APPARATUS, SYSTEMS, AND METHODS FOR TISSUE OXIMETRY AND PERFUSION IMAGING

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority from U.S. provisional patent application serial number 61/434,014 filed on January 19, 2011, incorporated herein by reference in its entirety.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

[0002] Not Applicable

INCORPORATION-BY-REFERENCE OF MATERIAL SUBMITTED ON A COMPACT DISC

[0003] Not Applicable

NOTICE OF MATERIAL SUBJECT TO COPYRIGHT PROTECTION

[0004] A portion of the material in this patent document is subject to copyright protection under the copyright laws of the United States and of other countries. The owner of the copyright rights has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the United States Patent and Trademark Office publicly available file or records, but otherwise reserves all copyright rights whatsoever. The copyright owner does not hereby waive any of its rights to have this patent document maintained in secrecy, including without limitation its rights pursuant to 37 C.F.R. § 1.14.

BACKGROUND OF THE INVENTION

[0005] 1. Field of the Invention

[0006] This invention pertains generally to tissue oximetry, and more particularly to tissue oximetry and perfusion imaging.

[0007] 2. Description of Related Art

[0008] Patients' skin integrity has long been an issue of concern for nurses and in nursing homes. Maintenance of skin integrity has been identified by the American Nurses Association as an important indicator of quality nursing care. Meanwhile, ulcers, and specifically venous and pressure ulcers, remain major health problems, particularly for hospitalized older adults. Detecting early wound formation is an extremely challenging and expensive problem.

[0009] When age is considered along with other risk factors, the incidences of these ulcers are significantly increased. Overall incidence of pressure ulcers for hospitalized patients ranges from 2.7% to 29.5%, and rates of greater than 50% have been reported for patients in intensive care settings. In a multicenter cohort retrospective study of 1,803 older adults discharged from acute care hospitals with selected diagnoses, 13.2% (i.e., 164 patients) demonstrated an incidence of stage I ulcers. Of those 164 patients, 38 (16%) had ulcers that progressed to a more advanced stage.

[0010] Pressure ulcers additionally have been associated with an increased risk of death within one year after hospital discharge. The estimated cost of treating pressure ulcers ranges from $5,000 to $40,000 for each ulcer, depending on severity. Meanwhile, venous ulcers can also cause significant health problems for hospitalized patients, especially in older adults. As many as 3% of the population suffer from leg ulcers, while this figure rises to 20% in those over 80 years of age. The average cost of treating a venous ulcer is estimated at $10,000, and can easily rise as high as $20,000 without effective treatment and early diagnosis.

[0011] Once a patient has been afflicted by a venous ulcer, the likelihood of the wound recurring is also extremely high, and ranges from 54% to 78%. This means that venous ulcers can have severely negative effects on those who suffer from them, significantly reducing quality of life and requiring extensive treatment. The impact of venous ulcers is often underestimated, despite accounting for as much as 2.5% of the total health care budget.

[0012] The high cost and incidence rates of venous ulcers, coupled with the

difficulty in treating them, mark an extremely good opportunity to introduce a low cost, non-invasive system capable of early detection. While traditional laser Doppler systems are able to deliver relatively accurate and reliable information, they cannot be used for continuous monitoring of patients, since they require bulky and extremely expensive equipment. Such solutions that are too expensive or difficult to deploy significantly limit adoption.

[0013] Hence, there is a need to develop a monitoring and preventive solution to scan the tissue and measure tissue perfusion status as a measure for the level of oxygen distribution and penetration thorughout the tissue as an indicator of tissue health. Accordingly, an object of the present invention is the use of photoplethysmographic in conjunction with pressure sensor signals to monitor perfusion levels of patients suffering from or at risk of venous ulcers.

BRIEF SUMMARY OF THE INVENTION

[0014] The systems and methods of the present invention include a compact perfusion scanner configured to scan and map tissue blood perfusion as a mean to detect and monitor the development of ulcers. The device incorporates a platform, a digital signal processing unit, a serial connection to a computer, pressure sensor, pressure metering system, an LED and photodiode sensor pair and a data explorer visual interface.

[0015] The systems and methods of the present invention provide effective preventive measures by enabling early detection of ulcer formation or inflammatory pressure that would otherwise have not been detected for an extended period, thus increasing risk of infection and higher stage ulcer development.

[0016] In a preferred embodiment, the compact perfusion scanner and method of characterizing tissue health status according to the present invention incorporates pressure sensing components in conjunction with the optical sensors to monitor the level of applied pressure on target tissue for precise skin/tissue blood perfusion measurements and oximetry. The systems and methods of the present invention enable new capabilities including but not limited to: measurement capabilities such as perfusion imaging and perfusion

mapping (geometric and temporal), signal processing and pattern recognition, automatic assurance of usage via usage tracking and pressure imaging, as well as data fusion.

[0017] One particular benefit of the sensor-enhanced system of the present invention is the ability to better manage each individual patient, resulting in a timelier and more efficient practice in hospitals and even nursing homes. This is applicable to patients with a history of chronic wounds, diabetic foot ulcers, pressure ulcers or post-operative wounds.

[0018] In addition, alterations in signal content may be integrated with the activity level of the patient, the position of patient's body and standardized assessments of symptoms. By maintaining the data collected in these patients in a signal database, pattern classification, search, and pattern matching algorithms may be used to better map symptoms with alterations in skin characteristics and ulcer development.

[0019] An aspect is an apparatus for monitoring perfusion oxygenation of a target tissue region of a patient, comprising: a scanner comprising: a planar sensor array; the sensor array configured to be positioned in contact with a surface of the target tissue region; the sensor array comprising one or more LED's configured to emit light into the target tissue region at a wavelength keyed for hemoglobin; the sensor array comprising one or more photodiodes configured to detect light reflected from the LED's; and a data acquisition controller coupled to the one or more LED's and to the one or more photodiodes for controlling the emission and reception of light from the sensor array to obtain perfusion oxygenation data associated with the target tissue region.

[0020] Another aspect is a system for monitoring perfusion oxygenation of a target tissue region of a patient, comprising: (a) a scanner comprising: a planar sensor array; the sensor array configured to be positioned in contact with a surface of the target tissue region; the sensor array comprising one or more light sources configured to emit light into the target tissue region at a wavelength keyed for hemoglobin; the sensor array comprising one or more

sensors configured to detect light reflected from the light sources; a pressure sensor coupled to the sensor array; the pressure sensor configured to obtain pressure readings of the sensor array's contact with a surface of the target tissue region; and (b) a data acquisition controller coupled to the one or more sensors and for controlling the emission and reception of light from the sensor array to obtain perfusion oxygenation data associated with the target tissue; and (c) a processing module coupled to the data acquisition controller; (d) the processing module configured to control sampling of the pressure sensor and sensor array for simultaneous acquisition of perfusion oxygenation data and pressure sensor data to ensure proper contact of the scanner with the surface of the target tissue region.

[0021] A further aspect is a method for performing real-time monitoring of perfusion oxygenation of a target tissue region of a patient, comprising: positioning a sensor array in contact with a surface of the target tissue region; emitting light from lights sources in the sensor array into the target tissue region at a wavelength keyed for hemoglobin; receiving light reflected from the light sources; obtaining pressure data associated with the sensor array's contact with a surface of the target tissue region; obtaining perfusion oxygenation data associated with the target tissue region; and sampling the perfusion oxygenation data and pressure data to ensure proper contact of the sensor array with the surface of the target tissue region.

[0022] It is appreciated that the systems and methods of the present invention are not limited to the specific condition of ulcer or wound, but may have broad application in all forms of wound management, such as skin diseases or treatments.

[0023] Further aspects of the invention will be brought out in the following portions of the specification, wherein the detailed description is for the purpose of fully disclosing preferred embodiments of the invention without placing limitations thereon.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING(S)

[0024] The invention will be more fully understood by reference to the following drawings which are for illustrative purposes only:

[0025] FIG. 1 shows a preferred embodiment of a perfusion oxygenation monitoring (POM) system for analyzing a region of tissue in accordance with the present invention

[0026] FIGS. 2A and 2B illustrate front and right perspective views of the perfusion hardware printed circuit board of the present invention.

[0027] FIG. 3 illustrates an exemplary LED emitter in accordance with the present invention.

[0028] FIG. 4 illustrates LED driver circuit in accordance with the present invention.

[0029] FIG. 5 illustrates an exemplary photodiode read circuit configured for reading the signal from photodiode sensor array.

[0030] FIG. 6 illustrates a calibration setup for calibration of the pressure sensor.

[0031] FIG. 7 shows a plot of results from the pressure verification trials of weights of 50g, 100g, 200g and 500g on a single sensor.

[0032] FIG. 8 is a plot showing measured pressure response curve, interpolated curve (exponential), and the point where the pressure sensor is specified to saturate.

[0033] FIG. 9 shows results from pressure verification trials on a second 1-pound sensor.

[0034] FIG. 10 is a plot showing raw pressure response curves, and various fits.

[0035] FIG. 11 illustrates a PC setup for running the perfusion oxygenation monitoring (POM) system of the present invention.

[0036] FIG. 12 shows a screenshot of the hardware configuration module interface in accordance with the present invention.

[0037] FIG. 13 shows a screenshot of the graphical user interface in

accordance with the present invention.

[0038] FIG. 14 shows an exemplary interpolation performed via a Kriging algorithm.

[0039] FIG. 15 shows a schematic diagram of a marker pattern used for testing the feature extraction module.

[0040] FIG. 16 illustrates the setup of FIG. 15 overlaid on an image.

[0041] FIG. 17 illustrates a block diagram of a method for outputting a mapped and interpolated perfusion image.

[0042] FIG. 18 shows an example of heterodyning used to help eliminate in-band noise in accordance with the present invention.

[0043] FIG. 19 is a plot of the theoretical response of the subtraction method of FIG. 18 in relation to noise and correction frequency.

[0044] FIG. 20 is a plot of the frequency response of the subtraction method shown on a dB scale.

[0045] FIG. 21 shows results from employing noise subtraction on a high frequency LED drive signal, and averaging several LED drive periods to obtain similar data rates as before.

[0046] FIG. 22 illustrates a zoomed view of FIG. 21.

[0047] FIG. 23 shows a sample of the time domain signals used for comparison of neck and thumb tissue measurements.

[0048] FIG. 24 shows the frequency domain representation of the measured signals.

[0049] FIG. 25 shows results from extracted plethysmograph signals of the forehead.

[0050] FIG. 26 shows a comparison of readings of extracted plethysmograph signals from under the knuckle on the thumb.

[0051] FIG. 27 shows results from varying pressure using the reflectance sensor on the neck.

[0052] FIG. 28 shows the results from both over and to the side of the black tape.

DETAILED DESCRIPTION OF THE INVENTION

[0053] FIG. 1 shows a preferred embodiment of a perfusion oxygenation monitoring (POM) system 10 for analyzing a region of tissue 52 of a patient 18 in accordance with the present invention. System 10 generally comprises six primary components: red/infrared LED array 44, photodiode array 46, pressure sensor 50, pressure metering system 48(which includes amplification and filtering circuitry), data acquisition unit 40, digital signal processing module 12 and application module 14 having a user interface.

