



(12)发明专利申请

(10)申请公布号 CN 105877764 A

(43)申请公布日 2016.08.24

(21)申请号 201610192712.7

(51)Int.Cl.

(22)申请日 2012.01.19

A61B 5/1455(2006.01)

(30)优先权数据

A61B 5/026(2006.01)

61/434,014 2011.01.19 US

A61B 5/00(2006.01)

(62)分案原申请数据

201280005865.6 2012.01.19

(71)申请人 加利福尼亚大学董事会

地址 美国加利福尼亚州

(72)发明人 马吉德·萨拉夫扎德 威廉·凯泽

芭芭拉·贝茨-詹森

阿里礼萨·梅尔尼亚

比詹·马帕尔 弗兰克·王

(74)专利代理机构 北京天昊联合知识产权代理

有限公司 11112

代理人 麦善勇 张天舒

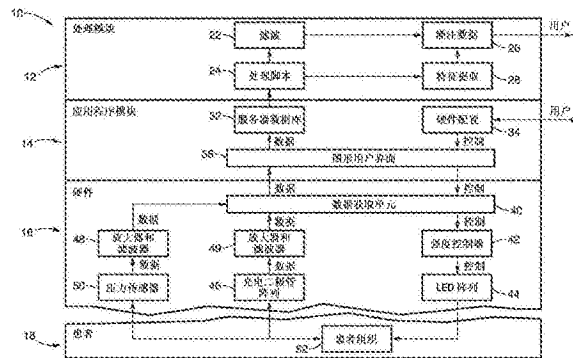
权利要求书3页 说明书35页 附图25页

(54)发明名称

组织的血氧定量测定和灌注成像的设备、系统及方法

(57)摘要

公开了一种表征组织的健康状态的小型灌注扫描器和方法,其将压力传感元件与光学传感器结合来监测施加到靶组织上的压力水平用以进行精确的皮肤/组织的血液灌注测量和血氧定量测定。该系统和方法能够进行灌注成像和灌注映像(几何和时间)、信号处理和模式识别、灌注数据的噪声消除和数据融合、扫描器的位置和压力读取。



1. 一种用于检测患者的靶组织区域的溃疡形成的设备,包括:
扫描器,包括:
传感器阵列,该传感器阵列配置为被定位成与所述靶组织区域的表面接触,该传感器阵列包括:
一个或多个光源,其配置为以针对血红蛋白的波长向所述靶组织区域发射光;以及
一个或多个光电二极管,其配置为检测从所述靶组织区反射的光;以及
数据获取控制器,其耦合到所述一个或多个光源和所述一个或多个光电二极管,用于控制来自所述传感器阵列的光的发射和接收以获得与所述靶组织区域的组织健康相关的灌注氧合数据;以及
处理模块,其与所述数据获取控制器耦合,其中所述处理模块被配置为基于所述灌注氧合数据和位置读数产生所述靶组织的灌注氧合图,
其中,所述灌注氧合图提供所述靶组织区域的溃疡形成的指标。
2. 根据权利要求1所述的设备,其中,所述灌注氧合图提供所述靶组织区域的褥疮形成的指标。
3. 根据权利要求1所述的设备,其中,所述灌注氧合图提供所述靶组织区域的静脉溃疡形成的指标。
4. 根据权利要求1所述的设备,其中,所述灌注氧合图提供所述靶组织区域的糖尿病足部溃疡形成的指标。
5. 根据权利要求1-4中任一权利要求所述的设备,还包括:
耦合到所述传感器阵列的压力传感器;以及
所述压力传感器配置为获得所述传感器阵列与所述靶组织区域的表面接触的压力读数,
其中,所述扫描器配置为在获得灌注氧合数据的同时获得压力传感器读数,以确保所述传感器阵列与靶组织区域的表面的适当接触。
6. 根据权利要求5所述的设备,
其中,所述压力传感器和所述传感器阵列被连接到印刷电路板(PCB)的第一侧;以及
其中,所述数据获取控制器在与所述第一侧相反的第二侧上连接至PCB。
7. 根据权利要求1-4中任一权利要求所述的设备,
其中,所述一个或多个光源包括一个或多个发光二极管(LED);以及
其中,所述数据获取控制器包括强度控制器,用于控制所述一个或多个LED的输出。
8. 根据权利要求7所述的设备,
其中,所述一个或多个LED的每一个包括配置用于发射红光和红外光的双发射器;以及
其中,所述双发射器共享一个公共阳极。
9. 根据权利要求8所述的设备,其中,所述强度控制器包括LED驱动器电路,该LED驱动器电路被配置为允许所述双发射器的红光LED发射器和红外光LED发射器被独立驱动。
10. 根据权利要求9所述的设备,其中,所述驱动器电路包括:
放大器;以及
场效应晶体管,其被配置为提供负反馈。
11. 根据权利要求5所述的设备,其中,所述处理模块进一步被配置为控制所述压力传

传感器和所述传感器阵列的采样,以同时获取压力传感器数据和灌注氧合数据。

12. 根据权利要求5所述的设备,其中,所述处理模块进一步被配置成控制所述压力传感器和所述传感器阵列的采样,以同时获取选自包括压力传感器数据、灌注氧合数据和位置数据的组的两个或两个以上的数据参数。

13. 根据权利要求12所述的设备,还包括图形用户界面,该图形用户界面被配置为同时显示所述两个或两个以上的数据参数。

14. 根据权利要求5所述的设备,还包括图形用户界面,该图形用户界面被配置为显示所述灌注氧合数据和所述压力传感器数据。

15. 根据权利要求14所述的设备,其中,所述图形用户界面被配置为允许用户输入以操作传感器阵列和压力传感器的设置。

16. 根据权利要求1-4中任一权利要求所述的设备,其中,所述处理模块被配置为接收靶组织的图像,并在该图像上叠加灌注氧合图。

17. 根据权利要求1-4中任一权利要求所述的设备,其中,所述处理模块还包括滤波模块,该滤波模块配置为通过将当所述一个或多个光源处于关闭状态时记录的数据从当所述一个或多个光源处于开启状态时记录的数据中减去除来滤除带内噪声。

18. 一种检测患者的靶组织区域的溃疡形成的方法,包括:

将与所述靶组织区域的表面接触的传感器阵列定位;

以针对血红蛋白的波长将来自所述传感器阵列中的光源的光发射到所述靶组织区域;

接收从所述靶组织区域反射的光;

获得所述传感器阵列与所述靶组织区域的表面接触相关的压力数据;

获得与所述靶组织区域的组织健康相关的灌注氧合数据;

获取来自所述传感器阵列的读数以获得所述扫描器的位置数据;

对所述位置数据进行内插,以产生所述靶组织的灌注氧合图,其中所述灌注氧合图提供所述靶组织区域的溃疡形成的指标;以及

对所述灌注氧合数据和压力数据进行采样,以确保所述传感器阵列与所述靶组织区域的表面的适当接触。

19. 根据权利要求18所述的方法,其中,所述灌注氧合图提供所述靶组织区域的褥疮形成的指标。

20. 根据权利要求18所述的方法,其中,所述灌注氧合图提供所述靶组织区域的静脉溃疡形成的指标。

21. 根据权利要求18所述的方法,其中,所述灌注氧合图提供所述靶组织区域的糖尿病足部溃疡形成的指标。

22. 根据权利要求18-21中任一权利要求所述的方法,

其中,所述传感器阵列中的所述光源包括配置为以针对血红蛋白的波长向所述靶组织区域发射光的一个或多个发光二极管(LED);以及

其中,所述传感器阵列包括被配置为检测从所述靶组织区域反射的光的一个或多个光电二极管。

23. 根据权利要求22所述的方法,

其中,所述一个或多个LED中的每一个LED包括被配置为发射红光和红外光的双发射

器；

其中，所述双发射器共用一个公共阳极；以及

其中，该方法进一步包括独立地驱动所述双发射器的所述红光LED发射器和所述红外光LED发射器。

24. 根据权利要求18-21中任一权利要求所述的方法，还包括同时显示所述灌注氧合数据和所述压力数据。

25. 根据权利要求18-21中任一权利要求所述的方法，其中，对位置数据进行内插的步骤包括向所获取的位置数据应用Kriging算法。

26. 根据权利要求18-21中任一权利要求所述的方法，还包括：

同时获取所述压力数据、所述灌注氧合数据、和所述位置数据；以及

同时显示所述压力数据、所述灌注氧合数据、和所述位置数据。

27. 根据权利要求18-21中任一权利要求所述的方法，还包括：

获得所述靶组织的图像；以及

将所述灌注氧合图叠加在所述图像上。

28. 根据权利要求18-21中任一权利要求所述的方法，还包括：

使所述一个或多个光源在所述一个或多个光源开启的周期和所述一个或多个光源关闭的周期之间循环；以及

通过将在所述一个或多个光源处于关闭状态时记录的数据从在所述一个或多个光源处于开启状态时记录的数据中减去除带内噪声。

组织的血氧定量测定和灌注成像的设备、系统及方法

[0001] 本申请是申请号为201280005865.6、申请日为2012年1月19日、发明名称为“组织的血氧定量测定和灌注成像的设备、系统及方法”的专利申请的分案申请。

[0002] 相关申请的交叉引用

[0003] 本申请要求于2011年1月19日提交的序列号为61/434,014的美国临时专利申请的优先权,其全部内容以引文方式并入本申请。

[0004] 有关联邦政府资助的研究或开发的声明

[0005] 不适用

[0006] 通过引用将光盘上所提交的资料结合到本申请中

[0007] 不适用

[0008] 材料受版权保护的通知

[0009] 本专利文件中的一部分材料受美国和其他国家的版权法的版权保护。该版权的所有者不反对任何人对该专利文件或专利公开进行拓制,因为其在美国专利商标局中是能够公开获得的文件或记录,除此之外保留其他任何形式的版权。因此,该版权所有者不会放弃其任何使得该专利文件处于保密状态的权利,其权利包括但不限于根据37C.F.R. §1.14的权利。

技术领域

[0010] 本发明总的来讲涉及组织的血氧定量测定,更具体地讲,涉及组织的血氧定量测定和灌注成像。

背景技术

[0011] 病人皮肤的完整性长期以来一直是护士和疗养院关注的问题。维护皮肤的完整性已由美国护士协会(American Nurses Association)确定为优良护理的一个重要指标。同时,溃疡,特别是静脉溃疡和褥疮,仍然是主要的健康问题,特别是对住院的老年人而言。检测早期伤口形成是一个极具挑战性和昂贵的问题。

[0012] 当年龄与其他危险因素一起被考虑时,这些溃疡的发生率显著增加。住院患者褥疮的总体发病率范围从2.7%至29.5%,已经有记载说重症监护室的患者的比率大于50%。在对于选定的诊断急性护理医院出院的1803个老年人的多中心队列的回顾性研究中,13.2%(即,164例患者)表现出第一阶段溃疡的发病率。在这164名患者中,38例(16%)的溃疡发展到更高阶段。

[0013] 褥疮另外还与出院后一年内死亡风险的增加相关。治疗褥疮的估计成本为每个溃疡需要\$5,000至\$40,000不等,视病情轻重决定。同样,对于住院治疗的患者而言,尤其是中老年人,静脉溃疡也可引起严重的健康问题。多达3%的人口患有腿部溃疡,而这一数字在80岁以上的人群中上升到20%。治疗静脉性溃疡的平均成本估计为\$10,000,并且在没有进行有效的治疗和早期诊断时可很容易上升为高达\$20,000。

[0014] 一旦患者受到静脉性溃疡的折磨,伤口再发的可能性也非常高,从54%至78%不

等。这意味着,静脉溃疡对患此病的患者而言具有极其严重的消极影响,大大降低生活质量,并要求广泛的治疗。尽管占医疗总预算高达2.5%,静脉溃疡的影响往往被低估。

[0015] 静脉溃疡的高成本和发病率,加上治疗的难度,标志着引入能够早期检测的低成本、非侵入性系统的一个非常好的机会。虽然传统的激光多普勒系统能够提供相对准确和可靠的信息,但是它们不能被用于连续监测患者,因为它们需要非常笨重、极其昂贵的设备。这样的解决方案太昂贵或难以部署,从而大大限制了使用。

[0016] 因此,需要开发一种监测和预防的解决方案,以扫描组织和测量组织灌注状态作为衡量氧分布水平的度量,以及测量整个组织的穿透性以作为组织健康的指标。因此,本发明的一个目的是使用与压力传感器信号结合的光体积描记来监测患有或可能患有静脉性溃疡的患者的灌注水平。

发明内容

[0017] 本发明的系统和方法包括作为检测和监控溃疡发展的装置的一种配置为扫描和映射组织血液灌注的小型灌注扫描器。该装置集成了平台、数字信号处理单元、与计算机的串行连接、压力传感器、压力计量系统、LED和光电二极管传感器对以及数据资源管理器(explorer)可视化界面。

[0018] 本发明的系统和方法通过能够早期检测到溃疡形成或炎症性压力来提供有效的预防措施,这些溃疡形成或炎症性压力通过其他方法可能在很长一段时间也检测不到,从而增加了感染和发展到更高阶段溃疡风险。

[0019] 在一个优选的实施例中,根据本发明的表征组织的健康状态的小型灌注扫描器和方法,将压力传感元件与光学传感器结合来监测施加到靶组织上的压力水平用以进行精确的皮肤/组织的血液灌注测量和血氧定量测定。本发明的系统和方法能够启用包括但不限于以下新的性能:诸如灌注成像和灌注映像(几何和时间)的测量性能、信号处理和模式识别、经由使用跟踪和压力成像的使用的自动保证、以及数据融合。

[0020] 本发明的传感器增强系统的一个特别的好处是能够更好地管理每个病人,从而能够更及时和更有效地在医院、甚至疗养院使用。这适用于有慢性伤口病史、糖尿病足部溃疡、褥疮或手术后的伤口的病人。

[0021] 另外,信号内容的改变可与病人的活动水平、病人身体的位置和标准化的评估症状结合。通过在信号数据库中保持这些患者中采集的数据,可以使用模式分类、搜索、和模式匹配算法来更好地使症状与皮肤特性和溃疡发展的改变进行映射。

[0022] 本发明的一个方面是一种用于监测患者的靶组织区域的灌注氧合的装置,包括:扫描器,该扫描器包括:平面传感器阵列;该传感器阵列被定位成与靶组织区域的表面接触;该传感器阵列包括一个或多个LED,所述一个或多个LED配置成以针对血红蛋白的波长向靶组织区域发射光;所述传感器阵列包括一个或多个光电二极管,所述一个或多个光电二极管配置成检测从LED反射的光;以及数据获取控制器,其与所述一个或多个LED以及所述一个或多个光电二极管耦合,用于控制来自传感器阵列的光的发射和接收,以获得与靶组织区域相关的灌注氧合数据。

[0023] 本发明的另一方面是一种用于监测患者的靶组织区域的灌注氧合的系统,包括:(a)扫描器,包括:平面传感器阵列,该传感器阵列被定位成与靶组织区域的表面接触;该传

传感器阵列包括一个或多个光源,所述一个或多个光源配置成以针对血红蛋白的波长向靶组织区域发射光;该传感器阵列包括一个或多个传感器,所述一个或多个传感器配置成检测从光源发射的光;压力传感器,其与传感器阵列耦合;该压力传感器配置成获取传感器阵列与靶组织区域表面接触的压力读数;和(b)数据获取控制器,其与所述一个或多个传感器耦合,用于控制来自传感器阵列的光的发射和接收,以获得与靶组织相关的灌注氧合;和(c)处理模块,其与数据获取控制器耦合;(d)该处理模块配置成控制压力传感器和传感器阵列的采样,以便同时获取灌注氧合数据和压力传感器数据,以确保扫描器与靶组织区域的表面的适当接触。

[0024] 本发明的再一个方面是一种用于实时监测患者的靶组织区域的灌注氧合的方法,包括:将传感器阵列定位成与靶组织区域的表面相接触;从传感器阵列中的光源向靶组织区域发射针对血红蛋白的波长的光;接收从光源反射的光;获得与传感器阵列和靶组织区域的表面的接触相关联的压力数据;获得与靶组织区域相关联的灌注氧合;以及对灌注氧合数据和压力数据进行采样,以确保传感器阵列与所述靶组织区域的表面的适当接触。

[0025] 可以理解,本发明的系统和方法并不限于溃疡或伤口的具体情况,在所有形式的伤口管理上可以具有更广阔的应用,如皮肤疾病或治疗。

[0026] 将在本说明书的以下部分中呈现本发明的其它方面,其中,所述详细描述是为了充分披露本发明的优选实施例,而不将其限制于这些优选实施例。

附图说明

[0027] 将参照以下仅仅用于示例目的的附图来更充分地理解本发明。

[0028] 图1示出了根据本发明的用于分析组织的区域的灌注氧合监测(POM)系统的一个优选实施例。

[0029] 图2A和图2B示出了根据本发明的灌注硬件印刷电路板的正面透视图和右侧透视图。

[0030] 图3示出了根据本发明的示例性LED发射器。

[0031] 图4示出了根据本发明的LED驱动器电路。

[0032] 图5示出了被配置用于从光电二极管传感器阵列读取信号的示例性光电二极管读取电路。

[0033] 图6示出了用于校准压力传感器的校准设置。

[0034] 图7示出了一个传感器上的重量为50克、100克、200克和500克的压力验证试验的结果的示图。

[0035] 图8是示出了测得的压力响应曲线、内插曲线(指数)、和该压力传感器被指定为饱和的点的示图。

[0036] 图9示出了第二1磅传感器的压力验证试验的结果。

[0037] 图10是示出了原始压力响应曲线以及各种拟合的示图。

[0038] 图11示出了用于运行本发明的灌注氧合监测(POM)系统的PC设置。

[0039] 图12示出了根据本发明的硬件配置模块界面的屏幕截图。

[0040] 图13示出了根据本发明的图形用户界面的屏幕截图。

[0041] 图14示出了经由Kriging算法进行的示例性内插。

- [0042] 图15示出了用于测试特征提取模块的标记图案的示意图。
- [0043] 图16示出了叠加在图像上的图15的设置。
- [0044] 图17示出了用于输出经过映射和内插的灌注图像的方法的框图。
- [0045] 图18示出了根据本发明的用于消除带内噪声的外差(heterodyning)的示例。
- [0046] 图19是与噪声和校正频率相关的图18的减法方法的理论响应的曲线图。
- [0047] 图20是以dB为单位示出的减法方法的频率响应的曲线图。
- [0048] 图21示出了对高频LED驱动信号进行噪声扣除并且对几个LED驱动周期进行平均以获得与之前类似的数据速率的结果。
- [0049] 图22示出了图21的放大示意图。
- [0050] 图23示出了用于比较颈部和指部组织测量结果的时域信号的采样。
- [0051] 图24示出了所测得信号的频域表示。
- [0052] 图25示出了从前额提取的体积描记信号的结果。
- [0053] 图26示出了从拇指的关节下提取的体积描记信号的读数比较。
- [0054] 图27示出了使用颈部上的反射传感器获得的变化压力的结果。
- [0055] 图28示出了来自黑胶带上方和黑胶带侧面的结果。

具体实施方式

[0056] 图1示出根据本发明的用于分析患者18的组织52的区域的灌注氧合监视(POM)系统的一个优选实施例。系统100总的来讲包括四个主要组件:红光/红外LED阵列44、光电二极管阵列46、压力传感器50、压力计量系统48(包括放大和滤波电路)、数据获取单元40、数字信号处理模块12和具有用户界面的应用程序模块14。

[0057] 系统10包括优选地处于手持式外壳(未示出)内的感测硬件组件16,其包括发射器/传感器阵列(44、46、50)和数据获取单元40。可以以各种阵列形式物理地配置与数据获取单元40(例如,通过电缆或无线连接)耦合的LED阵列44和光电二极管阵列46。数据获取单元40优选地能够与大量的LED和光电二极管连接。信号放大和滤波单元49可以用于在光电二极管信号/数据被数据获取单元40接收前对其进行调整。在一个优选实施例中,光电二极管信号放大和滤波单元49可以包括图5所示并且以下将进行详细描述的光电二极管读取电路120。

[0058] 感测/扫描硬件组件16还可包括强度控制器42,用于控制LED阵列44的输出。强度控制器42优选地包括图4所示并且将在以下进行详细描述LED驱动器电路100。

[0059] 数据获取单元40还与PC 154(见图11)上的应用程序模块14连接,从而允许用户经由通过图形用户界面36观察到的硬件配置模块34来对LED阵列44和来自光电二极管阵列46的信号的信令以及采样速率进行配置。从DAC 40获取的数据优选地存储在数据库32中,以用于后续处理。

[0060] 压力传感器50配置为测量从硬件包16施加给患者组织的压力,从而可以获取压力读数以保证进行测量时施加给皮肤52的压力是一致的和适当的。压力传感器50可以与预调节或测量电路48耦合,该预调节或测量电路48包括放大和滤波电路,以在信号被数据获取控制器40接收到之前对其进行处理。

[0061] LED阵列44配置为在靶组织52上投射针对血红蛋白的波长的光,而光电二极管传

传感器阵列46测量穿过组织52的光量。

[0062] 然后,信号处理模块12进一步经由处理脚本24和滤波模块22对所获取的数据进行处理和滤波。信号处理模块12还包括特征提取模块28,其可以是输出到可视化界面36,用于进一步的处理和可视化。灌注数据模块26将数据转换成体积描记波形,该体积描记波形可以在监视器等(未示出)上显示。界面36和处理模块12还可以配置为输出组织和捕获的灌注数据26的重叠图像。

[0063] 为了产生对应于脱氧和氧合血红蛋白吸收的光的波长,系统12优选地使用发光二极管作为发射源阵列44。在一个优选实施例中,系统10采用OSI Optoelectronics的DLED-660/880-CSL-2双光发射器组合。这种双发射器在一个封装中结合了红光(660nm)和红外光(880nm)LED。每个红光/红外光LED对需要20mA电流源,并且分别具有正向电压2.4/2.0V。可以理解,还可以使用其他光源。

[0064] 为了测量光体积描记(photoplethysmograph),从LED阵列44反射的光由光电二极管阵列46进行检测。在一个优选实施例中,采用OSI Optoelectronics的PIN-8.0-CSL光电二极管。该光电二极管的光谱范围为350nm到1100nm,分别具有为0.33和0.55到660nm和900nm光的响应。

[0065] 图2A和图2B示出了灌注硬件印刷电路板(PCB)60的正面透视图和右侧透视图。PCB 60包括位于光电二极管62的两个阵列46之间的两个LED对64的LED阵列44。板60还包括压力传感器50,用以监测施加在靶组织52上的压力。

[0066] 如图2A所示,光学传感器(例如,LED阵列44和光电二极管阵列46)位于PCB 60的正面侧,并且被配置为面对和(相对于透明盖(未示出)直接或邻近地)按压在靶组织52上。

[0067] 参照图2B,驱动电路(例如,连接器头70)位于PCB 60的背面侧68,从而很安全地不会与测试对象和PCB的容纳阵列传感器部分的正面(右侧)接触。阵列44、46定位成使得连接器头70和对应引线72及电缆74(其与数据获取单元40耦合)不会干扰该装置的使用。

[0068] 图2A中示出的阵列44、46为位于四个光电二极管62之间的两个LED 64。然而,应当理解,该阵列可以包括任何数量的LED,并且可以包括至少一个LED发射器64和一个光电二极管接收器的平面构造。

[0069] 图3示出了一个示例性的LED发射器64(OSI Optoelectronics的DLED-660/880CSL-2),其具有660nm的红光发射器84和880nm的红外光发射器82。

