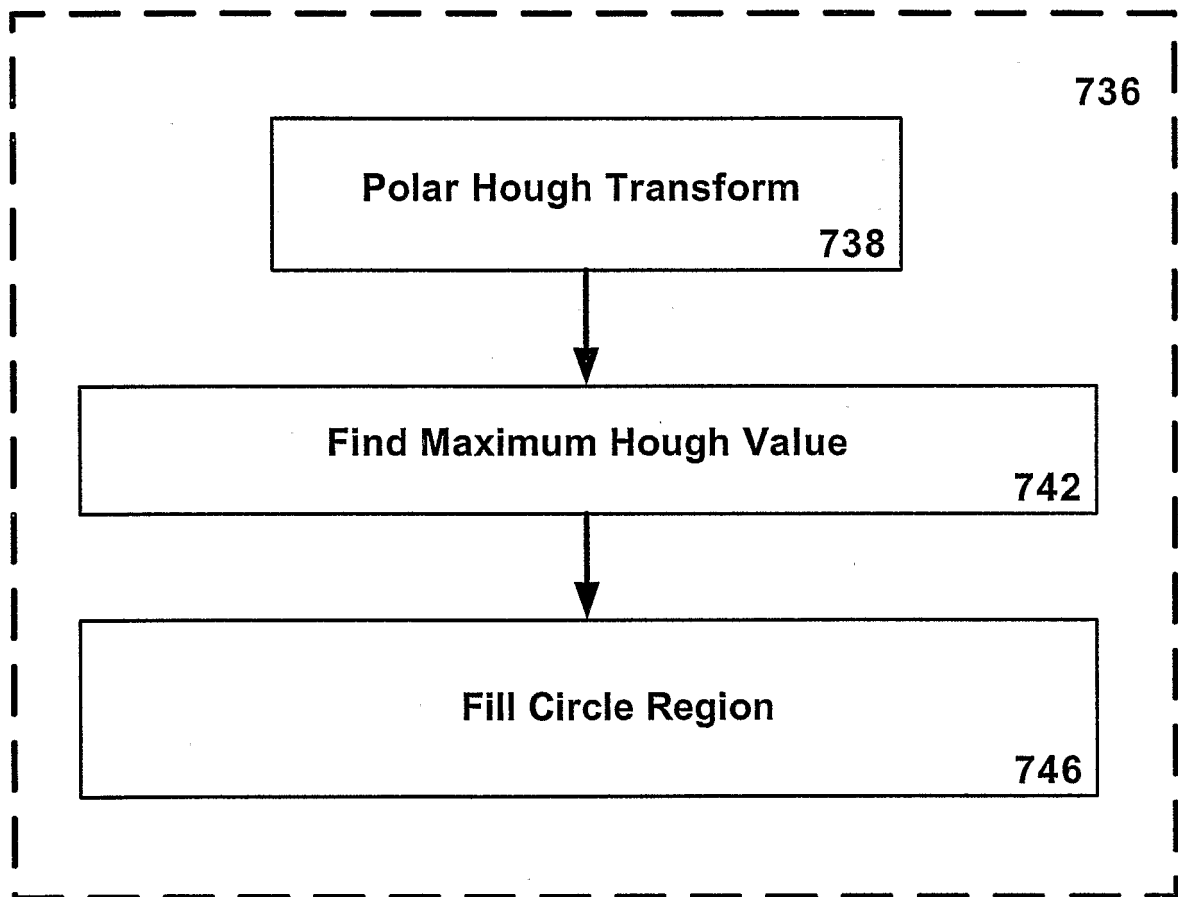


FIG. 8D

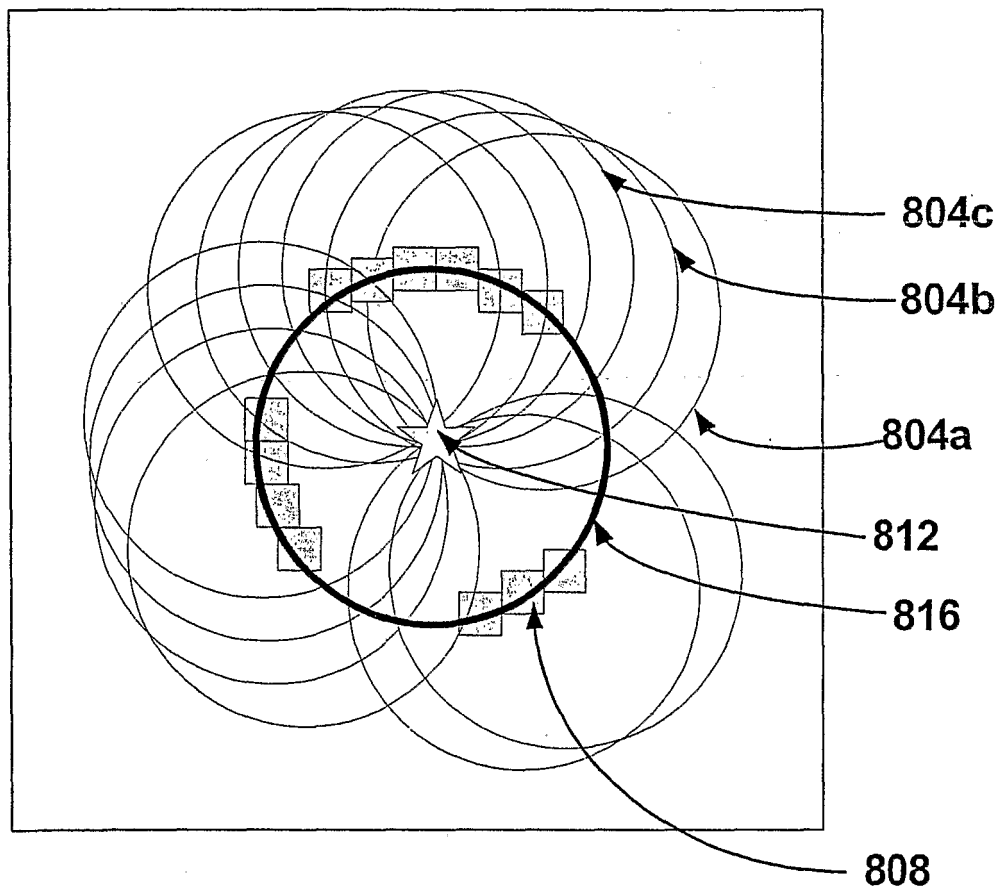
11/22

