



(12)发明专利申请

(10)申请公布号 CN 111386550 A

(43)申请公布日 2020.07.07

(21)申请号 201880074141.4

(74)专利代理机构 北京市柳沈律师事务所
11105

(22)申请日 2018.11.15

代理人 金玉洁

(30)优先权数据

62/586,611 2017.11.15 US

(51)Int.Cl.

G06T 7/20(2017.01)

(85)PCT国际申请进入国家阶段日

2020.05.15

G06T 7/579(2017.01)

(86)PCT国际申请的申请数据

PCT/US2018/061300 2018.11.15

(87)PCT国际申请的公布数据

WO2019/099684 EN 2019.05.23

(71)申请人 谷歌有限责任公司

地址 美国加利福尼亚州

(72)发明人 A.安杰洛瓦 M.威克 R.马朱利安

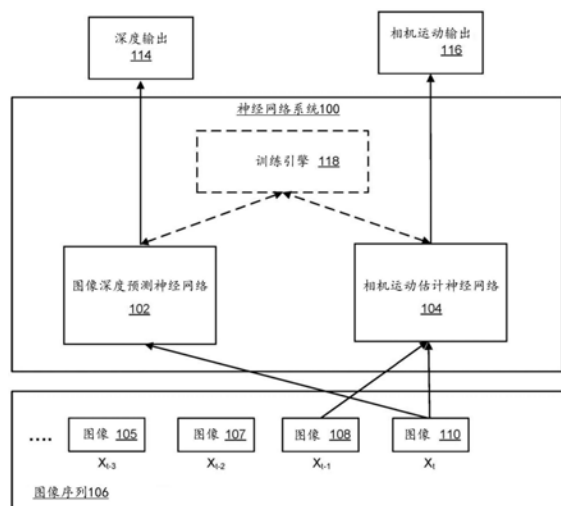
权利要求书3页 说明书12页 附图4页

(54)发明名称

图像深度和自我运动预测神经网络的无监督学习

(57)摘要

一种系统包括由一个或多个计算机实现的神经网络,其中神经网络包括图像深度预测神经网络和相机运动估计神经网络。神经网络被配置为接收图像序列。神经网络被配置为使用图像深度预测神经网络来处理图像序列中的每个图像,以为每个图像生成表征图像的深度的相应深度输出,并且使用相机运动估计神经网络来处理图像序列中的图像子集,以生成表征子集中的图像之间的相机的运动的相机的运动输出。已经使用无监督学习技术联合训练了图像深度预测神经网络和相机运动估计神经网络。



1. 一种系统,包括:

由一个或多个计算机实现的神经网络,其中所述神经网络包括图像深度预测神经网络和相机运动估计神经网络,其中所述神经网络被配置为接收图像序列,以及

使用所述图像深度预测神经网络处理所述图像序列中的每个图像,以为每个图像生成表征所述图像的深度的相应深度输出,以及

使用所述相机运动估计神经网络处理所述图像序列中的图像子集,以生成表征所述子集中的图像之间的相机的运动的相机运动输出;以及

其中已经使用无监督学习技术联合训练了所述图像深度预测神经网络和所述相机运动估计神经网络。

2. 根据权利要求1所述的系统,其中所述图像序列是由相机捕获的视频帧。

3. 根据权利要求1或2中任一项所述的系统,其中,所述深度输出包括所述图像中多个像素的每个像素的估计深度值,所述估计深度值表示在所述像素处描绘的场景距所述图像的焦平面的相应距离。

4. 根据权利要求1-3中任一项所述的系统,其中,所述相机运动输出是变换矩阵,所述变换矩阵将所述相机的位置和朝向从拍摄所述子集中的第一图像时的视点变换到拍摄所述子集中的第二图像时的视点。

5. 根据权利要求1-4中任一项所述的系统,其中所述图像深度预测神经网络包括卷积神经网络层。

6. 根据权利要求1-5中任一项所述的系统,其中所述相机运动估计神经网络包括卷积神经网络层。

7. 根据权利要求1-6中任一项所述的系统,其中所述图像子集包括所述图像序列中的两个图像。

8. 根据权利要求1-6中任一项所述的系统,其中所述图像子集包括所述图像序列中的三个或更多个图像。

9. 用指令编码的一个或多个计算机存储介质,当所述指令由一个或多个计算机执行时,所述指令使所述一个或多个计算机实现权利要求1-8中任一项所述的系统。

10. 一种用于训练权利要求1-8中任一项所述的神经网络的方法,所述方法包括:

获得包括图像序列的训练数据;并且

对于所述图像序列中的每个特定图像:

使用所述图像深度预测神经网络处理所述特定图像,以生成表征所述特定图像的第一深度的第一深度估计;

使用深度预测神经网络处理序列中所述特定图像之后的第二图像,以生成表征所述第二图像的第二深度的第二深度估计;

使用所述相机运动估计神经网络处理所述特定图像和所述第二图像,以生成第一变换矩阵,所述第一变换矩阵将相机的位置和朝向从拍摄所述特定图像时的视点变换到拍摄所述第二图像时的视点;以及

基于所述第一深度估计、所述第二深度估计和所述第一变换矩阵,反向传播损失函数的梯度估计,以联合调整所述深度预测神经网络和所述相机运动估计神经网络的参数的当前值。

11. 根据权利要求10所述的方法,其中所述损失函数包括基于3D的点云对准损失分量,所述基于3D的点云对准损失分量最小化从所述特定图像和所述第二图像生成的两个点云之间的点到点距离。

12. 根据权利要求11所述的方法,其中反向传播所述损失函数的梯度估计包括:

通过重复地估计最佳拟合变换来计算所述基于3D的点云对准损失分量,所述最佳拟合变换最小化第一点云中的点和它们在第二点云中的相应点之间的点到点距离。

13. 根据权利要求10至12中任一项所述的方法,其中,所述损失函数包括图像重构损失分量,所述图像重构损失分量保持 (i) 所述特定图像和从所述第二图像生成的第一重构图像,以及 (ii) 所述第二图像和从所述特定图像生成的第二重构图像的光度一致性。

14. 根据权利要求13所述的方法,其中反向传播所述损失函数的梯度估计包括:

通过 (i) 基于所述第一深度估计和所述第一变换矩阵,解析地计算指示所述第一重构图像中的有效像素坐标的有效性掩码,以及 (ii) 基于所述第二深度估计和所述第一变换矩阵的逆,解析地计算指示所述第二重构图像中的有效像素坐标的第二有效性掩码,来计算所述重构损失分量。

15. 根据权利要求13或14所述的方法,其中,所述损失函数包括结构化相似性损失分量,所述结构化相似性损失分量保持 (i) 所述特定图像和所述第一重构图像中的块的相似性,以及 (ii) 所述第二图像和所述第二重构图像中的块的相似性。

16. 根据权利要求10至15中任一项所述的方法,其中,所述损失函数还包括深度平滑性损失分量,所述深度平滑性损失分量允许 (i) 在所述特定图像中存在急剧变化的像素坐标处的所述第一深度估计中的急剧变化,以及 (ii) 在所述第二图像中存在急剧变化的像素坐标处的所述第二深度估计中的急剧变化。