[0070] 图4示出了根据本发明的LED驱动电路100。LED驱动电路100被配置为使得LED封装件64中的红光LED 88和红外光LED 82独立地驱动,即使这两种LED共阳极,通过引线80共用V_{DD}接线。

[0071] 驱动电路100包括低噪声放大器,其耦合到LED 64。在一个优选的实施例中,放大器110包括Linear Technologies的LT6200芯片。但是,应当理解的是,也可以采用在本领域中可用的其它放大器。LED驱动电路100还包括一个p-沟道MOS场效应晶体管(FET)112(例如,松下的MTM76110),它提供了负反馈。当输入端电压增大,50欧姆电阻器102两端的电压也增大。这将导致较大的电流消耗,通过LED 64,使它更加明亮。在2V,大约为40mA电流会通过LED 64,提供最佳的亮度。如果输入端电压增大太多,在LED 64上的电压降将不足以使其关闭,但仍然会有大量的电流流过LED 64和电阻102,造成大的热量积聚。出于这个原因,理想的是,输入电压保持在3V以下,以尽量减少过热并且防止部件损坏。如果放大器110供电

时放大器110的输入是浮动的,则输入端的100k下拉电阻器104和输出端的1k负载电阻器108确保电路100保持关闭状态。1k负载电阻器108还能够保证放大器110提供轨至轨输出电压。1 μ F电容器114确保了输出保持稳定,但是为LED 64快速切换提供了足够带宽。为了提供进一步的稳定,可以对驱动电路100进行修改,使得在电容器114上包括米勒补偿。这种变化提高了在低频率的驱动电路100的相位裕,允许更可靠的操作。

[0072] 图5示出了示例性的光电二极管读取电路120,其配置为读取来自光电二极管传感器阵列46的信号。在一个优选的实施例中,光电二极管62可以包括一个OS1 Optoelectronics的PIN-8.0-DP1光电二极管、PIN 4.0DP1光电二极管、或者PIN-0.8-DP1光电二极管(对于相同的反向偏置电压,它具有较低的电容)。

[0073] 光电二极管读取电路120通过一个简单的电流到电压的运算放大器124工作,如图14示出。运算放大器124(例如,来自Linear Technologies的LT6200)的正相输入引脚由分压器122驱动,提供2.5V(V_{DD} 的一半)。负相引脚被连接到光电二极管62(被反向偏置),并且通过反馈连接到放大器124的输出端。

[0074] 该反馈由具有2.7PF电容器129和100千欧姆的电阻器130的简单的低通滤波器126控制。0.1 μ F的电容器128用来去除分压器与地的耦合。该电路对光电二极管的电流输出进行放大并且将其转换成电压,从而允许数据获取单元经由其电压输出模块读取电压。

[0075] 可以理解,仅仅为了示例性的目的,示出了LED驱动电路100和光电二极管读取电路120的各个组成部分,并且可以根据需要使用其他模块或其他类型的组件。

[0076] 在本发明的一个实施例中,数据获取控制器40包括与NL 9104 3M gate FPGA机箱耦合的美国国家仪器公司的CompactRIO 9014实时控制器。数据获取控制器40使用用于电流输出、电流输入和电压输入的三组模块来与LED阵列44和光电二极管46连接。

[0077] 在一个实施例中,控制器40包括一个处理器、实时操作系统、存储器,并且支持经由USB的额外存储(全部未示出)。控制器40还可以包括一个以太网端口(未示出),用于连接到PC 154的用户界面。控制器40包括FPGA背板、电流输出模块(如N19263)、电流输入模块(如N19203)、以及允许来自光电二极管/放大器模块的多电压输入的电压输入模块(如N19205)。

[0078] POM系统10优选地采用压力传感器50来测量压力,并确保一致的结果(例如11B.FlexiForce传感器)。由于压力变化对体积描记测量带来的混杂影响,来自压力传感器50的读数提供了一个度量,根据该度量用户可以将传感器硬件16施加至患者的皮肤52。

[0079] 压力传感器50优选地附接在LED阵列44后面,并且测量用于施加到目标位置的压力。压力传感器50优选地被配置在指定的范围内提供精确的压力测量结果,例如,范围从零到约1磅,该范围涵盖了当使用POM传感硬件16时可以合理地施加的压力范围。

[0080] 压力传感器50用于指导用户更一致地操作扫描器16,从而使传感器/扫描器16在每次测量时以类似的方式定位。因此,取得的血氧定量测定数据被证实为是由来自压力传感器50的读数准确地获取的。

[0081] 在一个优选的实施例中,对压力传感器50进行校准,以便确保压力传感器提供可重复的、很好理解的测量结果,这些测量结果可直接解释成原始压力值。图6示出了校准装置140,其用于校准压力传感器50。橡胶加压器144被压成平坦表面,并用于分配FlexiForce传感器50的压力敏感区域上的重量。重量142被用于分配传感器50的活性区域上的重量。使

用从50克到500克的范围内的四个重量进行实验。通过加压器144向压力传感器50直接施加压力,并记录它的输出。

[0082] 图7至图10中的结果显示一个非线性的、但稳定的趋势,该数据可以用来将压力传感器的未来的任何测量值转换成绝对压力值。

[0083] 图7示出了一个传感器上的重量为50克、100克、200克和500克的压力验证试验的结果的示图。图8是示出了测得的压力响应曲线、内插曲线(指数)、和该压力传感器被指定为饱和的点的示图。图9示出了第二1磅传感器的压力验证试验的结果。为了进行该实验,施加了额外的中间重量水平(例如,150克和300克)。图10是示出了原始压力响应曲线以及各种拟合的示图。对于进行测试的两个传感器而言,指数拟合作为最佳拟合。

[0084] 虽然系统10最好使用来自压力传感器50的数据来验证扫描器在靶组织位点52上的适当布置,但是可以理解,在一个备选实施例中,用户可以简单地放弃压力监测而手动地监测压力(例如,依靠触感或在重力作用下将扫描器16简单地放置在组织位置52上)。

[0085] 参照图11,用户优选地通过PC 154与数据获取和控制单元40进行交互,PC 154运行处理模块和包括图形用户界面36(例如LabVIEW等)的应用程序模块14。在一个优选的实施例中,PC 154通过以太网连接(图中未示出)与数据获取单元40进行通信。可选地,PC 154通过诸如WIFI、蓝牙等的无线连接(未示出)与数据获取单元40进行通信。数据获取单元40上生成的数据文件还可以通过FTP连接转移到PC 154,以进行临时存储和进一步处理。

[0086] 对于图11所示的PC 154接口,LED阵列的各个LED 64投射针对血红蛋白的波长的光,光电二极管传感器62测量经过并且被组织52反射的光的量。数据获取单元40通常包括耦合到LED 164的数字TTL输出152和用于光电二极管62的模拟DC输入150。然后,数字处理模块12进一步对该数据进行处理和滤波,该数据然后被传输到图形用户界面36进行进一步处理和可视化。然后,该数据可以被转换成体积描记波形,以被显示。

[0087] 图12示出了硬件配置模块34界面的屏幕截图160。可以选择输入,以调节字段166中的LED阵列44的参数、字段164中的电压通道设置、字段162中的电流通路设置、以及诸如采样周期、压力采样周期等的其他参数。

[0088] 图13示出了图形用户界面36的屏幕截图170,该图形用户界面36还用作数据管理和资源管理器,以使得用户能够容易地读取灌注传感器并且观察各种信号。屏幕截图170示出了从血氧定量测定传感器(光电二极管阵列46和LED阵列44)、从压力传感器50捕获的数据和通过扫描光电二极管阵列46和LED阵列44捕获的跟踪/位置数据的整合。屏幕截图170示出了第一窗口172和第二窗口174,第一窗口172显示了体积描记波形(图13所示的2秒),第二窗口174显示了扫描器已经进行的绝对的x和y轴移动。图形用户界面36还能够将这映射到测量得到的 SpO_2 数据(例如,通过拖动显示窗口172和174中的一个)。屏幕截图170右侧的压力表176是压力传感器50读数的压力表,显示了大约所施加最大压力的一半。压力表176优选地以彩色编码条显示用于用户正在施加的压力大小与最大可测量压力的比(当施加压力变大时,编码条颜色从蓝到绿到红)。压力表176优选地映射到针对不同位置的最佳压力值。

[0089] 为了提供具有更多信息量的局部区域的灌注图,可以使用传感器跟踪数据进行血氧仪数据的内插。光学血氧仪传感器16提供绝对的 SpO_2 读数,从而给出被氧合的血液百分比。当与其提取位置相关时,该信号可以用于产生血液氧合图。在一个优选的实施例中,用

于产生 SP_{O_2} 读数的LED阵列44还用于确定位置。然而,应当理解,其他光学传感器(例如,激光(未示出))可独立于LED SP_{O_2} 读数来获得位置读数。在这种配置中,低功率激光(类似于激光跟踪鼠标)用于以很快的时间间隔对一个小区域进行成像,然后检测该图像已经移动的移动量。该信号然后被转换成二维的“X”和“Y”位置和位移测量结果。

[0090] 在一个优选的实施例中,经由Kriging算法进行内插,使用血氧仪传感器16映射数据点从而在测试区域上跟踪传感器16的移动。Kriging是通常用于空间相关信息的线性最小二乘内插方法。使用内插方法来估计值填补扫描可能丢失的空白点。内插数据被编译到彩色编码图像中,并且显示给用户。这样能够对原始数据进行准确的、各向异性的内插,这使得最终的结果更容易可视化。图14示出了示例内插。传感器硬件16的运动在该示例中几乎是一维的,在x轴上呈线性趋势。这是由于在该方向上的点的方差低(注意,相比于在Y方向上的位移1400,在X方向上有大约40的总位移)。

[0091] 为了使得所收集的血液的血氧定量测定数据可视化,处理软件12优选包括特征提取模块28,其可以检测到在照片上的标记(marker),然后正确地对齐、并叠加在血液的血氧定量测定数据26上(参见图1、图17)。在一个优选的方法中,特征提取模块28提取图像(例如,从扫描站点的一个照相机拍摄的照片),并将灌注数据直接叠加在其所提取的位置上。

[0092] 图15示出了用于测试特征提取模块28的标记图案200的示意图。图16示出叠加在图像205上的图15的设置。三个标记(202、204和206)用于一个给定的扫描区域208的定界点。第一标记202用来确定图像的旋转角度。第二标记206用来确定图像的左边界(图像位置)。第三标记204用来确定图像的宽度。标记(202、204和206)可以是任何颜色,但绿色是理想的颜色,因为它很容易区别于所有肤色。为了清楚地说明特征提取软件,绿色小塑料盒被用来代表点202、204和206(参见图16),快速编辑图像205从而把它们三个以一个合适模式布置。除了这个操作之外,通过该软件快速地产生所有其他图像。网格208作为样本数据,以更清楚地说明该工作正在做什么。

[0093] 在一个实施例中,移动应用程序(未示出)可用于方便地捕获和整合图片用于处理软件12。该应用程序允许用户快速使用移动设备(如,智能手机等)拍摄图片,并且会是图片自动通过蓝牙发送而被处理软件12捕获。然后,图像可与映射系统整合。

[0094] 图17示出用于输出一个映射和内插的灌注图像(例如,利用处理模块12)的方法220的框图。在本文中所附的源代码附录中,显示了执行方法220的代码示例。可以理解,所提供的代码仅仅是如何执行本发明的方法的一个例子。

[0095] 首先,在步骤222(通过处理脚本24)提取来自数据获取单元40的获取数据(该数据可能存储在服务器32上)。然后,所提取的数据被用于同时提取每个测量点的位置数据、灌注数据和压力数据。处理软件12可以同时采样位置、灌注、压力读数(例如,以3Hz间隔),以创建每个时间间隔的匹配的一组压力、位置和血氧测量值。

[0096] 为了从灌注模块228所记录的原始数据生成有用的信息和指标,使用一些算法。

[0097] 在步骤230,从数据中提取特征(例如,通过特征提取模块28)。然后,在步骤232映射对应于硬件传感器16位置的位置数据。完成扫描后,在步骤234将血氧定量测定数据映射到与从步骤232所获得的传感器的位置数据对应的适当坐标。在步骤236,对映射的数据进行内插(例如,使用在图14中示出的Kriging算法)。内插数据可被编译成彩色编码图像,并显示给用户,和/或灌注数据然后可被叠加在扫描站点的背景图像上(例如,图像205),如图

15和图16所述。

[0098] 在灌注侧,在步骤224对所提取数据执行RF噪声滤波。然后,在步骤226中去除运动噪声以在步骤228取得灌注数据。可通过滤波模块22执行步骤224和226。

[0099] 在图18所示的优选方法中,外差用来帮助消除带内噪声。LED阵列44关闭时记录的数据被从LED阵列44开启时的相邻数据中减去(减法方法)。这产生高频噪声,但消除了低频带内噪声,带内噪声是一个更大的问题。然后通过一个低通滤波器对被引入的额外的高频率噪声进行滤波。该算法是可配置的,以允许保存PPG信号的高频信息。

[0100] 如图18所示,来自标记为1和2的区域的噪声信息被用来计算出现在区域3的噪声。这可通过单侧方法或双侧方法来实现。

[0101] 对于单侧方法而言,只使用来自区域1的先前噪声信息,并且相关的噪声水平被假定为与区域1和3中的相同。对于双侧方法,平均来自区域1和2的噪声。最后,使用目标数据点(3)之前和随后的所有有用的噪声周期的数据,通过内插法尝试对3处的噪声进行内插。对这些区域的测量数据进行平均,以产生针对每个LED 64脉冲的单点。然后,在结束时,对结果进行低通滤波,以去除高频噪声。

[0102] 图19是与噪声和校正频率相关的图18的减法方法的理论响应的曲线图,该理论响应是通过以下步骤确定的:将宽频率范围的正弦噪声添加到方波信号上、应用噪声消除方法(校正方法)、以及测量剩余噪声与原始噪声的比例。然后,针对给定频率,在所有相位上平均测量结果。图20是以dB为单位显示的减法方法的频率响应的曲线图。

[0103] 对于图19和图20所示的频率响应曲线图,频率被相对于模拟的LED驱动信号的频率进行了归一化,其中1表示噪声与驱动信号频率相同,2表示噪声频率是驱动频率的二倍,以此类推。

[0104] 图21和图22是显示与未执行噪声消除技术情景相比对高频LED驱动信号应用了上述的图18的噪声消除(减法)方法所提取的体积描记信号的曲线图。图21示出了来自在高频LED驱动信号上应用噪声消除并且在几个LED驱动周期上进行平均以获得与之前类似的数据率的结果。注意,在图21中放大了图22中的1.5s左右的成功的噪声消除,显示了通过差分噪声消除去除的噪声尖峰。这些示图示出了本发明的噪声消除方法在去除带内噪声方面是有效的。

[0105] 用体积描记测量的频域信号来进行频域分析/实验。这些实验不仅显示了心率频率的高幅值成分,也显示了其谐波。这在不同位置之间表现出良好的一致性。

[0106] 为了验证频域中所示的谐波不是噪声或抖动的结果,而是表示脉冲波形的实际分量,构建了一个正弦波。该正弦波是通过计算针对每个单独的脉冲波形峰值的频率处的正弦波的总和来构建的。该叠加的目的是对波形中的频率抖动的影响进行建模,同时去除任何由于脉冲波形形状导致的频率分量。

[0107] 图23和图24示出了信号比较。图23示出用于比较的时域信号的样本。在相同压力下,对颈部测量与拇指测量进行了比较。图24示出了所测得的信号的频域表示。注意128BPM(2.13Hz)的二次谐波、207BPM(3.45Hz)的三次谐波。这些结果表明,如下所示的谐波的确是脉冲波形所固有的,而不是噪声或频率抖动的结果。

[0108] 使用本发明的灌注系统10在多个身体位置(包括颈部、拇指和前额)上进行了实验。在图25至图27中显示了所提取的体积描记信号的样本,图25至图27清楚地显示出该灌

注系统成功地去除了运动和环境噪声并且从不同身体位置提取了体积描记信号。

[0109] 图25显示来自所提取的前额的体积描记信号的结果。以使用压力传感器测量的电阻的形式给出了压力值。较小的电阻表示较高的施加压力。

[0110] 图26示出了来自拇指上的关节下的提取的体积描记信号的读数的比较。测量中，压力以外的所有因素均保持不变。适度的压力显然会导致更好的波形。

[0111] 图27示出使用颈部上的反射传感器导致的变化压力的结果。下面的实验显示了该系统中所施加的压力与灌注信号的整合和融合的重要性，这是由于传感器阵列施加到靶组织上的压力对灌注读数有着重大的影响，如在以下附图中示出。结果显示，当施加中度(0.15M至70k-ohm)压力时，颈部和拇指给出最好的结果，而采用低的压力(0.15M-ohm以上)时额头产生最好的结果。这可能是由于颈部和拇指比额头组织软。

[0112] 还在黑胶带上测试了灌注系统10，作为标记组织上的位置的手段。黑胶带用于作为皮肤上的标记来进行测试。传感器用来测量胶带上的信号，并且刚好处于黑胶带的侧面。可以看到皮肤上的印记，其中不在黑胶带上使用反射传感器。

[0113] 图28示出从黑色带侧的上方和侧面获得的结果。结果表明，使用简单的一块黑胶带能够有效地引起较大的信号差异，因此其可以用作特定的身体位置的标记。

[0114] 本发明的实施例可参考根据本发明的方法和系统的流程示图，和/或算法、公式或其他计算表达式(这些也可被实现为计算机程序产品)来进行说明。在这方面，流程图的每块或步骤、流程图中的块(和/或步骤)的组合、算法、公式或计算表达式可以通过各种手段(例如硬件，固件，和/或包括在计算机可读程序代码逻辑中实现的一个或多个计算机程序指令的软件)来实施。将理解的是，任何这样的计算机程序指令可以被加载到计算机，包括但不限于通用计算机或专用计算机或用以产生机器的其他可编程处理设备，使得在计算机或其他可编程处理装置上执行的计算机程序指令能够产生用于实现在一个或多个流程图中的一个或多个功能块中指定的功能。

[0115] 因此，流程图的功能块、算法、公式或计算表达式支持用于执行特定功能的手段组合、用于执行特定功能的步骤的组合、以及用于执行特定功能的诸如在计算机可读程序代码逻辑装置中体现的计算机程序指令。还应当理解，本文所描述的流程图示总的每个功能块、算法、公式或计算表达式及其组合可以通过执行特定功能或步骤的基于专用硬件的计算机系统或者专用硬件和计算机可读程序代码逻辑装置的组合来实现。

[0116] 而且，这些计算机程序指令(诸如在计算机可读程序代码逻辑中体现的)还可以存储在能够给计算机或其他可编程处理设备发出指令的计算机可读存储器中，使得存储在计算机可读存储器中的指令产生包括实现一个或多个流程图中的一个或多个功能块指定的功能的指令装置的产品。计算机程序指令还可以加载到计算机或其他可编程处理设备上以引起要在计算机或其他可编程设备上执行的一系列操作步骤，从而产生计算机实现过程，使得执行计算机或其他可编程处理设备的指令提供实现一个或多个流程图的一个或多个功能块、一个或多个算法、一个或多个公式、或者一个或多个计算表达式指定的功能的步骤。

[0117] 从上面的讨论可以理解，本发明可以以不同的方式体现，包括以下内容：

[0118] 1. 一种用于监测患者的靶组织区域的灌注氧合的设备，包括：扫描器，其包括：平面传感器阵列；该传感器阵列配置为被定位成与靶组织区域的表面接触；该传感器阵列包

括配置为向靶组织区域发射针对血红蛋白的波长的光的一个或多个LED;该传感器阵列包括配置为检测从所述LED反射的光的一个或多个光电二极管;以及数据获取控制器,其耦合到所述一个或多个LED和一个或多个光电二极管,用于控制来自所述传感器阵列的光的发射和接收以获得与靶组织区域相关的灌注氧合数据。

[0119] 2. 根据实施例1所述的设备,所述扫描器还包括:耦合到所述传感器阵列的压力传感器;所述压力传感器配置为获得所述传感器阵列与所述靶组织区域的表面接触的压力读数;其中,所述扫描器配置为在获得灌注氧合数据的同时获得压力传感器读数,以确保扫描器与靶组织区域的表面的适当接触。

[0120] 3. 根据实施例2所述的设备:其中,所述压力传感器和传感器阵列被连接到印刷电路板(PCB)的第一侧;以及其中,所述数据获取控制器在与所述第一侧相反的第二侧上连接至PCB。

[0121] 4. 根据实施例1所述的设备,其中,每个LED包括配置用于发射红光(660nm)和红外光(880nm)的双发射器。

[0122] 5. 根据实施例4所述的设备:其中,所述一个或多个LED耦合到驱动器电路;以及其中,所述驱动器电路被配置为允许红光LED发射器和红外光LED发射器在共享一个公共阳极的同时被独立驱动。

[0123] 6. 根据实施例5所述的设备,其中,所述驱动器电路包括放大器;以及场效应晶体管,其被配置为提供负反馈。

[0124] 7. 根据实施例2所述的设备,还包括:处理模块,其耦合到所述数据获取控制器;所述处理模块被配置为控制所述压力传感器和传感器阵列的采样,以同时获取压力传感器数据和灌注氧合数据。

[0125] 8. 根据实施例7所述的设备,其中,所述处理模块被配置为获得来自传感器阵列的读数以获得所述扫描器的位置数据。

[0126] 9. 根据实施例8所述的设备,其中,所述处理模块被配置为产生靶组织的灌注氧合图。

[0127] 10. 根据实施例8所述的设备,其中,所述处理模块被配置成控制所述压力传感器和传感器阵列的采样,以同时获取选自包括压力传感器数据、灌注氧合数据和位置数据的组的两个或两个以上的数据参数,从而同时显示所述的两个或两个以上的数据参数。

[0128] 11. 一种用于监测患者的靶组织区域的灌注氧合的系统,包括:(a)扫描器,包括:平面传感器阵列;所述传感器阵列配置为被定位成与靶组织区域的表面接触;所述传感器阵列包括配置为向所述靶组织区域发射针对血红蛋白的波长的光的一个或多个光源;所述传感器阵列包括配置为检测从所述光源反射的光的一个或多个传感器;压力传感器,其耦合到所述传感器阵列;所述压力传感器配置为获得所述传感器阵列与所述靶组织区域的表面接触的压力读数;以及(b)数据获取控制器,其耦合到所述一个或多个传感器并且用于控制来自所述传感器阵列的光的发射和接收以获得与所述靶组织相关的灌注氧合数据;以及(c)处理模块,其耦合到所述数据获取控制器;(d)所述处理模块配置为控制所述压力传感器和传感器阵列的采样以同时获取灌注氧合数据和压力传感器数据,从而确保所述扫描器与所述靶组织区域的表面的适当接触。

[0129] 12. 根据实施例11所述的系统:其中,所述传感器阵列包括配置来向靶组织区域发

射针对血红蛋白的波长的光的一个或多个LED;其中,所述传感器阵列包括配置为检测从所述LED反射的光的一个或多个光电二极管。

[0130] 13. 根据实施例12所述的系统:其中,所述一个或多个LED中的每一个LED包括被配置为发射红光(660nm)和红外光(880nm)的双发射器;其中,所述一个或多个LED耦合到驱动器电路;其中,所述驱动器电路被配置为允许红光LED发射器和红外光LED发射器在共享公共阳极的同时被独立地驱动。