**FIG. 8E**

12/22

**FIG. 8F**

13/22

**FIG. 9**

14/22

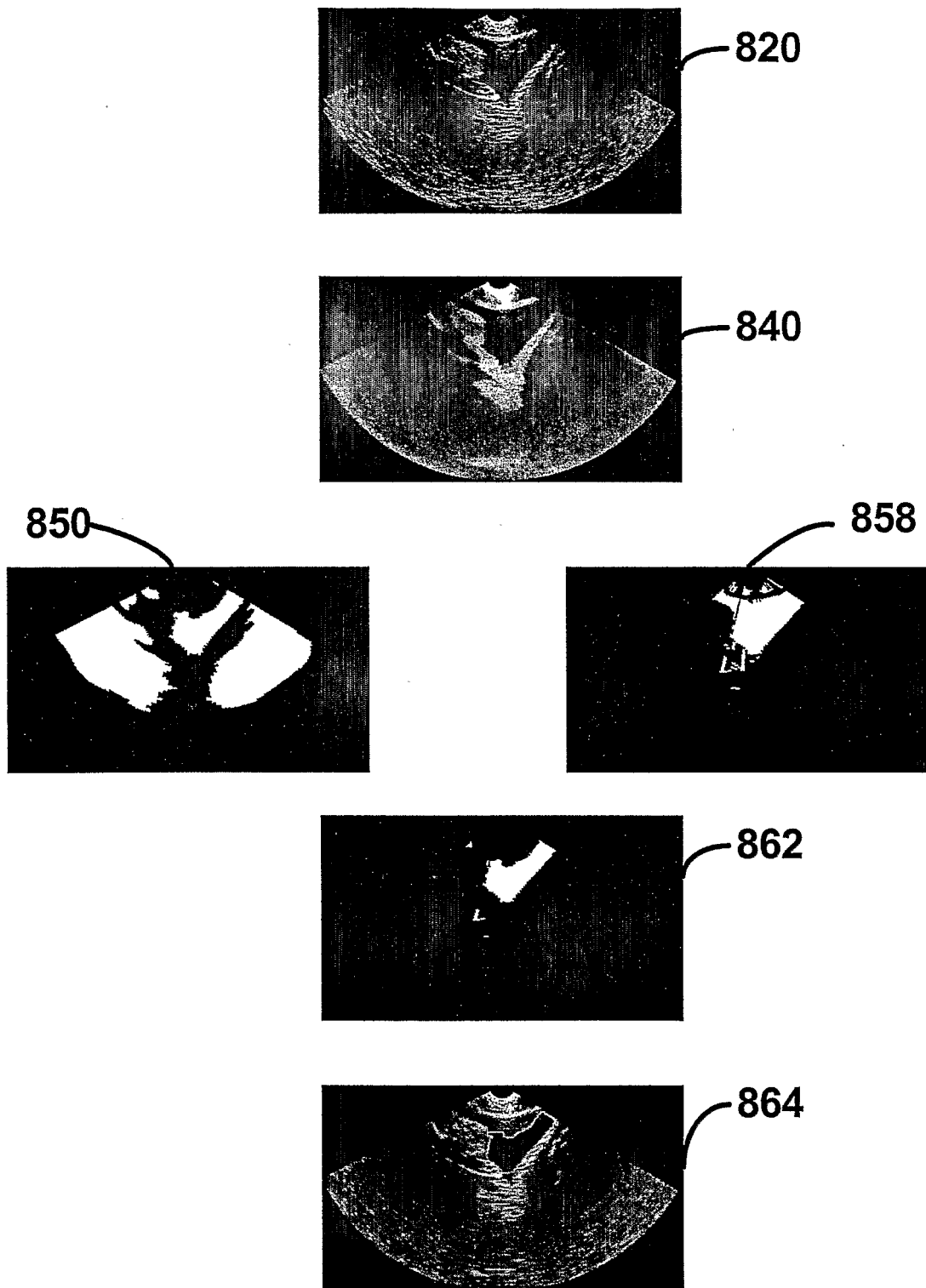