17. 根据权利要求10至16中任一项所述的方法,其中,所述损失函数是所述损失函数的各个分量的加权组合。

18. 根据权利要求10-17中任一项所述的方法,还包括使用小批量随机优化来调整所述图像深度预测神经网络和所述相机运动估计神经网络的参数的当前值。

19. 根据权利要求10-17中任一项所述的方法,还包括使用随机梯度优化来调整所述图像深度预测神经网络和所述相机运动估计神经网络的参数的当前值。

20. 根据权利要求10-19中任一项所述的方法,其中,所述图像序列是由相机捕获的视频帧。

21. 根据权利要求10-20中任一项所述的方法,其中所述第二图像紧接着所述图像序列中的所述特定图像。

22. 根据权利要求10-21中任一项所述的方法,其中,所述第一深度估计包括所述特定图像中的多个像素的每个像素的估计深度值,所述估计深度值表示在所述像素处描绘的场景距所述特定图像的焦平面的相应距离。

23. 根据权利要求10-22中任一项所述的方法,其中,所述第二深度估计包括所述第二图像中的多个像素的每个像素的估计深度值,所述估计深度值表示在所述像素处描绘的场景距所述第二图像的焦平面的相应距离。

24. 存储指令的一个或多个计算机存储介质,当所述指令被一个或多个计算机执行时,使得所述一个或多个计算机执行权利要求10-23中任一项所述的相应方法的操作。

25. 一个或多个计算机以及一个或多个存储指令的存储设备,当所述指令被所述一个或多个计算机执行时,使得所述一个或多个计算机执行权利要求10-23中任一项所述的相应方法的操作。

图像深度和自我运动预测神经网络的无监督学习

[0001] 相关申请的交叉引用

[0002] 本申请要求2017年11月15日提交的序列号62/586,611的美国临时申请的优先权。在先申请的公开被视为本申请的公开的部分,并通过引用结合于本申请的公开中。

背景技术

[0003] 本说明书涉及使用神经网络处理图像。

[0004] 机器学习模型接收输入并基于接收到的输入生成输出,例如预测输出。一些机器学习模型是参数模型,并且基于接收到的输入和模型的参数值生成输出。

[0005] 一些机器学习模型是深度模型,深度模型采用多层模型来为接收到的输入生成输出。例如,深度神经网络是深度机器学习模型,其包括输出层和一个或多个隐藏层,每个隐藏层对接收到的输入应用非线性变换以生成输出。

发明内容

[0006] 一般,本说明书中描述的主题的一个创新方面可以体现在包括由一个或多个计算机实现的神经网络的系统中。神经网络包括图像深度预测神经网络和相机运动估计神经网络。神经网络被配置为接收图像序列。神经网络被配置为使用图像深度预测神经网络来处理图像序列中的每个图像,以为每个图像生成(i)表征图像的深度的相应深度输出。神经网络被配置为使用相机运动估计神经网络来处理图像序列中的图像子集,以生成表征该子集中的图像之间的相机的运动的相机运动输出。已经使用无监督学习技术联合训练了图像深度预测神经网络和相机运动估计神经网络。

[0007] 前述和其他实施例中的每个可选地包括单独的或组合的以下特征中的一个或多个。图像序列可以是由相机捕获的视频帧。深度输出可以包括图像中多个像素的每个像素的估计深度值,该估计深度值表示在该像素处描绘的场景距图像的焦平面的相应距离。相机运动输出可以是变换矩阵,该变换矩阵将相机的位置和朝向从拍摄子集中的第一图像时的视点变换到拍摄子集中的第二图像时的视点。图像深度预测神经网络可以包括卷积神经网络层。相机运动估计神经网络可以包括卷积神经网络层。图像子集可以包括图像序列中的两个图像。图像子集可以包括图像序列中的三个或更多个图像。无监督学习技术可以包括下面描述的关于训练神经网络的方法的操作。

[0008] 本说明书中描述的主题的另一创新方面可以体现在编码有指令的一个或多个计算机存储介质中,当指令由一个或多个计算机执行时,该指令使得一个或多个计算机实现上述系统。

[0009] 本说明书中描述的主题的另一创新方面可以体现在训练上述系统的神经网络的方法中。该方法包括:获得包括图像序列的训练数据,并对图像序列中的每个特定图像执行以下操作。

[0010] 对于序列中的每个特定图像,该方法包括使用图像深度预测神经网络处理该特定图像,以生成表征该特定图像的第一深度的第一深度估计;使用深度预测神经网络处理序

列中特定图像之后的第二图像,以生成表征第二图像的第二深度的第二深度估计;使用相机运动估计神经网络处理特定图像和第二图像,以生成第一变换矩阵,该第一变换矩阵将相机的位置和朝向从拍摄特定图像时的视点变换到拍摄第二图像时的视点;以及基于第一深度估计、第二深度估计和第一变换矩阵,反向传播损失函数的梯度估计,以联合调整深度预测神经网络和相机运动估计神经网络的参数的当前值。

[0011] 前述和其他实施例中的每个可选地包括单独的或组合的以下特征中的一个或多个。

[0012] 损失函数可以包括基于3D的点云对准损失分量,其最小化从特定图像和第二图像生成的两个点云之间的点到点距离。

[0013] 损失函数可以包括图像重构损失分量,其保持 (i) 特定图像和从第二图像生成的相应重构图像,以及 (ii) 第二图像和从第一图像生成的相应第二重构图像的光度 (photometric) 一致性。

[0014] 损失函数可以包括结构化相似性损失分量,其保持 (i) 特定图像和重构图像中的块的相似性,以及 (ii) 第二图像和第二重构图像中的块的相似性。

[0015] 损失函数还可以包括深度平滑性损失分量,其允许 (i) 在特定图像中存在急剧变化的像素坐标处的第一深度估计中的急剧变化,以及 (ii) 在第二图像中存在急剧变化的像素坐标处的第二深度估计中的急剧变化。

[0016] 损失函数可以是损失函数的各个分量的加权组合。损失函数可以是以下分量中的一个或多个的加权组合: (i) 基于3D的点云对准损失分量、(ii) 图像重构损失分量、(iii) 结构化相似性损失分量,以及 (iv) 深度平滑性损失分量。例如,损失函数可以是 (i) 基于3D的点云对准损失分量、(ii) 图像重构损失分量、(iii) 结构化相似性损失分量和 (iv) 深度平滑性损失分量的加权组合。

[0017] 反向传播损失函数的梯度估计可以包括:通过重复地估计最佳拟合 (best-fit) 变换来计算基于3D的点云对准损失分量,该最佳拟合变换最小化第一点云中的点和它们在第二点云中的相应点之间的点到点距离。

[0018] 反向传播损失函数的梯度估计可以包括通过 (i) 基于第一深度估计和第一变换矩阵,解析地计算指示第一重构图像中的有效像素坐标的有效性掩码 (validity mask), 以及 (ii) 基于第二深度估计和第一变换矩阵的逆,解析地计算指示第二重构图像中的有效像素坐标的第二有效性掩码,来计算重构损失分量。

[0019] 在一些情况下,该方法可以包括使用小批量 (mini-batch) 随机优化来调整图像深度预测神经网络和相机运动估计神经网络的参数的当前值。在一些情况下,该方法可以包括使用随机梯度优化来调整图像深度预测神经网络和相机运动估计神经网络的参数的当前值。

[0020] 训练图像序列可以是由相机捕获的视频帧。第二图像可以紧跟在图像序列中的特定图像之后。第一深度估计可以包括特定图像中多个像素的每个像素的估计深度值,该估计深度值表示在该像素处描绘的场景距特定图像的焦平面的相应距离。第二深度估计可以包括第二图像中多个像素的每个像素的估计深度值,该估计深度值表示在该像素处描绘的场景距第二图像的焦平面的相应距离。

