

(12) 发明专利申请

(10) 申请公布号 CN 102762344 A

(43) 申请公布日 2012. 10. 31

(21) 申请号 201080024680. 0

代理人 毛力

(22) 申请日 2010. 05. 26

(51) Int. Cl.

(30) 优先权数据

B25J 9/16 (2006. 01)

12/474, 778 2009. 05. 29 US

G06T 7/00 (2006. 01)

(85) PCT申请进入国家阶段日

2011. 11. 28

(86) PCT申请的申请数据

PCT/US2010/036171 2010. 05. 26

(87) PCT申请的公布数据

W02010/138565 EN 2010. 12. 02

(71) 申请人 考戈奈克斯技术和投资股份有限公司

地址 美国加利福尼亚州

(72) 发明人 A·沃莱克 D·迈克尔

(74) 专利代理机构 上海专利商标事务所有限
公司 31100

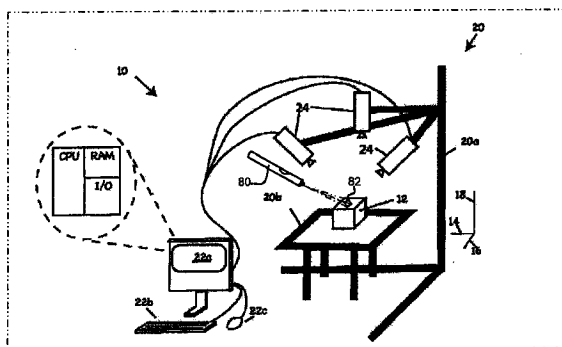
权利要求书 2 页 说明书 25 页 附图 5 页

(54) 发明名称

用于实用 3D 视觉系统的方法和设备

(57) 摘要

一种用于在 3D 成像系统 (10) 中指定感兴趣区域的方法和系统, 该 3D 成像系统包括多个照相机 (24), 这多个照相机包括至少第一和第二照相机, 其中每个照相机 (24) 具有沿照相机不同轨道排列的视场, 该方法包括以下步骤: 在多个照相机的视场内的位置呈现零件 (12); 在多个照相机中的每个照相机的视场内在零件 (12) 上标识感兴趣区域 (82); 以及针对多个照相机中的每个照相机: (i) 采集包括感兴趣区域的该零件 (12) 的至少一幅图像; (ii) 在与该至少一幅图像中的感兴趣区域 (82) 相关联的照相机 (24) 的视场内标识照相机 (24) 专属感兴趣视场; 以及 (iii) 存储所述感兴趣视场以供随后使用。



1. 一种用于在 3D 成像系统中指定感兴趣区域的方法,所述 3D 成像系统包括多个照相机,所述多个照相机包括至少第一和第二照相机,其中每个照相机具有沿照相机不同轨道排列的视场,所述方法包括以下步骤:

在所述多个照相机的视场内的位置呈现零件;

在所述零件上指示感兴趣区域,所述感兴趣区域在所述多个照相机中的每个照相机的视场内;

对于所述多个照相机中的每个照相机:

(i) 采集所述零件的包括所述感兴趣区域的至少一幅图像;

(ii) 在与所述至少一幅图像中的感兴趣区域相关联的照相机的视场内标识照相机专属感兴趣视场;以及

(iii) 存储所述感兴趣视场以供随后使用。

2. 如权利要求 1 所述的方法,其特征在于,在零件上指示所述感兴趣区域的所述步骤包括在所述零件上指示限定所述感兴趣区域的至少三个点。

3. 如权利要求 2 所述的方法,其特征在于,指示至少三个点的步骤包括使用激光设备来按顺序指示所述至少三个点中的每一个点,且其中采集至少一幅图像的步骤包括针对所述至少三个点中的每一个点采集单独图像。

4. 如权利要求 3 所述的方法,其特征在于,标识照相机专属感兴趣视场的步骤包括使用多幅采集到的图像来标识所述感兴趣视场。

5. 如权利要求 3 所述的方法,其特征在于,还包括以下步骤:对于每个点,确定何时所述点已在每个照相机所采集的至少一幅图像中被标识,并指示何时所述点已在每个照相机所采集的至少一幅图像中被标识。

6. 如权利要求 3 所述的方法,其特征在于,还包括以下步骤:对于每个点,确定何时所述点已在至少一个照相机所采集的至少一幅图像中被标识同时所述点尚未在至少另一个照相机所采集的至少一幅图像中被标识。

7. 如权利要求 1 所述的方法,其特征在于,在所述零件上标识感兴趣区域的步骤包括,使用激光设备通过经由激光束照亮所述零件的至少一部分来在所述零件的表面上指示所述感兴趣区域。

8. 如权利要求 7 所述的方法,其特征在于,所述激光设备包括按钮,所述按钮可被选择以指示所述零件的当前被照亮部分对应于定义信息的至少一些感兴趣区域,且应当分析该按钮被选择时刻的所述零件的图像以获得与所述零件的当前被照亮部分相关联的位置信息,所述方法还包括监测来自所述激光设备的指示的步骤。

9. 如权利要求 8 所述的方法,其特征在于,所述激光设备被编程以在所述按钮被选择时使所述零件上的照射束的外观发生变化,监测来自所述激光设备的指示的步骤包括分析照相机图像以标识所述零件上的照射束的外观的变化。

10. 如权利要求 1 所述的方法,其特征在于,还包括向系统用户提供指示何时已针对所述多个照相机中的每一个标识照相机专属感兴趣视场的反馈的步骤。

11. 如权利要求 2 所述的方法,其特征在于,采集至少一幅图像的所述步骤包括迅速连续地采集图像,且其中指示至少三个点的步骤包括使用激光设备来独立地照亮三个点中的每一个点。

12. 如权利要求 11 所述的方法,其特征在于,指示至少三个点的所述步骤包括,对于每个点,当所述激光设备照亮所述点时,指示被照亮位置是定义感兴趣区域的那些点之一。

13. 一种用于在 3D 成像系统中指定感兴趣区域的方法,所述 3D 成像系统包括多个照相机,所述多个照相机包括至少第一和第二照相机,其中每个照相机具有沿照相机不同轨道排列的视场,所述方法包括以下步骤:

在所述照相机的视场内的位置呈现零件;

使用激光设备按顺序并独立地在所述零件上指示多个点,所述多个点限定感兴趣区域;

对于所述多个照相机中的每一个,当在所述零件上指示所述点时,采集所述零件的包括所述感兴趣区域的图像;

使用每个照相机的多幅图像中的所述点来标识所述视场中的与所述感兴趣区域相对应的感兴趣视场;以及

将所述视场中的感兴趣视场存储为照相机专属感兴趣视场。

14. 一种用于训练三维视觉系统以限定照相机专属感兴趣视场的系统,在所述照相机专属感兴趣视场中在零件上指示了感兴趣区域,所述系统包括:

多个照相机,每个照相机具有视场且沿照相机不同轨道排列以使所述感兴趣区域在照相机的视场内,每个照相机用于获得所述零件在照相机的视场内的图像;以及

机器视觉处理器,被编程为针对每个照相机执行以下步骤:

(i) 接收图像;

(ii) 使用所接收的至少一幅图像,将用来产生图像的照相机的与所述感兴趣区域相关联的视场的一部分标识为感兴趣视场;以及

(iii) 将与所述感兴趣区域相关联的感兴趣视场存储为照相机专属感兴趣视场。

15. 如权利要求 14 所述的系统,其特征在于,还包括激光指示器设备,用于将激光束引导至所述零件以在所述零件上指示所述感兴趣区域。

16. 如权利要求 15 所述的系统,其特征在于,所述激光指示器设备包括按钮,所述按钮可被选择以指示所述零件的当前被照亮部分对应于定义信息的至少一些感兴趣区域,且应当分析所述按钮被选择时刻的所述零件的图像以获得与所述零件的当前被照亮部分相关联的位置信息,所述处理器被进一步编程以执行监测来自所述激光指示器设备的关于应当分析图像的指示的步骤。

17. 如权利要求 16 所述的系统,其特征在于,所述激光指示器设备被编程以在所述按钮被选择时使所述零件上的照射束的外观发生变化,且监测来自所述激光指示器设备的指示的步骤包括分析照相机图像以标识所述零件上的照射束的外观的变化。

18. 如权利要求 16 所述的系统,其特征在于,所述激光指示器设备包括在按钮被选择时发送信号的发射器。

用于实用 3D 视觉系统的方法和设备

[0001] 发明背景

[0002] 本发明涉及机器视觉,更具体地涉及三维(3D)机器视觉。本发明可应用于制造、质量控制以及机器人技术等等领域中。

[0003] 机器视觉指的是对图像进行自动分析以确定图像中表示的对象的特性。机器视觉常用于自动生产线中,在自动生产线中,分析部件图像以便于零件分拣,并确定零件放置和对准以进行组装。当机器人是用于自动组装的装置且图像自动分析用于便于零件分拣、放置以及对准时,该系统被称为视觉引导的机器人。机器视觉还用于机器人导航,例如用于当机器人在各种环境中行进时确保对场景的识别。

[0004] 尽管在文献中对三维(3D)分析进行了长期讨论,但当今大多数机器视觉系统依赖于二维(2D)图像分析。这通常使待检查的对象以受约束的取向和位置“被呈现”给该视觉系统。传送带通常用于此目的。被组装或检查的对象通常以特定已知、稳定的 3D 配置但以未知的位置和取向被放置在该传送带上,并被移至该视觉系统的视场内。基于对象在视场中的 2D 姿态(即位置和取向),并考虑它被放置在传送带上(因此确保它的“放置状态”和与视觉系统照相机的距离),该系统应用 2D 几何测量以确定该对象的准确 3D 姿态和/或与预期外观的一致性。

[0005] 本申请的受让人的先前著作中提供了使用此类 2D 视觉分析的示例,这些著作包括标题为“Methods and apparatus for machine vision inspection using single and multiple templates or patterns(用于使用单个和多个模板或图案进行机器视觉检查的方法和设备)”的美国专利 6,748,104、标题为“Machine vision methods for inspection of leaded components(用于对经引导的部件进行检查的机器视觉方法)”的美国专利 6,639,624,标题为“Nonfeedback-based machine vision methods for determining a calibration relationship between a camera and a moveable object(用于确定照相机与可移动对象之间的校准关系的非基于反馈的机器视觉方法)”的美国专利 6,301,396、标题为“Machine vision calibration targets and methods of determining their location and orientation in an image(机器视觉校准靶和确定它们在图像中的位置和取向的方法)”的美国专利 6,137,893、标题为“Machine vision methods using feedback to determine calibration locations of multiple cameras that image a common object(使用反馈来确定对常见物体成像的多个照相机的校准位置的机器视觉方法)”的美国专利 5,978,521、标题为“Machine vision methods using feedback to determine an orientation, pixel width and pixel height of a field of view(使用反馈来确定视场的取向、像素宽度和像素高度的机器视觉方法)”的美国专利 5,978,080、标题为“Nonfeedback-based machine vision method for determining a calibration relationship between a camera and a moveable object(用于确定照相机与可移动物体之间的校准关系的非基于反馈的机器视觉方法)”的美国专利 5,960,125、标题为“Fast high-accuracy multi-dimensional pattern localization(快速高准确度多维图案局部化)”的美国专利 6,856,698、标题为“Fast high-accuracy multi-dimensional pattern

inspection(快速高精度多维图案检查)”的美国专利 6,850,646、以及标题为“Fast high-accuracy multi-dimensional pattern inspection(快速高精度多维图案检查)”的美国专利 6,658,145 等等。