[0131] 14. 根据实施例11所述的系统,还包括:图形用户界面;其中,所述图形用户界面被配置为显示灌注氧合数据和压力传感器数据。

[0132] 15. 根据实施例14所述的系统,所述处理模块进一步被配置为从传感器阵列获得读数,以便获得扫描器的位置数据。

[0133] 16. 根据实施例15所述的系统,其中,所述处理模块进一步被配置为对位置数据进行内插,以生成靶组织的灌注氧合图。

[0134] 17. 根据实施例16所述的系统,其中,所述处理模块被配置成控制所述压力传感器和传感器阵列的采样,以同时获取选自包括压力传感器数据、灌注氧合数据和位置数据的组的两个或两个以上的数据参数,从而同时显示所述的两个或两个以上的数据参数。

[0135] 18. 根据实施例16所述的系统,其中,所述处理模块被配置为接收靶组织的图像,并在该图像上叠加灌注氧合图。

[0136] 19. 根据实施例14所述的系统,其中,所述图形用户界面被配置为允许用户输入以操作传感器阵列和压力传感器的设置。

[0137] 20. 根据实施例11所述的系统,其中,所述处理模块进一步包括:滤波模块;所述滤波模块配置为通过将当所述一个或多个光源处于关闭状态时记录的数据从当所述一个或多个光源处于开启状态时记录的数据中减去来滤除带内噪声。

[0138] 21. 一种对患者的靶组织区域的灌注氧合进行实时监测的方法,包括:将传感器阵列定位成与靶组织区域的表面接触;将来自所述传感器阵列中的光源的针对血红蛋白的波长的光发射到所述靶组织区域;接收从所述光源反射的光;获得传感器阵列与靶组织区域的表面接触相关的压力数据;获得与靶组织区域相关的灌注氧合数据;以及对灌注氧合数据和压力数据进行采样,以确保传感器阵列与所述靶组织区域的表面的适当接触。

[0139] 22. 根据实施例21所述的方法:其中,所述传感器阵列包括配置为向靶组织区域发射针对血红蛋白的波长的光的一个或多个LED;及其中,所述传感器阵列包括被配置为检测从LED反射的光的一个或多个光电二极管。

[0140] 23. 根据实施例22所述的方法:其中,所述一个或多个LED中的每一个LED包括被配置为发射红光(660nm)和红外光(880nm)的双发射器;该方法进一步包括在红光LED发射器和红外光LED发射器共用一个公共阳极的同时独立地驱动红光LED发射器和红外光LED发射器。

[0141] 24. 根据实施例21所述的方法,还包括:同时显示灌注氧合数据和压力传感器数据。

[0142] 25. 根据实施例21所述的方法,还包括:获取来自所述传感器阵列的读数以获得所述扫描器的位置数据。

[0143] 26. 根据实施例25所述的方法,还包括:对位置数据进行内插,以产生靶组织的灌

注氧合图。

[0144] 27. 根据实施例26所述的方法, 其中, 对位置数据进行内插的步骤包括向所获取的位置数据应用Kriging算法。

[0145] 28. 根据实施例26所述的方法, 还包括: 对压力传感器和传感器阵列进行采样, 以同时获取压力传感器数据、灌注氧合数据和位置数据; 以及同时显示压力传感器数据、灌注氧合数据和位置数据。

[0146] 29. 根据实施例26所述的方法, 还包括: 接收靶组织的图像; 以及将所述灌注氧合图叠加在所述图像上。

[0147] 30. 根据实施例21所述的方法, 还包括: 提供图形用户界面, 以允许用户输入; 以及根据所述用户输入控制传感器阵列和压力传感器的采样设置。

[0148] 31. 根据实施例21所述的方法, 还包括: 使所述一个或多个光源在所述一个或多个光源开启的周期和所述一个或多个光源关闭的周期之间循环; 以及通过将所述一个或多个光源处于关闭状态时记录的数据从在所述一个或多个光源处于开启状态时记录的数据中减去来滤除带内噪声。

[0149] 虽然上面的描述包含许多细节, 这些不应该被解释为限制本发明的范围, 而仅仅是提供对本发明的一些优选实施例的示例说明。因此, 应当理解, 本发明的范围完全包括对于本领域技术人员显然的其它实施例, 因此, 本发明的范围仅由所附的权利要求书限定, 其中某个元素的单数形式不是旨在表示“一个且仅一个”, 除非明确说明, 则表示“一个或多个”。那些对于本领域的普通技术人员已知的, 上述优选实施例中的各元素的所有结构、化学和功能等同物以引文方式明确地并入本文, 并且旨在包含于当前权利要求中。此外, 设备或方法没有必要解决本发明所要解决的每个问题。此外, 在本公开中没有元素、组件或方法步骤的目的是要贡献给公众, 无论该元素、组件或方法是否明确记载在权利要求中。本文中没有任何权利要求的元素是要根据35U.S.C.112, 第六款的规定进行解释, 除非明确地使用短语“用于…装置”来记载该元素。

[0150] 源代码附录

[0151] 通过举例而不是限制的方式提交下面的源代码, 作为本发明的信号处理的一个实施例中。本领域技术人员将容易地理解, 可以以各种其它方式进行信号处理, 这从本文的描述中是容易理解的, 并且这些信号处理的方法不限于下面列出的源代码中所示的方法。

[0152] %clear all;

[0153] clc;

[0154] %-----%

[0155] %Detect Heart Rate, Perfusion&SpO2

[0156] %-----%

[0157] %%Input File

[0158] %Perfusion=zeros(52,1);

[0159] %for ll=0:51

[0160] %inputfile

[0161] =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(11));
[0162] inputfile='gen3\gen3r10';
[0163] samplingRate=10e3;%Sampling Rate in Hz
[0164] period=5e-3;%Period in s
[0165] duty=2.5e-3;%Duty Cycle in s
[0166] totalTime=10;%Total File Time in s
[0167] offsetR=2.5e-3;%Red light offset in s
[0168] offsetlR=0e-3;%Red light offset in s
[0169] transTime=1.2e-4;%Rise/Fall time in s
[0170] %%Heuristics for Peak Detection&Blood Oximetry
[0171] RED_sens=0.42;%Photodiode sensitivity@660nm in A/V
[0172] lR_sens=0.61;%Photodiode sensitivity@880nm in A/V
[0173] MAX_HEART_RATE=220;
[0174] MIN_SAMP=1/((period*5)*MAX_HEART_RATE/60);%Fastest heartrate
allowed
[0175] %%Read Input File into Matlab
[0176] sensorselect=3;
[0177] if sensorselect==1%5mm
[0178] [PD1,PD2,PD3,PD4]=textread(inputfile,'%f%f%f%f%*[\n]',
[0179] 'delimiter',',,');%PD1->central photodiode(Channel 0);PD2->Drive
[0180] signal(Channel 1)
[0181] elseif sensorselect==2%10mm
[0182] [PD2,PD1,PD3,PD4]=textread(inputfile,'%f%f%f%f%*[\n]',
[0183] 'delimiter',',,');%PD1->central photodiode(Channel 0);PD2->Drive
[0184] signal(Channel 1)
[0185] elseif sensorselect==3
[0186] [PD2,PD3,PD1,PD4]=textread(inputfile,'%f%f%f%f%*[\n]',
[0187] 'delimiter',',,');%PD1->central photodiode(Channel 0);PD2->Drive
[0188] signal(Channel 1)
[0189] elseif sensorselect==4
[0190] [PD2,PD3,PD4,PD1]=textread(inputfile,'%f%f%f%f%*[\n]',
[0191] 'delimiter',',,');%PD1->central photodiode(Channel 0);PD2->Drive
[0192] signal(Channel 1)
[0193] end
[0194] PD 1=-PD 1;
[0195] %if trial==3
[0196] %PD1=PD1(length(PD1)/2+1:end);
[0197] %end
[0198] %Data=DownloadFromDB();
```

```

[0199] %PD 1=Data(1:end,1);
[0200] %PD2=Data(1:end,2);
[0201] No_R1R_Waves=totalTime/period;%Total#of RED+1R square waves
[0202] %%Noise Cancellation
[0203] %%-----%
[0204] %%1.single-sided subtraction
[0205] %%-----%
[0206] averageRed=zeros(No_R1R_Waves,1);
[0207] averageRedStep1=zeros(No_R1R_Waves,1);
[0208] averageRedStep2=zeros(No_R1R_Waves,1);
[0209] average1R=zeros(No_R1R_Waves,1);
[0210] averageNoise_1=zeros(No_R1R_Waves,1);%1 st off portion in each
period
[0211] averageNoise_2=zeros(No_R1R_Waves,1);%2nd off portion
[0212] for i=0:No_R1R_Waves-1
[0213] for j=1:(duty-transTime)*samplingRate%Average every period
[0214] averageRed(i+1,1)=averageRed(i+1,1)+
[0215] PD 1(ceil(i*period*samplingRate+j+offsetR*samplingRate+transTime*
samplingRate));
[0216] %average1R(i+1,1)=average1R(i+1,1)+
[0217] PD 1(floor(i*period*samplingRate+j+offset1R*samplingRate+transTime*
samplingRate));
[0218] end
[0219] %for j=1:(duty/2)*samplingRate%Average every period,no
[0220] transition time because LED is already on,changes are very short
[0221] %averageRedStep1(i+1,1)=averageRed(i+1,1)+
[0222] PD 1(ceil(i*period*samplingRate+j+offsetR*samplingRate+transTime*
samplingRate));
[0223] %averageRedStep2(i+1,1)=averageRed(i+1,1)+
[0224] PD 1(ceil(i*period*samplingRate+j+offsetR*samplingRate+transTime*
samplingRate+floor((duty/2)*samplingRate)));
[0225] %%average1R(i+1,1)=average1R(i+1,1)+
[0226] PD 1(floor(i*period*samplingRate+j+offset1R*samplingRate+transTime*
samplingRate));
[0227] %end
[0228] for j=1:(period-duty-transTime)*samplingRate%Averaging the off
[0229] portion for noise subtraction
[0230] %averageNoise_1(i+1,1)=averageNoise_1(i+1,1)+
[0231] PD 1(floor(i*period*samplingRate+j+transTime*samplingRate));

```

```

[0232] averageNoise_1(i+1,1)=averageNoise_1(i+1,1)+
[0233] PD1(max(2,floor(i*period*samplingRate+j+transTime*samplingRate-
(period-duty-offsetR-transTime)*samplingRate)));
[0234] %averageNoise_2(i+1,1)=averageNoise_2(i+1,1)+
[0235] PD1(floor(i*period*samplingRate+j+(offsetR+duty)*samplingRate));
[0236] end
[0237] averageRed(i+1,1)=averageRed(i+1,1)/floor((dutytransTime)*
[0238] samplingRate);
[0239] %average1R(i+1,1)=average1R(i+1,1)/((dutytransTime)*
[0240] samplingRate);
[0241] %averageRedStep1(i+1,1)=averageRedStep1(i+1,
[0242] 1)/floor((duty/2)*samplingRate);
[0243] %averageRedStep2(i+1,1)=averageRedStep2(i+1,
[0244] 1)/floor((duty/2)*samplingRate);
[0245] averageNoise_1(i+1,1)=averageNoise_1(i+1,1)/floor((period-
dutytransTime)*samplingRate);%Use period/2 when using both red and 1R
[0246] %averageNoise_2(i+1,1)=averageNoise_2(i+1,1)/((period/2-
dutytransTime)*samplingRate);
[0247] end
[0248] averageRed_1=averageRed-averageNoise_1;
[0249] averageRed_step=averageRedStep2-averageRedStep1;
[0250] %average1R_1=average1R-averageNoise_2;
[0251] averageRed_4=zeros(No_R1R_Waves/5,1);
[0252] average1R_4=zeros(No_R1R_Waves/5,1);
[0253] for i=1:(No_R1R_Waves/5)
[0254] for j=1:5
[0255] averageRed_4(i)=averageRed_4(i)+averageRed_1((i-1)*5+j);
[0256] %average1R_4(i)=average1R_4(i)+average1R_1((i-1)*5+j);
[0257] end
[0258] averageRed_4(i)=averageRed_4(i)/5;
[0259] %average1R_4(i)=average1R_4(i)/5;
[0260] end
[0261] %%-----%%
[0262] %%2.double-sided subtraction
[0263] %%-----%%
[0264] averageNoise_Red=(averageNoise_1+averageNoise_2)./2;%Average the
off portion on two sides of one on portion
[0265] averageNoise_1R=(averageNoise_1(2:end)+averageNoise_2(1:end 1))./2;
[0266] average1R_2=zeros(No_R1R_Waves,1);

```

```

[0267] averageRed_2=averageRed-averageNoise_Red;
[0268] averageIR_2(1:end-1)=averageIR(1:end-1)-averageNoise_IR;
[0269] averageIR_2(end)=averageIR(end)-averageNoise_2(end);%Last period of
IR uses single-sided subtraction
[0270] %%-----%
[0271] %3.interpolation subtraction
[0272] %%-----%
[0273] %Noise_raw=zeros(totalTime*samplingRate,1);%Store the
[0274] low-pass-filtered off portion continuously
[0275] %x_Noise=zeros(floor(offsetR*samplingRate*transTime*
[0276] samplingRate)+floor(offsetLR*samplingRate(
[0277] offsetR*samplingRate+
[0278] (duty+transTime)*samplingRate))*No_RIR_Waves,1);%coordinates of
Noise_raw
[0279] %x_Noise_x=0;
[0280] %Noise_raw_0=zeros(totalTime*samplingRate,1);
[0281] %
[0282] %for i=0:No_RIR_Waves-1
[0283] %for j=1:period*samplingRate
[0284] %if(((j<=offsetR*samplingRate)&&(j>transTime*samplingRate))
[0285] ||((j>(offsetR*samplingRate+(duty+transTime)*samplingRate))&&(j<=
offsetLR*samplingRate)))%load off portion to Noise_raw
[0286] %Noise_raw_0(floor(i*period*samplingRate+j))=
[0287] PD1(floor(i*period*samplingRate+j));
[0288] %end
[0289] %end
[0290] %end
[0291] %
[0292] %order=50;%Pre-lowpass filter for spline interpolation
[0293] %cutoff=200/samplingRate;%Cut off frequency=100 Hz
[0294] %y1=fir1(order,cutoff,'low');
[0295] %PD1_LPF=filtfilt(y1,1,Noise_raw_0);
[0296] %
[0297] %for i=0:No_RIR_Waves-1
[0298] %for j=1:period*samplingRate
[0299] %if(((j<=offsetR*samplingRate)&&(j>transTime*samplingRate))
[0300] ||((j>(offsetR*samplingRate+(duty+transTime)*samplingRate))&&(j<=
offsetLR*samplingRate)))%load off portion to Noise_raw
[0301] %x_Noise_x=x_Noise_x+1;

```

```

[0302] %Noise_raw(x_Noise_x)=
[0303] PD1_LPF(floor(i*period*samplingRate+j));
[0304] %
[0305] %
[0306] x_Noise(x_Noise_x)=floor(i*period*samplingRate+j);
[0307] end
[0308] %end
[0309] %end
[0310] %
[0311] %
[0312] %Noise=
[0313] interp1(x_Noise,Noise_raw(1:x_Noise_x),1:samplingRate*totalTime,'
spline');%Noise interpolation
[0314] %PD_N=PD1-Noise';
[0315] %
[0316] %
[0317] %averageRed_3_1=zeros(No_RIR_Waves,1);
[0318] %averageIR_3_1=zeros(No_RIR_Waves,1);
[0319] %for i=0:No_RIR_Waves-1%Average data in each square wave period
[0320] %for j=1:floor((duty-transTime)*samplingRate)
[0321] %averageRed_3_1(i+1,1)=averageRed_3_1(i+1,1)+
[0322] PD_N(floor(i*period*samplingRate+j+offsetR*samplingRate+transTime*
samplingRate));
[0323] %averageIR_3_1(i+1,1)=averageIR_3_1(i+1,1)+
[0324] PD_N(floor(i*period*samplingRate+j+offsetIR*samplingRate+transTime*
samplingRate));
[0325] %end
[0326] %averageRed_3_1(i+1,1)=averageRed_3_1(i+1,1)/(floor
((dutytransTime)*samplingRate));
[0327] %averageIR_3_1(i+1,1)=averageIR_3_1(i+1,1)/(floor((dutytransTime)*
samplingRate));
[0328] %end
[0329] %averageIR_3_1(end)=averageIR_3_1(end-1);%Abandon the last one of
[0330] IR_3 to eliminate error caused by interpolation
[0331] %%Create a Low-pass and Filter Waveforms
[0332] averageRed=averageRed_1;%_1,_2,_3,_4 corrcspond to single-sided
[0333] subtraction,double-sided subtraction,interpolation subtraction&
[0334] average of every 5 points
[0335] averageIR=zeros(length(averageRed_1),1);

```

```

[0336] order=100;
[0337] cutoff=10/(1/period);
[0338] y=fir1(order,cutoff,'low');
[0339] x=filtfilt(y,1,averageRed);
[0340] z=filtfilt(y,1,averageIR);
[0341] [dec,lib]=wavedec(averageRed,2,'db 10');
[0342] a2=wrcoef('a',dec,lib,'db10',2);
[0343] %Perfusion(11+1)=mean(x);
[0344] %%end
[0345] %%End of Loop
[0346] %%Pre-LPF for interpolation
[0347] %%order=100;
[0348] %%cutoff1=40/(1/period);
[0349] %%y1=fir1(order,cutoff1,'low');
[0350] %%x1=filtfilt(y1,1,averageRed);
[0351] %%z1=filtfilt(y1,1,averageIR);
[0352] %
[0353] %%freqz(y)%view filter
[0354] %%
[0355] numavg=100;
[0356] runavg=ones(1,numavg)/numavg;
[0357] x_avg=filtfilt(runavg,1,averageRed);
[0358] z_avg=filtfilt(runavg,1,averageIR);
[0359] %x=x-x_avg;
[0360] %z=z-z_avg;
[0361] time=(1:No_RIR_Waves)/(No_RIR_Waves)*totalTime;
[0362] %-----%
[0363] %Red LED
[0364] %-----%
[0365] figure;
[0366] subplot(2,1,1)
[0367] hold on;
[0368] plot(time,averageRed*1E3,'-k','linewidth',2);
[0369] plot(time,x*1E3,'-r','linewidth',2);
[0370] plot(time,x_avg*1E3,'-b','linewidth',2);
[0371] hold off;
[0372] ylabel('Recived Signal[mV]','fontsize',14,'fontweight','bold')
[0373] xlabel('Time[s]','fontsize',14,'fontweight','bold')
[0374] set(gca,'linewidth',2,'fontsize',10,'fontweight','bold')

```

```
[0375] legend('Red LED','Red LED(LPF)','Running Average',
[0376] 'Orientation','horizontal')
[0377] title('Red LED','fontsize',14,'fontweight','bold')
[0378] box on;
[0379] heart_beat_RED=x-x_avg;
[0380] wavelet_RED=a2-smooth(a2,200);
[0381] %heart_beat_RED=wavelet_RED;
[0382] %%Detect Heat Beat Peaks FAIL 202C VERSION
[0383] %temp=sign(diff(heart_beat_RED));
[0384] %%temp=sign(diff(x(order+numavg/2:end-numavg/2-1)));
[0385] %temp2=(temp(1:end-1)-temp(2:end))./2;
[0386] %loc=find(temp2~=0);
[0387] %loc=[loc(1);loc(find(diff(loc)>MIN_SAMP/2)+1)];
[0388] %peaks1=loc(find(temp2(loc)>0))+1;
[0389] %peaks1=peaks1(find(heart_beat_RED(peaks1)>0));
[0390] %valleys1=loc(find(temp2(loc)<0))+1;
[0391] %valleys1=valleys1(find(heart_beat_RED(valleys1)<0));
[0392] %peak detection that actually works:
[0393] peaks=[];
[0394] widthp=50;
[0395] for j=1:totalTime/period
[0396] if heart_beat_RED(j)==max(heart_beat_RED(max(1,jwidthp):
[0397] min(totalTime/period,j+widthp)))
[0398] peaks(end+1)=j;
[0399] end
[0400] end
[0401] valleys=[];
[0402] widthv=50;
[0403] for j=1:totalTime/period
[0404] if heart_beat_RED(j)==min(heart_beat_RED(max(1,jwidthv):
[0405] min(totalTime/period,j+widthv)))
[0406] valleys(end+1)=j;
[0407] end
[0408] end
[0409] diffzs=[];
[0410] widthd=25;
[0411] diff_hb=diff(heart_beat_RED);
[0412] for j=1:totalTime/period-1
[0413] if abs(diff_hb(j))==min(abs(diff_hb(max(1,jwidthd):
```

```
[0414] min(totalTime/period-1,j+widthd)))
[0415] diffzs(end+1)=j;
[0416] end
[0417] end
[0418] killthese=[];
[0419] for j=1:numel(diffzs)
[0420] for k=1:numel(peaks)
[0421] if abs(diffzs(j)-peaks(k))<25
[0422] killthese(end+1)=j;
[0423] end
[0424] for k=1:numel(valleys)
[0425] if abs(diffzs(j)-valleys(k))<25
[0426] killthese(end+1)=j;
[0427] end
[0428] end
[0429] end
[0430] peakspacing(j)=min(abs(diffzs(j)-peaks));
[0431] valleypacing(j)=min(abs(diffzs(j)-valleys));
[0432] end
[0433] diffzs(killthese)=[];
[0434] peakspacing(killthese)=[];
[0435] %clean up peaks/valleys to make them match 1:1
[0436] delp=[];
[0437] for i=1:length(peaks)-1
[0438] valid=0;
[0439] for j=1:length(valleys)
[0440] if peaks(i+1)>valleys(j)&&peaks(i)<valleys(j)
[0441] valid=1;
[0442] break
[0443] end
[0444] end
[0445] if valid==0&&heart_beat_RED(peaks(i+1))<heart_beat_RED(peaks(i))
[0446] delp(end+1)=i+1;
[0447] elseifvalid==0
[0448] delp(end+1)=i;
[0449] end
[0450] end
[0451] peaks(delp)=[];
[0452] delv=[];
```

```
[0453] for i=1:length(valleys)-1
[0454] valid=0;
[0455] forj=1:length(peaks)
[0456] if valleys(i+1)>peaks(j)&&valleys(i)<peaks(j)
[0457] valid=1;
[0458] break
[0459] end
[0460] end
[0461] ifvalid==0&&
[0462] heart_beat_RED(valleys(i+1))>heart_beat_RED(valleys(i))
[0463] delv(end+1)=i+1;
[0464] elseif valid==0
[0465] delv(end+1)=i;
[0466] end
[0467] end
[0468] valleys(delv)=[];
[0469] %finish of cleanup
[0470] mdiffzs=median(heart_beat_RED(diffzs));
[0471] mpeaks=median(heart_beat_RED(peaks));
[0472] mvalleys=median(heart_beat_RED(valleys));
[0473] secondpeak=(mdiffzs-mvalleys)/(mpeaks-mvalleys);
[0474] peakspacing=median(peakspacing);
[0475] valleyspacing=median(valleyspacing);
[0476] subplot(2,1,2)
[0477] hold on;
[0478] plot(time,heart_beat_RED*1E3,'-k','linewidth',2);
[0479] %ylim([-1.51.5])
[0480] plot(time(peaks),heart_beat_RED(peaks)*1E3,'or','linewidth',2,'
markersize',12);
[0481] plot(time(valleys),heart_beat_RED(valleys)*1E3,'ob','linewidth',2,'
markersize',12);
[0482] plot(time(diffzs),heart_beat_RED(diffzs)*1E3,'og','linewidth',2,'
markersize',12);
[0483] hold off;
[0484] ylabel('Heart Beat[mV]','fontsize',14,'fontweight','bold')
[0485] xlabel('Time[s]','fontsize',14,'fontweight','bold')
[0486] set(gca,'linewidth',2,'fontsize',10,'fontweight','bold')
[0487] box on;
[0488] Heart_Rate_RED=length(peaks)/(time(end)-time(1))*60;
```