FIG. 10

15/22

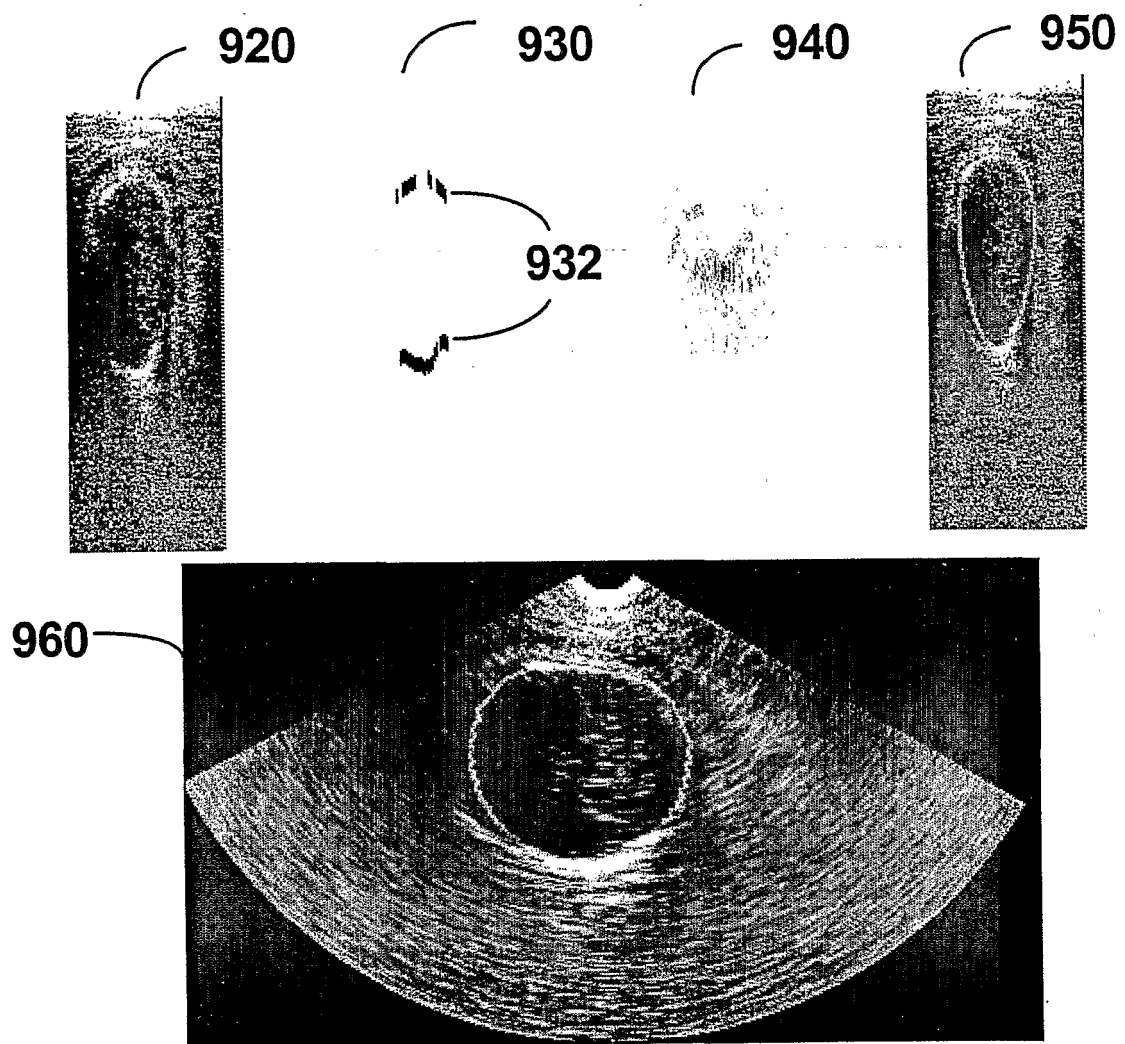


FIG. 11

FIG. 12