[0006] 在对机器人的依赖提高的情况下,从工厂车间到家庭的任何地方,对实用 3D 视觉系统的需求已经涌现。这是因为,在许多这些环境中,受检查的对象的整体位置和放置状态不一定受约束,例如否则可能是对象呈现在传送带上的情况。即,该对象的精确 3D 配置可能是未知的。

[0007] 为了适应 3D 场景中的姿态和位置的额外的自由度,3D 视觉工具即使不是必要的也是有用的。这些情况的示例包括标题为“System and method for registering patterns transformed in six degrees of freedom using machine vision(用于利用机器视觉对准在六个自由度中变换的图案的系统和方法)”的美国专利 6,771,808 和标题为“System and method for determining the position of an object in three dimensions using a machine vision system with two cameras(用于利用具有两个照相机的机器视觉系统来确定三维中的物体的位置的系统和方法)”的美国专利 6,728,582。

[0008] 现有技术中也已经提出了其它机器视觉技术。一些技术要求太多处理器能力而对于实时应用不实用。其它技术要求受检查对象通过复杂的对准程序,和/或要求该对象的许多特征在运行期间在视觉系统视场中同时可见。

[0009] 在机器视觉领域之外,现有技术还提供了基于接触来确定 3D 姿态的方法——诸如使用具有触摸传感器的 x、y、z 测量机器。然而,该方法要求接触,因此相对较慢且要求手动干预。还提供了基于电磁波的用于确定 3D 姿态的方法。这些方法不需要物理接触,但具有它们自身的缺点,诸如通常要求在受检查对象上固定发射器的不实用步骤。

[0010] 本发明的目的是提供用于机器视觉的改进方法和设备,更具体地是提供用于三维机器视觉的改进方法和设备。

[0011] 本发明的相关目的是提供具有包括但不限于制造、质量控制以及机器人的一系列实际应用的方法和设备。

[0012] 本发明的另一相关目的是提供允许确定例如三维空间中的位置和姿态的方法和设备。

[0013] 本发明的又一相关目的是提供对例如受检查对象的整体位置和放置状态施加减少的约束的方法和设备。

[0014] 本发明的又一相关目的是提供对受检查对象的对准要求最小的方法和设备。

[0015] 本发明的又一目的是提供能在现有和未来的机器视觉平台中实现的方法和设备。

发明内容

[0016] 除实现上述目的之外,本发明还特别提供用于通过对从对象的多幅图像收集的数据的三角测距来确定对象在三维上的姿态(例如沿 x、y 和 z 轴的位置、俯仰(pitch)、滚转(roll)和偏航(yaw))(或姿态的一个或多个特性)的方法和设备。

[0017] 因此,例如,在一个方面中,本发明提供用于 3D 机器视觉的方法,在该方法中,在校准步骤期间,设置以用于从各个不同视角获取对象的图像的多个照相机被校准以辨别映射功能,该映射功能在 3D 空间中标识从各个照相机的镜头发出的光线,这些光线对应于该

照相机视场中的像素位置。在训练步骤中,与照相机相关联的功能经训练以识别要采集的该对象的图像中的预期图案。运行时间步骤根据该对象的图像中的那些图案的像素方式位置以及根据在校准步骤期间辨别的映射,来对那些图案中的一个或多个在 3D 空间中的位置进行三角测距。

[0018] 本发明的其它方面提供如上所述的方法,其中运行时间步骤根据由多个照相机基本同时拍摄的该对象的图像来对位置进行三角测距。

[0019] 本发明的其它目的提供包括重新校准步骤的此类方法,在该重新校准步骤中使用该对象的运行时间图像来辨别上述映射功能,例如针对已经偏离校准的照相机。因此,例如,如果一个照相机产生的图像中的图案看起来处于与来自其它照相机的图像(例如当使用它们各自的 3D 光线映射时)不一致和/或显著不一致的位置(例如当映射至该照相机的 3D 光线时),利用来自那些其它照相机的图像确定的图案位置可用于重新校准该一个照相机。

[0020] 本发明的其它方面提供如上所述的方法,其中校准步骤包括将对准靶(诸如校准盘或其它物体上的靶心、十字准线或类似物等)定位在 3D 空间中的已知位置处,并记录或以其他方式表征(例如通过算法)那些位置与相应靶在照相机视场中的像素方式位置之间的关联。本发明的相关方面提供此类方法,其中使用那些对准靶、校准盘等等中的一个或多个例如通过同时成像来同时校准多个照相机。

[0021] 本发明的其它方面提供如上所述的方法,其中该校准步骤包括对考虑视场中的扭曲的每个照相机辨别映射功能。

[0022] 本发明的其它方面包括如上所述的方法,其中训练步骤包括训练与照相机相关联的功能以识别该对象的预期图案(诸如字母、数字、其它符号(诸如对准靶)、隅角或其它可辨别的特征(诸如暗斑和亮斑)),并且例如用于这些特征的测量技术和搜索/检测模型在本领域中是已知的。

[0023] 本发明的其它相关方面提供此类方法,其中训练步骤包括关于“模型点”训练上述功能,“模型点”即在运行时间被检查对象上的图案的 3D 空间中的预期位置(例如按照绝对项或相对项来表示)。在运行时间步骤期间,可将该信息与从那些图像辨别的三角测距的 3D 位置组合地使用,以辨别该对象的姿态。

[0024] 根据本发明的诸方面,关于这些图案的预期位置(即模型点)的训练包括查明每个此类图案的基准点(或“原点”)的 2D 姿态。对于预期出现在两个或更多个照相机视场中的图案而言,如下所述,此类基准点便于三角测距,以在 3D 空间中确定那些图案的位置(因此对象的位置)。

[0025] 本发明的相关方面提供此类方法,其中关于预期图案的训练包括在不同照相机之间(在与每个照相机相关联的功能内)利用相同模型来训练相同的预期图案。这具有确保在运行时间发现的图案的基准点(或原点)在由那些不同照相机获得的图像之间将一致的好处。

[0026] 本发明的其它相关方面提供此类方法,其中关于预期图案的训练包括在不同照相机之间(在与每个照相机相关联的功能内)针对相同图案利用不同模型。这样便于例如在姿态、视角和/或障碍物改变不同照相机对那些图案成像的方式时找出那些图案。

[0027] 本发明的相关方面提供此类方法,这些方法包括训练对如此建模的图案的基准点

(或原点)的选择。此类训练可例如由操作员例如使用激光指示器或其它方式来完成,以确保那些基准点(或原点)在由那些不同照相机获得的图像之间一致。

[0028] 本发明的相关方面提供此类方法,其中训练步骤包括,例如,通过利用与运行时间阶段期间运用的三角测距方法相似的三角测距方法来辨别图案的位置。替代地,可由操作者输入和/或通过其它测量方法辨别图案的预期(相对)位置。

[0029] 本发明的其它相关方面提供此类方法,其中训练步骤包括基于在来自另一照相机的图像中的该图案的先前标识来从来自一个(或多个)照相机的图像中找出预期图案。因此,例如,一旦操作者已经在从一个照相机拍摄的图像中标识预期图案,训练步骤就可包括在来自其它照相机的图案中找出同一图案。

[0030] 本发明的其它方面提供如上所述的方法,其中优选地,训练步骤采集每个照相机对该对象的多幅视图,以使在那些对象上找出的图案的原点被一致地定义。为解决多个图像之间潜在的 inconsistency,可使用对这些图案产生匹配分数最高的那些图像。这具有使该方法对找出处于任意姿态的零件更可靠的好处。

[0031] 在本发明的其它方面中,运行时间步骤包括对运行时间图像中的一个或多个图案的位置进行三角测距,该三角测距例如使用图案匹配或其它二维视觉工具,并使用在校准阶段期间辨别的映射来将各个照相机视场中的那些图案的像素方式位置与图案所依赖的上述 3D 光线相关联。

[0032] 根据本发明的相关方面,图案位置的三角测距可以通过“直接”的三角测距,例如根据图案所依赖的多条 3D 光线(来自多个照相机)的交点(或最小平方拟合点)确定给定图案的位置。或者,或作为附加,三角测距可以是“间接的”,其中给定图案的位置不仅根据图案所依赖的光线(或多条光线)来确定,而且根据(i)其它图案所依赖的光线和(ii)那些图案相对于彼此的位置(例如在训练阶段期间确定)来确定。

[0033] 本发明的其它方面提供如上所述的方法,其中,如果与照相机相关联的功能未能在训练或运行时间期间在对象的图像中找出预期图案,则与照相机相关联的功能“超时”,由此在例如这样的图案丢失、被遮挡或因其它原因未被检测到时,避免在位置确定时过度的延迟。

[0034] 本发明的其它方面与上述方法类似,其中使用 ID 矩阵码(或外观和/或位置是预定义或以其它方式已知的其它图案)来代替以上讨论的图案。在本发明的这些方面中,训练步骤被排除或简化。替代地,通过针对一般类型特征设计的视觉工具,根据训练阶段或运行时间图像辨别那些代码(或其它图案)的 2D 位置,以映射至 3D 位置。

[0035] 本发明的其它方面与上述方法类似,其中使用 ID 矩阵码(或外观和/或位置是预定义或以其它方式已知的其它图案)来代替以上讨论的图案。在本发明的这些方面中,训练步骤被排除或简化。替代地,通过针对一般类型特征设计的视觉工具,根据训练阶段或运行时间图像辨别那些代码(或其它图案)的 2D 位置,以映射至 3D 位置。

[0036] 本发明的其它方面提供机器视觉系统,例如根据以上方法操作且包括数字处理功能和照相机的机器视觉系统。本发明的这些和其它方面在附图和以下描述中显而易见。

[0037] 本发明的另一相关方面提供此类方法和设备,这些方法和设备允许检查对象,例如,以确定和验证对象的多个部分的相对位置。作为非限制性示例,例如,在组装、质量保证、维护或其它操作期间,可使用此类方法和设备来支持检查和验证。

[0038] 本发明的其它相关方面提供此类方法和设备,在一个或多个预期图案(例如与对象的零件/部分相关联的图案)不存在于运行时间图像中或存在于那些图像中、但却处于映射至不预期或不想要的 3D 位置的像素位置的实例中,这些方法和设备推断对象的零件(或其它部分)的不存在或错位。

[0039] 本发明的其它相关方面提供此类方法和设备,其中在运行时间步骤期间,该对象的零件或其它部分的位置基于与运行时间图像中找到的图案相对应的 3D 位置的子集来确定,且其中使用那些 3D 位置来确定另外其它图案的预期位置。可将那些其它图案的预期位置与它们实际的 3D 位置(例如根据运行时间图像确定的 3D 位置)作比较。在该比较中确定的位置差别超过指定公差时,系统可产生适当的通知(例如向操作者通知)。

[0040] 根据本发明的系统和方法的优点在于,它们比现有技术系统方法更容易使用和更实用,而且是基于视觉的,因此不需要与受检查对象接触或预先准备。(根据本发明的)此类系统和方法能容易地建立,然后利用“显示继续(show-and-go)”训练。

[0041] 此外,例如,相对于丢失和不正确的结果,上述系统和方法提供迅速的性能和稳健性。因此,例如,即使因为一些图案被一个或多个照相机的视图遮挡或因为那些图案的图像不能被及时采集而在一些运行时间图像中(以及在一些或任何情况下)未找到这些图案,根据本发明多个方面的方法和设备也能确定对象的姿态。作为另一示例,根据本发明多个方面的方法和设备通过使用与在运行时间图像中找出的图案相对应的 3D 位置的子集来进行三角测距,而提供相对于不正确的结果(例如由未对准的照相机引起)的稳健性:如果子集之一具有较低误差平方和,则可将该子集用于位置三角测距,而不是使用所有图案。