[0054] The system 10 comprises sensing hardware component 16 that includes arrays of emitters/sensors (44, 46, 50) and data acquisition unit 40, preferably in a handheld enclosure (not shown). The LED array 44 and photodiode arrays 46 coupled to the data acquisition unit 40 (e.g. through cabling or wireless connection) can be physically configured in a variety of arrays. The data acquisition unit 40 is preferably capable of interfacing with a large number of individual LEDs and photodiodes. Signal amplification and filtering unit 49 may be used to condition the photodiode signal/data prior to being received by the data acquisition unit 40. In a preferred embodiment, the photodiode signal amplification and filtering unit 49 may comprise a photodiode read circuit 120 shown in FIG. 5 and described in further detail below.

[0055] Sensing/scanning hardware component 16 may also include an intensity controller 42 for controlling the output of LED array 44. Intensity controller 42 preferably comprises LED driver circuit 100 shown in FIG. 4, and described in further detail below.

[0056] The data acquisition system 40 also interfaces with application module 14 on PC 154 (see FIG. 11), allowing a user to configure the LED array 44 signaling as well as sampling rate of the signal from photodiode array 46 via a hardware configuration module 34 that is viewed through the graphical user interface 36. Data acquired from DAC 40 is preferably stored in a database 32 for subsequent processing.

[0057] The pressure sensor 50 is configured to measure the pressure applied

from the hardware package 16 on to the patient's tissue, such that pressure readings may be acquired to maintain consistent and appropriate pressure to the skin 52 while measurements are being taken. The pressure sensor 50 may be coupled to pre-conditioning or metering circuitry 48 that includes amplification and filtering circuitry to process the signal prior to being received by the data acquisition controller 40.

[0058] The LED arrays 44 are configured to project light at wavelengths keyed for hemoglobin in the target tissue 52, and the photodiode sensor arrays 46 measure the amount of light that passes through tissue 52.

[0059] The signal processing module 12 then further processes and filters the acquired data via processing scripts 24 and filtering module 22. The signal processing module 12 further comprises a feature extraction module 28, which may be output to visual interface 36 for further processing and visualization. A perfusion data module 26 converts data into a Plethysmograph waveform, which may be displayed on a monitor or the like (not shown). The interface 36 and processing module 12 may also be configured to output an overlay image of the tissue and captured perfusion data 26.

[0060] In order to produce the wavelengths of light corresponding to deoxy and oxyhemoglobin absorption, the system 12 preferably uses light emitting diodes for the emitting source array 44. In a preferred embodiment, the system 10 incorporates the DLED-660/880-CSL-2 dual optical emitter combinations from OSI Optoelectronics. This dual emitter combines a red (660nm) and infrared (880nm) LED into a single package. Each red/infrared LED pair requires a 20mA current source and have a 2.4/2.0V forward voltage respectively. It is appreciated that other light sources may also be used.

[0061] In order to measure a photoplethysmograph, the light reflected from the LED array 44 is detected by the photodiode array 46. In a preferred embodiment, the PIN-8.0-CSL photodiode from OSI Optoelectronics is used. This photodiode has a spectral range of 350nm to 1100nm and has a responsivity of .33 and .55 to 660nm and 900nm light respectively.

[0062] FIGS. 2A and 2B illustrate front and right perspective views of the

perfusion hardware printed circuit board (PCB) 60. PCB 60 comprises LED array 44 of two LED pairs 64 spaced between two arrays 46 of photodiodes 62. The board 60 also comprises pressure sensor 50 to monitor the applied pressure on the target tissue 52.

[0063] As shown in FIG. 2A, the optical sensors (e.g. LED array 44 and photodiode array 46) are located on the front side 66 of the PCB 60 and are configured to face and press onto (either directly or adjacently with respect to transparent cover (not shown)) the target tissue 52.

[0064] Referring to FIG. 2B, driving circuitry, e.g. connector head 70, are located on the back side 68 of the PCB 60 safely out of contact with the test subject, and the front of the PCB (right) which houses the sensor portion of the array. The arrays 44, 46 are located such that connector head 70 and corresponding leads 72 and cables 74 (which couple to the data acquisition unit 40) do not interfere with using the device.

[0065] The arrays 44, 46 are shown in FIG. 2A as two LED's 64 positioned between four photodiodes 62. However, it is appreciated that the array may comprise any number of and planar configuration of at least one LED emitter 64 and one photodiode receiver.

[0066] FIG. 3 illustrates an exemplary LED emitter 64 (OSI Optoelectronics DLED-660/880 CSL-2) having 660nm red emitter 84 and 880nm Infrared emitter 82.

[0067] FIG. 4 illustrates LED driver circuit 100 in accordance with the present invention. LED driver circuit 100 is configured to allow the red LED 88 and infrared LED 82 in the LED package 64 to be driven independently, even though the LEDs are common anode, sharing a V_{DD} connection via leads 80.

[0068] Driver circuit 100 includes a low-noise amplifier 110 coupled to the LED 64. In a preferred embodiment, the amplifier 110 comprises a LT6200 chip from Linear Technologies. However, it is appreciated that other amplifiers available in the art may also be employed. LED driver circuit 100 further comprises a p-channel MOS field-effect transistor (FET) 112 (e.g. MTM76110 by Panasonic), which provides negative feedback. As voltage is increased at

the input, so is the voltage across the 50 ohm resistor 102. This results in larger current draw, which goes through the LED 64, making it brighter. At 2V, approximately 40mA is drawn through the LED 64, providing optimal brightness. If the voltage at the input is increased too far, the voltage drop across the LED 64 will be insufficient to turn it off, but there will still be a large amount of current flowing through the LED 64 and resistor 102, resulting in large heat buildup. For this reason, the input voltage is ideally kept below 3V to minimize overheating and prevent component damage. If the input to the op-amp 110 is floated while the amp 110 is powered, a 100k pull-down resistor 104 at the input and 1k load resistor 108 at the output ensure that the circuit 100 remains off. The 1k load resistor 108 also ensures that the amp 110 is able to provide rail to rail output voltage. The 1uF capacitor 114 ensures that the output remains stable, but provides enough bandwidth for fast LED 64 switching. To provide further stabilization, the driver circuit 100 may be modified to include Miller compensation on the capacitor 114. This change improves the phase margin for the driver circuit 100 at low frequencies, allowing more reliable operation.

[0069] FIG. 5 illustrates an exemplary photodiode read circuit 120 configured for reading the signal from photodiode sensor array 46. In a preferred embodiment, the photodiode 62 may comprise an OSI Optoelectronics PIN-8.0-DPI photodiode, PIN-4.0DPI photodiode, or alternatively PIN-0.8-DPI photodiode which has lower capacitance for the same reverse bias voltage.

[0070] The photodiode read circuit 120 operates via a simple current to voltage op-amp 124 as shown in Figure 14. The positive input pin of the op-amp 124 (e.g. LT6200 from Linear Technologies) is driven by a voltage divider 122, providing 2.5V (half of V_{DD}). The negative pin is hooked up to the photodiode 62, which is reverse biased, and through feedback to the output of the amplifier 124.

[0071] The feedback is controlled by a simple low pass filter 126 with a 2.7pF capacitor 129 and a 100 kilo-ohm resistor 130. The 0.1uF capacitor 128 is used to decouple the voltage divider from ground. The circuit amplifies the

current output of the photodiode and converts it to voltage, allowing the data acquisition unit to read the voltage via its voltage input module.

[0072] It is appreciated that the individual components of the LED driver circuit 100 and photodiode read circuit 120 are shown for exemplary purposes only, and that other models, or types of components may be used as desired.

[0073] In one embodiment of the present invention, the data acquisition controller 40 comprises National Instruments CompactRIO 9014 real-time controller coupled with an NI 9104 3M gate FPGA chassis. The data acquisition controller 40 interfaces with the LED arrays 44 and photodiodes 46 using three sets of modules for current output, current input, and voltage input.

[0074] In one embodiment, the controller 40 comprises a processor, real-time operating system, memory, and supports additional storage via USB (all not shown). The controller 40 may also include an Ethernet port (not shown) for connection to the user interface PC 154. The controller 40 comprises an FPGA backplane, current output module (e.g. NI 9263), current input module (e.g. NI 9203), and voltage input module (e.g. NI 9205) allowing multiple voltage inputs from photodiode/amplifier modules.

[0075] The POM system 10 preferably employs a pressure sensor 50 to measure pressure and ensure consistent results (e.g. 1lb. Flexiforce sensor). Due to the confounding effect varying pressure can have on plethysmograph measurements, readings from the pressure sensor 50 provide a metric from which the user can apply the sensor hardware 16 to the patient's skin 52.

[0076] The pressure sensor 50 is preferably attached behind the LED array 44, and measures the pressure used in applying it to a target location. The pressure sensor 50 is preferably configured to deliver accurate measurements of pressure in a specified range, e.g. a range from zero to approximately one pound, which encompasses the range of pressures that can reasonably be applied when using the POM sensing hardware 16.

[0077] The pressure sensor 50 is used to guide the user into operating the scanner 16 more consistently, so that the sensor/scanner 16 is positioned in a similar manner every measurement. The oximetry data that is taken is thus

verified to be accurately taken by readings from the pressure sensor 50.

[0078] In a preferred embodiment, the pressure sensor 50 is calibrated in order to ensure that the pressure sensor gives repeatable, well understood measurements that can be directly translated into raw pressure values. FIG. 6 illustrates a calibration setup 140 for calibration of the pressure sensor 50. A rubber pressure applicator 144 was filed down to a flat surface, and used to distribute the weight on the pressure sensitive region of the Flexiforce sensor 50. A weight 142 was used to distribute weight over the active region of the sensor 50. An experiment was conducted using 4 weights in a range from 50g to 500g. Pressure was applied directly to the pressure sensor 50 via applicator 144, and its outputs recorded.

[0079] The results in FIGS. 7-10 show a nonlinear but steady trend, which data can be used to translate any future measurement from the pressure sensor into an absolute pressure value.

[0080] FIG. 7 shows a plot of results from the pressure verification trials of weights of 50g, 100g, 200g and 500g on a single sensor. FIG. 8 is a plot showing measured pressure response curve, interpolated curve (exponential), and the point where the pressure sensor is specified to saturate. FIG. 9 shows results from pressure verification trials on a second 1-pound sensor. For this experiment, additional intermediate weight levels (e.g. 150g and 300g) were applied. FIG. 10 is a plot showing raw pressure response curves, and various fits. The exponential fit serves as the best fit for both sensors tested.

[0081] While the system 10 optimally uses data from the pressure sensor 50 to verify proper disposition of the scanner on the target tissue site 52, it is appreciated that in an alternative embodiment the user may simply forego pressure monitoring and monitor pressure manually (e.g. tactile feel or simply placing the scanner 16 on the tissue site 52 under gravity).

[0082] Referring to FIG. 11, the user preferably interacts with the data acquisition and control unit 40 through a PC 154 running the processing module 12 and application module 14 comprising graphic user interface 36 (e.g. LabVIEW or the like). In a preferred embodiment, the PC 154

communicates with the data acquisition unit 40 over via an Ethernet connection (not shown). Alternatively, PC 154 communicates with the data acquisition unit 40 via a wireless connection (not shown) such as WIFI, Bluetooth, etc. Data files generated on the data acquisition unit 40 may also be transferred to the PC 154 over an FTP connection for temporary storage and further processing.