```
[0489] %%-----%
[0490] %%1R LED
[0491] %%-----%
[0492] %figure;
[0493] %subplot(2,1,1)
[0494] %hold on;
[0495] %plot(time,averagelR*1E3,'-k','linewidth',2);
[0496] %plot(time,z*1E3,'-r','linewidth',2);
[0497] %plot(time,z_avg*1E3,'-b','linewidth',2);
[0498] %hold off;
[0499] %ylabel('Recived Signal[mV]','fontsize',14,'fontweight','bold')
[0500] %xlabel('Time[s]','fontsize',14,'fontweight','bold')
[0501] %set(gca,'linewidth',2,'fontsize',10,'fontweight','bold')
[0502] %legend('1R LED','1R LED(LPF)','Running Average',
[0503] 'Orientation','horizontal')
[0504] %title('1R LED','fontsize',14,'fontweight','bold')
[0505] %box on;
[0506] %
[0507] %heart_beat_1R=z-z_avg;
[0508] %
[0509] %%Detect Heat Beat Peaks
[0510] %temp=sign(diff(heart_beat_1R));
[0511] %%temp=sign(diff(z(order+numavg/2:end-numavg/2-1)));
[0512] %temp2=(temp(1:end-1)-temp(2:end))./2;
[0513] %loc=find(temp2~=0);
[0514] %loc=[loc(1);loc(find(diff(loc)>MIN_SAMP/2)+1)];
[0515] %peaks2=loc(find(temp2(loc)>0))+1;
[0516] %peaks2=peaks2(find(heart_beat_1R(peaks2)>0));
[0517] %valleys2=loc(find(temp2(loc)<0))+1;
[0518] %valleys2=valleys2(find(heart_beat_1R(valleys2)<0));
[0519] %
[0520] %subplot(2,1,2)
[0521] %hold on;
[0522] %plot(time,heart_beat_1R*1E3,'-k','linewidth',2);
[0523] %ylim([-1.51.5]);
[0524] %plot(time(peaks2),heart_beat_1R(peaks2)*1 E3,'or','linewidth',2,'
markersize',12);
[0525] %plot(time(valleys2),heart_beat_1R(valleys2)*1E3,'ob','linewidth',
2,'markersize',12);
```

```

[0526] %hold off;
[0527] %ylabel('Heart Beat[mV]','fontsize',14,'fontweight','bold')
[0528] %xlabel('Time[s]','fontsize',14,'fontweight','bold')
[0529] %set(gca,'linewidth',2,'fontsize',10,'fontweight','bold')
[0530] %box on;
[0531] %Heart_Rate_1R=length(peaks2)/(time(end)-time(1))*60
[0532] %%-----%
[0533] %%SpO2
[0534] %%-----%
[0535] %H_heart_beat_Red_peak=
[0536] interp1(peaks1,x(peaks1),1:length(time),'spline');%Interpolate the
peak value of heart beat(RED)for whole time range
[0537] %H_heart_beat_1R_peak=
[0538] interp1(peaks2,z(peaks2),1:length(time),'spline');%Interpolate the
peak value of heart beat(1R)for whole time range
[0539] %
[0540] %H_heart_beat_Red_valley=
[0541] interp1(valleys1,x(valleys1),1:length(time),'spline');%Interpolate
the valley value of heart beat(RED)for whole time range
[0542] %H_heart_beat_1R_valley=
[0543] interp1(valleys2,z(valleys2),1:length(time),'spline');%Interpolate
the valley value of heart beat(1R)for whole time range
[0544] %
[0545] %%Superposition
[0546] %x2=zeros(length(x1),1);
[0547] %z2=zeros(length(z1),1);
[0548] %for i=2:length(peaks1)-1
[0549] %x2(1:end-(peaks1(i)-peaks1(2)))=x2(1:end-(peaks1(i)-peaks1(2)))
[0550] +x1(peaks1(i)-peaks1(2)+1:end);
[0551] %z2(1:end-(peaks2(i)-peaks2(2)))=z2(1:end-(peaks2(i)-peaks2(2)))
[0552] +z1(peaks2(i)-peaks2(2)+1:end);
[0553] %end
[0554] %x2=x2/(length(peaks1)-2);
[0555] %z2=z2/(length(peaks2)-2);
[0556] %
[0557] %%H_heart_beat_Red=filtfilt(runavg,1,H_heart_beat_Red);
[0558] %%H_heart_beat_1R=filtfilt(runavg,1,H_heart_beat_1R);
[0559] %
[0560] %

```

```
[0561] %R_red=H_heart_beat_Red_valley./(H_heart_beat_Red_peak);
[0562] %R_IR=H_heart_beat_IR_valley./(H_heart_beat_IR_peak);
[0563] %
[0564] %R=(log(R_red)./log(R_IR))*(RED_sens/IR_sens);
[0565] %O2=(0.81-0.18.*R)./(0.63+0.11.*R)*100;
[0566] %SpO2=mean(O2)
[0567] %
[0568] %figure;
[0569] %hold on;
[0570] %plot(time,O2,'-r','linewidth',2);
[0571] %ylabel('SpO2','fontsize',14,'fontweight','bold')
[0572] %xlabel('Time[s]','fontsize',14,'fontweight','bold')
[0573] %set(gca,'linewidth',2,'fontsize',10,'fontweight','bold')
[0574] %ylim([90 110])
[0575] %box on;
[0576] x=[];
[0577] hrdata=[];
[0578] pdiff=[];
[0579] secpeak=[];
[0580] trial=1;
[0581] for trial=1:1
[0582] for filenum=1:1
[0583] for sensorselect=4
[0584] inputfile=['ir'+num2str(min(trial,2))'.'num2str(filenum)];
[0585] inputfile='all+';
[0586] %inputfile=['height\5s_stoy'+num2str(filenum)];
[0587] multilevel_extract;
[0588] hrdata(:,filenum)=heart_beat_RED;
[0589] dcdata(filenum)=median(x_avg);
[0590] %ifnanz(x(:,filenum))==0;break;end
[0591] r(filenum)=Heart_Rate_RED;
[0592] vs=min(numel(peaks),numel(valleys));
[0593] p2pdata(filenum)=median(heart_beat_RED(peaks(1:vs))heart_beat_RED
(valleys(1:vs)));
[0594] en=[];
[0595] for i=2:numel(valleys)-2
[0596] en(end+1)=sum(heart_beat_RED(valleys(i):valleys(i+1)).^2);
[0597] end
[0598] benergy(filenum)=median(en);
```

```
[0599] riset=[];
[0600] fallt=[];
[0601] if peaks(1)>valleys(1)
[0602] for i=1:vs-1
[0603] riset(end+1)=peaks(i)-valleys(i);
[0604] fallt(end+1)=valleys(i+1)-peaks(i);
[0605] end
[0606] else
[0607] for i=1:vs-1
[0608] riset(end+1)=peaks(i+1)-valleys(i);
[0609] fallt(end+1)=valleys(i)-peaks(i);
[0610] end
[0611] end
[0612] risetime(filenum)=median(riset);
[0613] falltime(filenum)=median(fallt);
[0614] for repeat=1:3
[0615] if peaks(1)<valleys(1);peaks(1)=[];end
[0616] end
[0617] for i=1:floor(numel(peaks)/2)
[0618] list_pdiff(i)=heart_beat_RED(peaks(2*i-1))-heart_beat_RED(peaks(2*
i));
[0619] end
[0620] pdiff(filenum)=median(list_pdiff);
[0621] secpeak(filenum)=secondpeak;
[0622] peakspace(filenum)=peakspacing;
[0623] valspace(filenum)=valleyspacing;
[0624] medpeak(filenum)=mpeaks-mvalleys;
[0625] end
[0626] %suffix='.pressure';
[0627] %presf=csvread([inputfile suffix]);
[0628] %presdata(filenum)=mean((presf(:,2)-.6)/2.8);
[0629] end
[0630] stoyrt(trial,:)=risetime*.005;
[0631] stoyft(trial,:)=falltime*.005;
[0632] stoyhr(trial,:)=r;
[0633] stoysecpeak(trial,:)=secpeak;
[0634] stoypeakspace(trial,:)=peakspace*.005;
[0635] stoyvalspace(trial,:)=valspace*.005;
[0636] stoymp(trial,:)=medpeak;
```

```
[0637] end
[0638] %stoyfts=stoyft./(min(stoyft')'*[1 1 1 1 1]);
[0639] %stoyrts=stoyrt./(min(stoyrt')'*[1 1 1 1 1]);
[0640] %stoysecpeaks=stoysecpeak./(min(stoysecpeak')'*[1 1 1 1 1]);
[0641] %stoymps=stoymp./(min(stoymp')'*[1 1 1 1 1]);
[0642] %
[0643] %
[0644] %for i=1:3;corrcoef(stoyhr(i,:),stoyrt(i,:))
[0645] %end
[0646] %for i=1:3;corrcoef(stoyhr(i,:),stoyft(i,:))
[0647] %end
[0648] %for i=1:3;corrcoef(stoyhr(i,:),stoysecpeak(i,:))
[0649] %end
[0650] %
[0651] %for i=1:3;corrcoef(stoybps(i,:),stoyrt(i,:))
[0652] %end
[0653] %for i=1:3;corrcoef(stoybps(i,:),stoyft(i,:))
[0654] %end
[0655] %for i=1:3;corrcoef(stoybps(i,:),stoysecpeak(i,:))
[0656] %end
[0657] %for i=1:3;corrcoef(stoybps(i,:),stoyhr(i,:))
[0658] %end
[0659] %
[0660] %for i=1:3;corrcoef(stoybpd(i,:),stoyrt(i,:))
[0661] %end
[0662] %for i=1:3;corrcoef(stoybpd(i,:),stoyft(i,:))
[0663] %end
[0664] %for i=1:3;corrcoef(stoybpd(i,:),stoysecpeak(i,:))
[0665] %end
[0666] %for i=1:3;corrcoef(stoybpd(i,:),stoyhr(i,:))
[0667] %end
[0668] %
[0669] %peaks=[];
[0670] %forj=1:4000
[0671] %if x(j,filenum)>5e-5&&x(j,filenum)==max(x(max(1,j75):
[0672] min(4000,j+75),filenum))
[0673] %peaks(end+1)=j;
[0674] %end
[0675] %end
```

```
[0676] %
[0677] %
[0678] %
[0679] %forj=1:4000
[0680] %if heart_beat_RED(j)>5e-5&&
[0681] heart_beat_RED(j)==max(heart_beat_RED(max(1,j-75):min(4000,j+75)))
[0682] %peaks(end+1)=j;
[0683] %end
[0684] %end
[0685] %t=1:4
[0686] %figure
[0687] %plot(t,stoy1bpd,'o',t,stoy2bpd,'o',t,stoy3bpd,'o')
[0688] %axis([.5 4.5 -1 1])
[0689] %set(gca,'XTick',1:4)
[0690] %set(gca,'XTickLabel',{'Rise Time''Fall Time''Second Peak Strength''
Heart Rate'})
[0691] %legend({'Trial 1''Trial 2''Trial 3'})
[0692] %title('Correlations:Metrics vs.Diastolic Blood Pressure,Henrik')
[0693] %ylabel('Correlation Coefficient')
[0694] %figure
[0695] %plot(t,stoy1bps,'o',t,stoy2bps,'o',t,stoy3bps,'o')
[0696] %axis([.5 4.5 -1 1])
[0697] %set(gca,'XTick',1:4)
[0698] %set(gca,'XTickLabel',{'Rise Time''Fall Time''Second Peak Strength''
Heart Rate'})
[0699] %legend({'Trial 1''Trial 2''Trial 3'})
[0700] %title('Correlations:Metrics vs.Systolic Blood Pressure,Henrik')
[0701] %ylabel('Correlation Coefficient')
[0702] %figure
[0703] %plot(t,stoy1hr,'o',t,stoy2hr,'o',t,stoy3hr,'o')
[0704] %axis([.5 4.5 -1 1])
[0705] %set(gca,'XTick',1:4)
[0706] %set(gca,'XTickLabel',{'Rise Time''Fall Time''Second Peak Strength''
Heart Rate'})
[0707] %legend({'Trial 1''Trial 2''Trial 3'})
[0708] %title('Correlations:Metrics vs.Heart Rate,Henrik')
[0709] %ylabel('Correlation Coefficient')
[0710] function[pointcoords]=rgbfind(filename)
[0711] im_unfiltered=imread(filename);%[y x rgb]
```

```
[0712] %h=fspecial('gaussian',10,10);
[0713] %im=imfilter(im_unfiltered,h);
[0714] im=im_unfiltered;
[0715] r=im(:,:,1);
[0716] g=im(:,:,2);
[0717] b=im(:,:,3);
[0718] %image(im);
[0719] %goal rgb=0,160,170
[0720] goalr=0;
[0721] goalg=160;
[0722] goalb=170;
[0723] tol=50;%goal offset tolerance
[0724] match=zeros(size(im,1),size(im,2),2);
[0725] for y=1:size(im,1)
[0726] for x=1:size(im,2)
[0727] if(r(y,x)>goalr+tol)||r(y,x)<goalr-tol)...
[0728] ||(g(y,x)>goalg+tol)||g(y,x)<goalg-tol)...
[0729] ||(b(y,x)>goalb+tol)||b(y,x)<goalb-tol)
[0730] %not a match
[0731] %match(y,x,:)= [0,0,0];
[0732] else
[0733] %match
[0734] match(y,x,:)= [1,0];
[0735] end
[0736] end
[0737] end
[0738] numblobs=0;
[0739] blob=[];
[0740] for y=1:size(im,1)
[0741] for x=1:size(im,2)
[0742] if match(y,x,1)==1
[0743] %these matches are already in blobs
[0744] if match(y-1,x+2,1)==1
[0745] match(y,x,2)=match(y-1,x+2,2);
[0746] blob(match(y-1,x+2,2)).x(end+1)=x;
[0747] blob(match(y-1,x+2,2)).y(end+1)=y;
[0748] elseif match(y-1,x+1,1)==1
[0749] match(y,x,2)=match(y-1,x+1,2);
[0750] blob(match(y-1,x+1,2)).x(end+1)=x;
```

```
[0751] blob(match(y-1,x+1,2)).y(end+1)=y;
[0752] elseif match(y-1,x,1)==1
[0753] match(y,x,2)=match(y-1,x,2);
[0754] blob(match(y-1,x,2)).x(end+1)=x;
[0755] blob(match(y-1,x,2)).y(end+1)=y;
[0756] elseif match(y-1,x-1,1)==1
[0757] match(y,x,2)=match(y-1,x-1,2);
[0758] blob(match(y-1,x-1,2)).x(end+1)=x;
[0759] blob(match(y-1,x-1,2)).y(end+1)=y;
[0760] elseif match(y,x-1,1)==1
[0761] match(y,x,2)=match(y,x-1,2);
[0762] blob(match(y,x-1,2)).x(end+1)=x;
[0763] blob(match(y,x-1,2)).y(end+1)=y;
[0764] %other matches require new blob
[0765] else%if match(y+1,x-1,1)==1
[0766] numblobs=numblobs+1;
[0767] match(y,x,2)=numblobs;
[0768] blob(numblobs).x=x;
[0769] blob(numblobs).y=y;
[0770] end
[0771] end
[0772] end
[0773] end
[0774] merged=zeros(1,numblobs);
[0775] figure();image(match(:,:,2)+1);
[0776] for y=size(im,1):-1:1
[0777] for x=size(im,2):-1:1
[0778] ifmatch(y,x,1)==1
[0779] %these matches are already in blobs
[0780] if(match(y,x+1,1)==1)&&(match(y,x,2)~=match(y,x+1,2))
[0781] merged(match(y,x,2))=match(y,x+1,2);
[0782] match(y,x,2)=match(y,x+1,2);
[0783] blob(match(y,x+1,2)).x(end+1)=x;
[0784] blob(match(y,x+1,2)).y(end+1)=y;
[0785] elseif match(y+1,x+1,1)==1&&match(y,x,2)~=match(y+1,x+1,2)
[0786] merged(match(y,x,2))=match(y+1,x+1,2);
[0787] match(y,x,2)=match(y+1,x+1,2);
[0788] blob(match(y+1,x+1,2)).x(end+1)=x;
[0789] blob(match(y+1,x+1,2)).y(end+1)=y;
```

```
[0790] elseif match(y+1,x,1)==1&&match(y,x,2)~=match(y+1,x,2)
[0791] merged(match(y,x,2))==match(y+1,x,2);
[0792] match(y,x,2)=match(y+1,x,2);
[0793] blob(match(y+1,x,2)).x(end+1)=x;
[0794] blob(match(y+1,x,2)).y(end+1)=y;
[0795] elseif match(y+1,x-1,1)==1&&match(y,x,2)~=match(y+1,x-1,2)
[0796] merged(match(y,x,2))==match(y+1,x-1,2);
[0797] match(y,x,2)=match(y+1,x-1,2);
[0798] blob(match(y+1,x-1,2)).x(end+1)=x;
[0799] blob(match(y+1,x-1,2)).y(end+1)=y;
[0800] end
[0801] end
[0802] end
[0803] end
[0804] for y=size(im,1):-1:1
[0805] for x=size(im,2):-1:1
[0806] if match(y,x,1)==1
[0807] if merged(match(y,x,2))>0
[0808] while merged(match(y,x,2))>0
[0809] match(y,x,2)=merged(match(y,x,2));
[0810] blob(match(y,x,2)).x(end+1)=x;
[0811] blob(match(y,x,2)).y(end+1)=y;
[0812] end;end;end;end;end
[0813] blob(find(merged))=[];
[0814] pointcoords=[];
[0815] for i=1:size(blob,2)
[0816] pointcoords(i,:)=[mean(blob(i).y);mean(blob(i).x)];
[0817] end
[0818] pointcoords=round(pointcoords);
[0819] figure();imshow(match(:,:,1));
[0820] figure();image(match(:,:,2)+1);
[0821] %+(match(:,:,2)>0)*3
[0822] end
[0823] function[exppic]=imoverlay(pcs,im,impic)
[0824] p1=pcs(1,:);
[0825] p2=pcs(2,:);
[0826] p3=pcs(3,:);
[0827] d1=p1(1)-p1(2);
[0828] d2=p2(1)-p2(2);
```

```
[0829] d3=p3(1)-p3(2);
[0830] s1=p1(1)+p1(2);
[0831] s2=p2(1)+p2(2);
[0832] s3=p3(1)+p3(2);
[0833] [a,v]=max([d1 d2 d3]);
[0834] [a,t]=min([s1 s2 s3]);
[0835] [a,r]=max([s1 s2 s3]);
[0836] %hyp=sqrt((pcs(v,1)-pcs(t,1))^2+(pcs(v,2)-pcs(t,2))^2);
[0837] %adj=sqrt((pcs(v,1)-pcs(r,1))^2+(pcs(v,2)-pcs(r,2))^2);
[0838] %angle=atand(adj/hyp);
[0839] ratio=(pcs(v,1)-pcs(t,1))/(pcs(t,2)-pcs(v,2));
[0840] angle=atand(ratio);
[0841] hangle=-1*(90-angle);
[0842] hoffset=(pcs(r,1)-pcs(t,1)-(pcs(t,2)-pcs(r,2))*tand(angle))*cosd
(angle);
[0843] scale=hoffset/size(im,2);
[0844] imout=imresize(im,scale);
[0845] padout=ones(size(imout));
[0846] padout=imrotate(padout,hangle);
[0847] imout=imrotate(imout,hangle);
[0848] sp=[0 0];
[0849] if hangle<0
[0850] for x=1:size(padout,2)
[0851] for y=size(padout,1):-1:1
[0852] if padout(y,x)==1
[0853] sp=[y x];
[0854] break
[0855] end
[0856] end
[0857] if sp;break;end
[0858] end
[0859] else
[0860] for y=size(padout,1):-1:1
[0861] for x=1:size(padout,2)
[0862] if padout(y,x)==1
[0863] sp=[y x];
[0864] break
[0865] end
[0866] end
```

```
[0867]   if sp;break;end
[0868]   end
[0869]   end
[0870]   offy=pcs(v,1)-sp(1);
[0871]   offx=pcs(v,2)-sp(2);
[0872]   exp=zeros(size(impic));
[0873]   exppic=exp;
[0874]   for y=1:size(padout,1)
[0875]     for x=1:size(padout,2)
[0876]       xcoord=max(1,offx+x);
[0877]       xcoord=min(xcoord,size(exp,2));
[0878]       ycoord=max(1,offy+y);
[0879]       ycoord=min(ycoord,size(exp,1));
[0880]       exp(ycoord,xcoord,:)=padout(y,x,:);
[0881]       exppic(ycoord,xcoord,:)=imout(y,x,:);
[0882]     end
[0883]   end
[0884]   image(impic);
[0885]   hold on
[0886]   hobject=image(exppic/255);
[0887]   hold off
[0888]   set(hobject,'AlphaData',exp(:,:1)/2);
[0889]   end
[0890]   function[imdata]=mapData(filename,ploten)
[0891]   %MAPDATA Summary of this function goes here
[0892]   %Detailed explanation goes here
[0893]   temp=csvread(filename);
[0894]   log_sp02=temp(1,:);
[0895]   log_pressure=temp(2,:);
[0896]   log_x=temp(3,:);
[0897]   log_y=temp(4,:);
[0898]   clear temp;
[0899]   vals=[];
[0900]   log_x=abs(min(log_x))+log_x;
[0901]   log_y=abs(min(log_y))+log_y;
[0902]   i=0;
[0903]   while i<numel(log_sp02)
[0904]     i=i+1;
[0905]     if log_sp02(i)<10
```

```
[0906] log_sp02(i)=[];
[0907] log_pressure(i)=[];
[0908] log_x(i)=[];
[0909] log_y(i)=[];
[0910] end
[0911] end
[0912] %for i=1:size(log_sp02,2)
[0913] grid=zeros(floor((max(log_y))/5)+1,floor((max(log_x))/5)+1);
[0914] [X,Y]=meshgrid(1:5:(max(log_x)),1:5:(max(log_y)));
[0915] while numel(log_sp02)>0
[0916] i=1;
[0917] xmatch=find(log_x==log_x(i));
[0918] ymatch=find(log_y==log_y(i));
[0919] match=intersect(xmatch,ymatch);
[0920] vals(end+1,:)= [log_x(i)log_y(i)max(log_sp02(match))];
[0921] %grid(log_y(i)+1,log_x(i)+1)=max(log_sp02(match));
[0922] log_sp02(match)=[];
[0923] log_pressure(match)=[];
[0924] log_x(match)=[];
[0925] log_y(match)=[];
[0926] end
[0927] %plot(sqrt(vals(:,1).^2+vals(:,2).^2),vals(:,3));
[0928] anisotropy=1;%range x/range y
[0929] alpha=0;%angle between axis/anisotropy in degrees
[0930] nu=1;%nu for covariance
[0931] vgrid=[55];
[0932] [kout evar]=vebyk(vals,vgrid,5,anisotropy,alpha,nu,1,0,0);
[0933] for i=1:size(kout,1)
[0934] if(size(grid,2)-1<kout(i,1)/5)|| (size(grid,1)-1<
[0935] kout(i,2)/5)
[0936] continue;
[0937] end
[0938] grid(kout(i,2)/5+1,kout(i,1)/5+1)=kout(i,3);
[0939] end
[0940] %image(grid);
[0941] imdata=[];
[0942] if ploten
[0943] figure();
[0944] surf(X,Y,grid);
```