[0042] 至少一些实施例包括一种用于在 3D 成像系统中指定感兴趣区域的方法,该 3D 成像系统包括多个照相机,这多个照相机至少包括第一和第二照相机,其中每个照相机具有沿不同照相机轨道排列的视场,该方法包括以下步骤:在多个照相机的视场内的某位置处呈现零件;在该零件上指示位于多个照相机中的每一个的视场内的感兴趣区域;针对多个照相机中的每一个:(i) 采集包括感兴趣区域的该零件的至少一幅图像,(ii) 在照相机的视场内标识与该至少一幅图像中的感兴趣区域相关联的照相机专属感兴趣视场;以及(iii) 存储该感兴趣视场以供以后使用。

[0043] 在一些实施例中,在零件上指示感兴趣区域的步骤包括在该零件上指示限定感兴趣区域的至少三个点。在一些实施例中,指示至少三个点的步骤包括使用激光设备来按顺序指示至少三个点中的每一个,且其中采集至少一幅图像的步骤包括针对至少三个点中的每一个采集单独图像。在一些实施例中,标识照相机专属感兴趣视场的步骤包括使用多幅采集到的图像来标识感兴趣视场。

[0044] 在一些情况下,该方法还包括以下步骤:对于每个点,确定何时该点已在每个照相机所采集的至少一幅图像中被标识,并指示何时该点已在每个照相机所采集的至少一幅图像中被标识。在一些情况下,该方法还包括以下步骤:对于每个点,确定何时该点已在至少一个照相机所采集的至少一幅图像中被标识同时该点尚未在至少另一个照相机所采集的至少一幅图像中被标识。

[0045] 在一些实施例中,在该零件上标识感兴趣区域的步骤包括,使用激光设备通过经由激光束照亮该零件的至少一部分来在该零件的表面上指示感兴趣区域。在一些实施例中,该激光设备包括按钮,该按钮可被选择以指示该零件的当前被照亮部分对应于定义信

息的至少一些感兴趣区域,且应当分析该按钮被选择时刻的该零件的图像以获得与该零件的当前被照亮部分相关联的位置信息,该方法还包括监测来自该激光设备的指示的步骤。

[0046] 在一些情况下,该激光设备被编程以在该按钮被选择时使该零件上的照射束的外观发生变化,监测来自激光设备的指示的步骤包括分析照相机图像以标识该零件上的照射束的外观的变化。在一些情况下,该方法还包括向系统用户提供反馈以指示何时已针对多个照相机中的每一个来标识照相机专属感兴趣视场的步骤。在一些实施例中,采集至少一幅图像的步骤包括迅速连续地采集图像,且其中指示至少三个点的步骤包括使用激光设备来独立地照亮三个点中的每一个点。在一些实施例中,指示至少三个点的步骤包括,对于每个点,当激光设备照亮该点时,指示被照亮位置是定义感兴趣区域的那些点之一。

[0047] 一些实施例包括用于在 3D 成像系统中指定感兴趣区域的方法,该 3D 成像系统包括多个照相机,该多个照相机包括至少第一和第二照相机,其中每个照相机具有沿照相机不同轨道排列的视场,该方法包括以下步骤:在照相机的视场内的一个位置呈现零件;使用激光设备按顺序并独立地在该零件上指示定义感兴趣区域的多个点;对于多个照相机中的每一个,在该零件上指示那些点的同时,采集包括感兴趣区域的该零件的图像;使用每个照相机的多幅图像中的那些点,在视场中标识与该感兴趣区域相对应的感兴趣视场;以及存储该视场中的该感兴趣视场作为照相机专属感兴趣视场。

[0048] 其它实施例包括一种用于训练三维视觉系统以限定照相机专属感兴趣视场的系统,其中在零件上指示了感兴趣区域,该系统包括:多个照相机,每个照相机具有视场且沿照相机不同的轨道排列,以使该感兴趣区域在该照相机的视场内,每个照相机用于在该照相机的视场内获得该零件的图像;以及机器视觉处理器,被编程为针对每个照相机执行以下步骤:(i) 接收图像;(ii) 使用所接收的至少一幅图像来标识用于产生与感兴趣区域相关联的图像的该照相机的视场的一部分作为感兴趣视场;以及(iii) 存储与该感兴趣区域相关联的感兴趣视场作为照相机专属感兴趣视场。

[0049] 在一些情况下,该系统还包括激光指示器设备,用于将激光束引导至该零件以在该零件上指示感兴趣区域。在一些情况下,该激光指示器设备包括按钮,该按钮可被选择以指示该零件的当前被照亮部分对应于定义信息的至少一些感兴趣区域,且应当分析该按钮被选择时刻的该零件的图像以获得与该零件的当前被照亮部分相关联的位置信息,该处理器进一步被编程以执行监测来自该激光指示器设备的关于应当分析那些图像的指示的步骤。在一些情况下,该激光指示器设备被编程以在该按钮被选择时使该零件上的照射束的外观发生变化,监测来自激光指示器设备的指示的步骤包括分析照相机图像以标识该零件上的照射束的外观的变化。在一些实施例中,该激光指示器设备包括在按钮被选择时发送信号的发射器。

附图说明

[0050] 通过参考附图可获得对本发明的更完整理解,在附图中:

[0051] 图 1 描绘了被配置用于实践本发明的数字数据处理器;

[0052] 图 2 是描绘根据本发明的系统的操作的流程图;

[0053] 图 3 描绘根据本发明的系统的校准;

[0054] 图 4A-4B 描绘根据本发明的系统的训练和 / 或运行时间操作;以及

[0055] 图 5 是示出用于训练 3D 成像系统中的多个照相机的不同感兴趣视场的方法的流程图。

具体实施方式

[0056] 图 1 描绘根据本发明的机器视觉系统 10, 其能通过对从对象 12 的多幅图像收集的数据的三角测距来确定该对象 12 在三维中的姿态。在所示实施例中, 该姿态是该对象在三维空间中的限定位置和取向, 或更准确地说, 该姿态是该对象沿 x、y 和 z 轴 14、16、18 的位置以及该对象相对于 x、y 和 z 轴的俯仰 (pitch)、滚转 (roll) 和偏航 (yaw)。在其它实施例中, 姿态可能限于这些空间特性的子集 (例如沿轴 16 的位置和偏航; 沿轴 14-18 的位置; 滚转等等)。所示轴 14-18 与视觉系统 10 的参照系 20 对准; 不过在其它实施例中, 也可使用其它参照系。所示实施例的参照系 20 在附图上通过利用加深阴影描绘的照相机支架 20a 和平台 20b 来表示; 不过, 在其它实施例中, 可采用其它构件或无此类构件。

[0057] 系统 10 还包括数字数据处理器 22 和图像采集设备 24。在此, 为简化起见而描述为 iMac®G5 个人计算机的数字数据处理器 22 可以是大型计算机、工作站、个人计算机 (例如运行 Windows®/ 英特尔奔腾 4 平台或其它)、专用视觉计算机、嵌入式处理器或运行私有、开源或其它操作系统的其它数字数据设备, 其被编程为或以其它方式被配置成根据本文的教导通过采集设备 24 提供的图像来确定对象 12 的姿态。该数字数据处理器可包括如图所示的显示器 22a、以及键盘 22b、鼠标 22c 以及本领域已知的所有其它类型的输入 / 输出设备。

[0058] 图像采集设备 24 可以是机器视觉照相机、摄像机、静态照相机或能在可见光或其它相关光谱中采集对象 12 的图像的其它设备。在不丧失一般性的情况下, 在下文中, 设备 24 通常被称为“照相机”——不过, 在实际中, 它们可包括任何方式的图像采集功能。在所示实施例中, 示出了三个此类设备 24, 不过在实际中, 可采用任意多个设备 (例如两个或更多个)。设置那些设备以从不同的各自视角采集对象 24 的图像。本领域普通技术人员还可理解, 在一些实施例中, 还可利用来自单个此类设备 24 的图像来确定受检查的对象的 3D 姿态, 因此并非所有实施例都需要来自多个照相机的图像。

[0059] 数字数据处理器 22 还包括被编程以进行根据本发明教导的操作的中央处理器 (CPU)、存储器 (RAM) 和本领域中已知类型的输入 / 输出 (I/O) 功能。

[0060] 具体而言, 在所示实施例中, 这些部件被配置成根据图 2 中所示方法提供 3D 机器视觉。在图 2 中, 在校准步骤 30 中, 多个照相机 24 中的每一个被校准以辨别各自的映射功能, 该映射功能在 3D 空间中标识从照相机透镜发出的且对应于该照相机视场中的像素位置的光线。

[0061] 参照图 3, 在此方面中, 校准步骤包括将对准靶 (诸如校准盘 40 或其它物体上的靶心、十字准线或类似物等) 定位在 3D 空间中的已知位置处, 并记录或以其他方式表征 (例如通过算法) 那些位置与相应靶在照相机视场中的像素方式位置之间的关联。优选地, 该映射考虑每个相应照相机 24 的视场中的变形。

[0062] 在所示实施例中, 对准靶、校准盘 40 等等用于例如通过同时成像来同时校准多个照相机 24。因此, 例如, 操作者可将靶放置在两个或更多个图像采集设备 24 的视场中, 这两个或更多个图像采集设备 24 对靶同时成像以进行校准。在使用校准盘 40 或类似物进行校

准的情况下,它们优选显示出位于原点的基准 41(例如与均匀棋盘图案不同的唯一图案),以使所有采集设备 24 能相对于具有指定取向的同一唯一基准点被校准。通过以一致方式如此校准多个设备 24,它们均可用于从它们的图像坐标(例如图案原点的像素方式位置)映射至同一基准点或坐标系。优选的此类基准 41 是不对称的,如附图中的 L 形状基准。

[0063] 通过非限制性示例,在以下专利文献中教导了用于此类校准的方法和设备,这些专利文献包括标题为“Methods and apparatus for machine vision inspection using single and multiple templates or patterns(用于使用单个和多个模板或图案进行机器视觉检查的方法和设备)”的美国专利 6,748,104、标题为“Machine vision methods for inspection of leaded components(用于对经引导的部件进行检查的机器视觉方法)”的美国专利 6,639,624,标题为“Nonfeedback-based machine vision methods for determining a calibration relationship between a camera and a moveable object(用于确定摄像机与可移动对象之间的校准关系的非基于反馈的机器视觉方法)”的美国专利 6,301,396、标题为“Machine vision calibration targets and methods of determining their location and orientation in an image(机器视觉校准靶和确定它们在图像中的位置和取向的方法)”的美国专利 6,137,893、标题为“Machine vision methods using feedback to determine calibration locations of multiple cameras that image a common object(使用反馈来确定对常见物体成像的多个摄像机的校准位置的机器视觉方法)”的美国专利 5,978,521、标题为“Machine vision methods using feedback to determine an orientation, pixel width and pixel height of a field of view(使用反馈来确定视场的取向、像素宽度和像素高度的机器视觉方法)”的美国专利 5,978,080、标题为“Nonfeedback-based machine vision method for determining a calibration relationship between a camera and a moveable object(用于确定摄像机与可移动物体之间的校准关系的非基于反馈的机器视觉方法)”的美国专利 5,960,125、标题为“Fast high-accuracy multi-dimensional pattern localization(快速高准确度多维图案局部化)”的美国专利 6,856,698、标题为“Fast high-accuracy multi-dimensional pattern inspection(快速高准确度多维图案检查)”的美国专利 6,850,646、以及标题为“Fast high-accuracy multi-dimensional pattern inspection(快速高准确度多维图案检查)”的美国专利 6,658,145,这些文献的教导通过引用结合于此。在后三篇专利中描述的方法和设备在本文中其它地方称为“PatMax”。