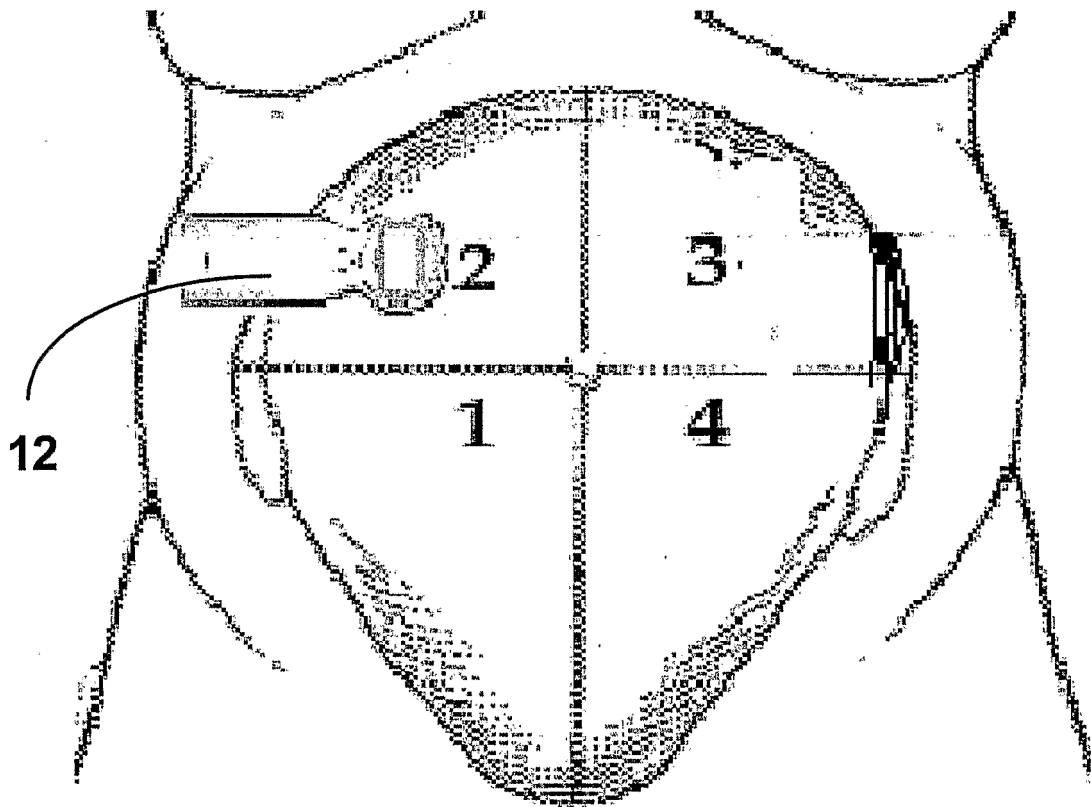


**Sonographer's
Outline →**

**Algorithm's
Outline →**

17/22