```
[0945] else
[0946]   imdat=((grid-min(min(grid)))*255/(max(max(grid))
[0947]   min(min(grid))));
[0948]   rgbdata=ind2rgb(round(imdat),jet(256));
[0949]   imwrite(rgbdata,'d_image.jpg','jpg')
[0950]   imdata=rgbdata;
[0951] end
[0952] end
```

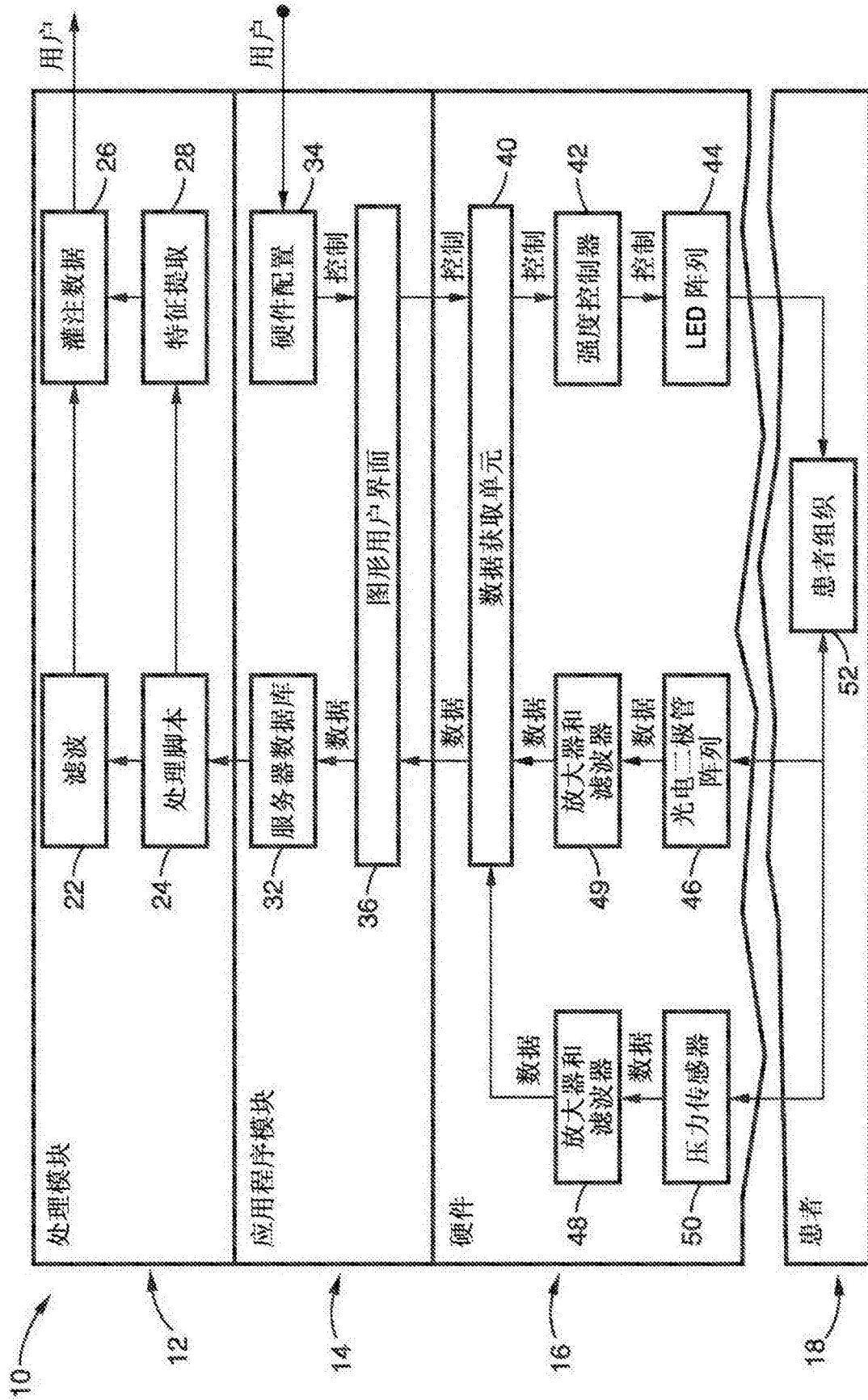


图1

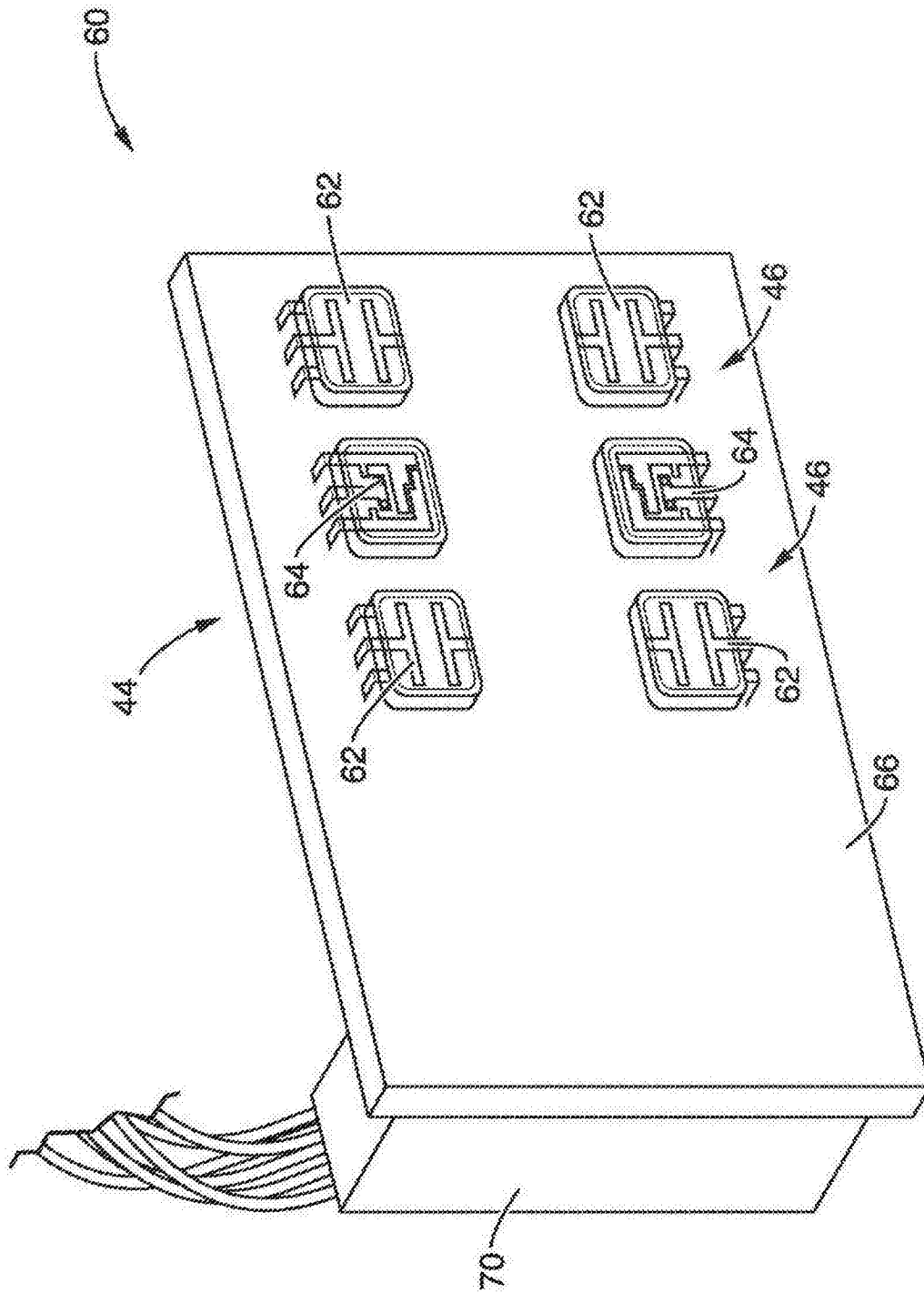


图2A

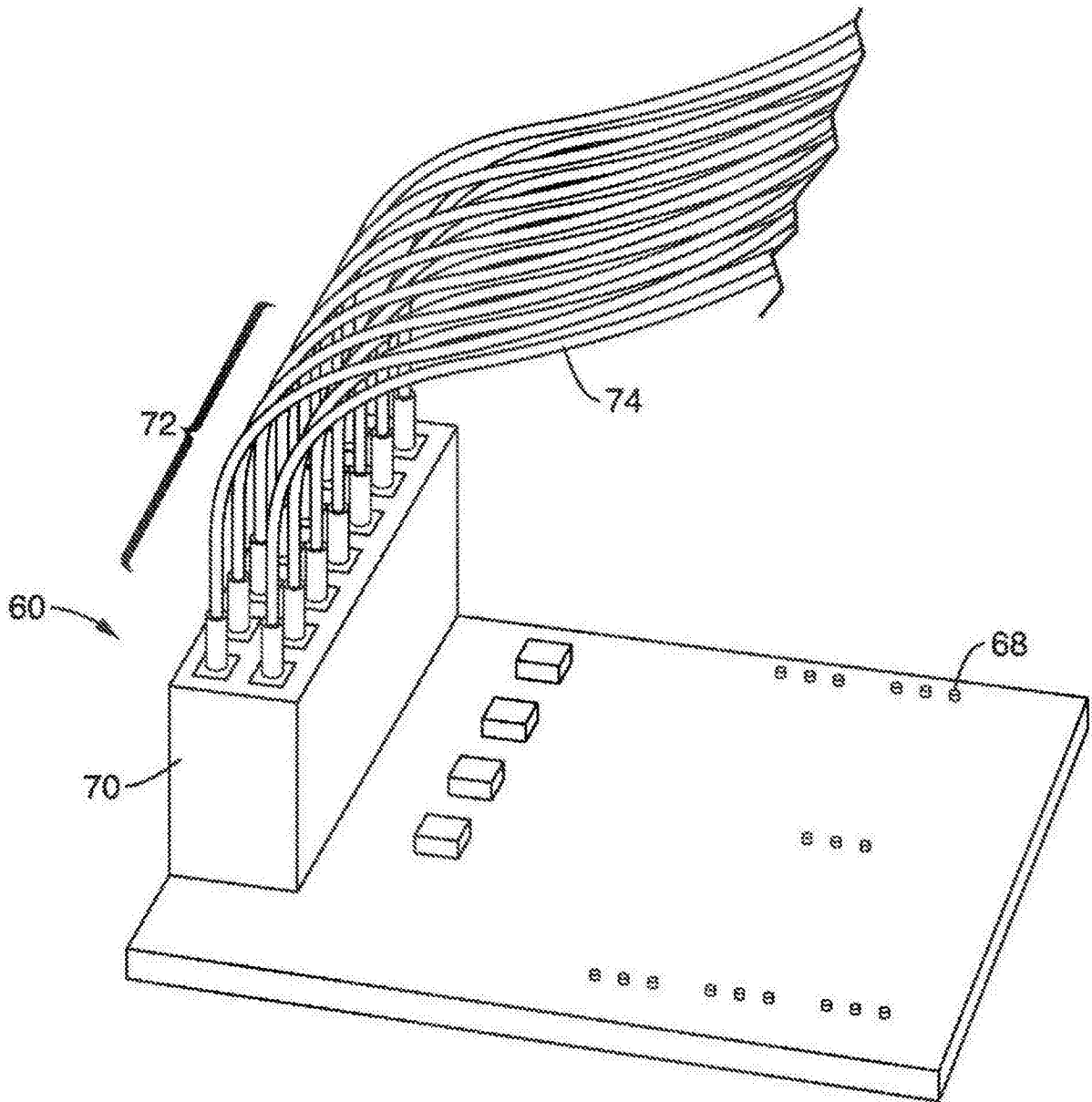


图2B

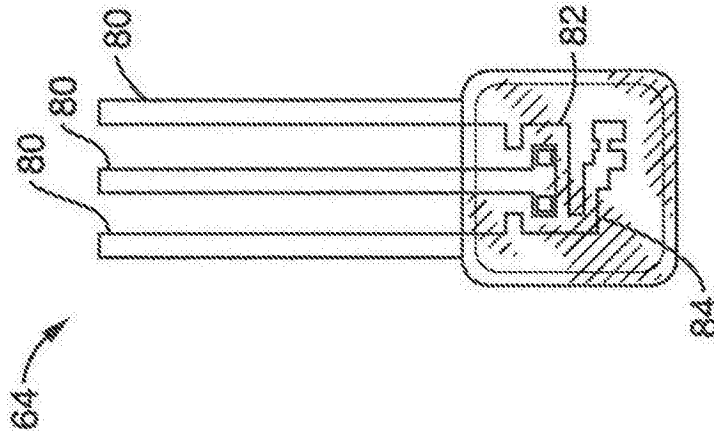


图3

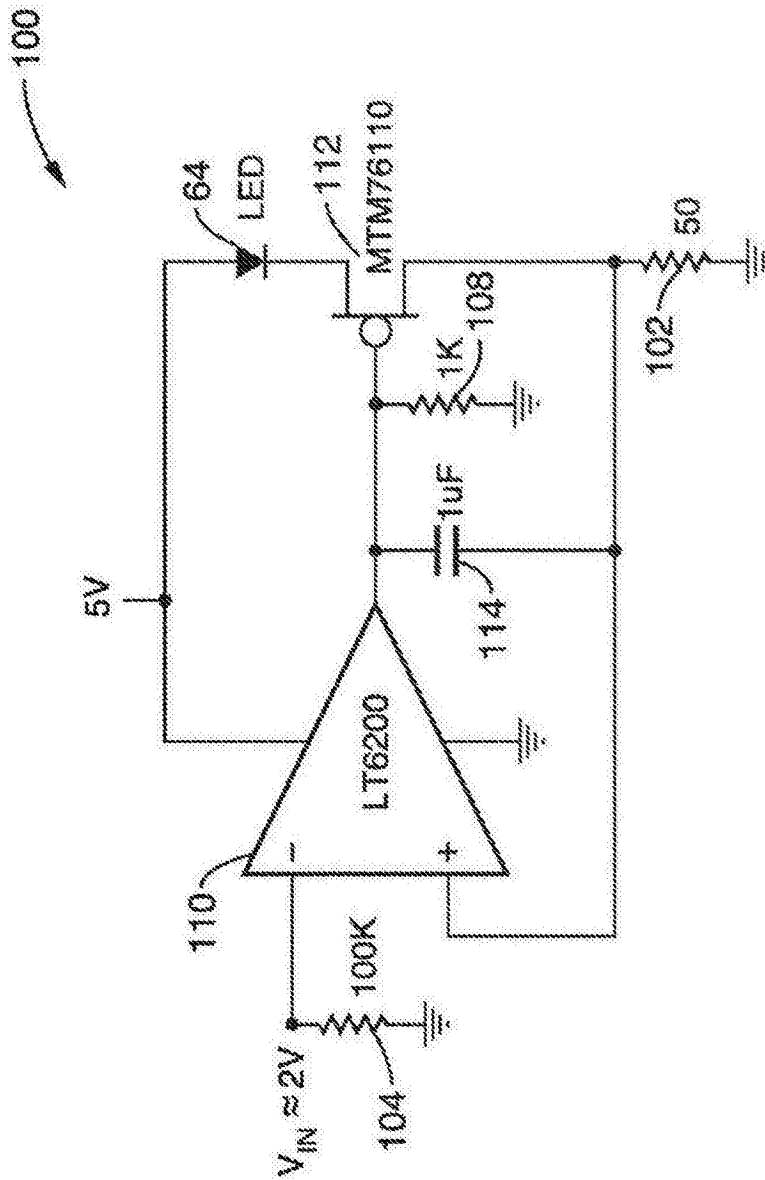


图4

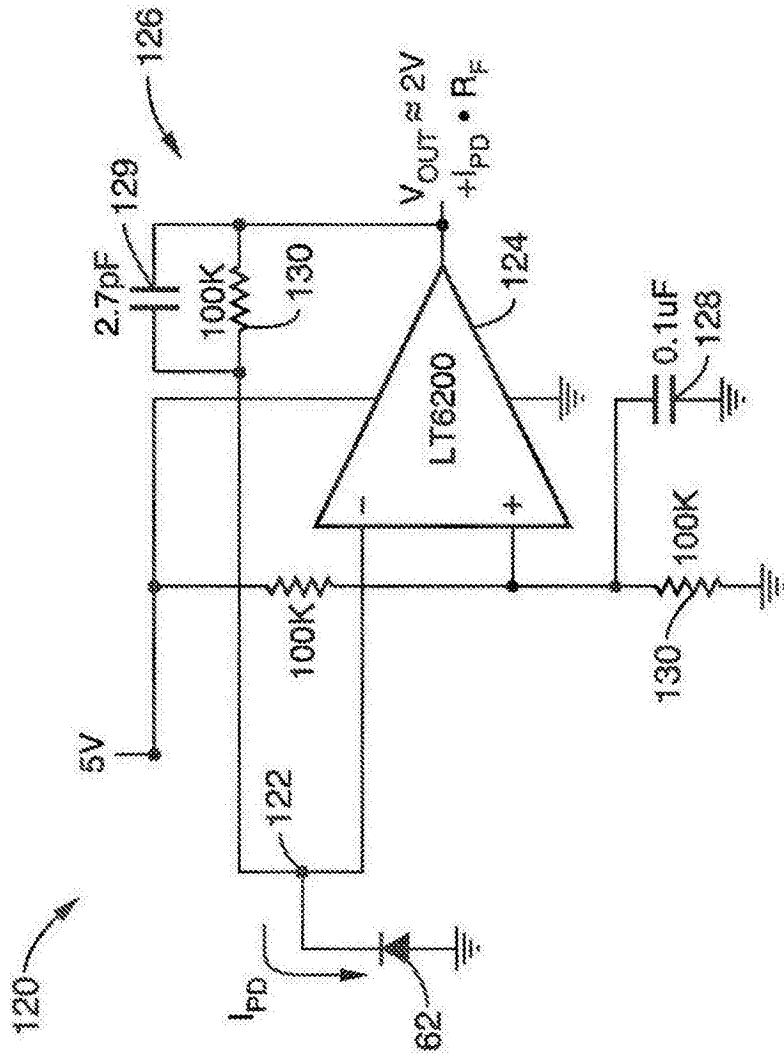


图5

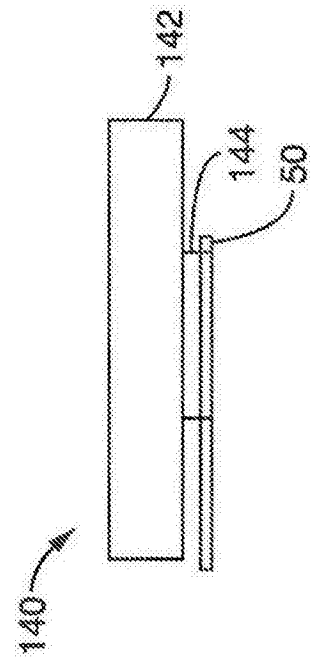


图6

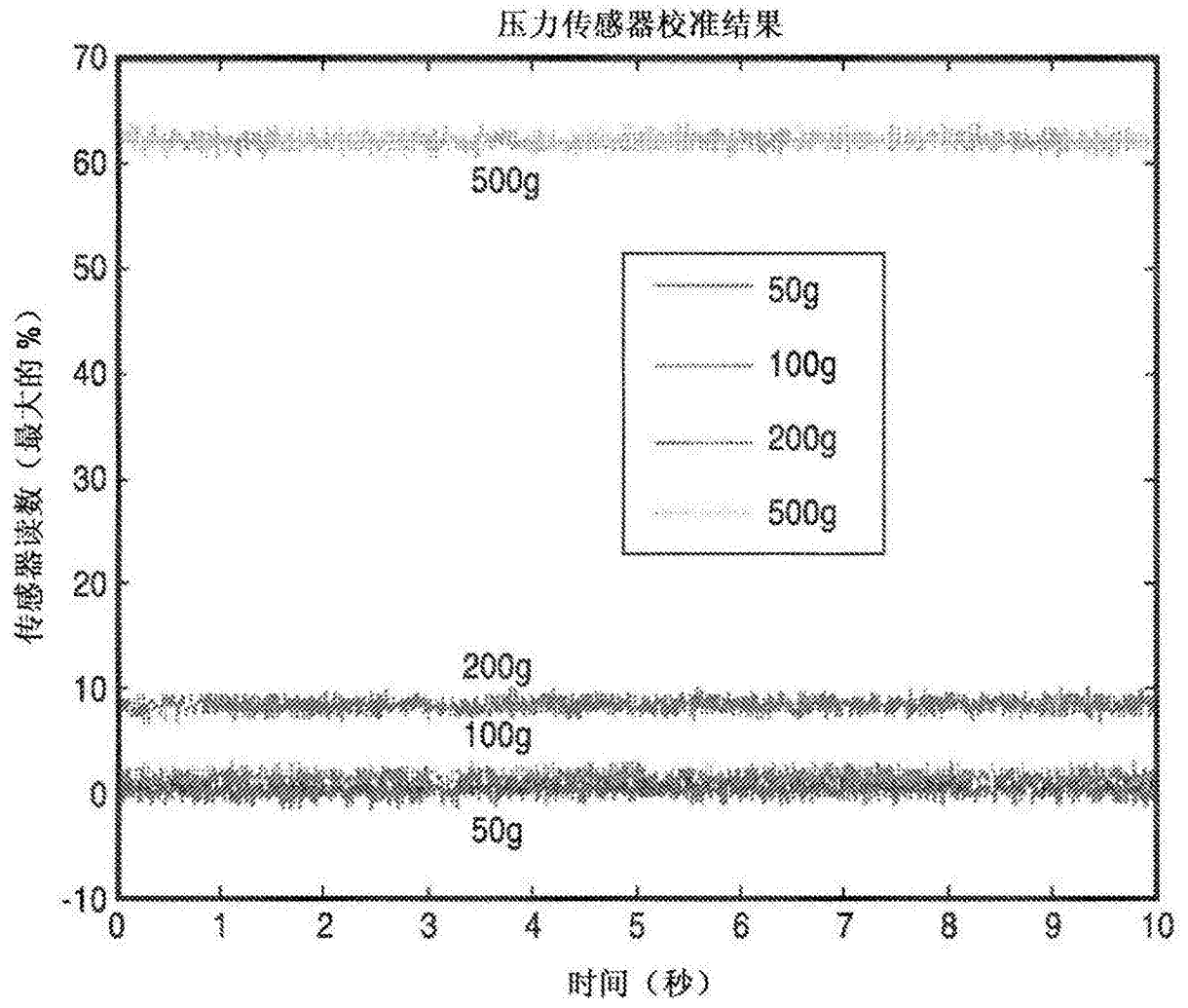