[0083] With respect to the PC 154 interface shown in FIG. 11, the individual LED's 64 of LED array 44 project light at wavelengths keyed for hemoglobin, and the photodiode sensors 62 measure the amount of light that passes through and is reflected from tissue 52. The data acquisition unit 40 generally comprises a digital TTL output 152 coupled to the LED's 64 and analog DC input 150 for photodiodes 62. The signal processing module 12 then further processes and filters this data, which is then transmitted to the graphical user interface 36 for further processing and visualization. The data may then be converted into a Plethysmograph waveform to be displayed.

[0084] FIG. 12 shows a screenshot 160 of the hardware configuration module 34 interface. Inputs can be selected for adjusting the LED array 44 parameters in fields 166, voltage channel settings in fields 164, current channel settings in fields 162, in addition to other parameters such as the sampling period, pressure sampling period, etc.

[0085] FIG. 13 shows a screenshot 170 of the graphical user interface 36 that also serves as data management and explorer to allow a user to easily read the perfusion sensors, and observe a variety of signals. The screenshot 170 shows integration of the data captured from blood oximetry sensors (photodiode array 46 and LED array 44), from pressure sensor 50, and the tracking/position data captured by the scanning the photodiode array 46 and LED array 44. The screenshot 170 shows a first window 172 that displays the Plethysmograph waveform (2 seconds shown in FIG. 13), and a second window 174 showing the absolute x and y axis movement that has been performed with the scanner. The graphical user interface 36 is also able to map this to the measured SPO_2 data (e.g. via toggling one of the display

windows 172 and 174). The bar 176 on the right of the screenshot 170 is the pressure gauge from pressure sensor 50 readings, showing approximately half of maximum pressure being applied. The gauge 176 preferably displays how much pressure the user is applying versus the maximum measurable pressure in a color coded bar (as more pressure is applied the bar changes from blue to green to red). The gauge 176 is preferably mapped to optimum pressure values for different locations.

[0086] In order to provide a more informative map of perfusion in a local region, interpolation of blood oximeter data may be conducted using sensor tracking data. The optical oximeter sensor 16 provides absolute SPO_2 readings, giving the percent of blood that is oxygenated. This information, when associated with the location it was taken from, can be used to generate a map of blood oxygenation. In a preferred embodiment, the LED array 44 used for generating SPO_2 readings is also used for determining location. However, it is appreciated that another optical sensor, e.g. laser (not shown), may be used to obtain location readings independently of the LED SPO_2 readings. In such configuration, a low-power laser (similar to a laser –tracking mouse) is used to image a small area at very fast intervals, and then detects movement by how that image has shifted. This information is then converted to two dimensional 'X' and 'Y' position and displacement measurements.

[0087] In a preferred embodiment, interpolation is performed via a Kriging algorithm, and data points are mapped using the oximeter sensor 16 to track movement of the sensor 16 over the test area. Kriging is a linear least squares interpolation method often used for spatially dependent information. The interpolation is used to fill in the blank spots that a scan may have missed with estimated values. The interpolated data is compiled into a color coded image, and displayed to the user. This allows an accurate, anisotropic interpolation of the raw data, which makes the end result much easier to visualize. An example interpolation is shown in FIG. 14. Movement of the sensor hardware 16 was mostly one dimensional in this example, resulting in a linear trend across the x axis. This is due to the low variance of points in that

direction (note the total displacement of approximately 40 in the X direction compared to 1400 in the Y).

[0088] To aid in visualizing the collected blood oximetry data, the processing software 12 preferably includes a feature extraction module 28 that that can detect markers on a picture, and then properly align and overlay blood oximetry data 26 (see FIGS. 1, 17). In a preferred method, the feature extraction module 28 takes images (e.g. pictures taken from a camera of the scan site), and superimposes the perfusion data directly over where it was taken from.

[0089] FIG. 15 shows a schematic diagram of a marker pattern 200 used for testing the feature extraction module 28. FIG. 16 illustrates the setup of FIG. 15 overlaid on an image 205. Three markers (202, 204 and 206) were used as delimiting points for a given scan area 208. A first marker 202 was used to determine rotation angle for the image. A second marker 206 was used to determine the left boundary (image position) for the image. A third marker 204 was used to determine the width of the image. The markers (202, 204 and 206) can be any color, but green is the ideal color, as it is easily distinguished from all skin tones. For a clear illustration of the feature extraction software, small plastic green boxes were used to represent points 202, 204, and 206 (see FIG. 16), and the image 205 was quickly edited to place three of them in a likely pattern. Aside from this manipulation, all other images were generated on the fly by the software. A grid 208 was used as sample data, to more clearly illustrate what is being done by the tool.

[0090] In one embodiment a mobile application (not shown) may be used to facilitate easy capture and integration of pictures for the processing software 12. The application allows a user to quickly take a picture with a mobile device (e.g. smartphone, or the like) and have it automatically sent over Bluetooth for capture by the processing software 12. The picture may then be integrated with the mapping system.

[0091] FIG. 17 illustrates a block diagram of a method 220 for outputting a mapped and interpolated perfusion image (e.g. with processing module 12).

An example of code for carrying out method 220 may be found in the Source Code Appendix attached hereto. It is appreciated that the provided code is merely one example of how to perform the methods of the present invention.

[0092] Acquired data from the data acquisition unit 40 (which may be stored on server 32) is first extracted at step 222 (via processing scripts 24). This extracted data is then used for simultaneously extracting location data, perfusion data and pressure data from each measurement point. The processing software 12 may simultaneously sample location, perfusion, and pressure readings (e.g. at 3Hz interval), in order to creating a matching set of pressure, position, and blood oxygen measurements at each interval.

[0093] In order to generate useful information and metrics from the raw data recorded by the perfusion module 228, a number of algorithms are used.

[0094] At step 230, features are extracted from the data (e.g. via the feature extraction module 28). Position data corresponding to the hardware sensor 16 location is then mapped at step 232. After a scan has been completed, the oximetry data is mapped at step 234 to appropriate coordinates corresponding to the obtained sensor position data from step 232. At step 236, the mapped data is interpolated (e.g. using the Kriging algorithm shown in FIG. 14). The interpolated data may be compiled into a color coded image, and displayed to the user, and/or the perfusion data may then overlayed on a background image (e.g. image 205) of the scan site as described in FIGS. 15 and 16.

[0095] On the perfusion side, RF noise filtering is then performed on the extracted data at step 224. Motion noise is then removed at step 226 to obtain the perfusion data at step 228. Steps 224 and 226 may be performed via filtering module 22.

[0096] In a preferred method illustrated in FIG. 18, heterodyning is used to help eliminate in-band noise. The data recorded from when the LED arrays 44 are off is subtracted from adjacent data from when LED arrays 44 are on (subtraction method). This creates high frequency noise, but removes low frequency in band noise, which is a larger issue. The additional high frequency noise that is introduced is then filtered out by a low pass filter. The algorithms

are configurable to allow the preservation of high frequency information of the PPG signals.

[0097] As illustrated in FIG. 18, relevant noise information from the areas marked *1* and *2* is used to calculate the noise that appears in area *3*. This may be done by either the single-sided method or the doubled-sided method.

[0098] For the single sided method, only the preceding noise information from area *1* is used, and the relevant noise level is assumed to be the same in area 1 and 3. For the double sided method, noise from areas *1* and *2* is averaged. Finally, interpolation of the noise at *3* is attempted via interpolation, using the data from all available noise periods, preceding and following the target data point (*3*). The measurement data is averaged in these areas to generate a single point for each LED 64 pulse. The result is then low-pass filtered at the end to remove high frequency noise.

[0099] FIG. 19 is a plot of the theoretical response of the subtraction method of FIG. 18 in relation to noise and correction frequency, determined by adding sinusoidal noise of a wide range of frequencies to a square wave signal, applying the noise cancellation method (correction method), and measuring the ratio of remaining noise to original noise. Measurements were averaged across all phases for a given frequency. FIG. 20 is a plot of the frequency response of the subtraction method shown on a dB scale.

[00100] For the frequency response plots shown in FIGS. 19 and 20, the frequency is normalized to the frequency of the simulated LED drive signal, with 1 meaning the noise is the same frequency as the drive signal and 2 meaning it is double the drive frequency, and so forth.

[00101] FIGS. 21 and 22 are plots showing the extracted plethysmograph signals employing the aforementioned noise cancelation (subtraction) method of FIG. 18 on a high frequency LED drive signal compared to the scenario when no noise cancellation technique is performed. FIG. 21 shows results from employing noise subtraction on a high frequency LED drive signal, and averaging several LED drive periods to obtain similar data rates as before. Note the successful noise reduction at around 1.5s. FIG. 22 is a zoomed

version of FIG. 21, showing the noise spike that is removed by differential noise subtraction. These plots show that the noise subtraction method of the present invention is effective in removing in band noise.

[00102] Frequency domain analysis/experiments were performed with the frequency domain signals of the plethysmograph measurements. The experiments revealed not only high magnitude elements at the heart rate frequency, but also its harmonics. This appears fairly consistent between locations.

[00103] In order to verify that the harmonics shown in the frequency domain were not the result of noise or jitter, but represented real components of the pulse waveform, a sinusoid wave was constructed. The sinusoid was created by summing sinusoids at the frequency for each separate pulse waveform peak. This superposition was intended to model the effects of frequency jitter in the waveform, while removing any frequency components due to the pulse waveform shape.

[00104] A comparison of signals is shown in FIGS. 23 and 24. FIG. 23 shows a sample of the time domain signals used for comparison. Neck measurements were compared to thumb measurements, taken at equal pressure. FIG. 24 shows the frequency domain representation of the measured signals. Note the second harmonic at 128BPM (2.13Hz), the third harmonic at 207BPM (3.45Hz), etc. The results demonstrate that the harmonics shown below are indeed intrinsic to the pulse waveform, and are not the result of noise or frequency jitter.

[00105] Experiments were performed on number of body locations, including neck, thumb and forehead using the perfusion system 10 of the present invention. Samples of extracted plethysmograph signals are reported in FIGS. 25-27, which clearly show that perfusion system successfully removes the motion and ambient noises and extracts the plethysmograph signal from different body location.

[00106] FIG. 25 shows results from extracted plethysmograph signals of the forehead. Pressure values are given in terms of resistance measured using

the pressure sensor. Smaller resistances indicate higher applied pressures.

[00107] FIG. 26 shows a comparison of readings of extracted plethysmograph signals from under the knuckle on the thumb. All factors except pressure were held constant between measurements. A moderate pressure clearly results in a better waveform.

[00108] FIG. 27 shows results from varying pressure using the reflectance sensor on the neck. The following experiments show the importance of the integration and fusion of applied pressure with perfusion signal in this system, since the pressure with which the sensor array is applied to the target tissue has a major impact on the perfusion readings as shown in the following figures. It appears that the neck and thumb give best results when moderate (0.15M to 70k-ohm) pressure is applied, while the forehead yield best results with low pressure (above 0.15M-ohm). This may be a result of the neck and thumb being softer tissue than the forehead.