Fig. 13



18/22

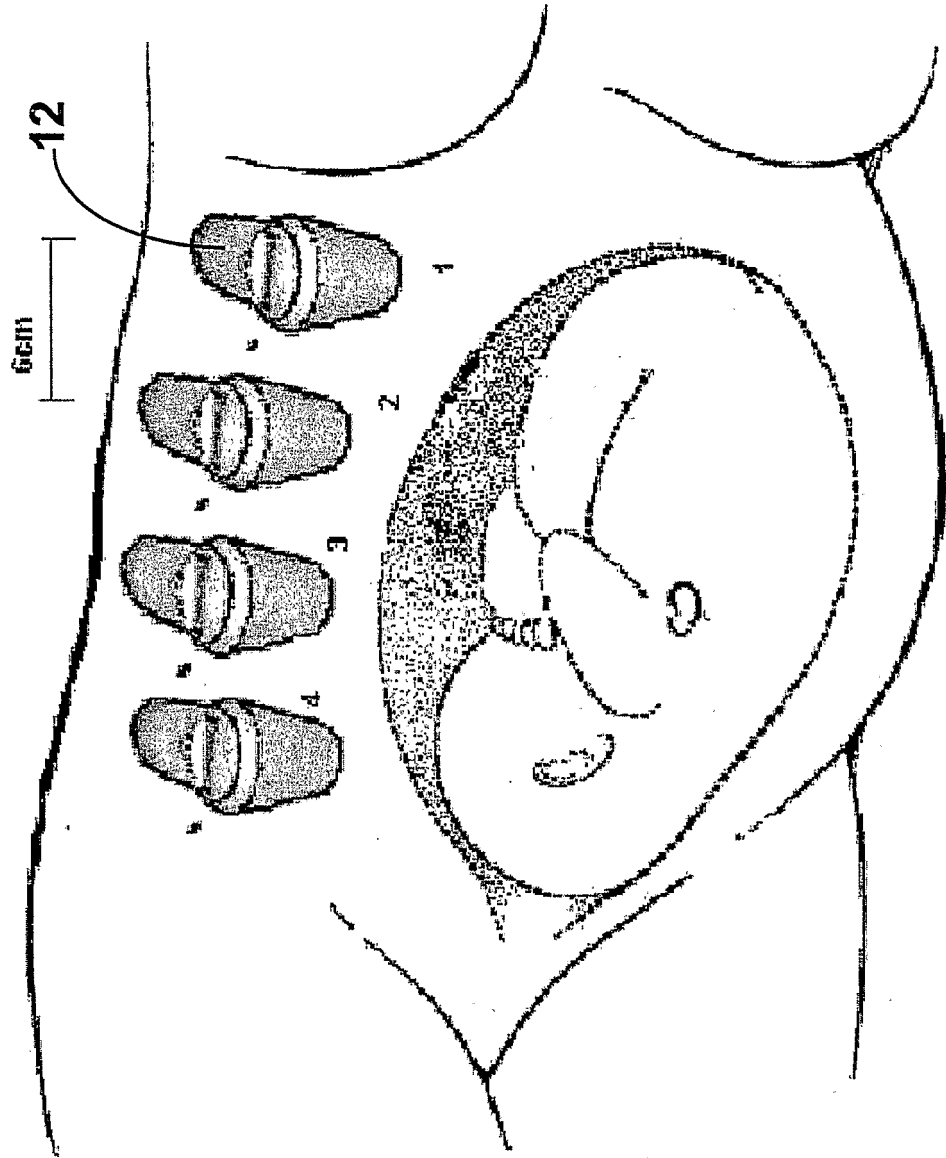
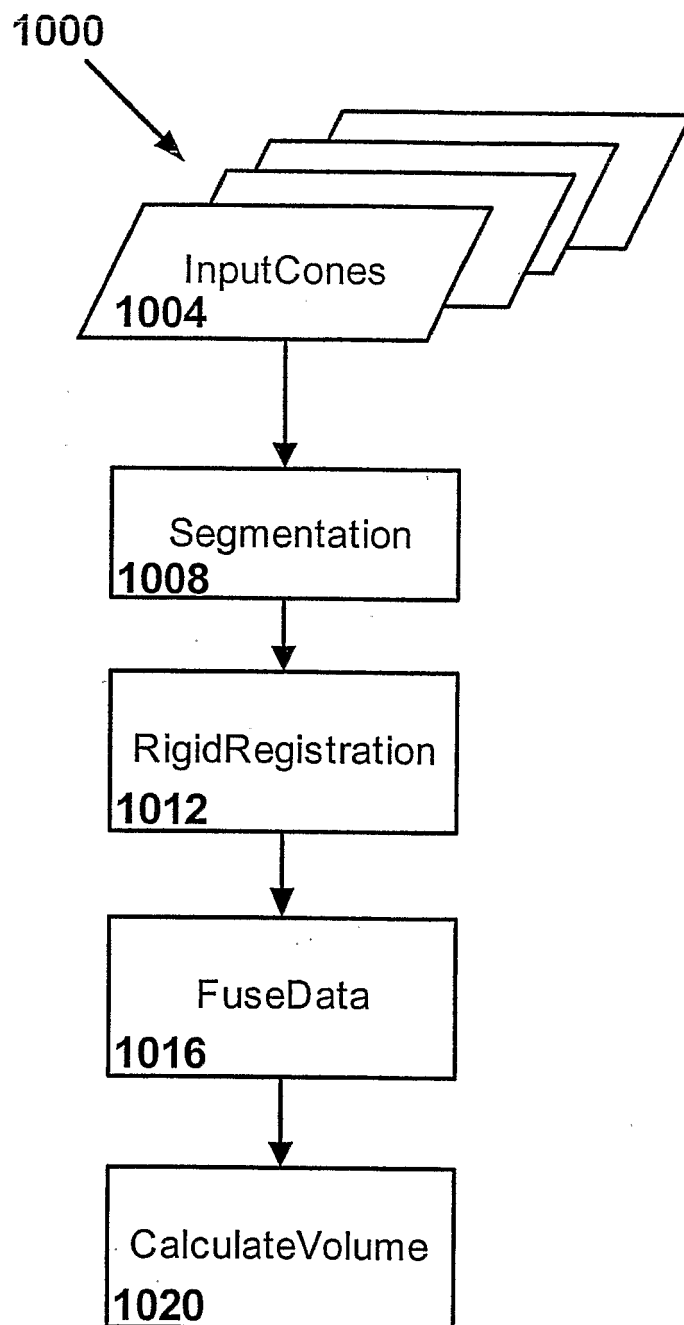