[0021] 本说明书中描述的主题可以在特定实施例中实现,以便实现以下优点中的一个或

多个。本说明书中描述的主题一般针对一种训练方案,用于从图像序列中无监督地学习深度和相机运动(或自我运动),例如由机器人代理的相机捕获的视频帧(即单目(monocular)视频)。由于训练方案只需要很少的信息源用于训练(例如,由相机拍摄的单目视频),因此在大的和变化的数据集上训练图像深度和相机运动估计神经网络成为可能。考虑到相机是最普遍的可用传感器,并且高质量相机便宜且易于部署,训练方案在由相机以无监督方式拍摄的任意单目视频上训练神经网络的能力,即,不需要单目视频的任何外部标记,打开了实际上无限制的训练数据,并且因此消除了感测伪影(artifact)(例如,感测标记数据中存在的噪声/误差)或传感器间校准问题。换句话说,训练方法使得更多的数据能够用作训练数据,并且不需要在数量上可能受限的特定标记的深度数据。

[0022] 尽管现有方法使用逐像素(pixel-wise)或基于梯度的损失,其仅考虑小局部邻域中的像素并导致质量恶化的预测,但是本说明书中描述的训练方案通过将基于3D的点云对准损失结合到训练损失函数中来考虑整个场景的3D几何形状。基于3D的损失会增强估计的3D点云和视频连续帧中的自我运动的一致性。

[0023] 此外,该训练方案组合新颖的基于3D的损失和基于使用来自相邻帧的估计深度和自我运动的帧重构的光度质量的结构化相似性损失。

[0024] 该训练方案还在图像重构损失中加入有效性掩码,以避免惩罚不存在有用信息的区域。先前的方法在损失中包括这样的区域,因此降低了神经网络的性能(例如,在图像的深度输出中引起边缘伪影)。通过解析地计算有效性掩码,以及将该掩码结合进图像重构损失中,该训练方案提高了深度和相机运动神经网络的性能。

[0025] 此外,训练方案采用深度平滑性损失,通过考虑输入图像的梯度来规范输入图像的深度估计。通过考虑输入图像的梯度,深度平滑性损失允许在输入图像中存在急剧变化的像素坐标处的深度的急剧变化,从而使得训练的深度预测神经网络能够生成更真实和准确的深度估计。

[0026] 使用上述训练方案训练的深度预测神经网络和相机运动估计神经网络,与使用传统方法训练的其他神经网络相比,能够以更高的质量(例如,更高的准确度、更高的保真度以及更真实)产生深度估计和相机自我运动估计。这样的估计可以用于机器人代理或自动控制,例如在导航中辅助。由训练的神经网络产生的估计不需要昂贵且难以校准的深度传感器以提供输入数据,并且可以使用便宜且易于部署的单目相机来获得。

[0027] 本说明书主题的一个或多个实施例的细节在附图和下面的描述中阐述。本主题的其他特征、方面和优点将从说明书、附图和权利要求中变得显而易见。

附图说明

[0028] 图1示出了示例神经网络系统的架构。

[0029] 图2是用于联合训练图像深度预测神经网络和相机运动估计神经网络的示例过程的流程图。

[0030] 图3是用于生成基于3D的点云对准损失的流程图。

[0031] 图4示出了如何使用ICP过程来惩罚变换矩阵和深度估计中的误差。

[0032] 不同附图中相同的附图标记和名称表示相同的元件。

具体实施方式

[0033] 本说明书描述了一种在一个或多个位置的一个或多个计算机上实现为计算机程序的神经网络系统,该神经网络系统被配置为使用图像深度预测神经网络和相机运动估计神经网络从输入图像序列中生成图像深度和自我运动估计。

[0034] 图1示出了神经网络系统100的示例架构。神经网络系统100是在一个或多个位置的一个或多个计算机上实现为计算机程序的系统的示例,其中可以实现下面描述的系统、组件和技术。

[0035] 通常,神经网络系统100被配置为接收图像序列106,并处理该序列中的每个图像,以生成表征图像的深度的深度输出,并处理该序列中的图像子集,以生成表征该子集中的图像之间的相机的运动的相机运动输出。例如,图像序列106可以包括由机器人代理的相机捕获的单目视频帧。作为另一示例,图像序列106可以包括使用移动设备的相机(例如,当电话用户在骑自行车时,由手持商业电话的相机)捕获的单目视频帧。序列106不一定包括来自视频的所有帧。例如,序列106可以包括视频的每隔一帧。

[0036] 为了生成深度输出和相机运动输出,神经网络系统100包括图像深度预测神经网络102(为简单起见,下文称为“深度网络102”)和相机运动估计神经网络104(下文称为“相机运动网络104”)。深度网络102和相机运动网络104中的每一个都包括一个或多个卷积神经网络层。在一些实现方式中,深度网络102和相机运动网络104是完全卷积神经网络。

[0037] 深度网络102被配置为根据深度网络102的深度参数的当前值,处理图像序列106中的每个图像,以为每个图像生成表征图像的深度的相应深度输出。为给定图像生成的深度输出包括给定图像中多个像素的每个像素的预测深度值。像素的深度值是(i)其中记录给定图像的平面和(i i)在该像素处描绘的场景(例如,该像素描绘的对象或对象的部分的实际或虚拟位置)之间的垂直距离的表示。其中记录图像的平面可以是焦平面,例如相机或虚拟相机的焦平面。

[0038] 在图1的示例中,深度网络102处理图像序列106中的图像110,以生成图像110的深度输出114。图像110可以表示为 X_t ,其中 t 是当由相机捕获到图像的时间。独立于序列106中的其他图像从当前图像110生成深度输出114。

[0039] 相机运动网络104被配置为处理图像序列106中的图像子集,以生成相机运动输出,该相机运动输出根据相机运动网络104的运动参数的当前值来表征该子集中的图像之间的相机的运动。

[0040] 在一些实现方式中,图像子集包括来自图像序列106的两个图像。例如,如图1所示,图像子集包括两个连续的图像 X_{t-1} (108)和 X_t (110),它们分别在时间 $t-1$ 和 t 由相机拍摄。相机运动网络104处理图像 X_{t-1} 和 X_t 以生成相机运动输出116,该输出是变换矩阵,该变换矩阵将相机的位置和朝向从拍摄图像 X_{t-1} 时的视点变换到拍摄图像 X_t 时的视点。换句话说,相机运动输出116表示从时间 $t-1$ 到时间 t 的相机的移动(位置和朝向)

[0041] 在一些其他实现方式中,图像子集包括来自图像序列106的三个或更多个图像。例如,图像子集可以包括图像 X_{t-2} 、 X_{t-1} 和 X_t 。在该示例中,给定图像子集,相机运动网络104生成相机运动输出,其表示从时间 $t-2$ 到时间 t 的相机的移动。

[0042] 为了有效地生成图像序列的深度输出和相机运动输出,神经网络系统100包括训练引擎118,训练引擎118是基于软件的系统、子系统或过程,该过程被编程为使用无监督学

习技术在训练数据上联合训练深度网络102和相机运动网络104。通常,训练引擎118将实现为安装在一个或多个位置的一个或多个计算机上的一个或多个软件模块或组件。在一些情况下,一个或多个计算机将专用于一个特定引擎;在其他情况下,可以在同一个或多个计算机上安装和运行多个引擎。