[0064] 在可任选的训练步骤 32 中,与每个相应照相机 24 相关联的模块(例如代码序列、子例程、功能、对象、其它数据结构和 / 或相关联的软件或其它功能)被训练成识别要在运行时间期间采集的对象 12 的图像中的预期图案。这些图案可以是字母、数字、其它符号(诸如对准靶)、隅角或其它特征(诸如暗斑和亮斑),期望这些特征在对象 12 的运行时间图像中可辨别,而且(例如)针对这些特征的测量技术和搜索 / 检测模型在本领域中是已知的。那些图案可以永久地是该对象的一部分或固定至该对象。然而,它们也可以是临时的,例如如同可移动校准靶的情况一样。实际上,它们甚至不需要与该对象在实体上关联。例如,它们可通过光学或其它方式投影到对象上,这些对象在训练和 / 或运行时间阶段期间通过激光器或其它设备成像。

[0065] 除训练与每个照相机 24 相关联的模块或其它功能以识别图案之外,所示实施例

的训练步骤 32 包括关于模型点位置训练它们,即那些图案例如在将在运行时间期间被检查的对象上相对于彼此(即在 3D 空间中)的预期位置。例如,这可通过利用与在运行时间阶段期间运用的三角测距方法相似的三角测距方法来实现。替代地,这些图案的预期(相对)位置可由操作者输入和/或通过其它测量方法(例如尺子、测径器、光学距离量规等等)来辨别。

[0066] 无论使用三角测距法还是其它方法,训练步骤 32 优选包括关于每个此类经训练图案的基准点(或“原点”)训练这些模块或其它功能或与每个照相机 24 关联的其它功能。对于预期出现在两个或更多个相机视场中的图案而言,关于此类基准点的训练便于在 3D 空间中对那些图案和/或对象的位置进行直接或间接的三角测距。

[0067] 在所实施例中,此类训练可通过使用相同模型(例如“PatMax”等等)训练不同照相机 24 之间的相同预期图案来实现。这具有确保在运行时间发现的图案的基准点(或原点)在由那些不同照相机获得的图像之间将一致的好处。

[0068] 在姿态、视角和/或障碍改变不同照相机 24 观察相同图案的方式的情况下,此类训练可包括在不同照相机之间利用不同模型(对于相同图案)。由于不同模型可能倾向于对相同图案标识不同基准点,所以所示实施例允许操作者训练针对相同图案选择相同基准点。

[0069] 例如,在训练步骤 32 期间,这可通过以下步骤实现:从多个照相机 24 同时采集该图案的图像(该图案将被用作步骤 34 中的搜索的模板),然后将激光指示器照射在该对象上。对于利用激光照射采集的图像,可计算激光点的 3D 位置,由此在这些图案的所有图像上限定一致的原点。(虽然在此关于将不同模型用于图案训练进行描述,但此技术也可应用于使用相同模型的实例)。为此,使用具有或不具有过多的激光指示器点的图像,可运行自动取阈值和斑点分析来找出在所有图像中的光斑的中心,由此确定一致的原点。如本文中别处所讨论,可使用三角测距来获得光斑的 3D 位置,由此允许将该技术用于所提供的(训练)对象上的多个图案,例如该对象不移动。

[0070] 例如,在训练步骤 32 期间,这可通过以下步骤实现:从多个照相机 24 同时采集该图案的图像(该图案将被用作步骤 34 中的搜索的模板),然后将激光指示器照射在该对象上。对于利用激光照射采集的图像,可计算激光点的 3D 位置,由此在这些图案的所有图像上限定一致的原点。(虽然在此关于将不同模型用于图案训练进行描述,但此技术也可应用于使用相同模型的实例)。为此,使用具有或不具有过多的激光指示器点的图像,可运行自动取阈值和斑点分析来找出在所有图像中的光斑的中心,由此确定一致的原点。如本文中别处所讨论,可使用三角测距来获得光斑的 3D 位置,由此允许将该技术用于所提供的(训练)对象上的多个图案,例如该对象不移动。

[0071] 根据本发明的一个优选实践,训练步骤 32 包括基于在来自另一照相机的图像中的该图案的先前标识来从来自一个(或多个)照相机 24 的图像中找出预期图案。因此,例如,一旦操作者已经在从一个照相机拍摄的图像中标识预期图案,训练步骤就可包括在来自其它照相机的图案中自动找出同一图案。

[0072] 在优选实施例中,优选地,步骤 32 包括对每个照相机 24 采集该对象的多个视图,以使在那些对象上找出的图案的原点被一致地定义。为解决多个图像之间潜在的不一致性,可使用对这些图案产生匹配分数最高的那些图像。这具有使该方法对找出处于任意姿

态的零件更稳健的好处。

[0073] 如上所述,训练步骤 32 是可选的:在本发明的一些实施例中,它在降低产量下采用,或完全不采用。例如,如果在运行时间时预期的图案易受经由斑点模型(例如寻找亮特征的模型)的搜索影响,则不需要图案训练——不过,仍可采用如上所述类型的位置训练。作为另一示例,如果图案是 ID 矩阵码(或使用外观和/或位置是预定义或以其它方式已知的其它图案来代替如上讨论的可训练图案),则也不需要图案训练。在此,例如,通过针对一般类型特征设计的视觉工具,根据训练阶段或运行时间图像辨别那些代码(或其它图案)的 2D 位置,以映射至 3D 位置。本发明的此类实现是有用的,因为工业零件可能总是已经具有 ID 矩阵码,且由此多个无需训练的传感器可输出 ID 矩阵码的 3D 位置。此外,由于 ID 矩阵码横跨矩形区域——所有传感器可输出 4 个隅角的 2D 位置;此外,如果 ID 矩阵利用特定工艺印刷,则我们可(借助代码的已知大小/类型)获知 3D 位置并计算 3D 姿态。

[0074] 在运行时间步骤 34,数字数据处理器 22 基于那些图案的表示在对象 12 的运行时间图像中的像素方式位置和根据校准步骤 32 期间辨别的映射对图案 42a-42c 中的一个或多个在对象 12 上的 3D 空间位置进行三角测距。在所示实施例中,对象 12 的那些运行时间图像优选由设备 24 同时或基本同时地采集。在此方面,基本同时指的是图像采集在时间上非常接近地进行,以致对象 12、设备 24、坐标系 20 等等的移动不会实际上影响图案在运行时间图像中的像素方式位置和/或由此确定的映射。这种同时采集可通过同时(或几乎同时)开启照相机 24 来实现,或通过其它方式来实现——其它方式包括例如,在照相机 24 快门打开的同时频闪地照亮所成像的对象。

[0075] 在所示实施例中,位置三角测距是如下完成的:使用图案匹配或其它二维视觉工具来辨别图案在运行时间图像中的像素方式位置,并使用在校准阶段期间辨别的映射来将那些图案在相应照相机 24 的视场中的像素方式位置与那些图案所依赖的上述 3D 光线相关联。使用此类 2D 视觉工具的示例包括上述通过引用结合的美国专利 6,748,104、6,639,624、6,301,396、6,137,893、5,978,521、5,978,080、5,960,125、6,856,698、6,850,646 以及 6,658,145。

[0076] 图案位置的三角测距可以通过“直接”的三角测距,例如根据图案所依赖的多条 3D 光线(来自多个照相机)的交点确定给定图案的位置。或者,或作为附加,三角测距可以是“间接的”,其中给定图案的位置不仅根据图案所依赖的光线(或多条光线)来确定,而且根据 (i) 其它图案所依赖的光线和 (ii) 那些图案在所成像对象上相对于彼此的位置(例如在训练阶段期间确定)。

[0077] 在所示实施例中,直接和/或间接三角测距可利用“最小平方拟合”或其它用于找出图案看起来所依赖的多条 3D 光线(来自多个照相机 24)之间的交点(或最近交点)的此类方法。例如,在从两个或多个照相机 24 采集的图像表明给定图案(或更准确地说,该图案的明显原点)依赖于两条或更多条光线的情况下,可采用最小平方拟合方法来确定那些光线在 3D 空间中的交点的位置或离该交点位置最近的点(即空间中最接近那些光线的点)。同样,在来自照相机 24 的图像表明多个图案的原点在多条光线上的情况下,可采用最小平方拟合方法利用那些图案在对象上的模型点来确定图案和/或对象本身的最可能位置。

[0078] 所示实施例利用优化器(或“解算器”)来找出光线和图案的最小平方(或均方根)拟合。这可以是本领域可获得的通用类型工具,和/或它可以按照以下详述的方式操

作。在任何情况下,在运行时间步骤 34 期间,给解算器提供从运行时间图像标识的图案(或更准确地说,图案原点)所依赖的 3D 光线的定义,以及(在相关的情况下)图案在对象上的位置或相对位置。

[0079] 通常,该信息定义过约束的系统(即通过光线定义和对象上的相对图案位置提供了比用来推断实际位置所需要的信息多的信息),所示系统利用该事实来实现稳健性。因此,例如,即使在图案从对象或其运行时间图像丢失的情况下(例如图案从一个或多个照相机视场被阻挡,或光照或其它条件不允许及时采集图案图像),运行时间步骤 34 也可确定对象姿态。而且,作为另一示例,运行时间步骤 34 可包括尝试由采集设备 24 找出的图案原点的子集(更准确地说,找出与图案原点相对应的位置的子集),以使光线与光线的模型点或三角测距之间的拟合的均方根(RMS)误差最小。如果子集之一具有较低误差平方和,则可将该子集用于位置三角测距,而不是使用所有图案原点。

[0080] 通常,该信息定义过约束的系统(即通过光线定义和对象上的相对图案位置提供了比用来推断实际位置所需要的信息多的信息),所示系统利用该事实来实现稳健性。因此,例如,即使在故意不考虑某些图案时(例如以通过将图案位置与另一发现图案所预测的位置相比较来检查/验证图案的位置),运行时间步骤 34 也可确定对象姿态。

[0081] 而且,作为另一示例,运行时间步骤 34 可包括尝试由采集设备 24 找出的图案原点的子集(更准确地说,找出与图案原点相对应的位置的子集),以使光线与光线的模型点或三角测距之间的拟合的均方根(RMS)误差最小。接着,该步骤可外推该子集中未包括的图案(即如上所述故意忽略的图案)的 3D 位置,并预测在它们相应照相机中的 2D 图像位置。所预测的图像位置可与实际测得的图像位置作比较:如果所预测的图像位置与实际测得的图像位置之间的距离超过一些用户指定的距离公差,则该系统可产生适当的警告或其它通知。替代地或作为附加,可将被忽略的图案的外推 3D 位置与通过三角测距确定的 3D 位置作比较;此外,在外推(预测)的位置与实际位置之间的距离不同的情况下,运行时间步骤 34 可包括产生适当的警告或其它通知。

[0082] 为了提高图案识别的速度,在训练和运行时间阶段期间,所示实施例可利用找出一个图案的位置来限制另一图案的搜索自由度。例如,如果第一照相机 24 在 15 度处找出该图案,且另一照相机大致处于与第一照相机相同的同一取向,则可能仅需要在 15+/-10 度的取向处寻找该图案。此外,给定来自一个照相机的原点位置,我们知道该原点将位于 3D 光线;由此,我们可将该光线投影到第二照相机的视场上,且仅沿该直线寻找该图案。