Fig. 14

19/22

**Fig. 15**

20/22

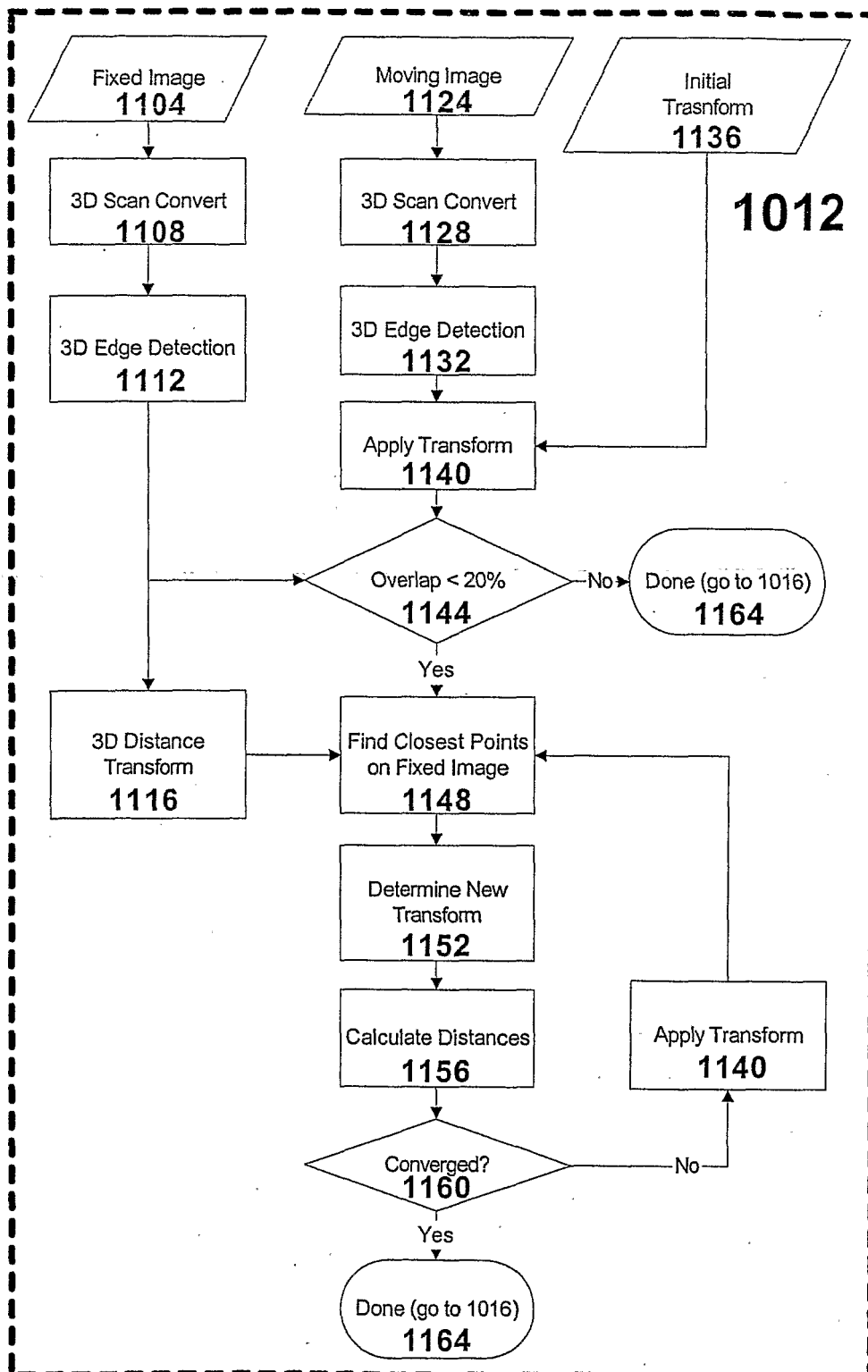
**Fig. 16**

Fig. 17A

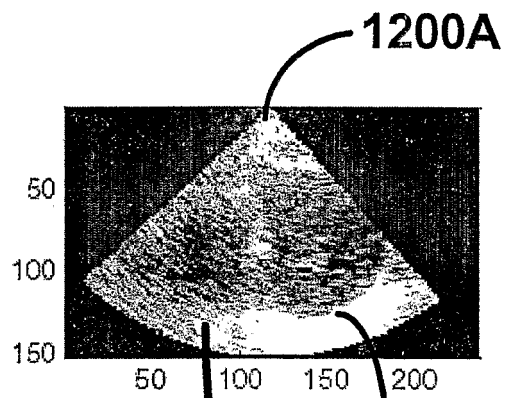