[00109] The perfusion system 10 was also tested on a black tape, as a means to mark locations on tissue. Black tape was used to test as a marker on the skin. The sensor was used to measure signals on the tape, and just to the side of it. An impression on the skin can be seen where the reflectance sensor was used off the tape.

[00110] FIG. 28 shows the results from both over and to the side of the black tape. The results show that using a simple piece of black tape is effective in causing large signal differences, and could therefore be used as a marker for specific body locations.

[00111] Embodiments of the present invention may be described with reference to flowchart illustrations of methods and systems according to embodiments of the invention, and/or algorithms, formulae, or other computational depictions, which may also be implemented as computer program products. In this regard, each block or step of a flowchart, and combinations of blocks (and/or steps) in a flowchart, algorithm, formula, or computational depiction can be implemented by various means, such as hardware, firmware, and/or software including one or more computer program instructions embodied in computer-

readable program code logic. As will be appreciated, any such computer program instructions may be loaded onto a computer, including without limitation a general purpose computer or special purpose computer, or other programmable processing apparatus to produce a machine, such that the computer program instructions which execute on the computer or other programmable processing apparatus create means for implementing the functions specified in the block(s) of the flowchart(s).

[00112] Accordingly, blocks of the flowcharts, algorithms, formulae, or computational depictions support combinations of means for performing the specified functions, combinations of steps for performing the specified functions, and computer program instructions, such as embodied in computer-readable program code logic means, for performing the specified functions. It will also be understood that each block of the flowchart illustrations, algorithms, formulae, or computational depictions and combinations thereof described herein, can be implemented by special purpose hardware-based computer systems which perform the specified functions or steps, or combinations of special purpose hardware and computer-readable program code logic means.

[00113] Furthermore, these computer program instructions, such as embodied in computer-readable program code logic, may also be stored in a computer-readable memory that can direct a computer or other programmable processing apparatus to function in a particular manner, such that the instructions stored in the computer-readable memory produce an article of manufacture including instruction means which implement the function specified in the block(s) of the flowchart(s). The computer program instructions may also be loaded onto a computer or other programmable processing apparatus to cause a series of operational steps to be performed on the computer or other programmable processing apparatus to produce a computer-implemented process such that the instructions which execute on the computer or other programmable processing apparatus provide steps for implementing the functions specified in the block(s) of the flowchart(s),

algorithm(s), formula(e), or computational depiction(s).

[00114] From the discussion above it will be appreciated that the invention can be embodied in various ways, including the following:

[00115] 1. An apparatus for monitoring perfusion oxygenation of a target tissue region of a patient, comprising: a scanner comprising: a planar sensor array; the sensor array configured to be positioned in contact with a surface of the target tissue region; the sensor array comprising one or more LED's configured to emit light into the target tissue region at a wavelength keyed for hemoglobin; the sensor array comprising one or more photodiodes configured to detect light reflected from the LED's; and a data acquisition controller coupled to the one or more LED's and to the one or more photodiodes for controlling the emission and reception of light from the sensor array to obtain perfusion oxygenation data associated with the target tissue region.

[00116] 2. The apparatus of embodiment 1, the scanner further comprising: a pressure sensor coupled to the sensor array; the pressure sensor configured to obtain pressure readings of the sensor array's contact with a surface of the target tissue region; wherein the scanner is configured to obtain pressure sensor readings while obtaining perfusion oxygenation data to ensure proper contact of the scanner with the surface of the target tissue region.

[00117] 3. The apparatus of embodiment 2: wherein the pressure sensors and sensor array are connected to a first side of a printed circuit board (PCB); and wherein the data acquisition controller is connected to the PCB on a second side opposite said first side.

[00118] 4. The apparatus of embodiment 1, wherein each LED comprises dual emitters configured for emitting red (660nm) and infrared (880nm) light.

[00119] 5. The apparatus of embodiment 4: wherein the one or more of the LED's are coupled driver circuit; and wherein the driver circuit is configured to allow the red LED emitter and infrared LED emitter to be driven independently while sharing a common anode.

[00120] 6. The apparatus of embodiment 5, wherein the driver circuit comprises an amplifier; and a field-effect transistor configured for providing negative

feedback.

[00121] 7. The apparatus of embodiment 2, further comprising: a processing module coupled to the data acquisition controller; the processing module configured to control sampling of the pressure sensor and sensor array for simultaneous acquisition of pressure sensor data and perfusion oxygenation data.

[00122] 8. The apparatus of embodiment 7, wherein the processing module is configured to obtain readings from the sensor array to obtain position data of the scanner.

[00123] 9. The apparatus of embodiment 8, wherein the processing module is configured to generate a perfusion oxygenation map of the target tissue.

[00124] 10. The apparatus of embodiment 8, wherein the processing module is configured to control sampling of the pressure sensor and sensor array for simultaneous acquisition of two or more data parameters selected from the group consisting of pressure sensor data, perfusion oxygenation data, and position data, to simultaneously display said two or more data parameters.

[00125] 11. A system for monitoring perfusion oxygenation of a target tissue region of a patient, comprising: (a) a scanner comprising: a planar sensor array; the sensor array configured to be positioned in contact with a surface of the target tissue region; the sensor array comprising one or more light sources configured to emit light into the target tissue region at a wavelength keyed for hemoglobin; the sensor array comprising one or more sensors configured to detect light reflected from the light sources; a pressure sensor coupled to the sensor array; the pressure sensor configured to obtain pressure readings of the sensor array's contact with a surface of the target tissue region; and (b) a data acquisition controller coupled to the one or more sensors and for controlling the emission and reception of light from the sensor array to obtain perfusion oxygenation data associated with the target tissue; and (c) a processing module coupled to the data acquisition controller; (d) the processing module configured to control sampling of the pressure sensor and sensor array for simultaneous acquisition of perfusion oxygenation data and

pressure sensor data to ensure proper contact of the scanner with the surface of the target tissue region.

[00126] 12. The system of embodiment 11: wherein the sensor array comprises one or more LED's configured to emit light into the target tissue region at a wavelength keyed for hemoglobin; and wherein the sensor array comprises one or more photodiodes configured to detect light reflected from the LED's.

[00127] 13. The system of embodiment 12: wherein each of the one or more LED's comprises dual emitters configured for emitting red (660nm) and infrared (880nm) light; wherein the one or more LED's are coupled to the driver circuit; and wherein the driver circuit is configured to allow the red LED emitter and the infrared LED emitter to be driven independently while sharing a common anode

[00128] 14. The system of embodiment 11, further comprising: a graphical user interface; wherein the graphical user interface is configured to display the perfusion oxygenation data and pressure sensor data.

[00129] 15. The system of embodiment 14, the processing module is further configured to obtain readings from the sensor array to obtain position data of the scanner.

[00130] 16. The system of embodiment 15, wherein the processing module is further configured to interpolate the position data to generate a perfusion oxygenation map of the target tissue.

[00131] 17. The system of embodiment 16, wherein the processing module is configured to control sampling of the pressure sensor and sensor array for simultaneous acquisition of two or more data parameters selected from the group consisting of pressure sensor data, perfusion oxygenation data, and position data, to simultaneously display the two or more data parameters.

[00132] 18. The system of embodiment 16, wherein the processing module is configured to receive an image of the target tissue, and overlay the perfusion oxygenation map over the image.

[00133] 19. The system of embodiment 14, wherein the graphical user interface is configured to allow user input to manipulate settings of the sensor array and

pressure sensor.

[00134] 20. The system of embodiment 11, wherein the processing module further comprises: a filtering module; the filtering module configure to filter in-band noise by subtracting data recorded when the one or more light sources are in an "off" state from data recorded when the one or more light sources are in an "on" state.

[00135] 21. A method for performing real-time monitoring of perfusion oxygenation of a target tissue region of a patient, comprising: positioning a sensor array in contact with a surface of the target tissue region; emitting light from lights sources in the sensor array into the target tissue region at a wavelength keyed for hemoglobin; receiving light reflected from the light sources; obtaining pressure data associated with the sensor array's contact with a surface of the target tissue region; obtaining perfusion oxygenation data associated with the target tissue region; and sampling the perfusion oxygenation data and pressure data to ensure proper contact of the sensor array with the surface of the target tissue region.

[00136] 22. A method as recited in embodiment 21: wherein the sensor array comprises one or more LED's configured to emit light into the target tissue region at a wavelength keyed for hemoglobin; and wherein the sensor array comprises one or more photodiodes configured to detect light reflected from the LED's.

[00137] 23. A method as recited in embodiment 22: wherein each of the one or more LED's comprises dual emitters configured for emitting red (660nm) and infrared (880nm) light; the method further comprising independently driving the red LED emitter and infrared LED emitter while the red LED emitter and infrared LED emitter share a common anode.

[00138] 24. A method as recited in embodiment 21, further comprising: simultaneously displaying the perfusion oxygenation data and pressure sensor data.

[00139] 25. A method as recited in embodiment 21, further comprising: acquiring readings from the sensor array to obtain position data of the

scanner.

[00140] 26. A method as recited in embodiment 25, further comprising: interpolating the position data to generate a perfusion oxygenation map of the target tissue.

[00141] 27. A method as recited in embodiment 26, wherein interpolating the position data comprises applying a Kriging algorithm to the acquired position data.

[00142] 28. A method as recited in embodiment 26, further comprising: sampling of the pressure sensor and sensor array for simultaneous acquisition of pressure sensor data, perfusion oxygenation data, and position data; and simultaneously displaying the pressure sensor data, perfusion oxygenation data, and position data.

[00143] 29. A method as recited in embodiment 26, further comprising: receiving an image of the target tissue; and overlaying the perfusion oxygenation map over the image.

[00144] 30. A method as recited in embodiment 21, further comprising: providing a graphical user interface to allow user input; and manipulating sampling settings of the sensor array and pressure sensor according to said user input.

[00145] 31. A method as recited in embodiment 21, further comprising: cycling the one or more light sources between a period when the one or more light sources are on, and a period when the one or more light sources are in an "off" state; and filtering in-band noise by subtracting data recorded from when the one or more light sources are off from data from when the one or more light sources are in an "on" state.

[00146] Although the description above contains many details, these should not be construed as limiting the scope of the invention but as merely providing illustrations of some of the presently preferred embodiments of this invention. Therefore, it will be appreciated that the scope of the present invention fully encompasses other embodiments which may become obvious to those skilled in the art, and that the scope of the present invention is accordingly to be

limited by nothing other than the appended claims, in which reference to an element in the singular is not intended to mean "one and only one" unless explicitly so stated, but rather "one or more." All structural, chemical, and functional equivalents to the elements of the above-described preferred embodiment that are known to those of ordinary skill in the art are expressly incorporated herein by reference and are intended to be encompassed by the present claims. Moreover, it is not necessary for a device or method to address each and every problem sought to be solved by the present invention, for it to be encompassed by the present claims. Furthermore, no element, component, or method step in the present disclosure is intended to be dedicated to the public regardless of whether the element, component, or method step is explicitly recited in the claims. No claim element herein is to be construed under the provisions of 35 U.S.C. 112, sixth paragraph, unless the element is expressly recited using the phrase "means for."