[0043] 特别地,在训练期间,训练引擎118获得包括训练图像序列的训练数据。训练图像可以是由机器人代理的相机捕获的视频帧。

[0044] 对于训练图像序列中的每个特定图像,训练引擎118重复执行以下操作。

[0045] 对于每个特定图像,训练引擎118根据深度网络102的深度参数的当前值,使用深度网络102处理特定图像,以生成表征该特定图像的第一深度的第一深度估计。训练引擎118根据深度网络102的深度参数的当前值,使用深度网络102处理序列中特定图像之后的第二图像,以生成表征该第二图像的第二深度的第二深度估计。训练引擎118使用相机运动网络104来处理特定图像和第二图像,以生成第一变换矩阵,该第一变换矩阵将相机的位置和朝向从拍摄特定图像时的视点变换到拍摄第二图像时的视点。然后,训练引擎118反向传播损失函数的梯度估计,以联合调整深度网络102和相机运动网络104的参数的当前值。基于第一深度估计、第二深度估计和第一变换矩阵来计算损失函数。

[0046] 通常,损失函数包括基于3D的点云对准损失分量,其最小化(i)从特定图像生成的第一点云中的点和(ii)从第二图像生成的第二点云中的它们的相应点之间的点到点距离。下面参考图3详细描述用于生成基于3D的点云对准损失分量的处理。

[0047] 损失函数还可以包括一个或多个以下损失分量:图像重构损失分量、结构化相似性损失分量和深度平滑性损失分量。图像重构损失分量保持(i)特定图像和从第二图像生成的第一重构图像,以及(ii)第二图像和从特定图像生成的第二重构图像的光度一致性。结构化相似性损失分量保持(i)特定图像和第一重构图像中的块的相似性,以及(ii)第二图像和第二重构图像中的块的相似性。深度平滑性损失分量允许(i)在特定图像中存在急剧变化的像素坐标处的第一深度估计中的急剧变化,以及(ii)在第二图像中存在急剧变化的像素坐标处的第二深度估计中的急剧变化。

[0048] 在训练期间,训练引擎118可以重复执行上述操作,以通过使用小批量随机优化或随机梯度优化方法来调整深度网络102和相机运动神经网络104的参数值,以最小化损失函数。

[0049] 下面参考图2更详细地描述用于联合训练深度网络102和相机运动网络104以及用于计算损失函数中每个分量的过程。

[0050] 在训练之后,神经网络系统100可以根据深度网络102的深度参数的训练值,使用深度网络102以为给定的输入图像生成深度输出,并且根据相机运动网络104的运动参数的训练值,使用相机运动网络104以为多个输入图像生成相机运动输出。

[0051] 在一些情况下,神经网络系统100可以向外部系统提供训练的深度网络102和训练的相机运动网络104或者网络参数的训练值。外部系统可以使用训练的深度网络102和训练的相机运动104以上述方式为输入图像序列生成深度输出和相机运动输出。

[0052] 作为特定示例,在训练之后,当机器人与环境交互时,系统100或外部系统可以使用深度输出和相机运动输出来控制机器人。通过使深度输出和相机运动输出可用作机器人的控制策略或规划器的输入,可以更有效地控制机器人来完成环境中的指定任务。

[0053] 图2是用于联合训练图像深度预测神经网络和相机运动估计神经网络(例如,图1的深度网络102和相机运动网络104)的示例过程200的流程图。为方便起见,过程200将被描述为由位于一个或多个位置的一个或多个计算机的系统执行。例如,适当编程的神经网络系统,例如图1的神经网络系统100,可以执行过程200。

[0054] 系统可以对训练数据中包括的不同的训练图像序列重复执行过程200,以便重复更新图像深度预测神经网络和相机运动估计神经网络的参数值。

[0055] 系统获得训练图像序列(步骤201)。例如,训练图像可以是由机器人代理的相机或移动设备的相机捕获的视频帧。训练图像序列不一定包括视频的所有连续帧。例如,训练图像序列可以包括视频的每隔一帧。

[0056] 对于训练图像序列中的每个特定图像 X_{t-1} ,系统重复执行以下操作(步骤202-218)。

[0057] 系统根据深度网络的深度参数的当前值,使用深度网络处理特定图像 X_{t-1} ,以生成表征特定图像 X_{t-1} 在时间 $t-1$ 的第一深度的第一深度估计 D_{t-1} (步骤202)。

[0058] 系统根据深度网络的深度参数的当前值,使用深度网络处理序列中特定图像 X_{t-1} 之后的第二图像 X_t ,以生成表征第二图像 X_t 在时间 t 的第二深度的第二深度估计 D_t (步骤204)。

[0059] 系统使用相机运动网络处理特定图像 X_{t-1} 和第二图像 X_t ,以生成第一变换矩阵 T_t (步骤206)。第一变换矩阵 T_t ,也称为自我运动,表示从时间 $t-1$ 到时间 t 的相机的移动(位置和朝向)。也就是说,第一变换矩阵 T_t 将相机的位置和朝向从拍摄特定图像 X_{t-1} 时的视点变换到拍摄第二图像 X_t 时的视点。

[0060] 为了构建用于联合训练深度和相机运动网络的损失函数,系统首先构建基于3D的点云对准损失分量,该分量最小化从特定图像 X_{t-1} 和第二图像 X_t 生成的两个点云之间的点到点距离(步骤208)。

[0061] 具体地,系统基于第二图像 X_t 的第二深度估计 D_t 为第二图像 X_t 生成3D点云 Q_t 。具有相应估计深度值 D_t^{ij} 的坐标 (i, j) 处的每个图像像素可以被投影到结构化3D点云 Q_t 中,如下所示:

$$[0062] \quad Q_t^{ij} = D_t^{ij} \cdot K^{-1}[i, j, 1]^T, \quad (1)$$

[0063] 其中 K 是相机固有矩阵,并且坐标是齐次的。

[0064] 给定表示从 $t-1$ 到 t 的相机的移动的第一变换矩阵 T_t ,系统变换3D点云 Q_t 以获得先前的图像(特定图像 X_{t-1})的点云 \hat{Q}_{t-1} 的估计,如下:

$$[0065] \quad \hat{Q}_{t-1} = T_t^{-1} \cdot Q_t, \quad (2)$$

[0066] 其中 T_t^{-1} 表示从 t 到 $t-1$ 的相机的移动的逆。

[0067] 类似地,系统基于特定图像 X_{t-1} 的第一深度估计 D_{t-1} 为特定图像 X_{t-1} 生成3D点云 Q_{t-1} 。具有相应估计深度值 D_{t-1}^{ij} 的坐标 (i, j) 处的每个图像像素可以被投影到结构化3D点云 Q_{t-1} 中,如下:

$$[0068] \quad Q_{t-1}^{ij} = D_{t-1}^{ij} \cdot K^{-1}[i, j, 1]^T, \quad (3)$$

[0069] 其中K是相机固有矩阵,并且坐标是齐次的。

[0070] 然后,系统转换3D点云 Q_{t-1} 以获得第二图像 X_t 的点云 \hat{Q}_t 的估计,如下:

$$[0071] \quad \hat{Q}_t = T_t \cdot Q_{t-1}, \quad (4)$$

[0072] 其中 T_t 表示从 $t-1$ 到 t 的相机的移动。

[0073] 基于3D的点云对准损失分量直接比较两个3D点云: \hat{Q}_{t-1} 与 Q_{t-1} ,或者 \hat{Q}_t 与 Q_t 。也就是说,在一些情况下,基于3D的点云对准损失分量可以比较(i)从第二图像的点云 Q_t 生成的特定图像的点云 \hat{Q}_{t-1} 的估计值与(ii)特定图像的点云 Q_{t-1} 。在一些情况下,基于3D的点云对准损失分量可以将(i)从特定图像的点云 Q_{t-1} 生成的第二图像的点云 \hat{Q}_t 的估计与(ii)第二图像的点云 Q_t 进行比较。