[0083] 如果两个图案容易混淆(即存在与在该照相机的视场中的该图案相似的两个实例),则所示实施例可尝试所有不同的可能的对应关系。例如,使用上述通过引用结合的标题为“Fast high-accuracy multi-dimensional pattern localization(快速高准确度多维图案局部化)”的美国专利 6,856,698、标题为“Fast high-accuracy multi-dimensional pattern inspection(快速高准确度多维图案检查)”的美国专利 6,850,646、标题为“Fast high-accuracy multi-dimensional pattern inspection(快速高准确度多维图案检查)”的美国专利 6,658,145 中描述的技术,诸如字符序列“P”、“ST”、“It”以及“便签”(来自 POST-IT® 便签标签)之类的图案均不同,因此当找到匹配时,我们知道它是正确的匹配。

[0084] 替代地,称为“斑点”分析的机器视觉工具可用于找出图案(例如如果它们是暗孔)。在此情况下,可假设斑点 #1 对应于 3D 模型点 1,而斑点 #2 对应于 3D 模型点 2 等等。

如果不起作用,则分析可继续至下一假设:斑点 #2 对应于 3D 模型点 #2,而斑点 #1 对应于 3D 模型点 #1。

[0085] 根据以下讨论可获得对所示实施例的三角测距过程的更完整理解。

[0086] 为了使 n 条 3D 光线相交(即为了找出使与 n 条 3D 光线的距离平方和最小的点),首先将每条光线表征为两个不同的正交面(由于从点到线的距离的平方是该点到与该光线相交的两个正交面的距离的平方和)。这通过以下的 C++ 代码为例:

[0087]

```
static void solveMinimizeSquareErrorEquations(  
    const cmStd vector<cc3Vect> &eqs,  
    const cmStd vector<double> &val,  
    cc3Vect &ans,  
    double &rmsError)  
{  
    int size = eqs.size();  
    assert(size==val.size());
```

[0088]

```
ccNMMatrix func(size,3);
ccNMMatrix vals(size,1);
int i;
for (i = 0; i < size; ++i)
{
    func.element(i,0,eqs[i].x());
    func.element(i,1,eqs[i].y());
    func.element(i,2,eqs[i].z());
    vals.element(i,0,val[i]);
}

ccNMMatrix func2 = func.Tmultiply(func);
ccNMMatrix vals2 = func.Tmultiply(vals);

double cnum;
ccNMMatrix result =
    ccNMMatrix::solve
(func2,vals2,true/*symmetric*/,cnum,HUGE_VAL);

//包含退化情况
if (func2.conditionNumber(false) > 1e12)
    throw ccMathError::Singular();

ans = cc3Vect(result.element(0,0),
               result.element(1,0),
               result.element(2,0));

double sumSqError = 0;
for (i = 0; i < size; ++i)
{
```

[0089]


```
        double error = eqs[i].dot(ans)-val[i];
        sumSqError += error*error;
    }
    rmsError = cfSqrt(sumSqError/size);
}

void cfComputePerpendiculars(const cc3Vect &start1,
                             const cc3Vect &end1,
                             cc3Vect &perp1,
                             cc3Vect &perp2)
{
    cc3Vect diff = end1 - start1;
    cc3Vect axes[3] = {cc3Vect(1,0,0),
                      cc3Vect(0,1,0),
                      cc3Vect(0,0,1)};

    cc3Vect crosses[3];
    double lens[3];
    int i;
    for (i = 0; i < 3; ++i)
        crosses[i] = diff.cross(axes[i]);
    for (i = 0; i < 3; ++i)
        lens[i] = crosses[i].dot(crosses[i]);
    int j;
    for (i = 0; i < 2; ++i)
        for (j = i+1; j < 3; ++j)
            if (lens[i] < lens[j])
            {
                double tmp = lens[i];
                lens[i] = lens[j];
            }
}
```

[0090]

```

        lens[j] = tmp;
        cc3Vect cross = crosses[i];
        crosses[i] = crosses[j];
        crosses[j] = cross;
    }
    perp1 = crosses[0] * 1./lens[0];
    perp2 = crosses[1] * 1./lens[1];
}

static void intersectLinesInThreeSpace
(const cmStd vector<cc3DLineSeg> &segs,
 cc3Vect &intersection,
 double &rmsError)
{
//找出使 segs.size()*2 个屏幕之间的平方和误差最小的点
    int size = segs.size();
    cmStd vector<cc3Vect> eqs(size*2);
    cmStd vector<double> vals(size*2);
    int i;
    for (i = 0; i < size; ++i)
    {

cfComputePerpendiculars(segs[i].p1(),segs[i].p2(),eqs[i*2],eqs[i*2+1]);
        vals[i*2] = segs[i].p1().dot(eqs[i*2]);
        vals[i*2+1] = segs[i].p1().dot(eqs[i*2+1]);
    }

    solveMinimizeSquareErrorEquations(eqs,vals,intersection,rmsError);
}

```

[0091] 为了求解将 3D 点最佳映射到相应 3D 光线的姿态,我们可以说我们使用以下方程(这些方程以 Maple math package 表示,该软件可从 Waterloo Maple 公司的子公司

Maplesoft 买到), 这些方程用于产生最优 C 代码。

[0092] 该方法求解使点 p (表示为 x, y, z) 与表示为 (px, py, pz, pt) 的平面之间的平方和误差最小的姿态 (按照变量 a, b, c, d, tx, ty, tz 表示)。注意, 每个 3D 光线对应于两个此类平面约束。该方法通过对代数误差函数的系数求和来计算平方和误差。然后, 该方法利用梯度下降来求解最优 a, b, c, d, tx, ty, tz 。注意, 由于存在 7 个变量 (a, b, c, d, tx, ty, tz) 且仅有 6 个自由度, 因此我们尝试了 4 种不同情况——其中 a 被设定为 1 且变量为 b, c, d 的情况; 其中 b 被设定为 1 且变量为 a, c, d 的情况; 其中 c 被设定为 1 且变量为 a, b, d 的情况; 以及其中 d 被设定为 1 且变量为 a, b, c 的情况。

```
[0093] quatRot :=
[0094] matrix(4,4,
[0095] [[a*a+b*b-c*c-d*d, 2*b*c-2*a*d, 2*a*c+2*b*d, tx],
[0096] [2*a*d+2*b*c, a*a-b*b+c*c-d*d, 2*c*d-2*a*b, ty],
[0097] [2*b*d-2*a*c, 2*a*b+2*c*d, a*a-b*b-c*c+d*d, tz],
[0098] [0,0,0, a*a+b*b+c*c+d*d]]);
[0099] val := matrix(1,4, [x, y, z, 1]);
[0100] dot := matrix(1,4, [px, py, pz, pt]);
[0101] pos := multiply(quatRot, transpose(val));
[0102] unit := (a*a+b*b+c*c+d*d);
[0103] weightMat := multiply(dot, pos);
[0104] weight := weightMat[1,1]/unit;
[0105] weightSq := expand(weight*weight);
[0106] weightSqDA := simplify(expand(diff(weightSq, a)*unit*unit));
[0107] weightSqDB := simplify(expand(diff(weightSq, b)*unit*unit));
[0108] weightSqDC := simplify(expand(diff(weightSq, c)*unit*unit));
[0109] weightSqDD := simplify(expand(diff(weightSq, d)*unit*unit));
[0110] weightSqDTX := simplify(expand(diff(weightSq, tx)*unit*unit));
[0111] weightSqDTY := simplify(expand(diff(weightSq, ty)*unit*unit));
[0112] weightSqDTZ := simplify(expand(diff(weightSq, tz)*unit*unit));
[0113] 鉴于下文将可进一步理解上述内容, 其中 GenericPoly() 是提取函数的系数的函数。因此, 如果该函数是  $x*x+2*x*y+y*y$ , 则其类函数 (generic function) 是  $f_{0x2y0}*x*x+f_{0x1y1}*x*y+f_{0x0y2}*y*y$ , 其中  $f_{0x2y0} = 1, f_{0x1y1} = 2, f_{0x0y2} = 1$ 。GenericPoly() 被包括在 MARS 中, MARS 即 Maple Matlab Resultant Solver 系统, 作为非限制性示例, 该系统例如可从 www.cs.unc.edu/~geom/MARS 公开和自由地获得。
[0114] read(`genericpoly.map`);
[0115] genPoly :=
[0116] GenericPoly(expand(eval(weightMat[1,1]*weightMat[1,1])),
[0117] array(1..7, [a, b, c, d, tx, ty, tz]), ' f' );
[0118] genPolyDA := simplify(expand(unit*diff(eval(genPoly[1]),
a)-eval(genPoly[1])*4*a));
```

[0119] `genPolyDB := simplify(expand(unit*diff(eval(genPoly[1]), b)-eval(genPoly[1])*4*b));`

[0120] `genPolyDC := simplify(expand(unit*diff(eval(genPoly[1]), c)-eval(genPoly[1])*4*c));`

[0121] `genPolyDD := simplify(expand(unit*diff(eval(genPoly[1]), d)-eval(genPoly[1])*4*d));`

[0122] `genPolyDTX := simplify(expand(diff(eval(genPoly[1]), tx)));`

[0123] `genPolyDTY := simplify(expand(diff(eval(genPoly[1]), ty)));`

[0124] `genPolyDTZ := simplify(expand(diff(eval(genPoly[1]), tz)));`

[0125] `C(genPolyDA);`

[0126] `C(genPolyDB);`

[0127] `C(genPolyDC);`

[0128] `C(genPolyDD);`

[0129] `C(genPolyDTX);`

[0130] `C(genPolyDTY);`

[0131] `C(genPolyDTZ);`

[0132] (注意, 由于误差函数不是 `derivative(weightMat*weightMat)`, 而实际上是 `(a*a+b*b+c*c+d*d)*derivative(weightMat*weightMat)-4*a*weightMat*weightMat` (用于计算相对于 a 的偏导数) (写为:

[0133] `unit*diff(eval(genPoly[1]), a)-eval(genPoly[1])*4*a);`)

[0134] 由于商的链式法则:

[0135] $\text{deriv}(F(x)/G(x)) = (G(x)*F'(x) - F(x)*G'(x)) / (G(x)*G(x))$

[0136] 注意, 对于此分析, 我们可以忽略分母的平方 (在链式法则针对偏导数的分母中), 因为分母 `(a*a+b*b+c*c+d*d)` 对所有偏导数一律适用:

[0137] $d((a*a+b*b+c*c+d*d)^2)/da = 4*a*(a*a+b*b+c*c+d*d)$

[0138] 作为对上述内容的进一步说明, 数值梯度下降方法利用误差函数以及该误差函数的导数。误差函数的导数可通过数字方式或符号方式来计算。对于数值方式计算的导数, 可仅将变量之一改变少量, 然后重新计算误差函数, 由此通过数值方式计算导数。对于通过符号方式计算的导数, 需要与该导数对应的符号函数——在此情况下我们具有该导数, 因为我们具有描述误差函数的代数表达, 而且我们能对该代数误差函数以符号方式求微分。

[0139] 在所示实施例中, C 数据结构保持该代数表达的所有系数, 如下:

[0140] `typedef struct ptRayGenericPoly`

[0141] `{`

[0142] `double fa0b0c0d0tx0ty0tz2;`

[0143] `double fa0b0c0d0tx0ty1tz1;`

[0144] `double fa0b0c0d0tx0ty2tz0;`

[0145] `double fa0b0c0d0tx1ty0tz1;`

[0146] `double fa0b0c0d0tx1ty1tz0;`

[0147] `double fa0b0c0d0tx2ty0tz0;`

[0148] double fa0b0c0d2tx0ty0tz1 ;
[0149] double fa0b0c0d2tx0ty1tz0 ;
[0150] double fa0b0c0d2tx1ty0tz0 ;
[0151] double fa0b0c0d4tx0ty0tz0 ;
[0152] double fa0b0c1d1tx0ty0tz1 ;
[0153] double fa0b0c1d1tx0ty1tz0 ;
[0154] double fa0b0c1d1tx1ty0tz0 ;
[0155] double fa0b0c1d3tx0ty0tz0 ;
[0156] double fa0b0c2d0tx0ty0tz1 ;
[0157] double fa0b0c2d0tx0ty1tz0 ;
[0158] double fa0b0c2d0tx1ty0tz0 ;
[0159] double fa0b0c2d2tx0ty0tz0 ;
[0160] double fa0b0c3d1tx0ty0tz0 ;
[0161] double fa0b0c4d0tx0ty0tz0 ;
[0162] double fa0b1c0d1tx0ty0tz1 ;
[0163] double fa0b1c0d1tx0ty1tz0 ;
[0164] double fa0b1c0d1tx1ty0tz0 ;
[0165] double fa0b1c0d3tx0ty0tz0 ;
[0166] double fa0b1c1d0tx0ty0tz1 ;
[0167] double fa0b1c1d0tx0ty1tz0 ;
[0168] double fa0b1c1d0tx1ty0tz0 ;
[0169] double fa0b1c1d2tx0ty0tz0 ;
[0170] double fa0b1c2d1tx0ty0tz0 ;
[0171] double fa0b1c3d0tx0ty0tz0 ;
[0172] double fa0b2c0d0tx0ty0tz1 ;
[0173] double fa0b2c0d0tx0ty1tz0 ;
[0174] double fa0b2c0d0tx1ty0tz0 ;
[0175] double fa0b2c0d2tx0ty0tz0 ;
[0176] double fa0b2c1d1tx0ty0tz0 ;
[0177] double fa0b2c2d0tx0ty0tz0 ;
[0178] double fa0b3c0d1tx0ty0tz0 ;
[0179] double fa0b3c1d0tx0ty0tz0 ;
[0180] double fa0b4c0d0tx0ty0tz0 ;
[0181] double falb0c0d1tx0ty0tz1 ;
[0182] double falb0c0d1tx0ty1tz0 ;
[0183] double falb0c0d1tx1ty0tz0 ;
[0184] double falb0c0d3tx0ty0tz0 ;
[0185] double falb0c1d0tx0ty0tz1 ;
[0186] double falb0c1d0tx0ty1tz0 ;

```
[0187]    double fa1b0c1d0tx1ty0tz0 ;
[0188]    double fa1b0c1d2tx0ty0tz0 ;
[0189]    double fa1b0c2d1tx0ty0tz0 ;
[0190]    double fa1b0c3d0tx0ty0tz0 ;
[0191]    double fa1b1c0d0tx0ty0tz1 ;
[0192]    double fa1b1c0d0tx0ty1tz0 ;
[0193]    double fa1b1c0d0tx1ty0tz0 ;
[0194]    double fa1b1c0d2tx0ty0tz0 ;
[0195]    double fa1b1c1d1tx0ty0tz0 ;
[0196]    double fa1b1c2d0tx0ty0tz0 ;
[0197]    double fa1b2c0d1tx0ty0tz0 ;
[0198]    double fa1b2c1d0tx0ty0tz0 ;
[0199]    double fa1b3c0d0tx0ty0tz0 ;
[0200]    double fa2b0c0d0tx0ty0tz1 ;
[0201]    double fa2b0c0d0tx0ty1tz0 ;
[0202]    double fa2b0c0d0tx1ty0tz0 ;
[0203]    double fa2b0c0d2tx0ty0tz0 ;
[0204]    double fa2b0c1d1tx0ty0tz0 ;
[0205]    double fa2b0c2d0tx0ty0tz0 ;
[0206]    double fa2b1c0d1tx0ty0tz0 ;
[0207]    double fa2b1c1d0tx0ty0tz0 ;
[0208]    double fa2b2c0d0tx0ty0tz0 ;
[0209]    double fa3b0c0d1tx0ty0tz0 ;
[0210]    double fa3b0c1d0tx0ty0tz0 ;
[0211]    double fa3b1c0d0tx0ty0tz0 ;
[0212]    double fa4b0c0d0tx0ty0tz0 ;
```

```
[0213] }ptRayGenericPoly ;
```

[0214] 所示实施例还利用添加代数表达式的系数的函数（那些函数以 3D 点 (x, y, z) 和相应的平面（通过 px, py, pz, pt 表征）作为输入）：

```
[0215] void ptRayGenericPoly_addToVals(ptRayGenericPoly*vals,
[0216]    double x,
[0217]    double y,
[0218]    double z,
[0219]    double px,
[0220]    double py,
[0221]    double pz,
[0222]    double pt) ;
```

[0223] 所示实施例还利用计算给定姿态 (a, b, c, d, tx, ty, tz) 处的误差的函数，其中 (a, b, c, d) 是 3D 旋转（滚转、俯仰、偏航）的四元表示，而 (tx, ty, tz) 是平移：

```
[0224] double ptRayGenericPoly_eval(const ptRayGenericPoly*vals,  
[0225] double a,  
[0226] double b,  
[0227] double c,  
[0228] double d,  
[0229] double tx,  
[0230] double ty,  
[0231] double tz);
```

[0232] 所示实施例还提供计算导数的函数：

```
[0233] double ptRayGenericPoly_DA(const ptRayGenericPoly*vals,  
[0234] double a,  
[0235] double b,  
[0236] double c,  
[0237] double d,  
[0238] double tx,  
[0239] double ty,  
[0240] double tz);
```

```
[0241] double ptRayGenericPoly_DB(const ptRayGenericPoly*vals,  
[0242] double a,  
[0243] double b,  
[0244] double c,  
[0245] double d,  
[0246] double tx,  
[0247] double ty,  
[0248] double tz);
```

```
[0249] double ptRayGenericPoly_DC(const ptRayGenericPoly*vals,  
[0250] double a,  
[0251] double b,  
[0252] double c,  
[0253] double d,  
[0254] double tx,  
[0255] double ty,  
[0256] double tz);
```

```
[0257] double ptRayGenericPoly_DD(const ptRayGenericPoly*vals,  
[0258] double a,  
[0259] double b,  
[0260] double c,  
[0261] double d,  
[0262] double tx,
```

```
[0263]         double ty,  
[0264]         double tz);  
[0265] double ptRayGenericPoly_DTX(const ptRayGenericPoly*vals,  
[0266]         double a,  
[0267]         double b,  
[0268]         double c,  
[0269]         double d,  
[0270]         double tx,  
[0271]         double ty,  
[0272]         double tz);  
[0273] double ptRayGenericPoly_DTY(const ptRayGenericPoly*vals,  
[0274]         double a,  
[0275]         double b,  
[0276]         double c,  
[0277]         double d,  
[0278]         double tx,  
[0279]         double ty,  
[0280]         double tz);  
[0281] double ptRayGenericPoly_DTZ(const ptRayGenericPoly*vals,  
[0282]         double a,  
[0283]         double b,  
[0284]         double c,  
[0285]         double d,  
[0286]         double tx,  
[0287]         double ty,  
[0288]         double tz);
```

[0289] 例如,添加系数的函数可按照与以下示例摘录一致的方式来表达;其完整函数根据本发明的教导和所提供的 Maple 代码可显而易见:

```
[0290] void ptRayGenericPoly_addToVals(ptRayGenericPoly*vals,  
[0291]         double x,  
[0292]         double y,  
[0293]         double z,  
[0294]         double px,  
[0295]         double py,  
[0296]         double pz,  
[0297]         double pt)  
[0298] {  
[0299]     double x_2 = x*x;  
[0300]     double y_2 = y*y;
```



```

[0301] double z_2 = z*z ;
[0302] double px_2 = px*px ;
[0303] double py_2 = py*py ;
[0304] double pz_2 = pz*pz ;
[0305] double pt_2 = pt*pt ;
[0306] vals-> fa0b0c0d0tx0ty0tz2+ = pz_2 ;
[0307] vals-> fa0b0c0d0tx0ty1tz1+ = 2*py*pz ;
[0308] vals-> fa0b0c0d0tx0ty2tz0+ = py_2 ;
[0309] vals-> fa0b0c0d0tx1ty0tz1+ = 2*pz*px ;
[0310] vals-> fa0b0c0d0tx1ty1tz0+ = 2*py*px ;
[0311] vals-> fa0b0c0d0tx2ty0tz0+ = px_2 ;
[0312] vals-> fa0b0c0d2tx0ty0tz1+ = 2*pt*pz+2*pz_2*z-2*py*y*pz-2*px*x*pz ;
[0313] vals-> fa0b0c0d2tx0ty1tz0+ = 2*pt*py-2*py_2*y-2*px*x*py+2*pz*z*py ;
[0314] vals-> fa0b0c0d2tx1ty0tz0+ = 2*pt*px-2*px_2*x-2*py*y*px+2*pz*z*px ;
[0315] vals-> fa0b0c0d4tx0ty0tz0+ = pz_2*z_2+px_2*x_2+py_2*y_2+pt_2+2*pz*z
*pt-
[0316] 2*px*x*pt-2*py*y*pt+2*px*x*py*y-2*px*x*pz*z-
[0317] 2*py*y*pz*z ;
[0318] vals-> fa0b0c1d1tx0ty0tz1+ = 4*pz*z*py+4*pz_2*y ;
[0319] ...

```

[0320] 注意,其起作用的方式是那些 fa...’ s 是系数,且那些变量的名称编码了每个以下多项式的次数

[0321] $F(a, b, c, d, tx, ty, tz) = fa0b0c0d0tx0ty0tz0+fa0b0c0d0tx0ty0tz1*tz+fa0b0c0d0tx0ty0tz2*tz^2+fa0b0c0d0tx0ty1tz1*ty*tz+\dots$

[0322] 而且从以下一般零误差函数开始

[0323] $fa0b0c0d0tx0ty0tz0 = 0$

[0324] $fa0b0c0d0tx0ty0tz1 = 0$

[0325] ...