[00147] SOURCE CODE APPENDIX

[00148] The following source code is submitted by way of example, and not of limitation, as an embodiment of signal processing in the present invention. Those skilled in the art will readily appreciate that signal processing can be performed in various other ways, which would be readily understood from the description herein, and that the signal processing methods are not limited to those illustrated in the source code listed below.

```
% clear all;
clc;

%-------------------------------------------------------------------------%
% Detect Heart Rate, Perfusion & SpO2
%-------------------------------------------------------------------------%

%% Input File
% Perfusion = zeros(52,1);
```

```
% for ll = 0:51
% inputfile
=strcat('3.2_s=10k_t=3s_p=5000u_duty=2500u_Richard_two_sensors_volararm
_ch0=min=offset=2500um_volar_arm_ch1=1cmCTtoCT=offset=0_',num2str(ll));

inputfile='gen3\gen3r10';

samplingRate = 10e3; % Sampling Rate in Hz
period = 5e-3; % Period in s
duty = 2.5e-3; % Duty Cycle in s
totalTime = 10; % Total File Time in s
offsetR = 2.5e-3; % Red light offset in s
offsetIR = 0e-3; % Red light offset in s
transTime = 1.2e-4; % Rise/Fall time in s


%% Heuristics for Peak Detection & Blood Oximetry
RED_sens = 0.42; % Photodiode sensitivity @ 660nm in A/W
IR_sens = 0.61; % Photodiode sensitivity @ 880nm in A/W
MAX_HEART_RATE = 220;
MIN_SAMP = 1/((period*5)*MAX_HEART_RATE/60); % Fastest heartrate
allowed

%% Read Input File into Matlab

sensorselect=3;
if sensorselect==1 %5mm

[PD1, PD2, PD3, PD4]=textread(inputfile, '%f%f%f%f%*[^\n]',
'delimiter',','); % PD1 -> central photodiode (Channel 0); PD2 -> Drive
signal (Channel 1)
elseif sensorselect==2 %10mm
```

```
[PD2, PD1, PD3, PD4]=textread(inputfile, '%f%f%f%f%*[^\n]',
'delimiter',','); % PD1 -> central photodiode (Channel 0); PD2 -> Drive
signal (Channel 1)

elseif sensorselect==3


[PD2, PD3, PD1, PD4]=textread(inputfile, '%f%f%f%f%*[^\n]',
'delimiter',','); % PD1 -> central photodiode (Channel 0); PD2 -> Drive
signal (Channel 1)

elseif sensorselect==4

 [PD2, PD3, PD4, PD1]=textread(inputfile, '%f%f%f%f%*[^\n]',
'delimiter',','); % PD1 -> central photodiode (Channel 0); PD2 -> Drive
signal (Channel 1)

end

PD1=-PD1;

% if trial==3
% PD1 = PD1(length(PD1)/2+1:end);
% end
% Data=DownloadFromDB();



% PD1 = Data(1:end,1);
% PD2 = Data(1:end,2);
No_RIR_Waves = totalTime/period; % Total # of RED+IR square waves

%% Noise Cancellation
```

```
% %--------------------------------------%
% % 1. single-sided subtraction
% %--------------------------------------%

averageRed = zeros(No_RIR_Waves, 1);
averageRedStep1 = zeros(No_RIR_Waves, 1);
averageRedStep2 = zeros(No_RIR_Waves, 1);
averageIR = zeros(No_RIR_Waves, 1);
averageNoise_1 = zeros(No_RIR_Waves, 1); % 1st off portion in each
period
averageNoise_2 = zeros(No_RIR_Waves, 1); % 2nd off portion

for i=0:No_RIR_Waves-1
for j=1:(duty-transTime)*samplingRate % Average every period

averageRed(i+1, 1) = averageRed(i+1, 1) +
PD1(ceil(i*period*samplingRate+j+offsetR*samplingRate+transTime*samplin
gRate));

%averageIR(i+1, 1) = averageIR(i+1, 1) +
PD1(floor(i*period*samplingRate+j+offsetIR*samplingRate+transTime*sampl
ingRate));

end

% for j=1:(duty/2)*samplingRate % Average every period, no
transition time because LED is already on, changes are very short
% averageRedStep1(i+1, 1) = averageRed(i+1, 1) +
PD1(ceil(i*period*samplingRate+j+offsetR*samplingRate+transTime*samplin
gRate));
```

```
% averageRedStep2(i+1, 1) = averageRed(i+1, 1) +
PD1(ceil(i*period*samplingRate+j+offsetR*samplingRate+transTime*samplin
gRate+floor((duty/2)*samplingRate)));
% %averageIR(i+1, 1) = averageIR(i+1, 1) +
PD1(floor(i*period*samplingRate+j+offsetIR*samplingRate+transTime*sampl
ingRate));
% end

for j=1:(period-duty-transTime)*samplingRate %Averaging the off
portion for noise subtraction
% averageNoise_1(i+1, 1) = averageNoise_1(i+1, 1) +
PD1(floor(i*period*samplingRate+j+transTime*samplingRate));

averageNoise_1(i+1, 1) = averageNoise_1(i+1, 1) +
PD1(max(2,floor(i*period*samplingRate+j+transTime*samplingRate-(period-
duty-offsetR-transTime)*samplingRate)));

%averageNoise_2(i+1, 1) = averageNoise_2(i+1, 1) +
PD1(floor(i*period*samplingRate+j+(offsetR+duty)*samplingRate));

end

averageRed(i+1, 1) = averageRed(i+1, 1)/floor((dutytransTime)*
samplingRate);
```

```
%averageIR(i+1, 1) = averageIR(i+1, 1)/((dutytransTime)*
samplingRate);
% averageRedStep1(i+1, 1) = averageRedStep1(i+1,
1)/floor((duty/2)*samplingRate);
% averageRedStep2(i+1, 1) = averageRedStep2(i+1,
1)/floor((duty/2)*samplingRate);

averageNoise_1(i+1, 1) = averageNoise_1(i+1, 1)/floor((period-dutytransTime)*
samplingRate); % Use period/2 when using both red and IR
%averageNoise_2(i+1, 1) = averageNoise_2(i+1, 1)/((period/2-dutytransTime)*
samplingRate);

end


averageRed_1 =averageRed-averageNoise_1;
averageRed_step = averageRedStep2-averageRedStep1;


%averageIR_1 = averageIR - averageNoise_2;

averageRed_4 = zeros(No_RIR_Waves/5, 1);
averageIR_4 = zeros(No_RIR_Waves/5, 1);


for i=1:(No_RIR_Waves/5)
for j=1:5
averageRed_4(i) = averageRed_4(i)+averageRed_1((i-1)*5+j);


% averageIR_4(i) = averageIR_4(i)+averageIR_1((i-1)*5+j);

end

averageRed_4(i) = averageRed_4(i)/5;
```

```
% averageIR_4(i) = averageIR_4(i)/5;
end

% %--------------------------------------%
% %2. double-sided subtraction
% %--------------------------------------%
averageNoise_Red = (averageNoise_1 + averageNoise_2) ./ 2; % Average
the off portion on two sides of one on portion

averageNoise_IR = (averageNoise_1(2:end) + averageNoise_2(1:end1))
./2;
averageIR_2 = zeros(No_RIR_Waves, 1);
averageRed_2 = averageRed - averageNoise_Red;
averageIR_2(1:end-1) = averageIR(1:end-1) - averageNoise_IR;
averageIR_2(end) = averageIR(end) - averageNoise_2(end); % Last period
of IR uses single-sided subtraction

% %--------------------------------------%
% 3. interpolation subtraction
% %--------------------------------------%
% Noise_raw = zeros(totalTime * samplingRate, 1); % Store the
low-pass-filtered off portion continously
% x_Noise = zeros(floor(offsetR*samplingRatetransTime*
samplingRate)+floor(offsetIR*samplingRate(
offsetR*samplingRate +
(duty+transTime)*samplingRate))*No_RIR_Waves,1); % coordinates of
Noise_raw
% x_Noise_x = 0;
% Noise_raw_0 = zeros(totalTime * samplingRate, 1);
```

```
%
% for i=0:No_RIR_Waves-1
% for j=1:period*samplingRate
% if (((j<=offsetR*samplingRate)&&(j>transTime*samplingRate))
|| ((j> (offsetR*samplingRate + (duty+transTime)*samplingRate)) && (j
<= offsetIR*samplingRate))) % load off portion to Noise_raw
% Noise_raw_0(floor(i*period*samplingRate+j)) =
PD1(floor(i*period*samplingRate+j));
% end
% end

% end
%
% order = 50; % Pre-low pass filter for spline interpolation
% cutoff = 200/samplingRate; % Cut off frequency = 100 Hz
% y1 = fir1(order, cutoff,'low');
% PD1_LPF = filtfilt(y1,1,Noise_raw_0);
%
% for i=0:No_RIR_Waves-1
% for j=1:period*samplingRate
% if (((j<=offsetR*samplingRate)&&(j>transTime*samplingRate))
|| ((j> (offsetR*samplingRate + (duty+transTime)*samplingRate)) && (j
<= offsetIR*samplingRate))) % load off portion to Noise_raw
% x_Noise_x = x_Noise_x + 1;
% Noise_raw(x_Noise_x) =
PD1_LPF(floor(i*period*samplingRate+j));

%
%
x_Noise(x_Noise_x) = floor(i*period*samplingRate+j);
end
```

```
% end
% end
%
%
% Noise =

interp1(x_Noise,Noise_raw(1:x_Noise_x),1:samplingRate*totalTime,'spline
'); % Noise interpolation
% PD_N = PD1 - Noise';
%
%
% averageRed_3_1 = zeros(No_RIR_Waves, 1);
% averageIR_3_1 = zeros(No_RIR_Waves, 1);


% for i=0:No_RIR_Waves-1 % Average data in each square wave period
% for j=1:floor((duty-transTime)*samplingRate)
% averageRed_3_1(i+1, 1) = averageRed_3_1(i+1, 1) +
PD_N(floor(i*period*samplingRate+j+offsetR*samplingRate+transTime*sampl
ingRate));
% averageIR_3_1(i+1, 1) = averageIR_3_1(i+1, 1) +
PD_N(floor(i*period*samplingRate+j+offsetIR*samplingRate+transTime*samp
lingRate));
% end
% averageRed_3_1(i+1, 1) = averageRed_3_1(i+1, 1)/(floor((dutytransTime)*
samplingRate));
% averageIR_3_1(i+1, 1) = averageIR_3_1(i+1, 1)/(floor((dutytransTime)*
samplingRate));
% end
% averageIR_3_1(end) = averageIR_3_1(end-1); % Abandon the last one of
IR_3 to eliminate error caused by interpolation
```

```
%% Create a Low-pass and Filter Waveforms
averageRed = averageRed_1; % _1,_2,_3,_4 correspond to single-sided
subtraction, double-sided subtraction, interpolation subtraction &
average of every 5 points

averageIR = zeros(length(averageRed_1),1);

order = 100;
cutoff = 10/(1/period);
y = fir1(order, cutoff,'low');
x = filtfilt(y, 1, averageRed);

z = filtfilt(y, 1, averageIR);
[dec,lib] = wavedec(averageRed,2,'db10');
a2 = wrcoef('a',dec,lib,'db10',2);

%Perfusion(ll+1) = mean(x);