[0074] 为了构造基于3D的点云对准损失分量,系统使用刚性配准(rigid registration)方法,例如迭代最近点(Iterative Closet Point, ICP)过程,来计算(i)最佳拟合变换 T' ,其最小化两个点云中相应点之间的点到点距离,以及(ii)残余(residual)配准误差 r ,其表示在应用了用于最小化点到点距离的ICP的最佳拟合变换 T' 之后相应点之间的残余距离。下面参考图3和图4详细描述ICP过程。

[0075] 系统如下构建基于3D的点云对准损失分量 L_{3D} :

$$[0076] \quad L_{3D} = \|T' - I\|_1 + \|r\|_1, \quad (5)$$

[0077] 其中 $\|\cdot\|_1$ 表示L1-norm(L1范数), I 是单位矩阵,并且 r 是残余配准误差。

[0078] 接下来,系统构建图像重构损失分量,其保持(i)特定图像 X_{t-1} 和从第二图像 X_t 生成的第一重构图像 \hat{X}_{t-1} ,以及(ii)第二图像 X_t 和从特定图像 X_{t-1} 生成的第二重构图像 \hat{X}_t 的光度一致性(步骤210)。

[0079] 特别地,系统将估计的点云 \hat{Q}_{t-1} 投影到图像 $t-1$ 处的相机上,作为 $K\hat{Q}_{t-1}$,其中K是相机固有矩阵。然后,系统将这种变换和投影与等式(1)相结合,以建立从时间 t 的图像坐标到时间 $t-1$ 的图像坐标的映射。基于该映射,系统基于深度输出 D_t 和第一变换矩阵 T_t ,通过扭曲 X_{t-1} 来重构图像 \hat{X}_t :

$$[0080] \quad \hat{X}_t^{ij} = X_{t-1}^{\hat{i}\hat{j}}, [\hat{i}, \hat{j}, 1]^T = KT_t(D_t^{ij} \cdot K^{-1}[i, j, 1]^T) \quad (6)$$

[0081] 类似地,系统基于深度输出 D_{t-1} 和 T_t^{-1} ,通过扭曲 X_t 重构图像 \hat{X}_{t-1} , T_t^{-1} 是第一变换矩阵 T_t 的逆。

[0082] 计算 \hat{X}_t 涉及创建从 X_t 到 X_{t-1} 中的图像坐标的映射。然而,由于相机的运动, X_t 中的一些像素坐标可能被映射到 X_{t-1} 中的图像边界之外的坐标。对于前向自我运动,这个问题在从 X_t 计算 \hat{X}_{t-1} 时通常很明显。在损失函数中包括这样的像素会降低深度和相机运动网络的性能。先前的方法要么忽略了这个问题,要么试图通过给网络添加通用掩码来解决这个问题,这样预计会排除由于任何原因而无法解释的区域。然而,这种方法并不有效,并且经常

导致深度图像中的边缘伪影。

[0083] 为了解决上述问题,这里系统从深度输出和第一变换矩阵(自我运动)解析地计算有效性掩码。对于训练图像序列中的每对图像 X_{t-1} 和 X_t ,系统可以创建一对有效性掩码 M_{t-1} 和 M_t 。 M_{t-1} 指示当从 X_t 重构 \hat{X}_{t-1} 时哪些像素坐标有效。 M_t 指示从 X_{t-1} 重构 \hat{X}_t 时哪些像素坐标有效。

[0084] 更具体地,系统基于第一深度估计 D_{t-1} 和第一变换矩阵 T_t ,解析地计算指示第一重构图像 \hat{X}_{t-1} 中的有效像素坐标的第一有效性掩码 M_{t-1} 。系统还基于第二深度估计 D_t 和第一变换矩阵的逆 T_t^{-1} ,解析地计算指示第二重构图像 \hat{X}_t 中的有效像素坐标的第二有效性掩码 M_t 。

[0085] 在计算有效性掩码之后,然后,系统将重构图像 \hat{X}_t 和 \hat{X}_{t-1} 分别与特定图像 X_t 和第二图像 X_{t-1} 进行比较,以产生基于光度一致性的图像重构损失分量,如下:

$$[0086] \quad L_{rec} = \sum_{ij} \|(X_t^{ij} - \hat{X}_t^{ij})M_t^{ij}\|, \quad (7)$$

[0087] 其中 X_t^{ij} 表示在第二图像 X_t 的坐标 (i, j) 处的图像像素, \hat{X}_t^{ij} 表示在重构图像 \hat{X}_t 的坐标 (i, j) 处的图像像素,而 M_t^{ij} 表示在有效性掩码 M_t 的坐标 (i, j) 处的图像像素。通过使用有效性掩码,系统可以避免在图像的看不见的区域中的虚假损失,从而避免惩罚不存在有用信息的区域。作为结果,系统可以提高深度和相机运动网络的预测/估计质量。

[0088] 随后,系统构建结构化相似性损失分量(步骤212)。一般,结构化相似性损失分量保持(i)特定图像和第一重构图像中的块的相似性,以及(ii)第二图像和第二重构图像中的块的相似性。

[0089] 特别地,系统旨在最小化如下定义的结构化相似性损失分量 L_{SSIM} :

$$[0090] \quad L_{SSIM} = \sum_{ij} [1 - SSIM(\hat{X}_t^{ij}, X_t^{ij})]M_t^{ij}, \quad (8)$$

[0091] 其中 $SSIM(x, y)$ 测量两个图像块 x 和 y 之间的相似性,其定义如下:

$$[0092] \quad SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x + \sigma_y + c_2)}, \quad (9)$$

[0093] 其中 μ_x, σ_x 是局部均值和方差。

[0094] 然后,系统构建深度平滑性损失分量来规范深度估计(步骤214)。深度平滑性损失分量允许(i)在特定图像 X_{t-1} 中存在急剧变化的像素坐标处第一深度估计 D_{t-1} 中的急剧变化,以及(ii)在第二图像 X_t 中存在急剧变化的像素坐标处第二深度估计 D_t 中的急剧变化。深度平滑性损失考虑了给定图像 X 的梯度,并且可以表示如下:

$$[0095] \quad L_{sm} = \sum_{ij} \|\partial_x D^{ij}\| e^{-\|\partial_x X^{ij}\|} + \|\partial_y D^{ij}\| e^{-\|\partial_y X^{ij}\|}, \quad (10)$$

[0096] 其中 X^{ij} 表示给定图像 X 的坐标 (i, j) 处的图像像素, D^{ij} 是对应于 X^{ij} 的估计深度,并且 ∂_x, ∂_y 是梯度。通过考虑图像 X 的梯度,深度平滑性损失允许在图像中存在急剧变化的像素坐标处的深度的急剧变化。