Fig. 17B

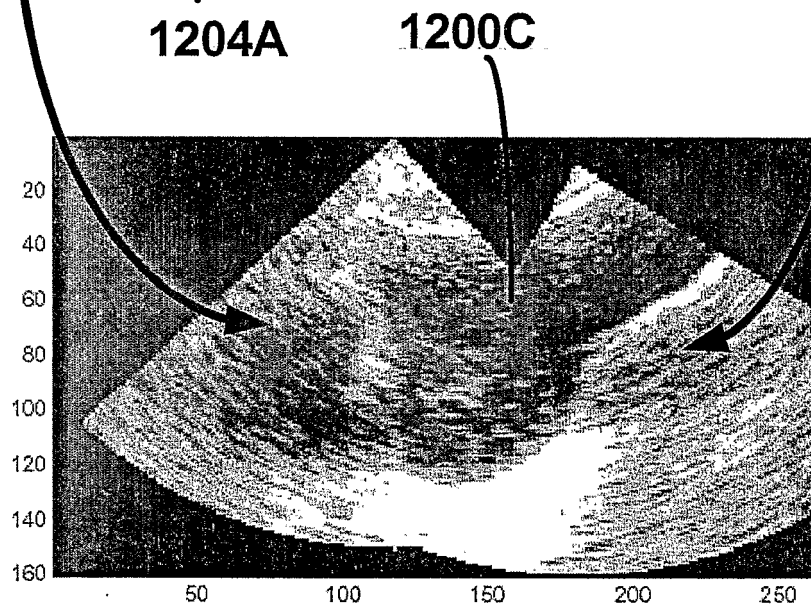
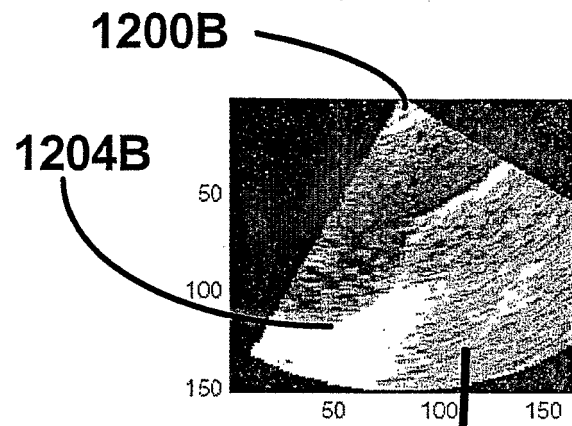
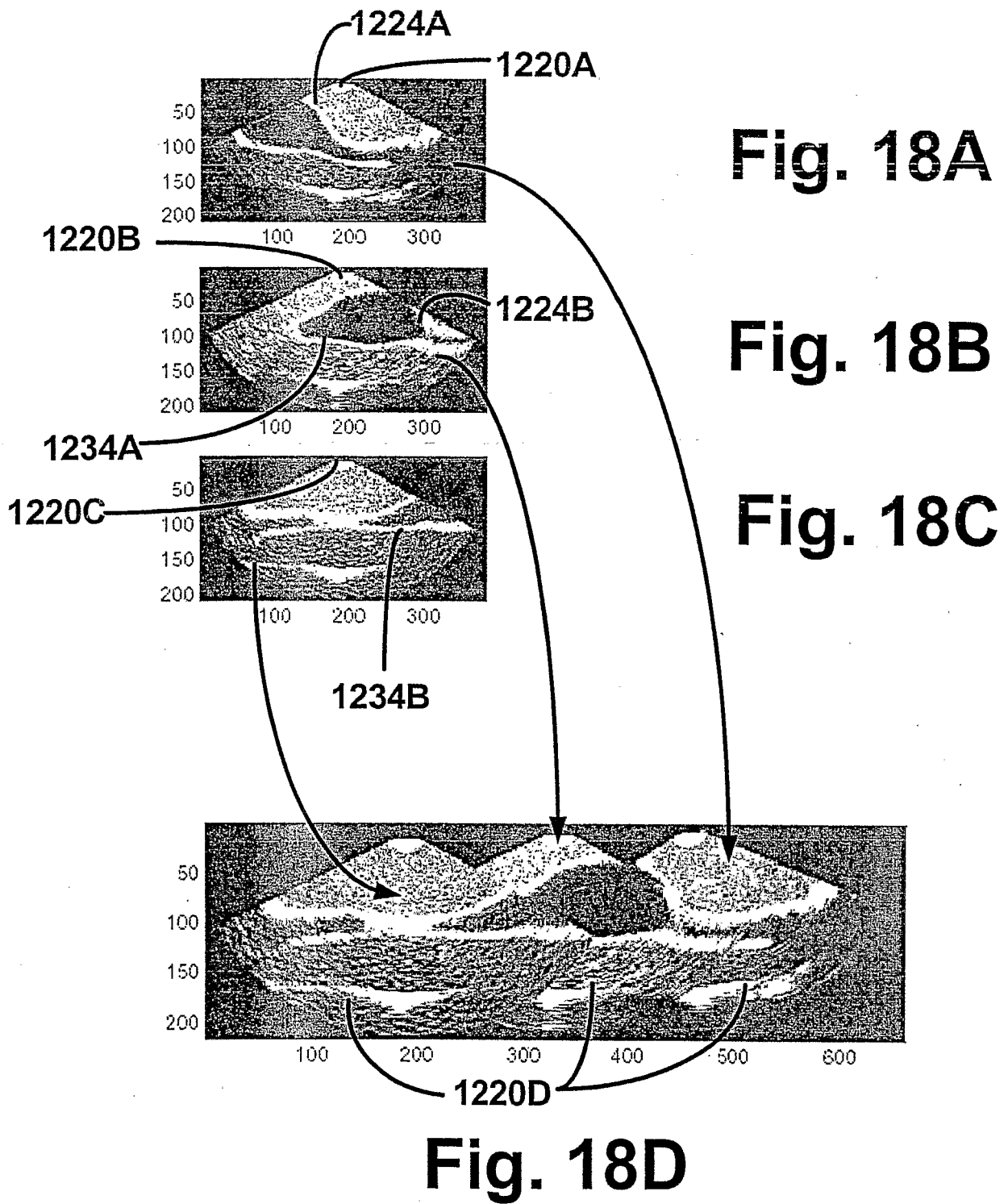