%end
%% End of Loop

% % Pre-LPF for interpolation
% % order = 100;
% % cutoff1 = 40 /(1/period);
% % y1 = fir1(order, cutoff1,'low');
% % x1 = filtfilt(y1, 1, averageRed);
% % z1 = filtfilt(y1, 1, averageIR);
%
% % freqz(y) % view filter
% %
```

```
numavg = 100;
runavg = ones(1, numavg)/numavg;
x_avg = filtfilt(runavg, 1, averageRed);
z_avg = filtfilt(runavg, 1, averageIR);


% x = x - x_avg;
% z = z - z_avg;

time = (1:No_RIR_Waves)/(No_RIR_Waves)*totalTime;

%--------------------------------------%
% Red LED
%--------------------------------------%


figure;
subplot(2, 1, 1)
hold on;
plot(time, averageRed*1E3, '-k', 'linewidth', 2);
plot(time, x*1E3, '-r', 'linewidth', 2);
plot(time, x_avg*1E3, '-b', 'linewidth', 2);
hold off;
ylabel('Recived Signal [mV]', 'fontsize', 14, 'fontweight', 'bold')
xlabel('Time [s]', 'fontsize', 14, 'fontweight', 'bold')
set(gca,'linewidth', 2, 'fontsize', 10, 'fontweight', 'bold')
legend('Red LED', 'Red LED (LPF)', 'Running Average',
'Orientation','horizontal')
title('Red LED', 'fontsize', 14, 'fontweight', 'bold')
box on;
```

```
heart_beat_RED = x-x_avg;
wavelet_RED = a2-smooth(a2,200);

%heart_beat_RED = wavelet_RED;
% % Detect Heat Beat Peaks FAIL 202C VERSION
% temp = sign(diff(heart_beat_RED));
% % temp = sign(diff(x(order+numavg/2:end-numavg/2-1)));
% temp2 = (temp(1:end-1)-temp(2:end))./2;
% loc = find(temp2 ~= 0);
% loc = [loc(1); loc(find(diff(loc) > MIN_SAMP/2)+1)];
% peaks1 = loc(find(temp2(loc) > 0))+1;

% peaks1 = peaks1(find(heart_beat_RED(peaks1) > 0));
% valleys1 = loc(find(temp2(loc) < 0))+1;
% valleys1 = valleys1(find(heart_beat_RED(valleys1) < 0));

%peak detection that actually works:

peaks=[];
widthp=50;
for j = 1:totalTime/period

if heart_beat_RED(j)==max(heart_beat_RED(max(1,jwidthp):
min(totalTime/period,j+widthp)))
peaks(end+1)=j;

end
end
```

```
valleys=[];
widthv=50;
for j = 1:totalTime/period

if heart_beat_RED(j)==min(heart_beat_RED(max(1,jwidthv):
min(totalTime/period,j+widthv)))
valleys(end+1)=j;

end
end

diffzs=[];
widthd=25;
diff_hb = diff(heart_beat_RED);
for j = 1:totalTime/period-1

if abs(diff_hb(j))==min(abs(diff_hb(max(1,jwidthd):
min(totalTime/period-1,j+widthd))))
diffzs(end+1)=j;

end
end

killthese=[];
for j=1:numel(diffzs)
for k=1:numel(peaks)
if abs(diffzs(j)-peaks(k))<25
killthese(end+1)=j;
```

```
end
for k=1:numel(valleys)
if abs(diffzs(j)-valleys(k))<25
killthese(end+1)=j;

end
end
end

peakspacing(j) = min(abs(diffzs(j)-peaks));
valleyspacing(j) = min(abs(diffzs(j)-valleys));

end

diffzs(killthese)=[];
peakspacing(killthese)=[];

%clean up peaks/valleys to make them match 1:1

delp=[];

for i = 1:length(peaks)-1
valid=0;
for j = 1:length(valleys)

if peaks(i+1)>valleys(j) && peaks(i)<valleys(j)

valid=1;

break
```

```
end
end
if valid==0 && heart_beat_RED(peaks(i+1))<heart_beat_RED(peaks(i))

delp(end+1)=i+1;
elseif valid==0
delp(end+1)=i;

end
end

peaks(delp)=[];

delv=[];

for i = 1:length(valleys)-1
valid=0;
for j = 1:length(peaks)

if valleys(i+1)>peaks(j) && valleys(i)<peaks(j)
valid=1;

break

end
end
if valid==0 &&
heart_beat_RED(valleys(i+1))>heart_beat_RED(valleys(i))
```

```
delv(end+1)=i+1;
elseif valid==0
delv(end+1)=i;

end
end

valleys(delv)=[];

%finish of cleanup

mdiffzs = median(heart_beat_RED(diffzs));
mpeaks = median(heart_beat_RED(peaks));
mvalleys = median(heart_beat_RED(valleys));

secondpeak = (mdiffzs-mvalleys)/(mpeaks-mvalleys);
peakspacing = median(peakspacing);
valleyspacing = median(valleyspacing);

subplot(2, 1, 2)
hold on;
plot(time, heart_beat_RED*1E3, '-k', 'linewidth', 2);

% ylim([-1.5 1.5])

plot(time(peaks), heart_beat_RED(peaks)*1E3, 'or', 'linewidth', 2,
'markersize', 12);
plot(time(valleys), heart_beat_RED(valleys)*1E3, 'ob', 'linewidth', 2,
'markersize', 12);
plot(time(diffzs), heart_beat_RED(diffzs)*1E3, 'og', 'linewidth', 2,
```

```
'markersize', 12);
hold off;
ylabel('Heart Beat [mV]', 'fontsize', 14, 'fontweight', 'bold')
xlabel('Time [s]', 'fontsize', 14, 'fontweight', 'bold')
set(gca,'linewidth', 2, 'fontsize', 10, 'fontweight', 'bold')
box on;
Heart_Rate_RED = length(peaks)/(time(end)-time(1))*60;


% %--------------------------------------%
% % IR LED
% %--------------------------------------%


% figure;
% subplot(2, 1, 1)
% hold on;
% plot(time, averageIR*1E3, '-k', 'linewidth', 2);
% plot(time, z*1E3, '-r', 'linewidth', 2);
% plot(time, z_avg*1E3, '-b', 'linewidth', 2);
% hold off;
% ylabel('Recived Signal [mV]', 'fontsize', 14, 'fontweight', 'bold')
% xlabel('Time [s]', 'fontsize', 14, 'fontweight', 'bold')
% set(gca,'linewidth', 2, 'fontsize', 10, 'fontweight', 'bold')
% legend('IR LED','IR LED (LPF)','Running Average',
'Orientation','horizontal')
% title('IR LED', 'fontsize', 14, 'fontweight', 'bold')
% box on;
%
% heart_beat_IR = z-z_avg;
%
% % Detect Heat Beat Peaks
% temp = sign(diff(heart_beat_IR));
```

```
% % temp = sign(diff(z(order+numavg/2:end-numavg/2-1)));
% temp2 = (temp(1:end-1)-temp(2:end))./2;
% loc = find(temp2 ~= 0);
% loc = [loc(1); loc(find(diff(loc) > MIN_SAMP/2)+1)];
% peaks2 = loc(find(temp2(loc) > 0))+1;
% peaks2 = peaks2(find(heart_beat_IR(peaks2) > 0));
% valleys2 = loc(find(temp2(loc) < 0))+1;
% valleys2 = valleys2(find(heart_beat_IR(valleys2) < 0));
%
% subplot(2, 1, 2)
% hold on;
% plot(time, heart_beat_IR*1E3, '-k', 'linewidth', 2);
% ylim([-1.5 1.5]);
% plot(time(peaks2), heart_beat_IR(peaks2)*1E3, 'or', 'linewidth', 2,
'markersize', 12);
% plot(time(valleys2), heart_beat_IR(valleys2)*1E3, 'ob', 'linewidth',
2, 'markersize', 12);
% hold off;
% ylabel('Heart Beat [mV]', 'fontsize', 14, 'fontweight', 'bold')
% xlabel('Time [s]', 'fontsize', 14, 'fontweight', 'bold')
% set(gca,'linewidth', 2, 'fontsize', 10, 'fontweight', 'bold')
% box on;
% Heart_Rate_IR = length(peaks2)/(time(end)-time(1))*60


% %--------------------------------------%
% % SpO2
% %--------------------------------------%
% H_heart_beat_Red_peak =
interp1(peaks1,x(peaks1),1:length(time),'spline'); % Interpolate the
peak value of heart beat (RED) for whole time range
% H_heart_beat_IR_peak =
```

```
interp1(peaks2,z(peaks2),1:length(time),'spline'); % Interpolate the
peak value of heart beat (IR) for whole time range
%

% H_heart_beat_Red_valley =
interp1(valleys1,x(valleys1),1:length(time),'spline'); % Interpolate
the valley value of heart beat (RED) for whole time range
% H_heart_beat_IR_valley =
interp1(valleys2,z(valleys2),1:length(time),'spline'); % Interpolate
the valley value of heart beat (IR) for whole time range
%
% % Superposition
% x2 = zeros(length(x1),1);
% z2 = zeros(length(z1),1);
% for i=2:length(peaks1)-1
% x2(1:end-(peaks1(i)-peaks1(2))) = x2(1:end-(peaks1(i)-peaks1(2)))

+ x1(peaks1(i)-peaks1(2)+1:end);
% z2(1:end-(peaks2(i)-peaks2(2))) = z2(1:end-(peaks2(i)-peaks2(2)))
+ z1(peaks2(i)-peaks2(2)+1:end);
% end
% x2 = x2/(length(peaks1)-2);
% z2 = z2/(length(peaks2)-2);
%
% % H_heart_beat_Red = filtfilt(runavg, 1, H_heart_beat_Red);
% % H_heart_beat_IR = filtfilt(runavg, 1, H_heart_beat_IR);
%
%
% R_red = H_heart_beat_Red_valley./(H_heart_beat_Red_peak);
% R_IR = H_heart_beat_IR_valley./(H_heart_beat_IR_peak);
%
```

```
% R = (log(R_red)./log(R_IR))*(RED_sens/IR_sens);
% O2 = (0.81-0.18.*R)./(0.63+0.11.*R)*100;
% SpO2 = mean(O2)
%
% figure;
% hold on;
% plot(time, O2, '-r', 'linewidth', 2);
% ylabel('SpO2', 'fontsize', 14, 'fontweight', 'bold')
% xlabel('Time [s]', 'fontsize', 14, 'fontweight', 'bold')
% set(gca,'linewidth', 2, 'fontsize', 10, 'fontweight', 'bold')
% ylim([90 110])
% box on;

x=[];
hrdata=[];
pdiff=[];
secpeak=[];


trial=1;
for trial = 1:1
for filenum = 1:1


for sensorselect=4
inputfile = ['ir+' num2str(min(trial,2)) '.' num2str(filenum)];
inputfile = 'all+';


%inputfile = ['height\5s_stoy' num2str(filenum)];

multilevel_extract;
hrdata(:,filenum) = heart_beat_RED;
dcdata(filenum) = median(x_avg);
```

```
%if nnz(x(:,filenum))==0; break; end

r(filenum) = Heart_Rate_RED;

 vs=min(numel(peaks),numel(valleys));
p2pdata(filenum) = median(heart_beat_RED(peaks(1:vs))heart_
beat_RED(valleys(1:vs)));

en=[];
for i=2:numel(valleys)-2
en(end+1) = sum(heart_beat_RED(valleys(i):valleys(i+1)).^2);

end

benergy(filenum)=median(en);

riset=[];
fallt=[];
if peaks(1)>valleys(1)

for i=1:vs-1
riset(end+1) = peaks(i)-valleys(i);
fallt(end+1) = valleys(i+1)-peaks(i);

end
else

for i=1:vs-1
riset(end+1) = peaks(i+1)-valleys(i);
```

```
fallt(end+1) = valleys(i)-peaks(i);


end
end


risetime(filenum)=median(riset);
falltime(filenum)=median(fallt);

for repeat=1:3
if peaks(1)<valleys(1); peaks(1)=[]; end
end

for i=1:floor(numel(peaks)/2)
list_pdiff(i) = heart_beat_RED(peaks(2*i-1))heart_
beat_RED(peaks(2*i));

end

 pdiff(filenum)=median(list_pdiff);

secpeak(filenum)=secondpeak;
peakspace(filenum)=peakspacing;