[0097] 系统构建损失函数(步骤216)。在一些实现方式中,损失函数仅包括基于3D的点云对准损失分量。在一些实现方式中,损失函数是基于3D的点云对准损失分量和(i)图像重构损失分量、(ii)结构化相似性损失分量或(iii)深度平滑性损失分量中的一个或多个的加权组合。例如,损失函数可以具有以下形式:

$$[0098] \quad L = \sum_s \alpha L_{rec}^s + \beta L_{3D}^s + \gamma L_{sm}^s + \omega L_{SSIM}^s, \quad (11)$$

[0099] 其中所有损失分量在四个不同的尺度s应用,尺度s范围从获得的训练数据中的输入图像的输入分辨率到具有较小分辨率的图像(例如,输入图像的宽度和高度为 $\frac{1}{4}$ 、 $\frac{1}{8}$ 或 $\frac{1}{10}$ 的图像)。 $\alpha, \beta, \gamma, \omega$ 是预定的超参数。例如, $\alpha=0.85, \beta=0.1, \gamma=0.05$,并且 $\omega=0.15$ 。又如, $\alpha=0.5, \beta=0.3, \gamma=0.2$,并且 $\omega=0$ 。再如, $\alpha=0.1, \beta=0.2, \gamma=0.3$,并且 $\omega=0$ 。

[0100] 然后,系统生成损失函数的梯度估计,并反向传播该梯度估计,以联合调整深度网络的深度参数的当前值和相机运动网络的运动参数的当前值(步骤218)。系统可以通过使用小批量随机优化或随机梯度优化方法来联合调整深度和运动参数的当前值以最小化损失函数。

[0101] 系统对训练图像序列中的其他图像重复上述步骤202-218,以确定深度网络的深度参数的训练值和相机运动网络的运动参数的训练值。

[0102] 图3是用于生成基于3D的点云对准损失分量的示例过程300的流程图。为方便起见,过程300将被描述为由位于一个或多个位置的一个或多个计算机的系统执行。例如,适当编程的神经网络系统或神经网络系统的组件,例如图1的神经网络系统100的训练引擎118,可以执行过程300。

[0103] 如上参考图2所述,为了生成基于3D的点云对准损失分量(此后也称为基于3D的损失),系统首先基于第二图像 X_t 的第二深度估计 D_t 为第二图像 X_t 生成3D点云 Q_t (步骤304)。具有相应估计深度值 D_t^{ij} 的坐标(i, j)处的每个图像像素可以使用等式(1)被投影到结构化3D点云 Q_t 中。

[0104] 给定第一变换矩阵的逆 T_t^{-1} ,系统变换3D点云 Q_t ,以使用等式(2)估计先前的图像(特定图像 X_{t-1})的点云 \hat{Q}_{t-1} (步骤306)。

[0105] 类似地,系统基于特定图像 X_{t-1} 的第一深度估计 D_{t-1} 为特定图像 X_{t-1} 生成3D点云 Q_{t-1} (步骤308)。具有相应估计深度值 D_{t-1}^{ij} 的坐标(i, j)处的每个图像像素可以被投影到基于等式(3)的结构化3D点云 Q_{t-1} 中。

[0106] 然后,系统使用第一变换矩阵 T_t 变换3D点云 Q_{t-1} ,以获得基于等式(4)的第二图像 X_t 的点云 \hat{Q}_t 的估计(步骤310)。

[0107] 为了构造基于3D的点云对准损失分量,系统使用迭代最近点(ICP)过程302来获得最佳拟合变换 T' ,该变换最小化两个点云之间的点到点距离。

[0108] 具体地,ICP过程302将两个点云A和B(例如,点云 \hat{Q}_{t-1} 和 Q_{t-1} 或者点云 \hat{Q}_t 和 Q_t)作为输入。为方便起见,以下描述描述了当采用两个点云 \hat{Q}_{t-1} 和 Q_{t-1} 作为输入时的ICP过程302,但是应当理解,ICP过程302对于两对点云是相同的。

[0109] 给定两个点云 \hat{Q}_{t-1} 和 Q_{t-1} , ICP过程302输出最佳拟合变换 T' (步骤312), 其最小化A中的变换点和它们在B中的相应点之间的距离:

$$[0110] \quad \underset{T'}{\operatorname{argmin}} \frac{1}{2} \sum_{i,j} \|T'.A^{ij} - B^{c(ij)}\|^2 \quad (12)$$

[0111] 其中 $c(\cdot)$ 表示通过ICP方法(例如, 通过使用简单的最近点试探)找到的点对点对应关系。系统还使用ICP过程302来为每个像素坐标 (i, j) 输出残余 $r^{ij} = A^{ij} \cdot T'^{-1} \cdot B^{c(ij)}$ (步骤314)。残余反映了在应用了用于最小化点到点距离的ICP的最佳拟合变换 T' 之后相应点之间的残余距离。

[0112] 系统如下构建基于3D的点云对准损失分量 L_{3D} :

$$[0113] \quad L_{3D} = \|T'_t - I\|_1 + \|r_t\|_1$$

[0114] 其中 $\|\cdot\|_1$ 表示L1-norm (L1范数), I 是单位矩阵, 并且 r 是残余配准误差(步骤316)。

[0115] 图4示出了系统如何使用ICP过程来惩罚第一变换矩阵(自我运动) T_t 和第二深度估计 D_t 中的误差。该图示出了带有侧视镜的汽车前部的俯视图。给定时间 t 的深度估计 D_t , 系统创建点云 Q_t , 然后通过第一变换矩阵 T_t 将其变换为先前的图像的的点云 \hat{Q}_{t-1} 的预测。如果来自深度和相机运动网络的第一变换矩阵 T_t 和 D_t 是完美的, \hat{Q}_{t-1} 将与 Q_{t-1} 完美地对准。当不是这种情况时, 系统用ICP将 \hat{Q}_{t-1} 对准 Q_{t-1} , 以生成最佳拟合变换矩阵 T'_t 和残余 r_t 。然后, 系统使用 T'_t 和 r_t 向着更好的初始对准来调整 T_t 和 D_t 。更具体地, 系统使用 T'_t 作为关于自我运动 T_t 的基于3D的损失的正梯度的近似值, 并且使用 r_t 作为关于深度估计 D_t 的基于3D的损失的正梯度的近似值。

[0116] 本说明书结合系统和计算机程序组件使用术语“配置”。一个或多个计算机组成的系统被配置为执行特定操作或动作, 这意味着该系统已经在其上安装了软件、固件、硬件或它们的组合, 这些软件、固件、硬件或它们的组合在操作中使得该系统执行这些操作或动作。对于被配置为执行特定操作或动作的一个或多个计算机程序, 这意味着该一个或多个程序包括当由数据处理装置执行时, 使该装置执行操作或动作的指令。

[0117] 本说明书中描述的主题和功能操作的实施例, 包括在本说明书中公开的结构及其结构等同物, 或者在它们中的一个或多个的组合, 可以在数字电子电路中、在有形体现的计算机软件或固件中、在计算机硬件中实现。本说明书中描述的主题的实施例可以被实现为一个或多个计算机程序(即编码在有形非暂时性存储介质上的计算机程序指令的一个或多个模块), 用于由数据处理装置执行或控制数据处理装置的操作。计算机存储介质可以是机器可读存储设备、机器可读存储基底、随机或串行存取存储器设备或者它们中的一个或多个的组合。可替代地或附加地, 程序指令可以被编码在人工生成的传播信号上, 例如, 机器生成的电、光或电磁信号, 其被生成以对信息进行编码, 用于到合适的接收器装置的传输, 以由数据处理装置执行。