Fig. 17C

22/22



专利名称(译)	基于3D超声的仪器，用于非侵入性测量羊水量		
公开(公告)号	EP1596718A2	公开(公告)日	2005-11-23
申请号	EP2003810833	申请日	2003-11-05
[标]申请(专利权)人(译)	诊断超声		
申请(专利权)人(译)	超声诊断CORPORATION		
当前申请(专利权)人(译)	超声诊断CORPORATION		
[标]发明人	CHALANA VIKRAM DUDYCHA STEPHEN MCMORROW GERALD J		
发明人	CHALANA, VIKRAM DUDYCHA, STEPHEN MCMORROW, GERALD, J.		
IPC分类号	A61B8/14 A61B8/00 A61B8/08 G01S7/52 G01S15/89 G06T5/00 G06T7/60		
CPC分类号	A61B8/00 A61B8/0866 A61B8/4254 A61B8/4455 A61B8/4472 A61B8/4477 A61B8/4483 A61B8/483 A61B8/5223 A61B8/565 G01S15/8993 G06T7/0012 G06T7/11 G06T7/12 G06T7/62 G06T2207/10136 G06T2207/20061 G06T2207/30004 G06T2207/30044 G16H50/30		
代理机构(译)	WILLIAMSON , BRIAN		
优先权	60/423881 2002-11-05 US PCT/US2003/024368 2003-08-01 WO 60/470525 2003-05-12 US 10/443126 2003-05-20 US 10/633186 2003-07-31 US PCT/US2003/014785 2003-05-09 WO		
其他公开文献	EP1596718A4		
外部链接	Espacenet		

摘要(译)

公开了一种包括收发器（10）的手持式3D超声仪器，其用于非侵入地且自动地测量子宫中的羊水量，需要最少的操作者干预。收发器包括具有触发器（14）和顶部按钮（16）的手柄（12），附接到手柄（12）的收发器壳体（18）和收发器圆顶（20）。使用2D图像处理算法，仪器向用户自动反馈获取3D图像集的位置。用户获取覆盖子宫中所有羊水的一个或多个3D数据集，然后使用优化的3D算法处理该数据，以输出针对任何胎儿脑头体积贡献校正的总羊水量。