valspace(filenum)=valleyspacing;

medpeak(filenum) = mpeaks-mvalleys;

end

% suffix = '.pressure';
% presf=csvread([inputfile suffix]);
```

```
% presdata(filenum)=mean((presf(:,2)-.6)/2.8);


end

stoyrt(trial,:)=risetime*.005;
stoyft(trial,:)=falltime*.005;
stoyhr(trial,:)=r;
stoysecpeak(trial,:)=secpeak;
stoypeakspace(trial,:)=peakspace*.005;
stoyvalspace(trial,:)=valspace*.005;
stoymp(trial,:)=medpeak;


end

% stoyfts=stoyft./(min(stoyft')'*[1 1 1 1 1]);
% stoyrts=stoyrt./(min(stoyrt')'*[1 1 1 1 1]);
% stoysecpeaks=stoysecpeak./(min(stoysecpeak')'*[1 1 1 1 1]);
% stoymps=stoymp./(min(stoymp')'*[1 1 1 1 1]);


%
%
% for i=1:3;corrcoef(stoyhr(i,:),stoyrt(i,:))
% end
% for i=1:3;corrcoef(stoyhr(i,:),stoyft(i,:))
% end
% for i=1:3;corrcoef(stoyhr(i,:),stoysecpeak(i,:))
% end
%
% for i=1:3;corrcoef(stoybps(i,:),stoyrt(i,:))
% end
```

```
% for i=1:3;corrcoef(stoybps(i,:),stoyft(i,:))
% end
% for i=1:3;corrcoef(stoybps(i,:),stoysecpeak(i,:))
% end
% for i=1:3;corrcoef(stoybps(i,:),stoyhr(i,:))
% end
%
% for i=1:3;corrcoef(stoybpd(i,:),stoyrt(i,:))
% end
% for i=1:3;corrcoef(stoybpd(i,:),stoyft(i,:))
% end
% for i=1:3;corrcoef(stoybpd(i,:),stoysecpeak(i,:))
% end
% for i=1:3;corrcoef(stoybpd(i,:),stoyhr(i,:))
% end


%
% peaks=[];
% for j = 1:4000
% if x(j,filenum)>5e-5 && x(j,filenum)==max(x(max(1,j75):
min(4000,j+75),filenum))
% peaks(end+1)=j;


% end
% end
%
%
%
% for j = 1:4000
% if heart_beat_RED(j)>5e-5 &&
heart_beat_RED(j)==max(heart_beat_RED(max(1,j-75):min(4000,j+75)))
```

```
% peaks(end+1)=j;
% end
% end


% t= 1:4
% figure
% plot(t,stoy1bpd,'o',t,stoy2bpd,'o',t,stoy3bpd,'o')
% axis([.5 4.5 -1 1])
% set(gca,'XTick',1:4)
% set(gca,'XTickLabel',{'Rise Time' 'Fall Time' 'Second Peak Strength'
'Heart Rate'})
% legend({'Trial 1' 'Trial 2' 'Trial 3'})
% title('Correlations: Metrics vs. Diastolic Blood Pressure, Henrik')
% ylabel('Correlation Coefficient')
% figure
% plot(t,stoy1bps,'o',t,stoy2bps,'o',t,stoy3bps,'o')
% axis([.5 4.5 -1 1])
% set(gca,'XTick',1:4)
% set(gca,'XTickLabel',{'Rise Time' 'Fall Time' 'Second Peak Strength'
'Heart Rate'})
% legend({'Trial 1' 'Trial 2' 'Trial 3'})
% title('Correlations: Metrics vs. Systolic Blood Pressure, Henrik')
% ylabel('Correlation Coefficient')
% figure
% plot(t,stoy1hr,'o',t,stoy2hr,'o',t,stoy3hr,'o')
% axis([.5 4.5 -1 1])
% set(gca,'XTick',1:4)
% set(gca,'XTickLabel',{'Rise Time' 'Fall Time' 'Second Peak Strength'
'Heart Rate'})
% legend({'Trial 1' 'Trial 2' 'Trial 3'})
% title('Correlations: Metrics vs. Heart Rate, Henrik')
```

```
% ylabel('Correlation Coefficient')


function [ pointcoords ] = rgbfind( filename )
im_unfiltered = imread(filename); %[y x rgb]


%h = fspecial('gaussian',10,10);
%im=imfilter(im_unfiltered,h);

im=im_unfiltered;

r = im(:,:,1);
g = im(:,:,2);
b = im(:,:,3);

% image(im);
%goal rgb = 0,160,170


goalr = 0;
goalg = 160;
goalb = 170;
tol=50; %goal offset tolerance
match=zeros(size(im,1),size(im,2),2);
for y = 1:size(im,1)

for x = 1:size(im,2)

if (r(y,x)>goalr+tol) || (r(y,x)<goalr-tol) ...
|| (g(y,x)>goalg+tol) || (g(y,x)<goalg-tol) ...
|| (b(y,x)>goalb+tol) || (b(y,x)<goalb-tol)
```

```
%not a match
%match(y,x,:)=[0,0,0];

else

%match

 match(y,x,:)=[1,0];

end
end
end

numblobs=0;
blob=[];
for y = 1:size(im,1)

for x = 1:size(im,2)

if match(y,x,1)==1

%these matches are already in blobs

if match(y-1,x+2,1)==1
match(y,x,2)=match(y-1,x+2,2);
blob(match(y-1,x+2,2)).x(end+1)=x;
blob(match(y-1,x+2,2)).y(end+1)=y;

elseif match(y-1,x+1,1)==1
```

```
match(y,x,2)=match(y-1,x+1,2);
blob(match(y-1,x+1,2)).x(end+1)=x;
blob(match(y-1,x+1,2)).y(end+1)=y;

elseif match(y-1,x,1)==1
match(y,x,2)=match(y-1,x,2);
blob(match(y-1,x,2)).x(end+1)=x;
blob(match(y-1,x,2)).y(end+1)=y;

elseif match(y-1,x-1,1)==1
match(y,x,2)=match(y-1,x-1,2);
blob(match(y-1,x-1,2)).x(end+1)=x;
blob(match(y-1,x-1,2)).y(end+1)=y;

elseif match(y,x-1,1)==1
match(y,x,2)=match(y,x-1,2);
blob(match(y,x-1,2)).x(end+1)=x;
blob(match(y,x-1,2)).y(end+1)=y;

%other matches require new blob
else%if match(y+1,x-1,1)==1
numblobs = numblobs+1;
match(y,x,2)=numblobs;
blob(numblobs).x=x;
blob(numblobs).y=y;

end

end
```

```
end
end

merged=zeros(1,numblobs);

figure();image(match(:,:,2)+1);

for y = size(im,1):-1:1
for x = size(im,2):-1:1

if match(y,x,1)==1

%these matches are already in blobs

if (match(y,x+1,1)==1) && (match(y,x,2)~=match(y,x+1,2))
merged(match(y,x,2))=match(y,x+1,2);
match(y,x,2)=match(y,x+1,2);
blob(match(y,x+1,2)).x(end+1)=x;
blob(match(y,x+1,2)).y(end+1)=y;

elseif match(y+1,x+1,1)==1 && match(y,x,2)~=match(y+1,x+1,2)
merged(match(y,x,2))=match(y+1,x+1,2);
match(y,x,2)=match(y+1,x+1,2);
blob(match(y+1,x+1,2)).x(end+1)=x;
blob(match(y+1,x+1,2)).y(end+1)=y;

elseif match(y+1,x,1)==1 && match(y,x,2)~=match(y+1,x,2)
merged(match(y,x,2))=match(y+1,x,2);
match(y,x,2)=match(y+1,x,2);
blob(match(y+1,x,2)).x(end+1)=x;
```

```
blob(match(y+1,x,2)).y(end+1)=y;

elseif match(y+1,x-1,1)==1 && match(y,x,2)~=match(y+1,x-1,2)
merged(match(y,x,2))=match(y+1,x-1,2);
match(y,x,2)=match(y+1,x-1,2);
blob(match(y+1,x-1,2)).x(end+1)=x;
blob(match(y+1,x-1,2)).y(end+1)=y;

end

end

end

end

for y = size(im,1):-1:1
for x = size(im,2):-1:1
if match(y,x,1)==1
if merged(match(y,x,2))>0

while merged(match(y,x,2))>0
match(y,x,2)=merged(match(y,x,2));
blob(match(y,x,2)).x(end+1)=x;
blob(match(y,x,2)).y(end+1)=y;

end;end;end;end;end

blob(find(merged))=[];
pointcoords=[];
for i=1:size(blob,2)
```

```
pointcoords(i,:)=[mean(blob(i).y);mean(blob(i).x)];

end

pointcoords=round(pointcoords);

figure();imshow(match(:,:,1));
figure();image(match(:,:,2)+1);

%+(match(:,:,2)>0)*3

end

function [ exppic ] = imoverlay( pcs, im, impic )
p1 = pcs(1,:);
p2 = pcs(2,:);
p3 = pcs(3,:);

d1=p1(1)-p1(2);
d2=p2(1)-p2(2);
d3=p3(1)-p3(2);

s1=p1(1)+p1(2);
s2=p2(1)+p2(2);
s3=p3(1)+p3(2);

[a,v] = max([d1 d2 d3]);
[a,t] = min([s1 s2 s3]);
[a,r] = max([s1 s2 s3]);
```

```
%hyp = sqrt( (pcs(v,1)-pcs(t,1))^2 + (pcs(v,2)-pcs(t,2))^2 );
%adj = sqrt( (pcs(v,1)-pcs(r,1))^2 + (pcs(v,2)-pcs(r,2))^2 );
%angle=atand(adj/hyp);


ratio = (pcs(v,1)-pcs(t,1)) / (pcs(t,2)-pcs(v,2));
angle=atand(ratio);
hangle = -1*(90 - angle);
hoffset=( pcs(r,1)-pcs(t,1)- (pcs(t,2)-pcs(r,2))*tand(angle) ) *
cosd(angle);
scale=hoffset/size(im,2);


imout = imresize(im,scale);
padout = ones(size(imout));
padout = imrotate(padout,hangle);
imout = imrotate(imout,hangle);


sp=[0 0];
if hangle<0
for x=1:size(padout,2)
for y=size(padout,1):-1:1
if padout(y,x)==1
sp = [y x];


break


end
end
if sp; break; end
```

```
end
else
for y=size(padout,1):-1:1
for x=1:size(padout,2)
if padout(y,x)==1
sp = [y x];

break

end
end
if sp; break; end

end
end

offy = pcs(v,1)-sp(1);

offx = pcs(v,2)-sp(2);

exp = zeros(size(impic));
exppic = exp;
for y=1:size(padout,1)

for x=1:size(padout,2)
xcoord = max(1,offx+x);
xcoord = min(xcoord,size(exp,2));
ycoord = max(1,offy+y);
ycoord = min(ycoord,size(exp,1));
```

```
exp(ycoord,xcoord,:)=padout(y,x,:);
exppic(ycoord,xcoord,:)=imout(y,x,:);


end
end


image(impic);
hold on
hobject = image(exppic/255);
hold off
set(hobject, 'AlphaData',exp(:,:,1)/2);


end

function [ imdata ] = mapData( filename, ploten )

%MAPDATA Summary of this function goes here
% Detailed explanation goes here


temp = csvread(filename);
log_spO2 = temp(1,:);
log_pressure = temp(2,:);
log_x = temp(3,:);
log_y = temp(4,:);
clear temp;
vals = [];


log_x = abs(min(log_x))+log_x;
log_y = abs(min(log_y))+log_y;
i=0;
```

```
while i<numel(log_spO2)

i=i+1;

if log_spO2(i)<10
log_spO2(i)=[];
log_pressure(i)=[];
log_x(i)=[];
log_y(i)=[];

end
end

% for i=1:size(log_spO2,2)

grid = zeros( floor((max(log_y))/5)+1 , floor((max(log_x))/5)+1 );
[X, Y] = meshgrid(1:5:(max(log_x)),1:5:(max(log_y)));
while numel(log_spO2)>0

i=1;
xmatch = find(log_x==log_x(i));
ymatch = find(log_y==log_y(i));
match = intersect(xmatch,ymatch);

 vals(end+1,:) = [log_x(i) log_y(i) max(log_spO2(match))];

% grid(log_y(i)+1,log_x(i)+1) = max(log_spO2(match));

log_spO2(match)=[];
log_pressure(match)=[];
```

```
log_x(match)=[];
log_y(match)=[];


end

%plot(sqrt(vals(:,1).^2 + vals(:,2).^2),vals(:,3));
anisotropy = 1; %range x / range y
alpha = 0; %angle between axis/anisotropy in degrees
nu = 1; %nu for covariance
vgrid = [5 5];
[kout evar] = vebyk(vals,vgrid,5,anisotropy,alpha,nu,1,0,0);
for i=1:size(kout,1)

if (size(grid,2)-1 < kout(i,1)/5) || (size(grid,1)-1 <
kout(i,2)/5)
continue;
end

 grid(kout(i,2)/5+1,kout(i,1)/5+1)=kout(i,3);

end

%image(grid);

imdata=[];
if ploten

figure();
surf(X,Y,grid);
```

```
else

 imdat = ( (grid-min(min(grid))) *255/(max(max(grid))
5

min(min(grid))) );
rgbdata = ind2rgb(round(imdat),jet(256));
imwrite(rgbdata,'d_image.jpg','jpg')
10 imdata=rgbdata;


end
end
15
```