[0326] 接着,针对每组点、平面(通过 $(x, y, z, px, py, pz, pt)$ 表征)均调用函数 `ptRayGenericPoly_addToVals()`,且它将单项式系数累加到误差和函数中。

[0327] 参照图 2 的可选步骤 36,例如,如果相机之一在校准之后推进,则所示系统 10 可重新校准失去对准的图像采集设备 24。(在此方面,将可理解在所示实施例中,优选照相机 24 在校准之后保持固定和/或静态,以保持在校准阶段期间确定的映射)。这可在步骤 34 中执行的位置三角测距期间检测,且可能在步骤 32 中执行的三角测距期间检测。为此,如果一个照相机 24 产生这样的图像,即在这些图像中在受检查对象的运行时间图像中找出的图案看起来处于与来自其它照相机的图像(例如当使用它们相应的 3D 光线映射时)不一致和/或显著不符的位置处(例如当映射至该照相机的 3D 光线),则可使用利用来自那些其它照相机的图像确定的图案位置来使用如上联系步骤 30 所述的同一校准方法来重新校准那个相机。

[0328] 在此方面,将可理解,在步骤 30(以及步骤 36)中确定的映射可分解为两个不同效果:透镜畸变,诸如枕形失真和其它图像失常,如果照相机 24 推近则该枕形失真和其它图像失常将保持恒定(因为它仅仅是透镜函数,且因为例如透镜可相对于 CCD、CMOS 或其它图像传感器被胶水固定在适当位置);以及照相机 24 相对于工作场所的姿态。如果照相机推近,则后者即姿态将改变。在步骤 36,例如,可归因于工作场所中的照相机姿态的映射的方面可被重新计算,例如不需要校准盘,因为透镜畸变被认为是保持恒定。

[0329] 所示系统 10 优选包括“超时”特征,该特征防止在运行时间阶段 34 期间采集的图像中未检测到预期图案实例下的过度延迟。在此方面,系统简单地将未在指定延迟间隔(例如由操作员设定或以其它方式设定)内检测到的图案视为未找到,并基于另一找到的图案来继续位置确定。这具有使系统 10 相对于丢失特征更稳健和其操作更及时的好处。

[0330] 在至少一些实施例中,特定的机器视觉过程可能需要系统用户在所获得的待检查或分析零件的图像中标识一个或多个特定区域,或通过系统用户在所获得的待检查或分析零件的图像中标识一个或多个特定区域来最优化。例如,在被检查的第一零件的情况下,系统用户可能知道感兴趣的图案将始终在照相机的视场内,因此该图案将出现在该照相机所产生的图像的左上四分之一部分。在此情况下,为了加速图案识别过程,照相机处理器将对该图案的搜索限制于所获得图像的左上四分之一部分将是最优的。这里,在第二照相机与第一照相机同时获得第一零件的实例的图像的情况下,该图案可能始终出现在第二照相机视场的右下四分之一部分中,因此处理来自第二照相机的图像的处理器将对该图案的搜索限制于该右下四分之一角落中会是最优的。同样,在系统包括第三、第四照相机的情况下,对图案识别的图像处理可能最优地被仅限制于照相机图像的多个部分。

[0331] 在至少一些实施例中,可构想在训练过程期间,用户可执行感兴趣(AOI)训练过程,利用定位在两个或更多个照相机的视场内的理想零件,用户手动地标识该零件上的多个点,这多个点在所获得的图像中限定应被检查或分析的感兴趣区域。图像是利用两个或更多个照相机获得的,其中用户所选择的多个点可由计算机 CPU/ 处理器分辨(参见图 1)。之后,处理器存储在所获得图像中选择的地点的坐标,以针对每个照相机定义与感兴趣区域相对应的各个“感兴趣视场(FOI)”。

[0332] 虽然各个图像中的点对应于被成像的零件上的相同选定点,但与选定点对应的感兴趣视场(即图像的多个部分)对每个照相机而言不同,因为每个照相机相对于该零件和选定点取向不同。因此,例如,虽然四个选定点可在利用第一照相机通过其沿第一轨道安排的视场产生的图像中限定正方形,但相同的四个点可能在利用第二照相机通过其沿第二轨道安排的视场产生的图像中限定矩形、梯形或其它四边形。

[0333] 应当理解,该 AOI 训练过程相对简单,且确保多照相机 3D 系统中的每个照相机被校准,以检查或分析零件的完全同一区域或部分。应当理解,在单个 AOI 训练过程期间,所有感兴趣的照相机特定域可被同时训练。

[0334] 在投产过程期间,可构想若干不同方法以用于指示或指定 AOI。例如,在一些情况下,系统用户可采用手动激光指示器 80 在零件 12 上指出多个选定点,且每次指定一个点。在此,虽然第一点是经由激光指示器选择的,但照相机可被控制以获得同时图像。

[0335] 在至少一些情况下,手动激光指示器可包括发射器和按钮,其中当选择该按钮时,指示器发射器向连接至计算机 22a 的接收器发射信号(再次参见图 1),指示与该时刻对应

的图像应当被检查以在激光照亮点的图像内标识多个位置。在其它情况下,激光器在所获得的一系列图像内的一些活动可指示计算机处理器已选择了点。例如,指示器 80 可打开,并产生稳定状态的激光束,而当选定指示器按钮时,该指示器 80 可闪烁两次。在此,在每个照相机在 AOI 训练过程期间迅速连续获得一系列图像的情况下,计算机处理器可被编程为将该闪烁的照亮点识别为该点已被选择的信号,且相关的图像应被检查以在激光照亮点的图像内标识多个位置。在其它情况下,用户可简单地通过键盘 22b 选择一键,以指示应分析特定时刻下的图像以标识每个图像中的照亮点的位置。

[0336] 接着,用户可使用指示器 80 来选择第二区域限定点,并使照相机同时获得图像,接着选择所需的第三点,等等。一旦包括定义 AOI 所需的所有点的图像已被捕获,计算机处理器就可标识各个不同照相机的 FOI 以供后续使用。在已存储照相机专属的 FOI 之后,在正常操作期间,当零件随后被成像时,处理器可仅分析特定照相机的图像中的 FOI。

[0337] 虽然一些实施例可能要求用户选择四个点来限定大致为直线的 AOI,但在至少一些实施例中,用户可斟酌使用更多或更少点来限定其它 AOI 形状。例如,可指定 / 选择三个点来规定三角形,五个点可规定五角形等等。

[0338] 作为另一实例,激光源可被编程以迅速地扫描激光束,以形成正方形(参见图 1 的 82)、矩形、三角形、圆形等等 AOI 边界,其中该 AOI 边界被导向并对着零件 12 的表面以限定 AOI。在此,每个照相机可能仅需要拍摄该零件的单幅图像,且照相机处理器可被编程以根据该图像来“学习”该照相机专属的 FOI。在此,同样,表明 AOI 已被选择的信号可通过激光指示器向计算机处理器的传输、当指示器按钮被选择时计算机处理器所感测到的一些活动(例如 AOI 边界的闪烁、AOI 边界强度的变化、从 AOI 边界到边界限定点集合的迅速变化)、计算机键盘按钮的选择等等来促成。

[0339] 作为另一其它实例,系统用户可在 AOI 训练程序期间将胶粘物或类似物涂敷到零件上。在胶粘物被涂敷以限定 AOI 之后,经由多个照相机获得图像,计算机处理器在每幅图像中标识与胶粘物位置相对应的照相机专属 FOI,随后相应地使用这些照相机专属 FOI。

[0340] 在至少一些实施例中,可构想每个照相机可在 AOI 训练程序期间操作,迅速连续地获得多幅图像。在此情况下,在至少一些实施例中,可在获得图像的同时在 AOI 周围使用激光器“绘制”方框、圆圈等等形状的 AOI 边界,其中计算机处理器被编程为使用来自每个照相机的多幅图像的激光点位置信息来创建照相机专属的 FOI。

[0341] 在至少一些实施例中,可构想计算机处理器可被编程为在 AOI 训练程序期间向系统用户提供至少一些反馈。例如,在系统包括三个照相机且用户经由激光指示器选择一个点的情况下,如果照相机获得图像但与一个或多个照相机相关联的图像不包括在用户所指示的时刻由激光所照亮的点,则处理器可通过蜂鸣或类似方式指示应当选择不同点,或处理器可不指示该点被俘获(即可能仅在所有照相机的图像中的点已被标识时才提供蜂鸣),由此提示用户选择另一点。作为另一实例,在与三个系统照相机中的两个照相机相关联的图像可用于分辨四个所选择点,且与第三照相机相关联的图像仅可用于在同时获得的图像中分辨两个所选择点的情况下,处理器可指示训练过程不成功。作为另一个实例,在使用激光在零件上“绘制”AOI 边界、同时照相机拍摄迅速连续的图像的情况下,处理器可被编程以识别轮廓已封闭(即出现交点),且可通过蜂鸣或类似方式指示封闭。反馈也可以是通过计算机显示屏 22a 的视觉方式。

[0342] 在至少一些实施例中,除使用 AOI 训练程序来限定照相机专属 AOI 以供后续使用之外,可在图案 / 模型训练会话期间使用类似的训练程序来表明零件上应当创建模型图案的区域 / 区。例如,用户可采用激光指示器 80 或类似物在包括特定感兴趣特征的零件上限定 AOI,然后计算机处理器可针对由用户指定的每个照相机专属 FOI 运行独立的模型产生过程。

[0343] 参照图 5,示出了示例性的 AOI 训练程序 50。在框 52,开始该 AOI 训练程序。在此,为了开始该程序,用户可通过计算机 22(或通过激光指示器 80)指示该程序应当开始(参见图 1)。在框 54,用户指示零件(参见图 1 中的 12)上的处于系统照相机 24 的视场内的 AOI 点。在本示例中,用来限定 AOI 的点通过如图 1 所示的激光指示 80 来指示,不过也可使用以上教导的任何其它方式或与以上教导等效的方式。

[0344] 在框 56,使用照相机 24 来获得零件 12 的图像。在框 58,通过计算机处理器来检查这些图像,以标识限定 AOI 的选定点。在框 60,处理器确定 AOI 限定点是否已经位于与每个照相机相关联的至少一个图像中。在 AOI 限定点尚未全部标识的情况下,控制向上返回至框 54,在框 54 处该过程继续循环。最优地,在框 55,可向系统用户提供反馈。一旦已经标识所有 AOI 限定点,就在框 62 存储每个照相机的 FOI,且在框 64 向用户提供表明 AOI 训练程序已针对当前 AOI 完成的反馈(例如蜂鸣)。在用户想要针对该零件限定第二 AOI 的情况下,可重复该过程 50。

[0345] 以上描述的是满足上述目标的方法和设备等等。将理解,在附图中示出和以上描述的方法和设备仅仅是本发明的实施例的示例,且包含对这些实施例的改变的其它实施例也落在本发明的范围内。