[0118] 术语“数据处理装置”指的是数据处理硬件, 并且包含用于处理数据的所有种类的装置、设备和机器, 例如包括可编程处理器、计算机或多个处理器或计算机。该装置还可以是或进一步包括专用逻辑电路, 例如FPGA(field programmable gate array, 现场可编程门阵列)或ASIC(application-specific integrated circuit, 专用集成电路)。除了硬件

之外,该装置可以可选地包括为计算机程序创建执行环境的代码,例如,构成处理器固件、协议栈、数据库管理系统、操作系统或它们中的一个或多个的组的代码。

[0119] 计算机程序,也可以被称为或描述为程序、软件、软件应用、应用、模块、软件模块、脚本或代码,可以用任何形式的编程语言编写,包括编译或解释语言,或者声明性或程序性语言;并且其可以以任何形式部署,包括作为独立程序或作为模块、组件、子例程或其他适合在计算环境中使用的单元。程序可以,但不一定,对应于文件系统中的文件。程序可以存储在保存其他程序或数据的文件的部分中,例如存储在标记语言文档中的一个或多个脚本,存储在专用于所讨论的程序的单个文件中,或者存储在多个协调文件中,例如存储一个或多个模块、子程序或部分代码的文件。计算机程序可以部署为在一个计算机或者位于一个站点或分布在多个站点并通过数据通信网络互连的多个计算机上执行。

[0120] 本说明书中描述的过程和逻辑流程可以由一个或多个可编程计算机执行,该计算机执行一个或多个计算机程序以通过对输入数据进行操作并生成输出来执行功能。该过程和逻辑流程也可以由专用逻辑电路,例如,FPGA或ASIC,或者由专用逻辑电路和一个或多个编程计算机的组合来执行。

[0121] 适合计算机程序的执行的计算机可以基于通用或专用微处理器或两者,或者任何其他类型的中央处理单元。一般,中央处理单元将从只读存储器或随机存取存储器或者两者接收指令和数据。计算机的基本元件是用于执行(performing)或执行(executing)指令的中央处理单元和用于存储指令和数据的一个或多个存储器设备。中央处理单元和存储器可以由专用逻辑电路补充或并入专用逻辑电路。一般,计算机还将包括一个或多个用于存储数据的大容量存储设备,例如磁盘、磁光盘或光盘,或者可操作地耦合到一个或多个大容量存储设备,以从一个或多个大容量存储设备接收数据或向一个或多个大容量存储设备传送数据,或者两者都包括。然而,计算机不需要这样的设备。此外,计算机可以嵌入到另一设备中,例如移动电话、个人数字助理(personal digital assistant,PDA)、移动音频或视频播放器、游戏控制台、全球定位系统(Global Positioning System,GPS)接收器或便携式存储设备,例如通用串行总线(universal serial bus,USB)闪存驱动器,仅举几个示例。

[0122] 适用于存储计算机程序指令和数据的计算机可读介质包括所有形式的非易失性存储器、介质和存储器设备,例如包括半导体存储设备,例如,EPROM、EEPROM和闪存设备;磁盘,例如内部硬盘或可移动磁盘;磁光盘;和CD-ROM和DVD-ROM。

[0123] 为了提供与用户的交互,本说明书中描述的主题的实施例可以在计算机上实现,该计算机具有用于向用户显示信息的显示器设备,例如CRT(cathode ray tube,阴极射线管)或LCD(liquid crystal display,液晶显示器),以及用户可以向计算机提供输入的键盘和定点设备,例如鼠标或轨迹球。也可以使用其他类型的设备来提供与用户的交互;例如,提供给用户的反馈可以是任何形式的感觉反馈,例如视觉反馈、听觉反馈或触觉反馈;并且可以以任何形式接收来自用户的输入,包括声音、语音或触觉输入。此外,计算机可以通过向用户使用的设备发送文档和从用户使用的设备接收文档来与用户交互;例如通过响应于从网络浏览器接收的请求,将网页发送到用户的设备上的网络浏览器。此外,计算机可以通过向个人设备(例如,运行消息传递应用的智能手机)发送文本消息或其他形式的消息,并接收来自用户的响应消息来与用户交互。

[0124] 用于实现机器学习模型的数据处理装置还可以包括,例如,用于处理机器学习训

练或产生的普通和计算密集型部分(即推理、工作负载)的专用硬件加速器单元。

[0125] 机器学习模型可以使用机器学习框架来实现和部署,例如TensorFlow框架、微软认知工具包(Microsoft Cognitive Toolkit)框架、Apache Singa框架或Apache MXNet框架。

[0126] 本说明书中描述的主题的实施例可以在计算系统中实现,该计算系统包括后端组件,例如作为数据服务器,或者包括中间件组件,例如应用服务器,或者包括前端组件,例如具有图形用户界面、网络浏览器或应用的客户端计算机,用户可以通过该应用与本说明书中描述的主题的实现进行交互,或者一个或多个这样的后端、中间件或前端组件的任意组合。系统的组件可以通过任何形式或介质的数字数据通信(例如通信网络)来互连。通信网络的示例包括局域网(local area network,LAN)和广域网(wide area network,WAN),例如互联网。

[0127] 计算系统可以包括客户端和服务端。客户端和服务端通常彼此远离,并且通常通过通信网络进行交互。客户端和服务端的关系是由运行在各自计算机上的计算机程序产生的,并且相互之间具有客户端-服务端关系。在一些实施例中,服务端将数据(例如,HTML页面)传输到用户设备,例如,出于向与设备交互的用户(充当客户端)显示数据并从用户接收用户输入的目的。在用户设备处生成的数据,例如用户交互的结果,可以在服务端处从设备接收。

[0128] 尽管本说明书包含许多具体的实施细节,但这些不应被解释为对任何发明的范围或可能要求保护的范围的限制,而是对特定发明的特定实施例的特征的描述。本说明书中在单独实施例的上下文中描述的某些特征也可以在单个实施例中组合实现。相反,在单个实施例的上下文中描述的各种特征也可以在多个实施例中单独或以任何合适的子组合来实现。此外,尽管特征可以在上面被描述为在某些组合中起作用,并且甚至最初被如此要求保护,但是在一些情况下,来自所要求保护的组合的一个或多个特征可以从该组合中被删除,并且所要求保护的组合可以针对子组合或子组合的变化。

[0129] 类似地,尽管在附图中以特定顺序描述了操作并且在权利要求中陈述了操作,但是这不应该被理解为要求这些操作以所示的特定顺序或顺序地来执行,或者要求所有示出的操作都被执行,以获得期望的结果。在某些情况下,多任务和并行处理可能是有利的。此外,上述实施例中的各种系统模块和组件的分离不应被理解为在所有实施例中都需要这样的分离,并且应当理解,所描述的程序组件和系统一般可以一起集成在单个软件产品中或者封装到多个软件产品中。

[0130] 已经描述了主题的特定实施例。其他实施例在所附权利要求的范围内。例如,权利要求中陈述的动作可以以不同的顺序执行,并且仍然获得期望的结果。作为一个示例,附图中描述的过程不一定需要所示的特定顺序或顺序来获得期望的结果。在一些情况下,多任务和并行处理可能是有利的。