CLAIMS

What is claimed is:

1. An apparatus for monitoring perfusion oxygenation of a target tissue region of a patient, comprising:

a scanner comprising:

a planar sensor array;

the sensor array configured to be positioned in contact with a surface of the target tissue region;

the sensor array comprising one or more LED's configured to emit light into the target tissue region at a wavelength keyed for hemoglobin;

the sensor array comprising one or more photodiodes configured to detect light reflected from the LED's; and

a data acquisition controller coupled to the one or more LED's and to the one or more photodiodes for controlling the emission and reception of light from the sensor array to obtain perfusion oxygenation data associated with the target tissue region.

2. An apparatus as recited in claim 1, the scanner further comprising:

a pressure sensor coupled to the sensor array;

the pressure sensor configured to obtain pressure readings of the sensor array's contact with a surface of the target tissue region;

wherein the scanner is configured to obtain pressure sensor readings while obtaining perfusion oxygenation data to ensure proper contact of the scanner with the surface of the target tissue region.

3. An apparatus as recited in claim 2:

wherein the pressure sensors and sensor array are connected to a first side of a printed circuit board (PCB); and

wherein the data acquisition controller is connected to the PCB on a second side opposite said first side.

4. An apparatus as recited in claim 1, wherein each LED comprises dual emitters configured for emitting red (660nm) and infrared (880nm) light.

5. An apparatus as recited in claim 4:

wherein the one or more of the LED's are coupled driver circuit; and

wherein the driver circuit is configured to allow the red LED emitter and infrared LED emitter to be driven independently while sharing a common anode.

6. An apparatus as recited in claim 5, wherein the driver circuit comprises an amplifier; and

a field-effect transistor configured for providing negative feedback.

7. An apparatus as recited in claim 2, further comprising:

a processing module coupled to the data acquisition controller;

the processing module configured to control sampling of the pressure sensor and sensor array for simultaneous acquisition of pressure sensor data and perfusion oxygenation data.

8. An apparatus as recited in claim 7, wherein the processing module is configured to obtain readings from the sensor array to obtain position data of the scanner.

9. An apparatus as recited in claim 8, wherein the processing module is configured to generate a perfusion oxygenation map of the target tissue.

10. An apparatus as recited in claim 8, wherein the processing module is configured to control sampling of the pressure sensor and sensor array for simultaneous acquisition of two or more data parameters selected from the group consisting of pressure sensor data, perfusion oxygenation data, and position data, to simultaneously display said two or more data parameters.

11. A system for monitoring perfusion oxygenation of a target tissue region of a patient, comprising:

(a) a scanner comprising:

a planar sensor array;

the sensor array configured to be positioned in contact with a surface of the target tissue region;

the sensor array comprising one or more light sources configured to emit light into the target tissue region at a wavelength keyed for hemoglobin;

the sensor array comprising one or more sensors configured to detect light reflected from the light sources;

a pressure sensor coupled to the sensor array;

the pressure sensor configured to obtain pressure readings of the sensor array's contact with a surface of the target tissue region; and

(b) a data acquisition controller coupled to the one or more sensors and for controlling the emission and reception of light from the sensor array to obtain perfusion oxygenation data associated with the target tissue; and

(c) a processing module coupled to the data acquisition controller;

(d) the processing module configured to control sampling of the pressure sensor and sensor array for simultaneous acquisition of perfusion oxygenation data and pressure sensor data to ensure proper contact of the scanner with the surface of the target tissue region.

12. A system as recited in claim 11:

wherein the sensor array comprises one or more LED's configured to emit light into the target tissue region at a wavelength keyed for hemoglobin; and

wherein the sensor array comprises one or more photodiodes configured to detect light reflected from the LED's.

13. A system as recited in claim 12:

wherein each of the one or more LED's comprises dual emitters configured for emitting red (660nm) and infrared (880nm) light;

wherein the one or more LED's are coupled to the driver circuit; and

wherein the driver circuit is configured to allow the red LED emitter and the infrared LED emitter to be driven independently while sharing a common anode

14. A system as recited in claim 11, further comprising:

a graphical user interface;

wherein the graphical user interface is configured to display the perfusion oxygenation data and pressure sensor data.

15. A system as recited in claim 14, the processing module is further configured to obtain readings from the sensor array to obtain position data of the scanner.

16. A system as recited in claim 15, wherein the processing module is further configured to interpolate the position data to generate a perfusion oxygenation map of the target tissue.

17. A system as recited in claim 16, wherein the processing module is configured to control sampling of the pressure sensor and sensor array for simultaneous acquisition of two or more data parameters selected from the group consisting of pressure sensor data, perfusion oxygenation data, and position data, to simultaneously display the two or more data parameters.

18. A system as recited in claim 16, wherein the processing module is configured to receive an image of the target tissue, and overlay the perfusion oxygenation map over the image.

19. A system as recited in claim 14, wherein the graphical user interface is configured to allow user input to manipulate settings of the sensor array and pressure sensor.

20. A system as recited in claim 11, wherein the processing module further comprises:

a filtering module;

the filtering module configure to filter in-band noise by subtracting data recorded when the one or more light sources are in an "off" state from data recorded when the one or more light sources are in an "on" state.

21. A method for performing real-time monitoring of perfusion oxygenation of a target tissue region of a patient, comprising:

positioning a sensor array in contact with a surface of the target tissue region;

emitting light from lights sources in the sensor array into the target tissue region at a wavelength keyed for hemoglobin;

receiving light reflected from the light sources;

obtaining pressure data associated with the sensor array's contact with a surface of the target tissue region;

obtaining perfusion oxygenation data associated with the target tissue region; and

sampling the perfusion oxygenation data and pressure data to ensure proper contact of the sensor array with the surface of the target tissue region.

22. A method as recited in claim 21:

wherein the sensor array comprises one or more LED's configured to emit light into the target tissue region at a wavelength keyed for hemoglobin; and

wherein the sensor array comprises one or more photodiodes configured to detect light reflected from the LED's.

23. A method as recited in claim 22:

wherein each of the one or more LED's comprises dual emitters configured for emitting red (660nm) and infrared (880nm) light;

the method further comprising independently driving the red LED emitter and infrared LED emitter while the red LED emitter and infrared LED emitter share a

common anode

24. A method as recited in claim 21, further comprising:

simultaneously displaying the perfusion oxygenation data and pressure sensor data.

25. A method as recited in claim 21, further comprising:

acquiring readings from the sensor array to obtain position data of the scanner.

26. A method as recited in claim 25, further comprising:

interpolating the position data to generate a perfusion oxygenation map of the target tissue.

27. A method as recited in claim 26, wherein interpolating the position data comprises applying a Kriging algorithm to the acquired position data.

28. A method as recited in claim 26, further comprising:

sampling of the pressure sensor and sensor array for simultaneous acquisition of pressure sensor data, perfusion oxygenation data, and position data; and

simultaneously displaying the pressure sensor data, perfusion oxygenation data, and position data.

29. A method as recited in claim 26, further comprising:

receiving an image of the target tissue; and

overlaying the perfusion oxygenation map over the image.

30. A method as recited in claim 21, further comprising:

providing a graphical user interface to allow user input; and

manipulating sampling settings of the sensor array and pressure sensor according to said user input.

31. A method as recited in claim 21, further comprising:

cycling the one or more light sources between a period when the one or more light sources are on, and a period when the one or more light sources are off; and

filtering in-band noise by subtracting data recorded from when the one or more light sources are in an "off" state from data from when the one or more light sources are in an "on" state.