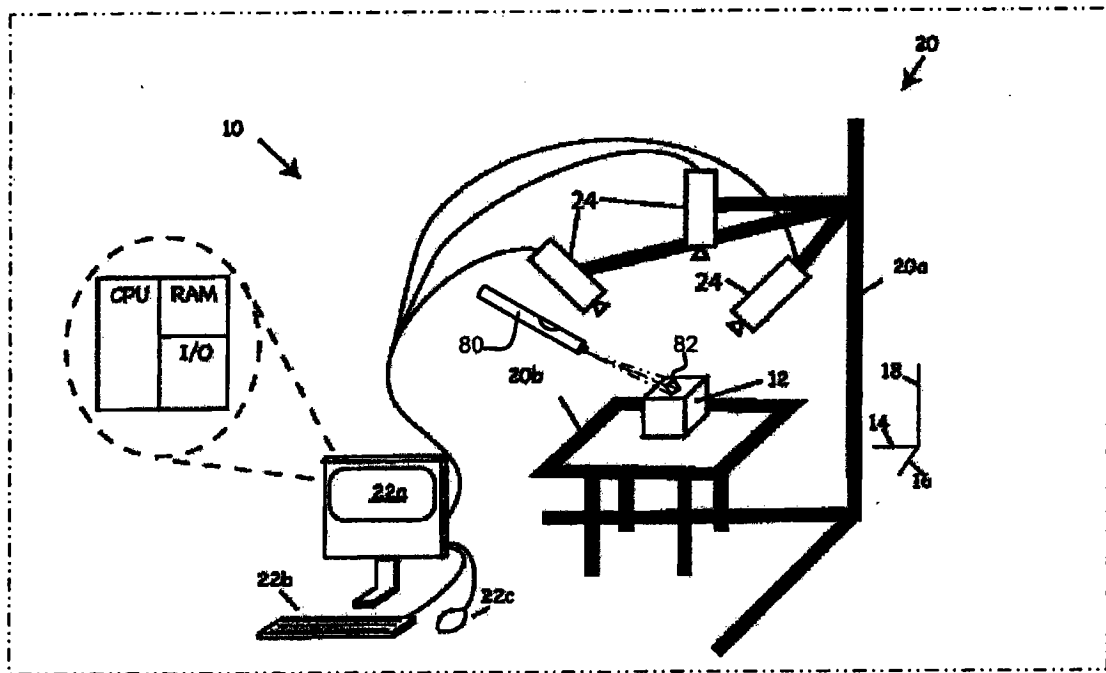


图 1

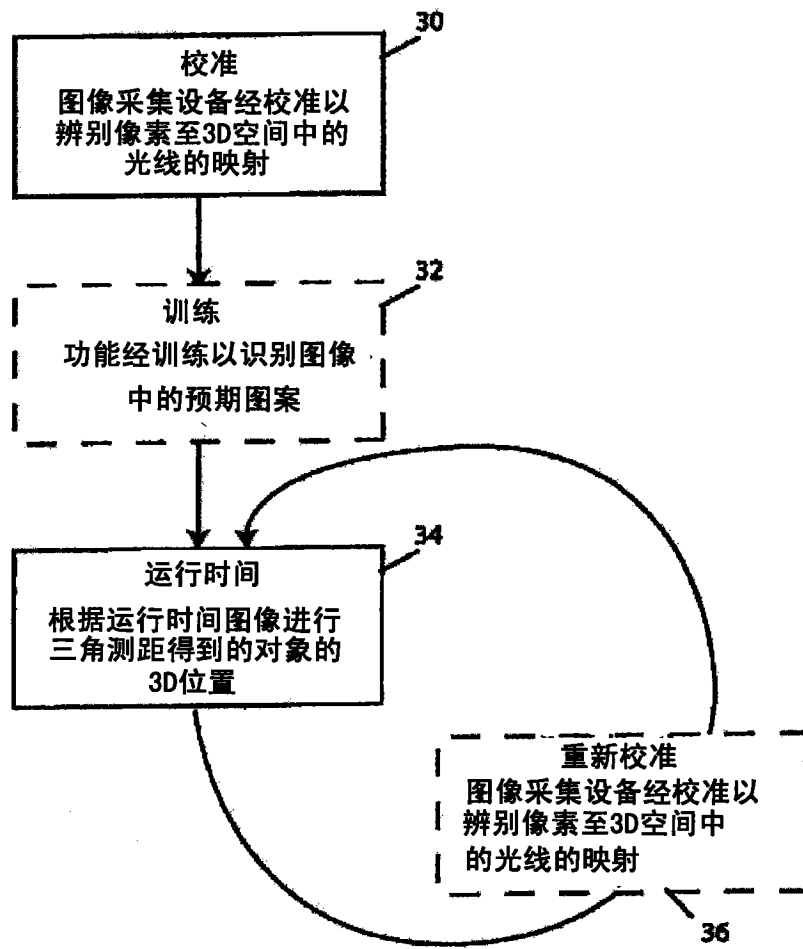


图 2

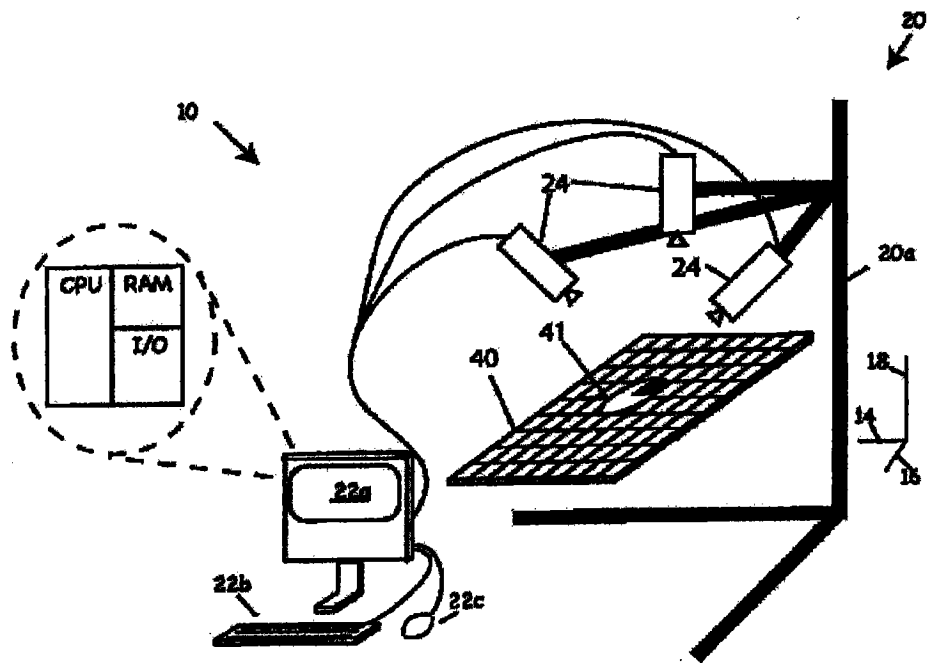


图 3

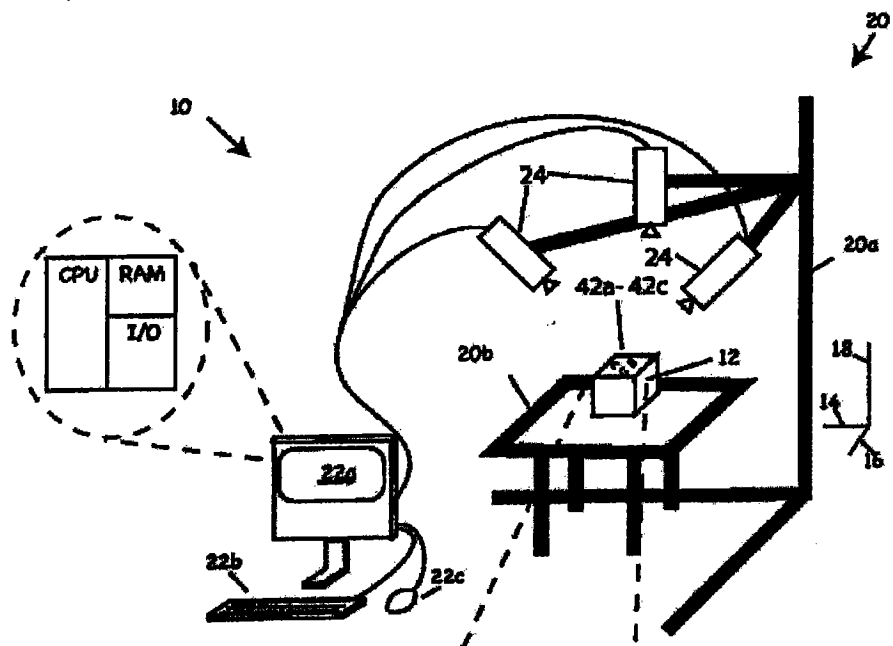


图 4A

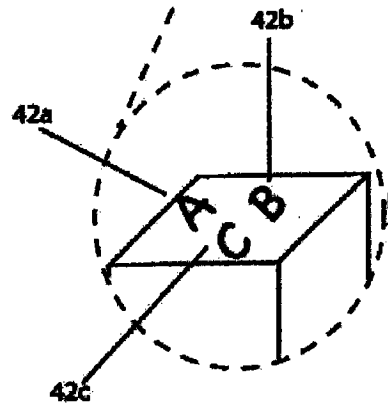


图 4B

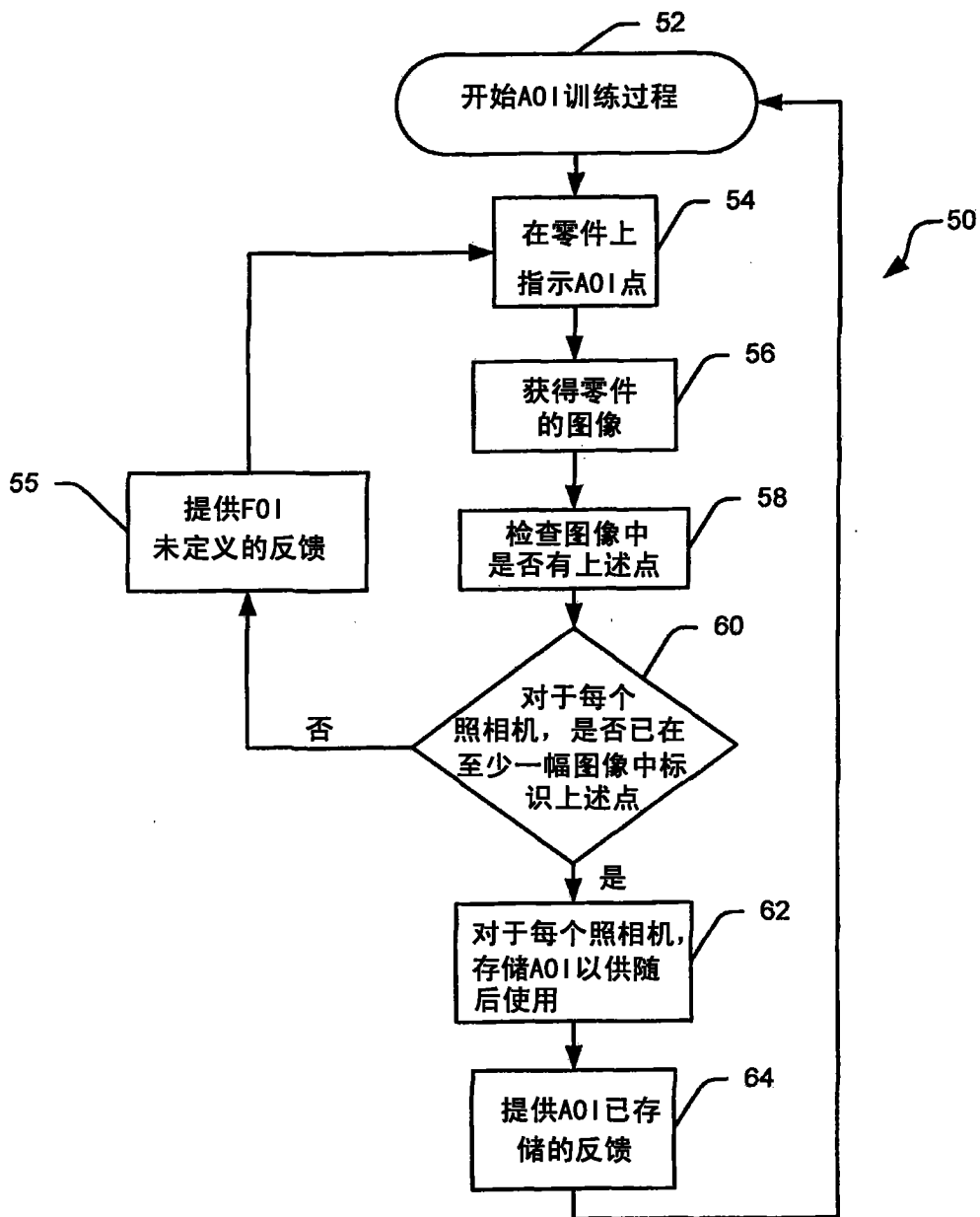


图 5