图7

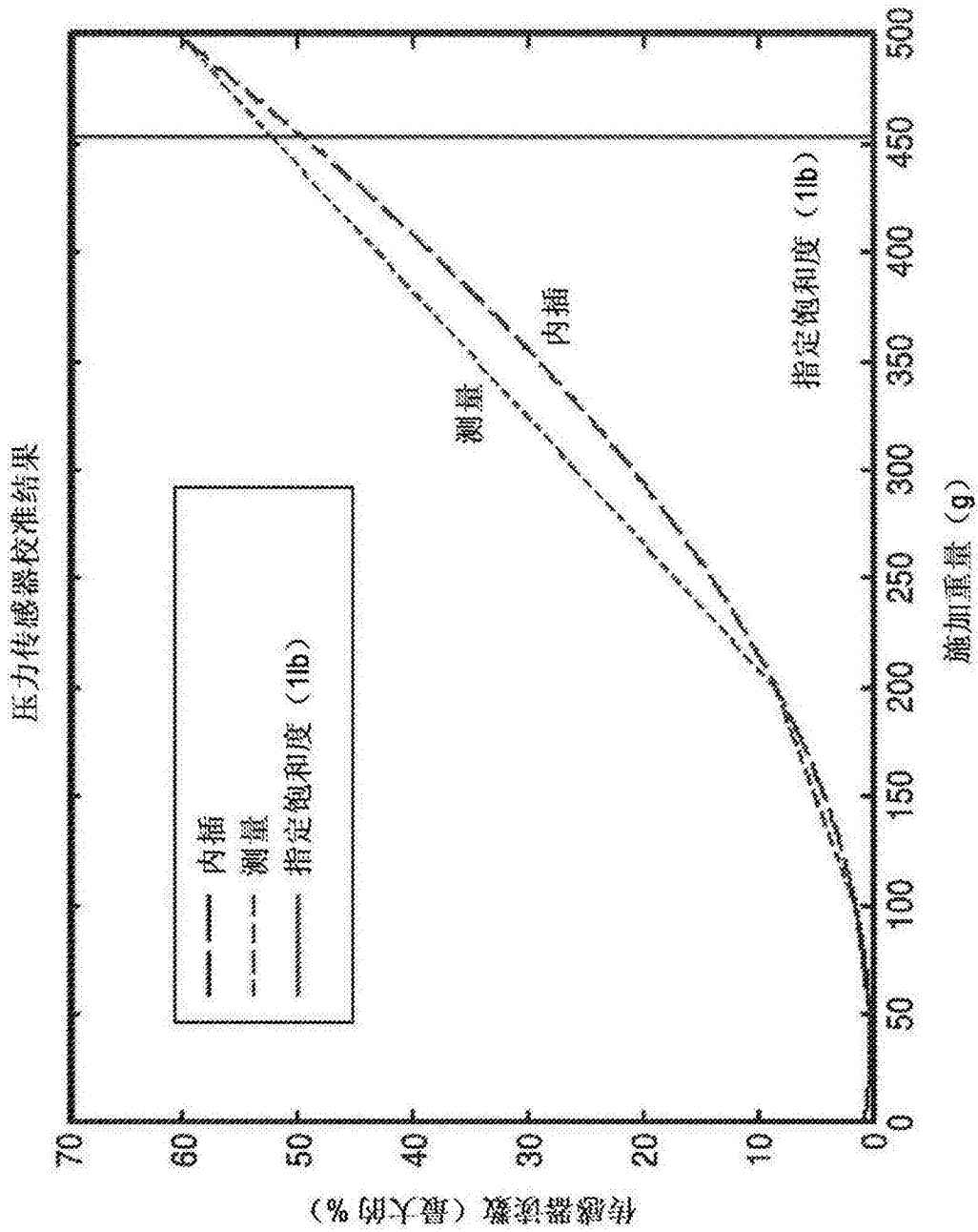


图8

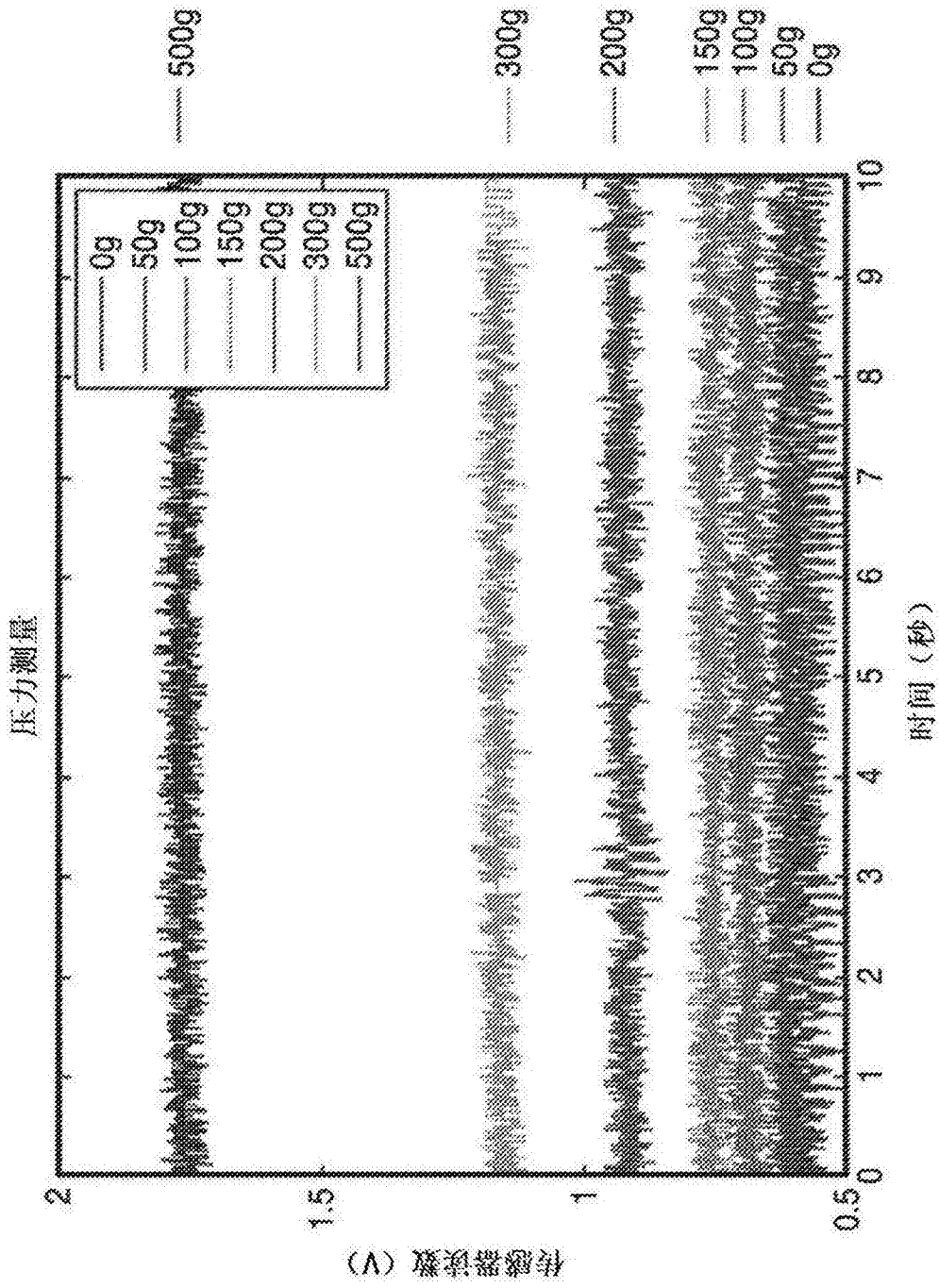


图9

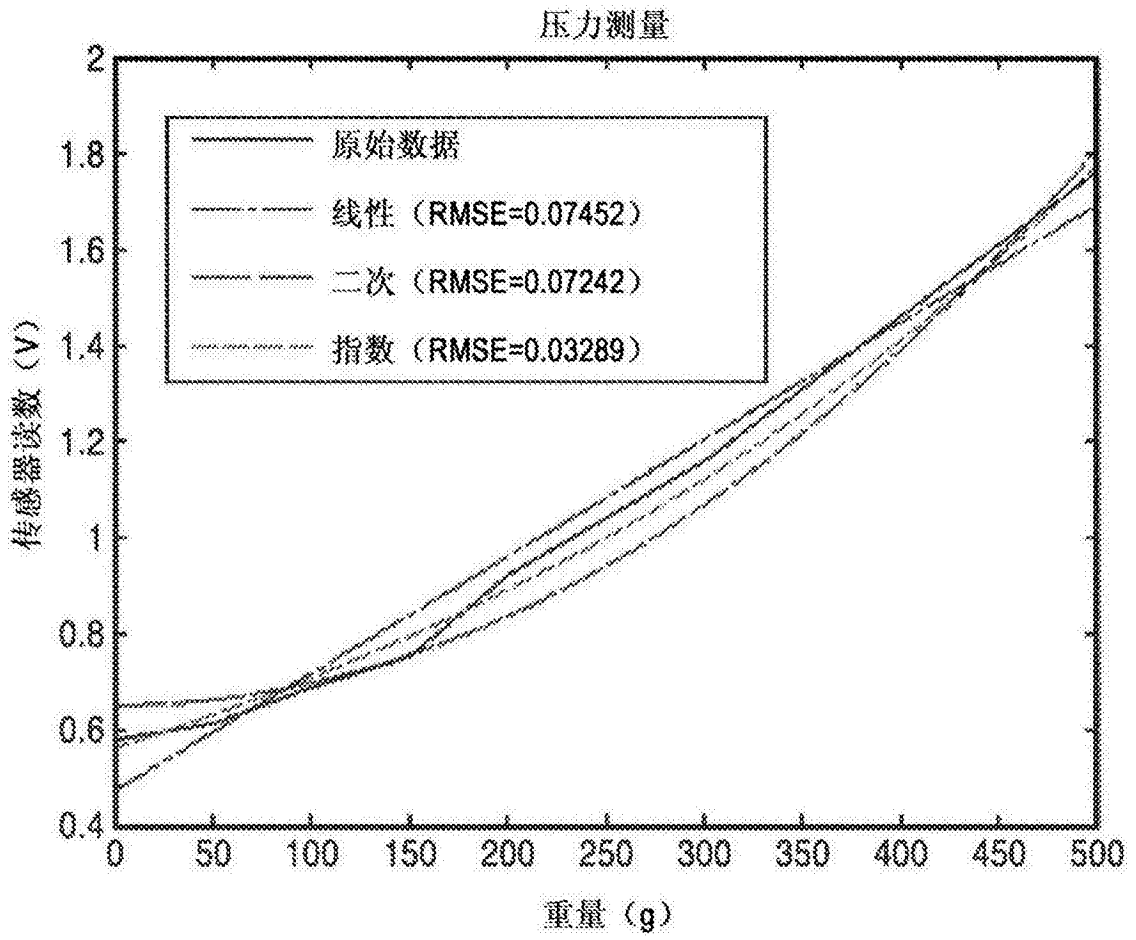


图10

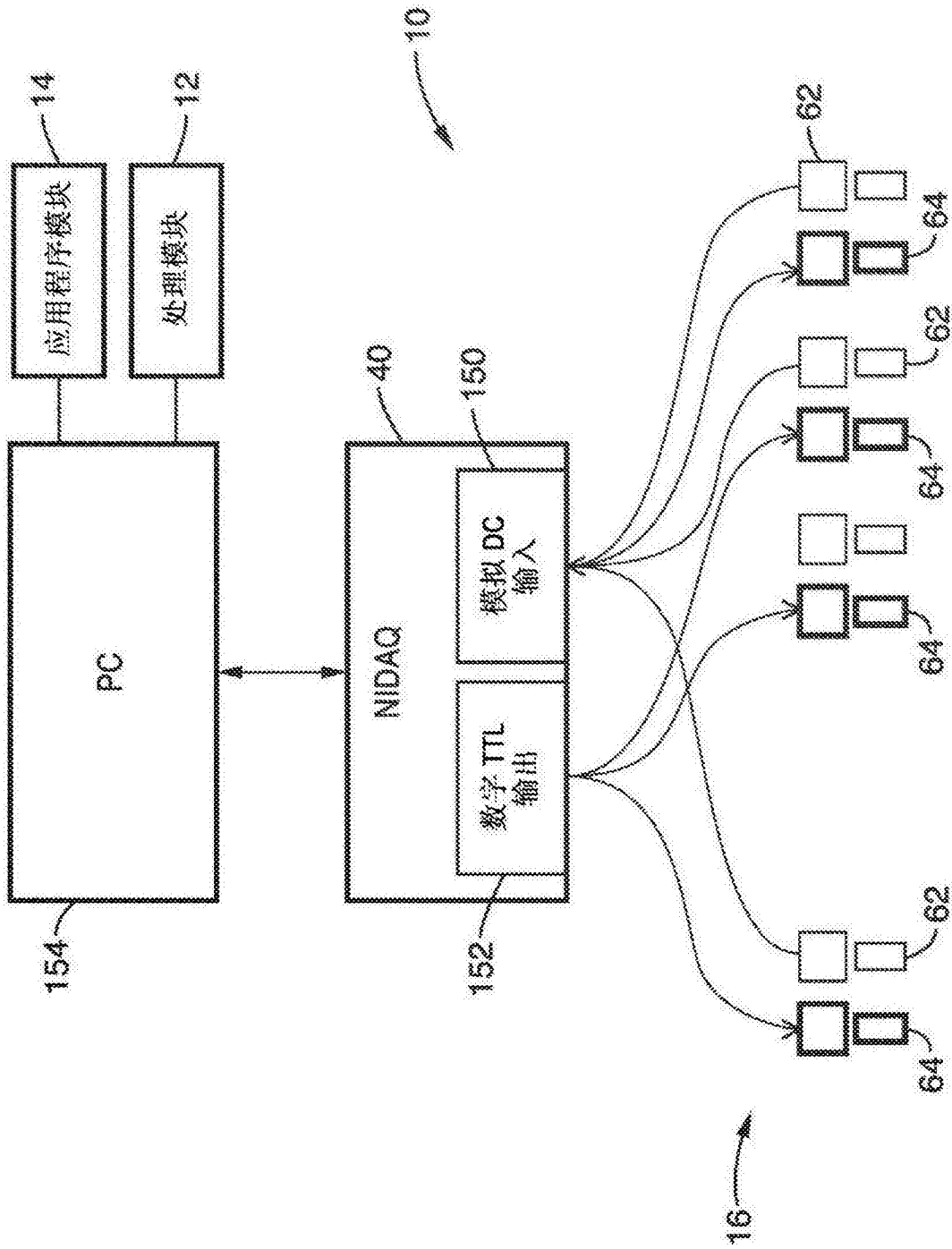
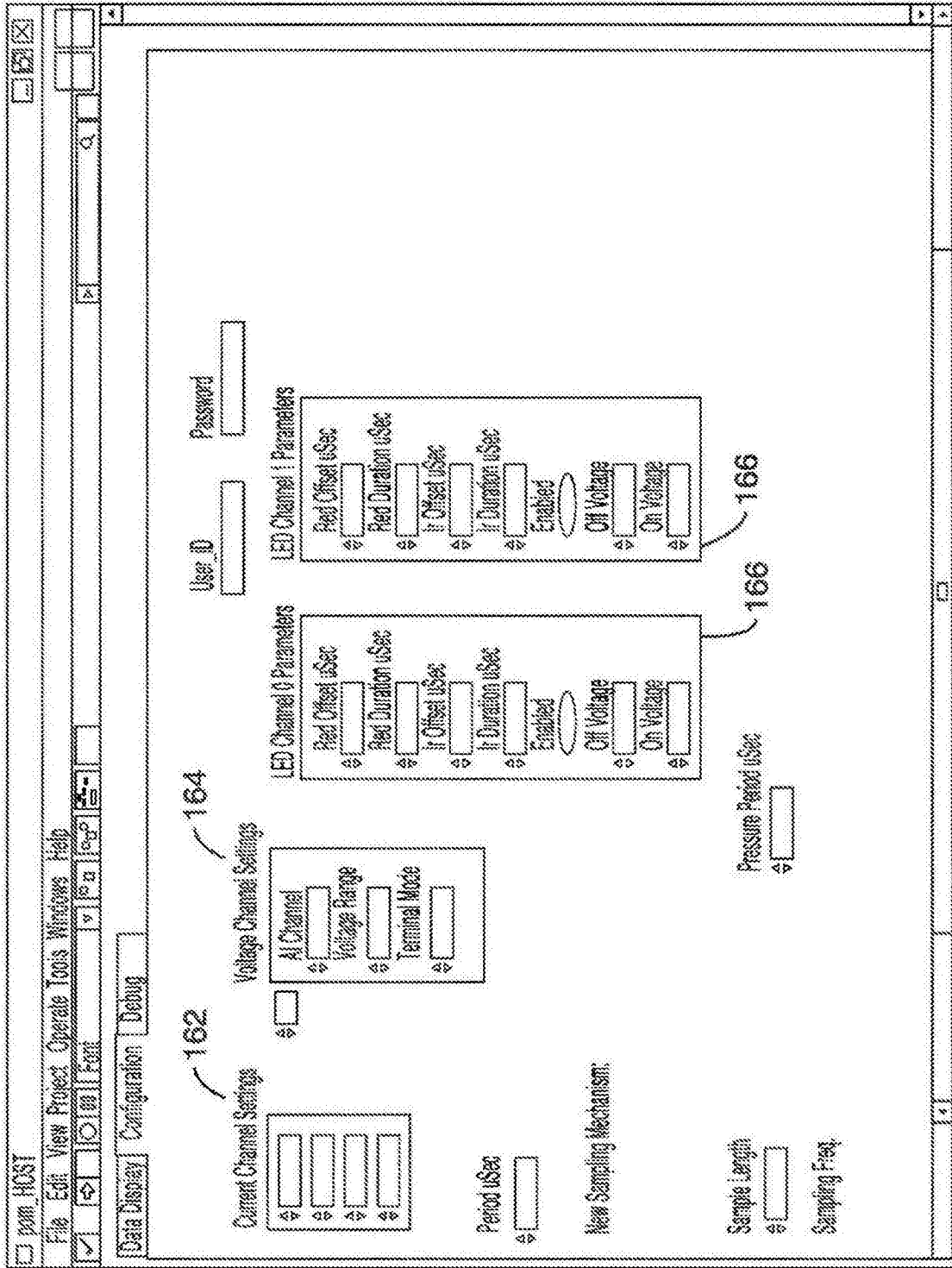


图11



160

图12

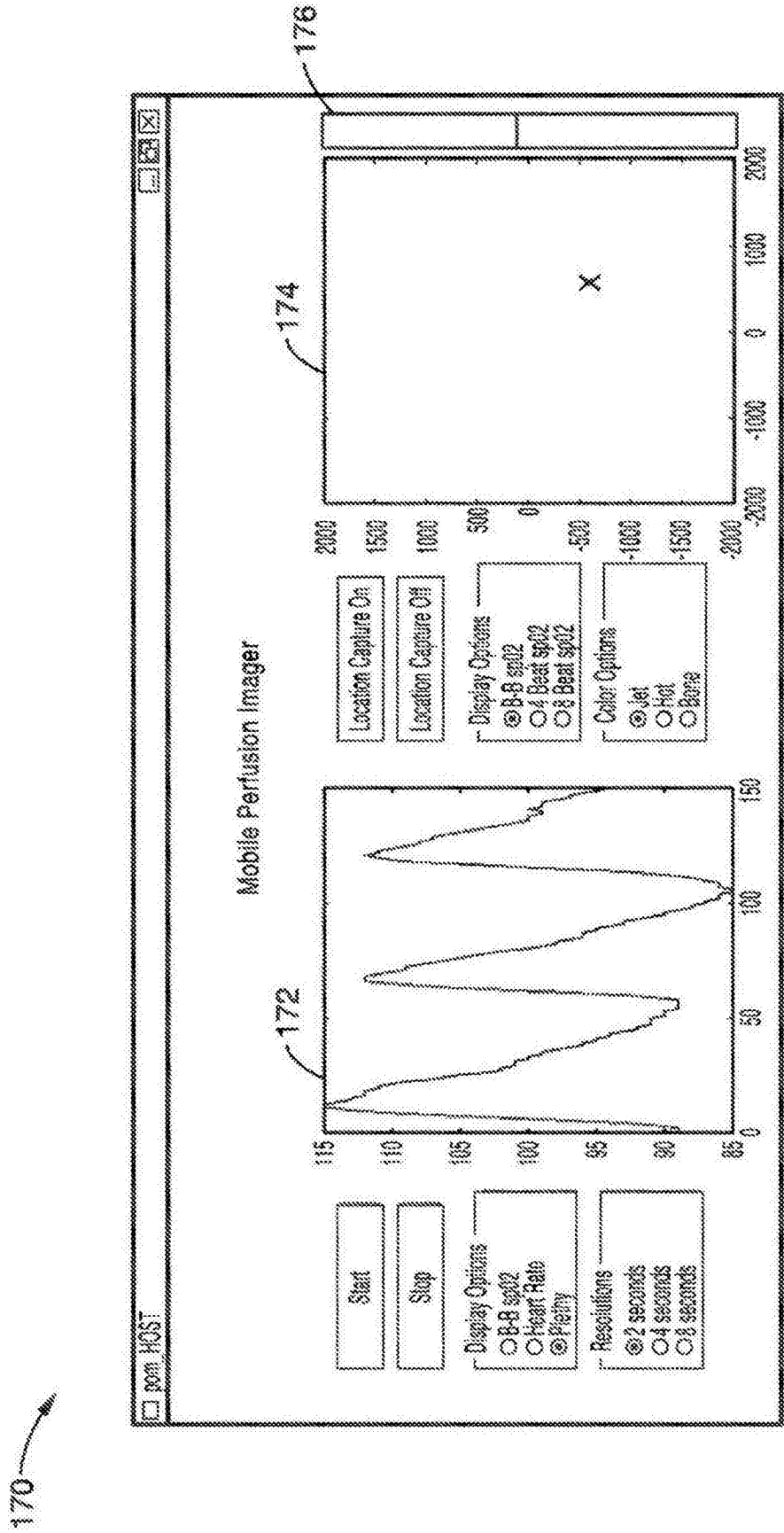


图13

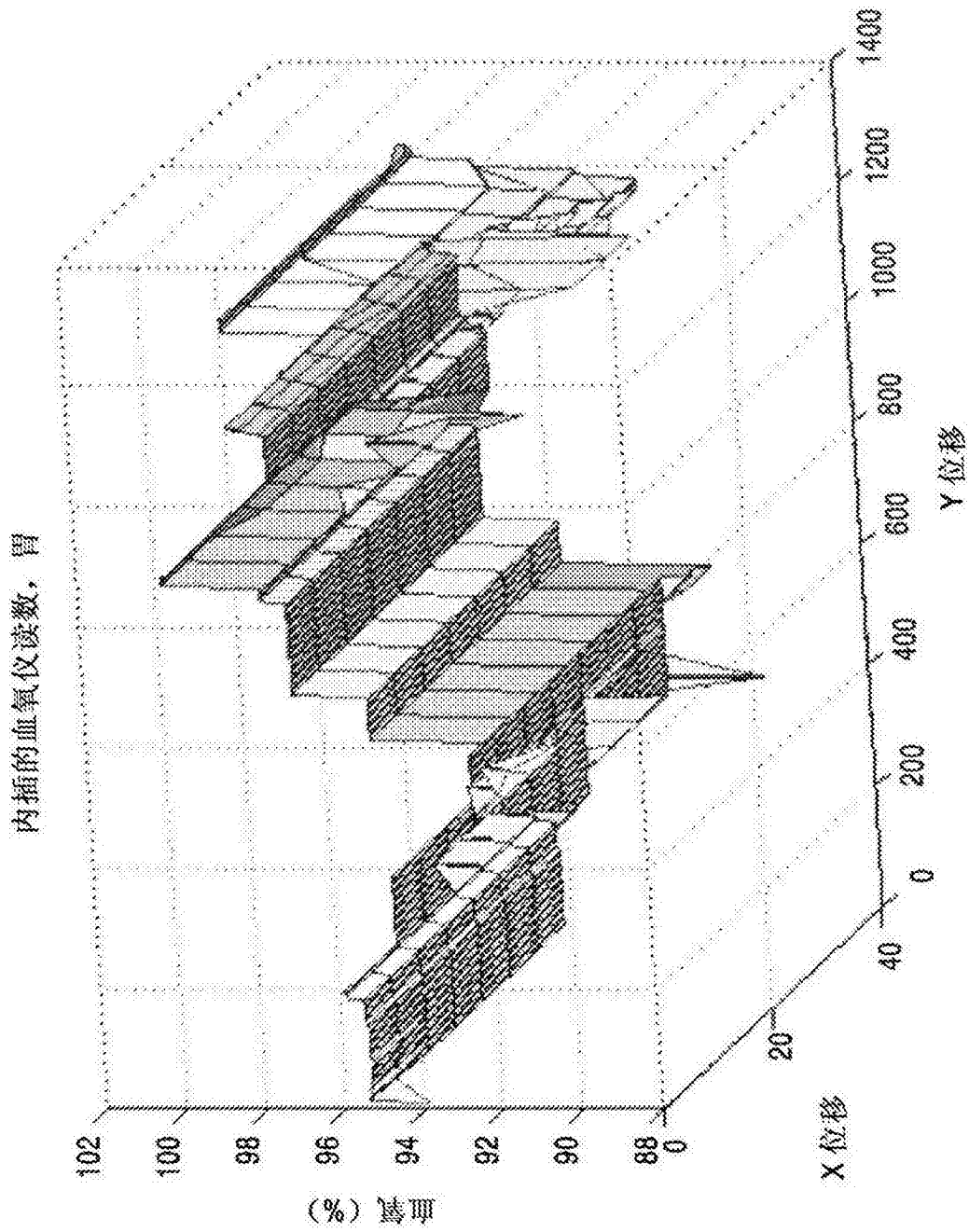


图14

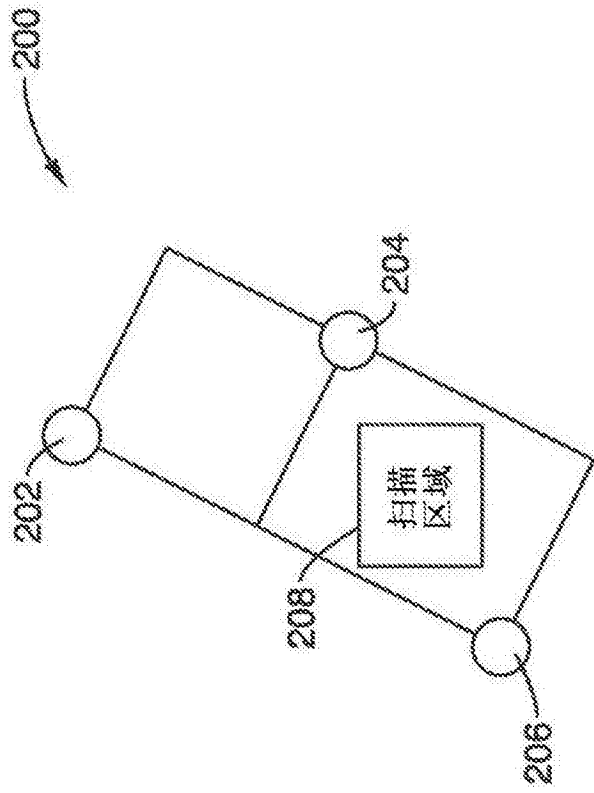


图15

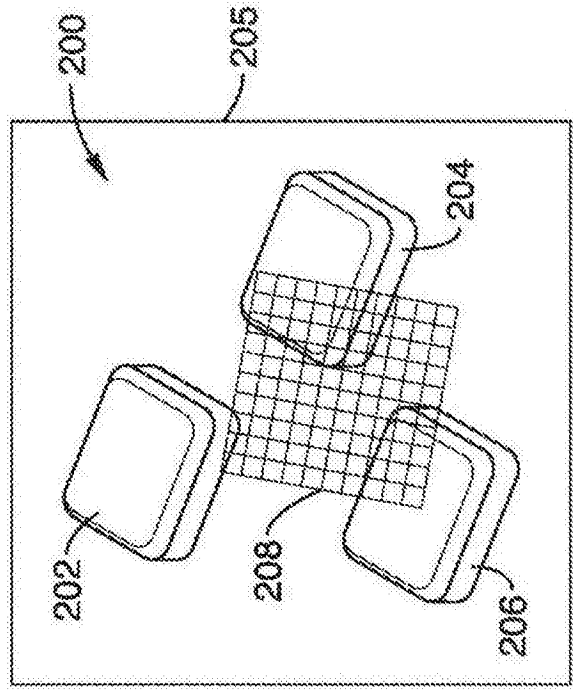


图16

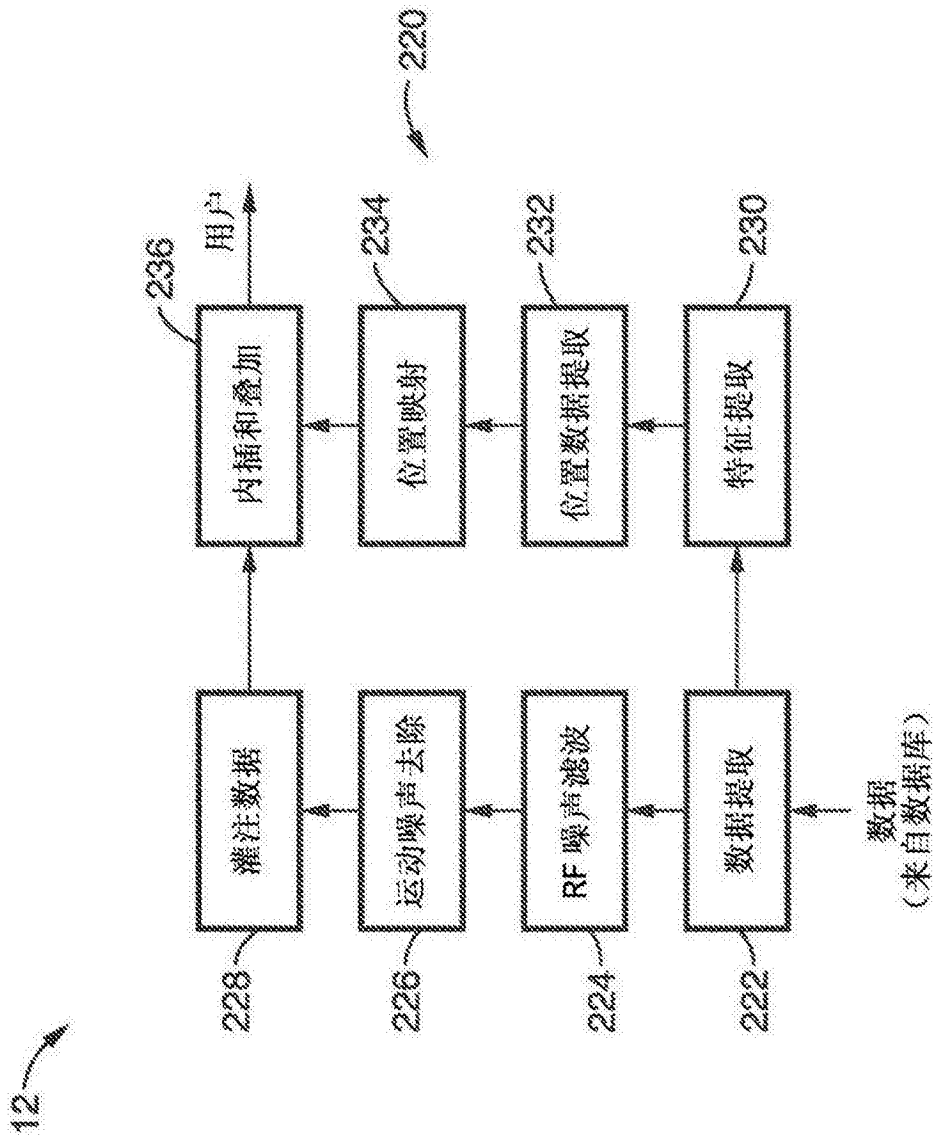


图17

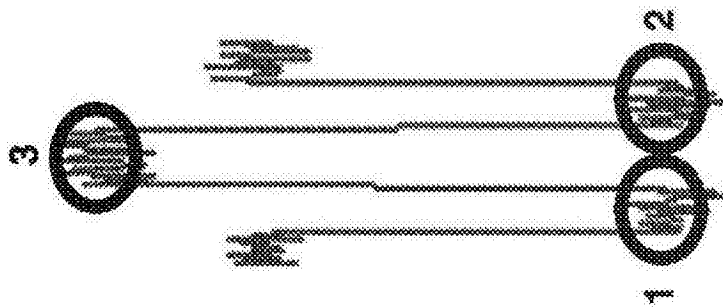


图18

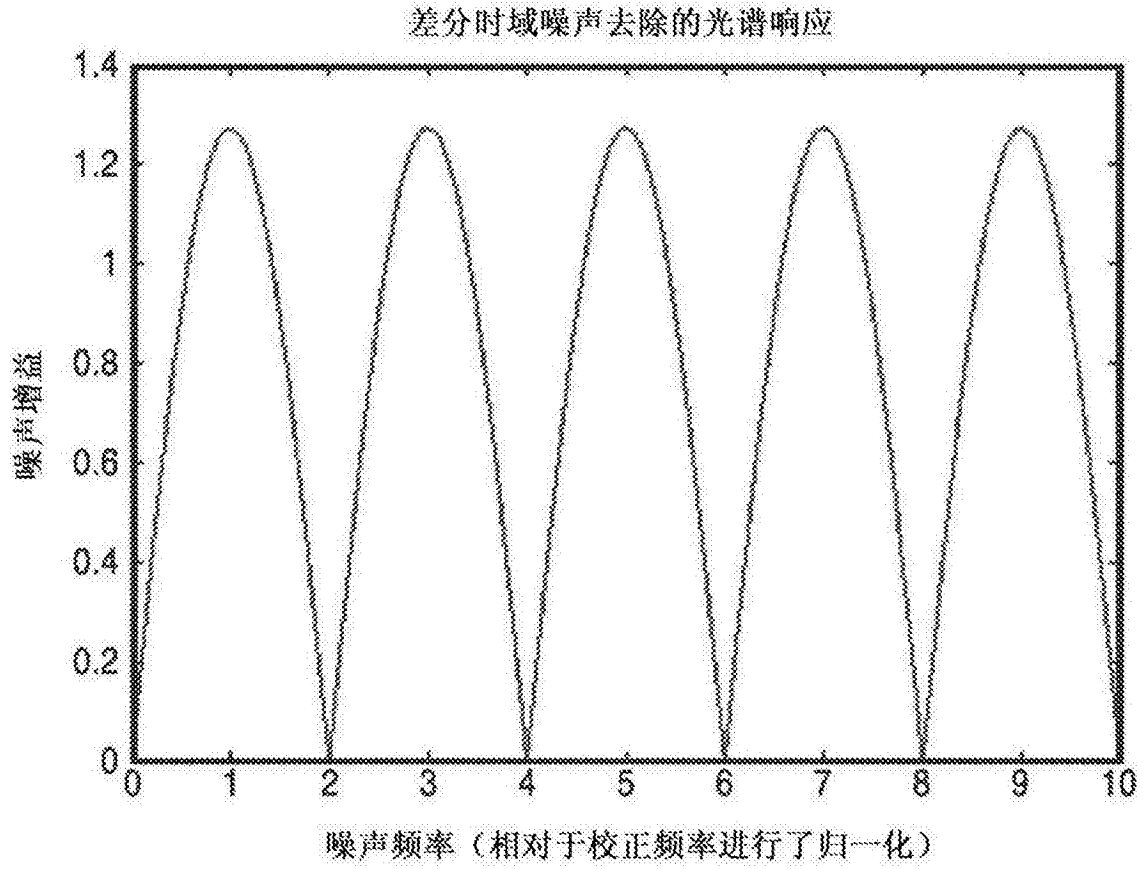


图19

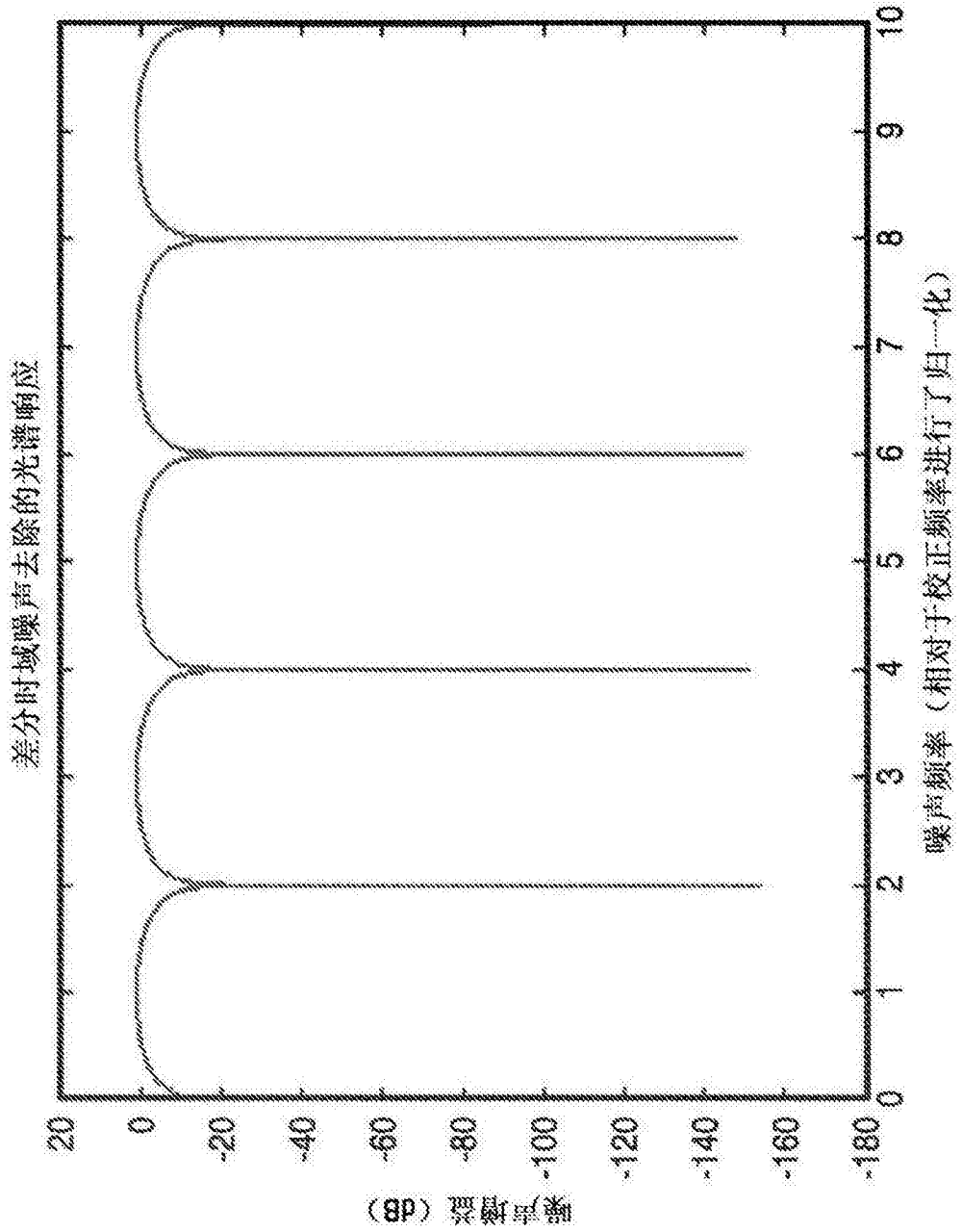


图20

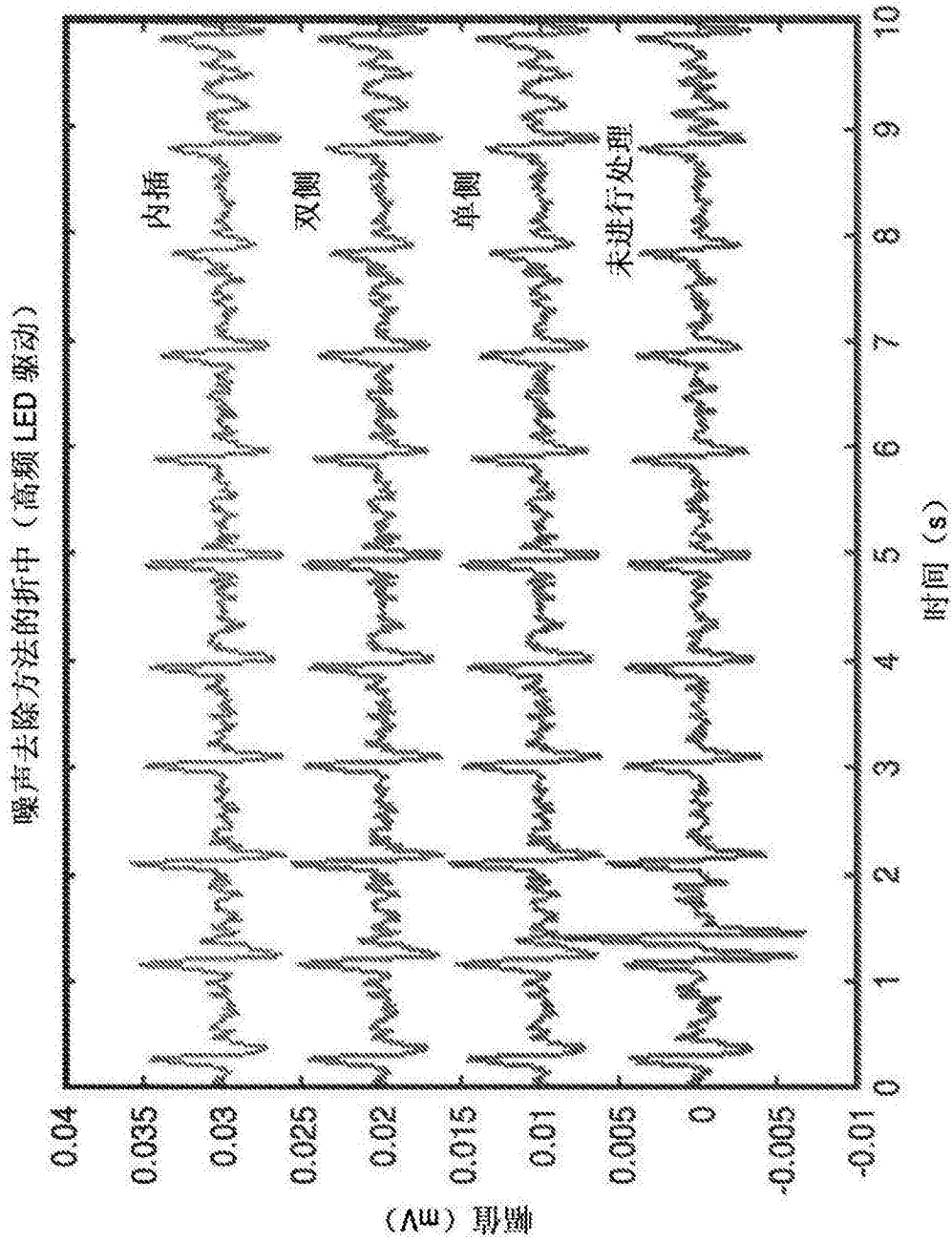


图21

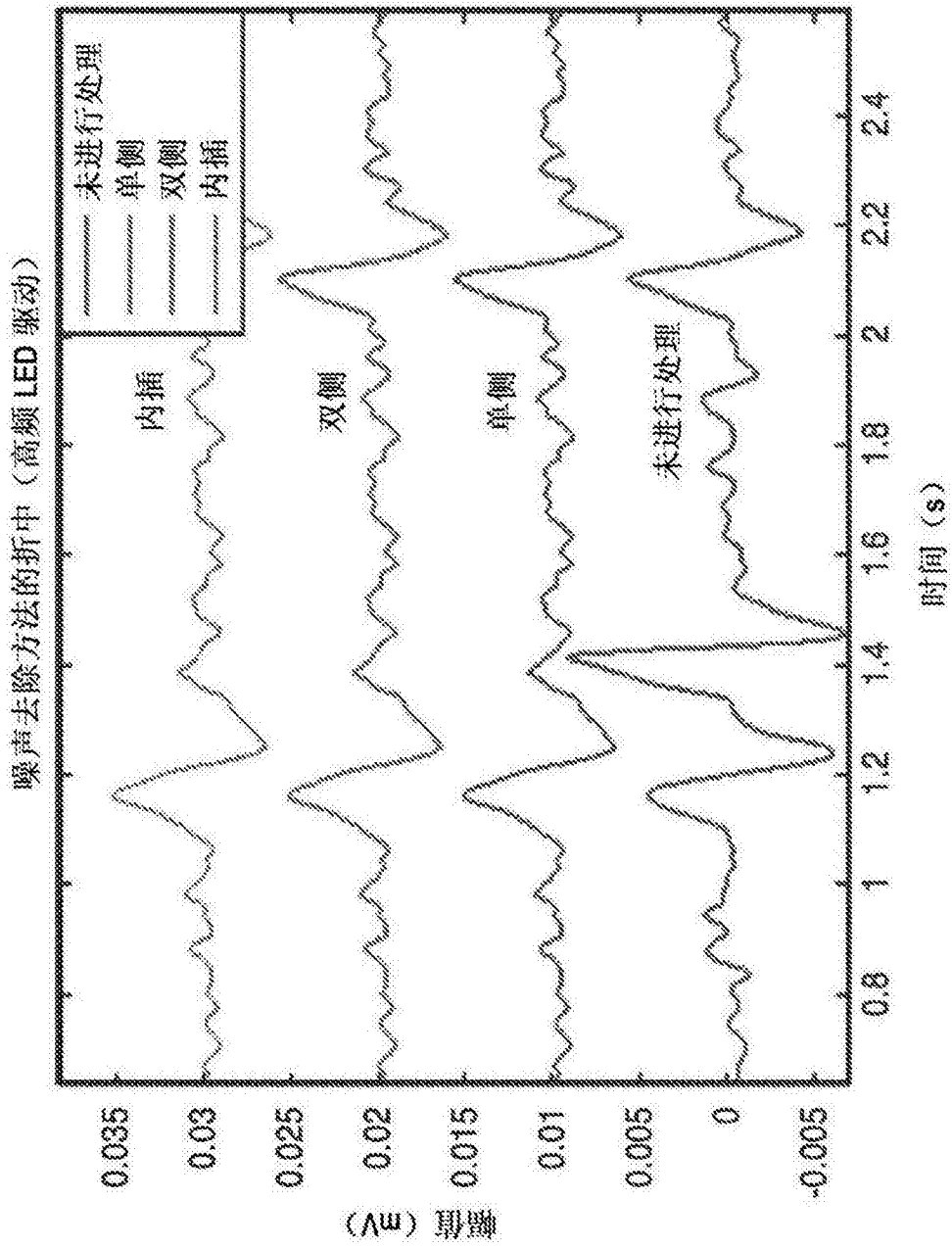


图22

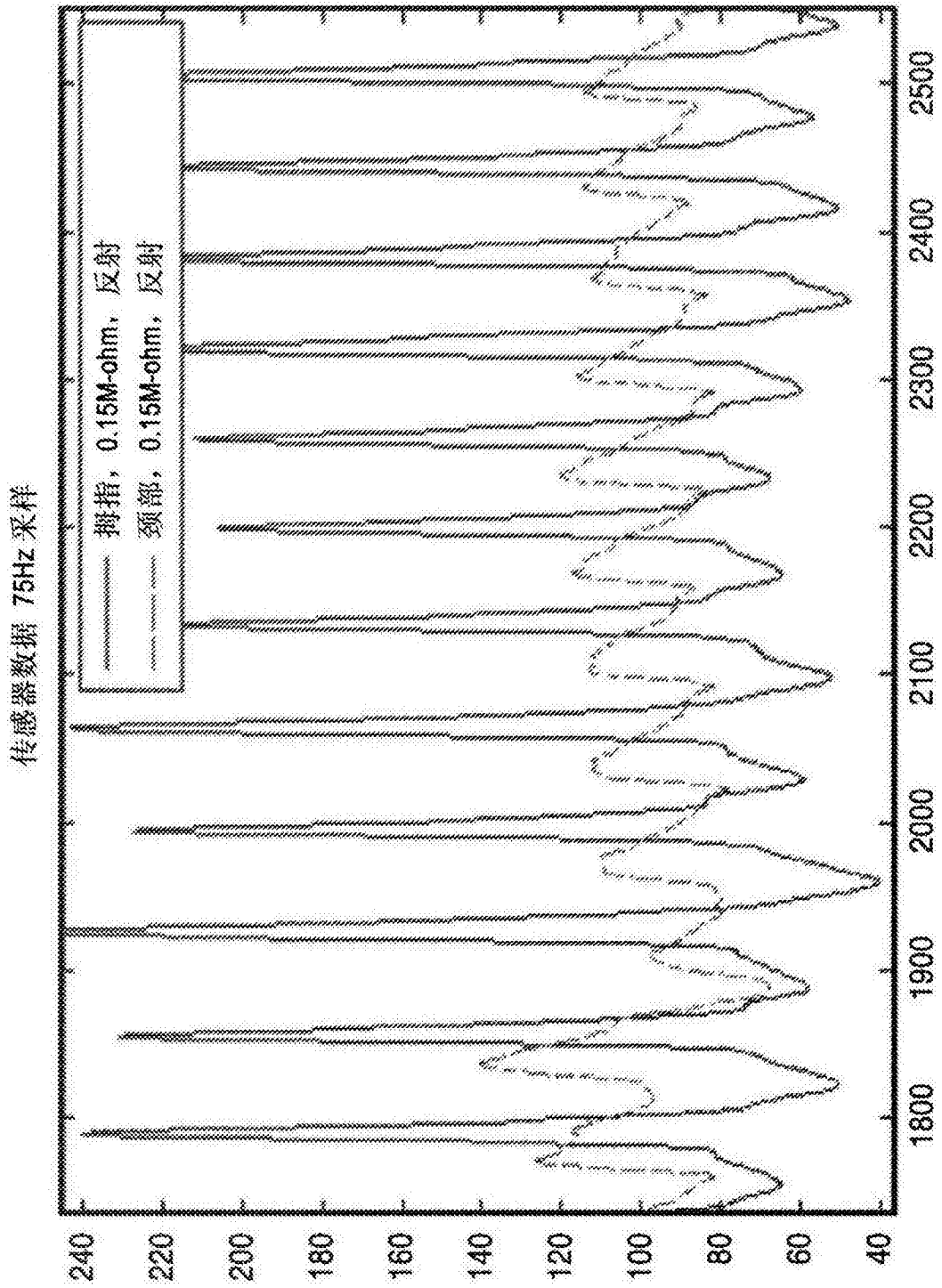


图23

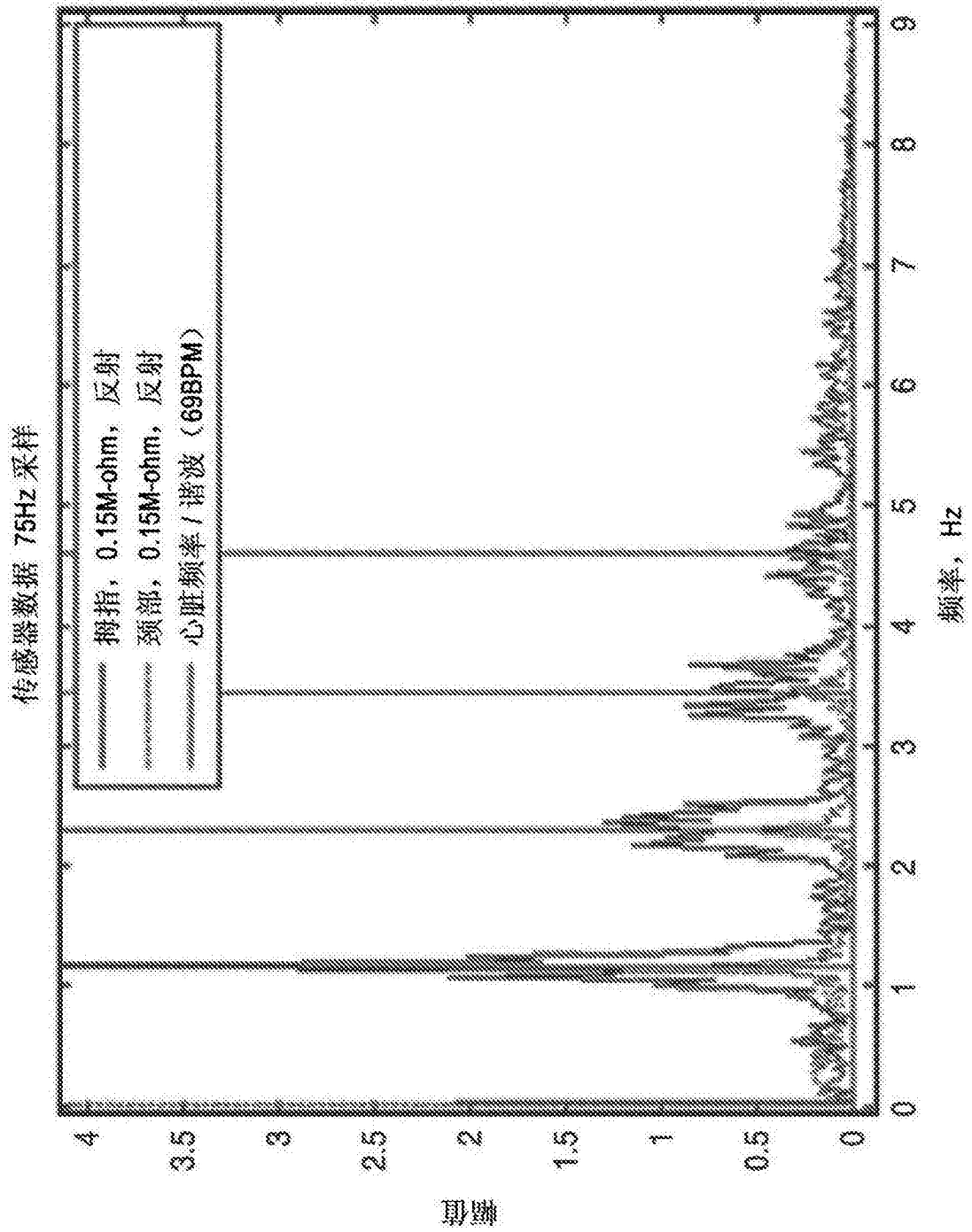


图24

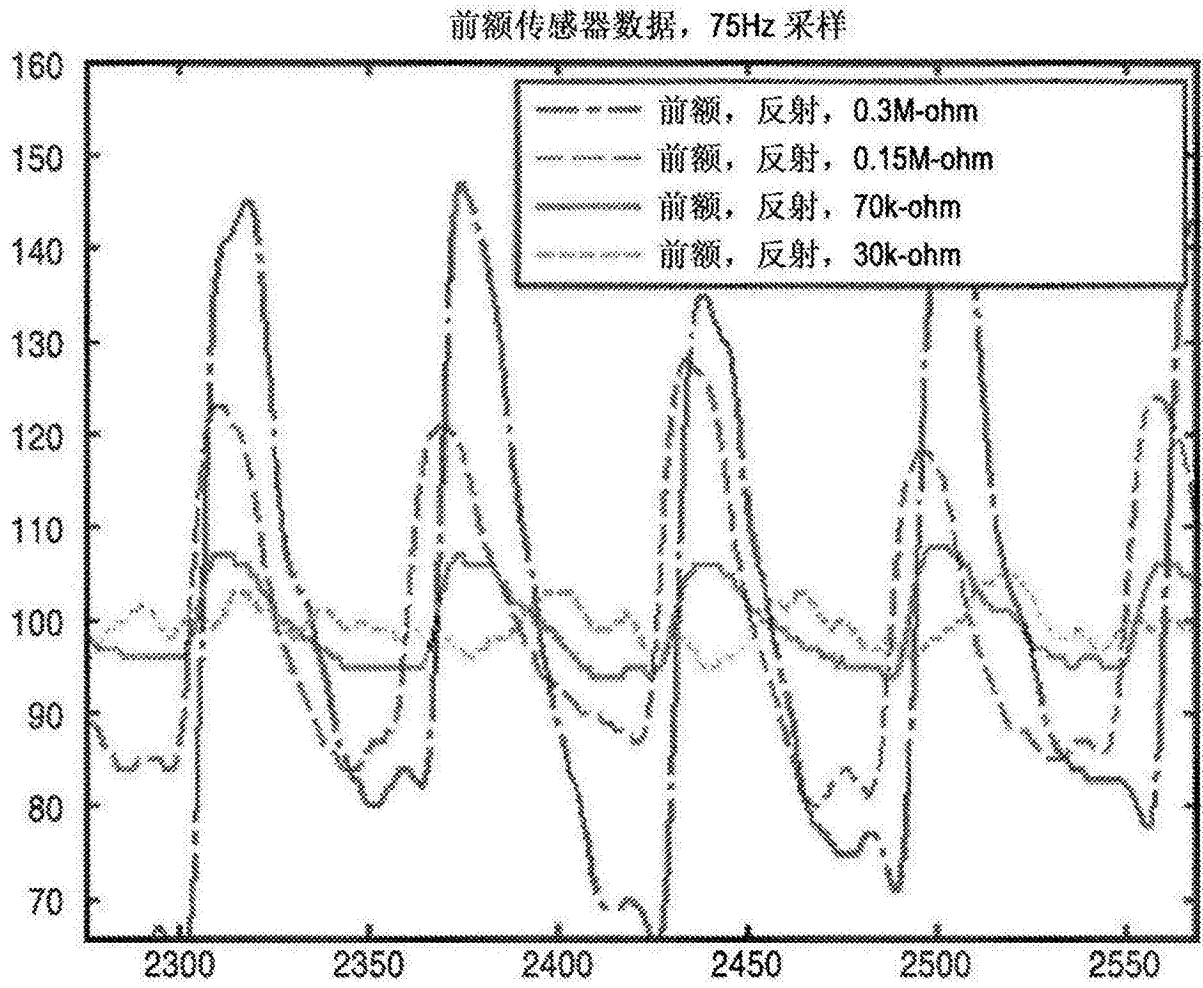


图25

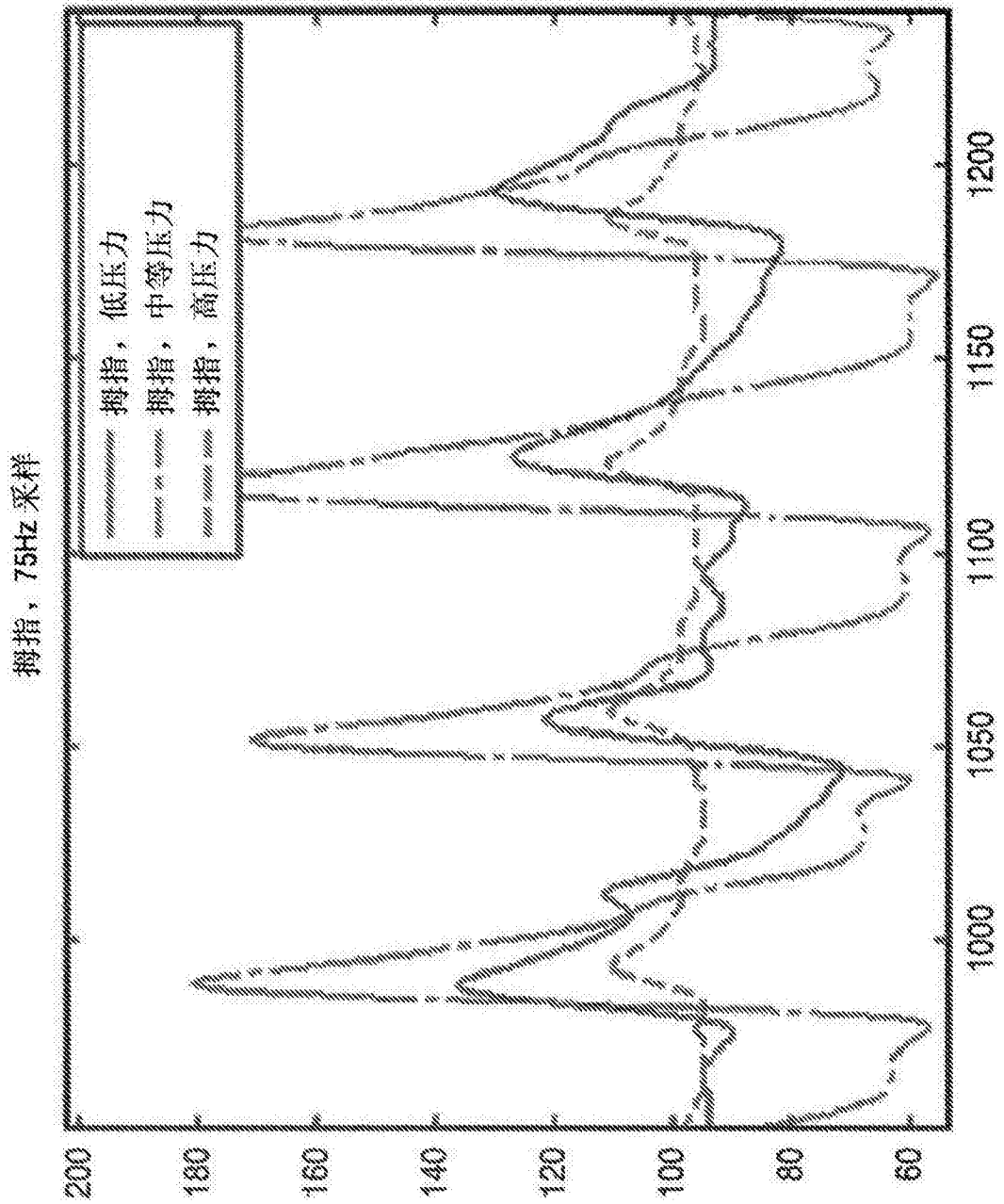


图26

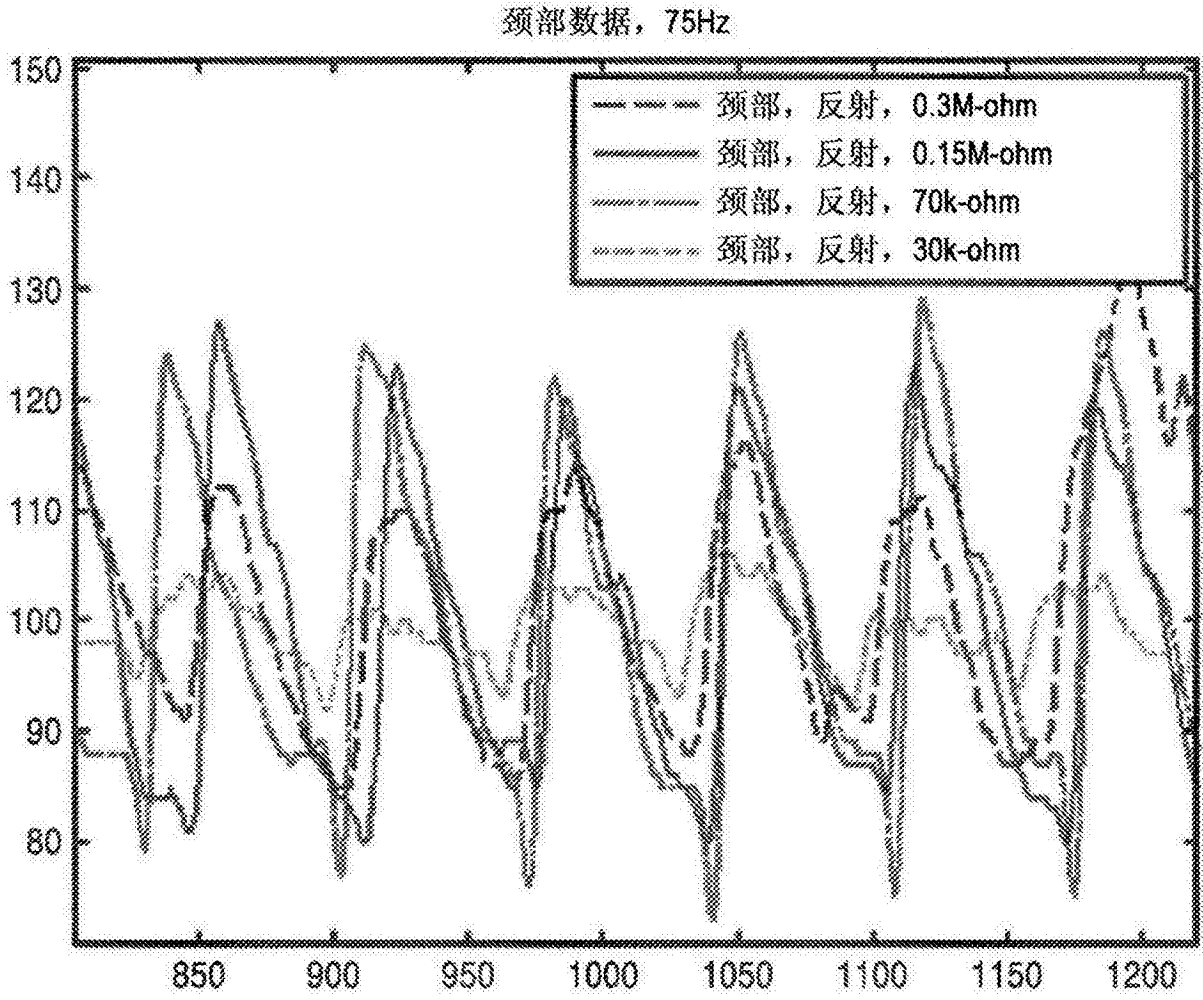


图27

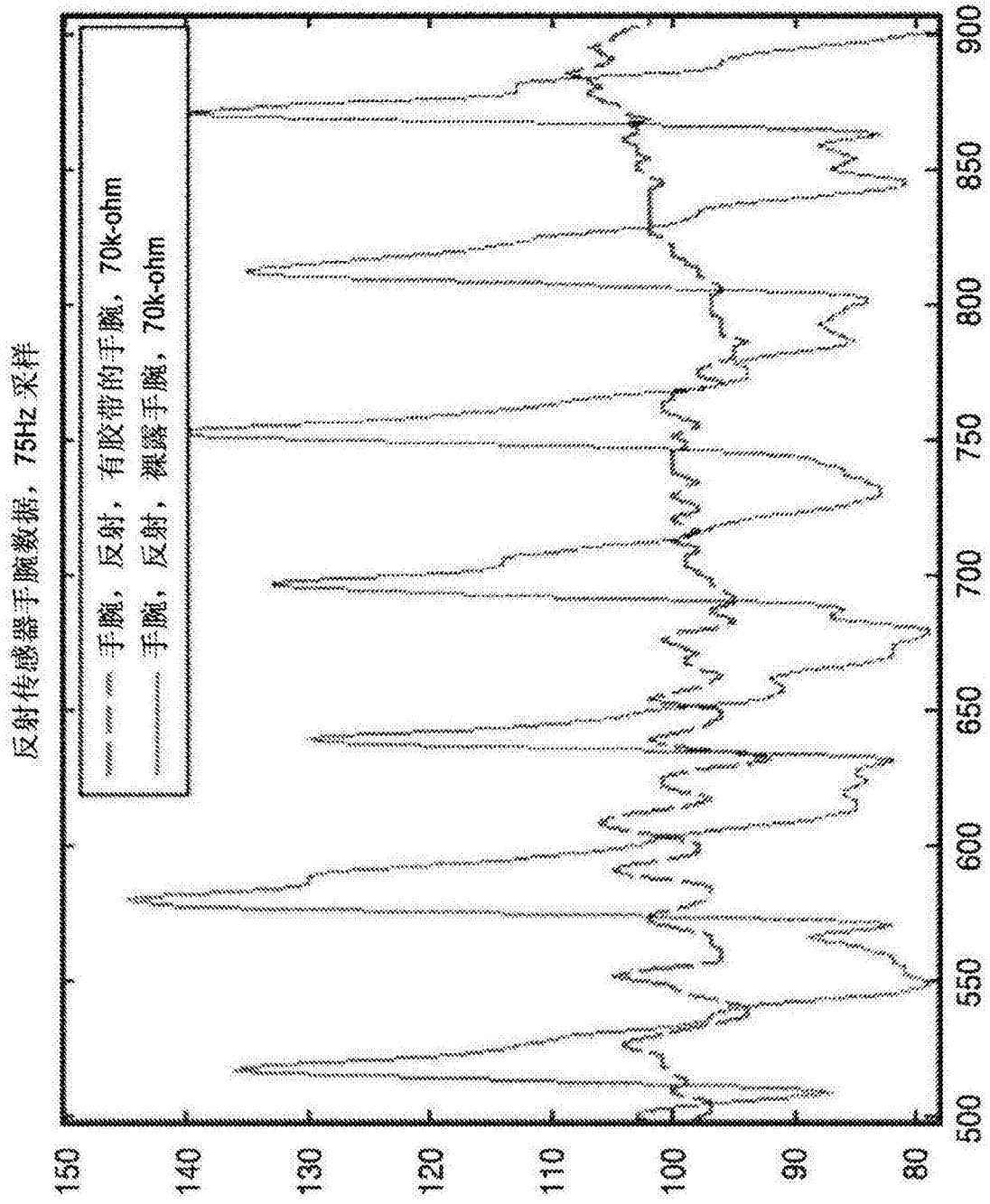


图28

专利名称(译)	组织的血氧定量测定和灌注成像的设备、系统及方法		
公开(公告)号	CN105877764A	公开(公告)日	2016-08-24
申请号	CN201610192712.7	申请日	2012-01-19
[标]申请(专利权)人(译)	加利福尼亚大学董事会		
申请(专利权)人(译)	加利福尼亚大学董事会		
当前申请(专利权)人(译)	加利福尼亚大学董事会		
[标]发明人	马吉德萨拉夫扎德 威廉凯泽 芭芭拉贝茨詹森 阿里礼萨梅尔尼亚 比詹马帕尔 弗兰克王		
发明人	马吉德·萨拉夫扎德 威廉·凯泽 芭芭拉·贝茨-詹森 阿里礼萨·梅尔尼亚 比詹·马帕尔 弗兰克·王		
IPC分类号	A61B5/1455 A61B5/026 A61B5/00		
CPC分类号	A61B5/0261 A61B5/14557 A61B5/6843 F04C2270/041 A61B5/02416 A61B5/14552 A61B5/7225 A61B5/743 A61B2562/166 G16H10/00 A61B5/447 A61B5/6814 A61B5/6822 A61B5/6826 A61B5/7203 A61B5/7271 A61B5/742 A61B5/7425 A61B2562/0247		
代理人(译)	张天舒		
优先权	61/434014 2011-01-19 US		
外部链接	Espacenet SIPO		

摘要(译)

公开了一种表征组织的健康状态的小型灌注扫描器和方法，其将压力传感元件与光学传感器结合起来监测施加到靶组织上的压力水平用以进行精确的皮肤/组织的血液灌注测量和血氧定量测定。该系统和方法能够进行灌注成像和灌注映像(几何和时间)、信号处理和模式识别、灌注数据的噪声消除和数据融合、扫描器的位置和压力读取。