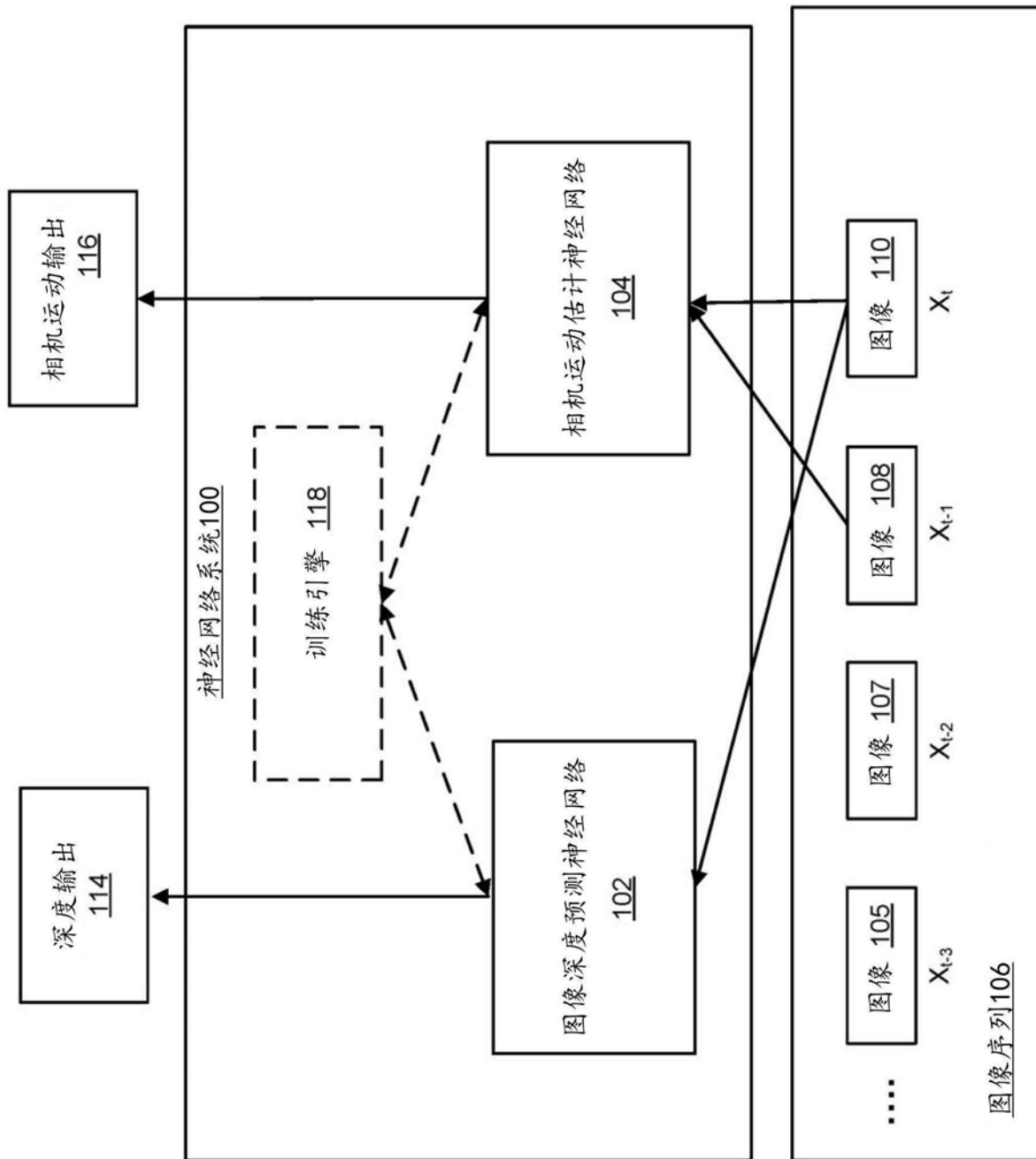


图1

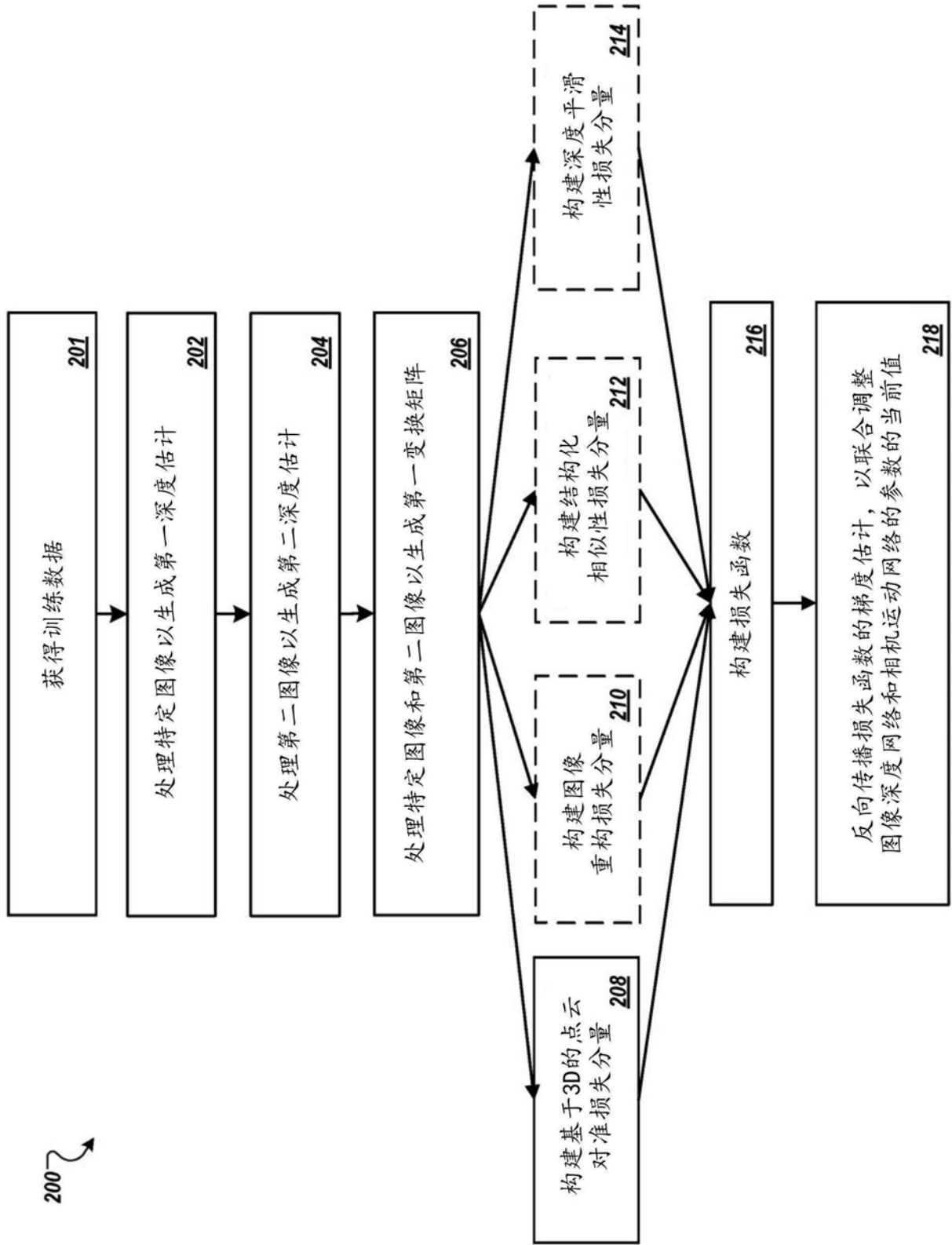


图2

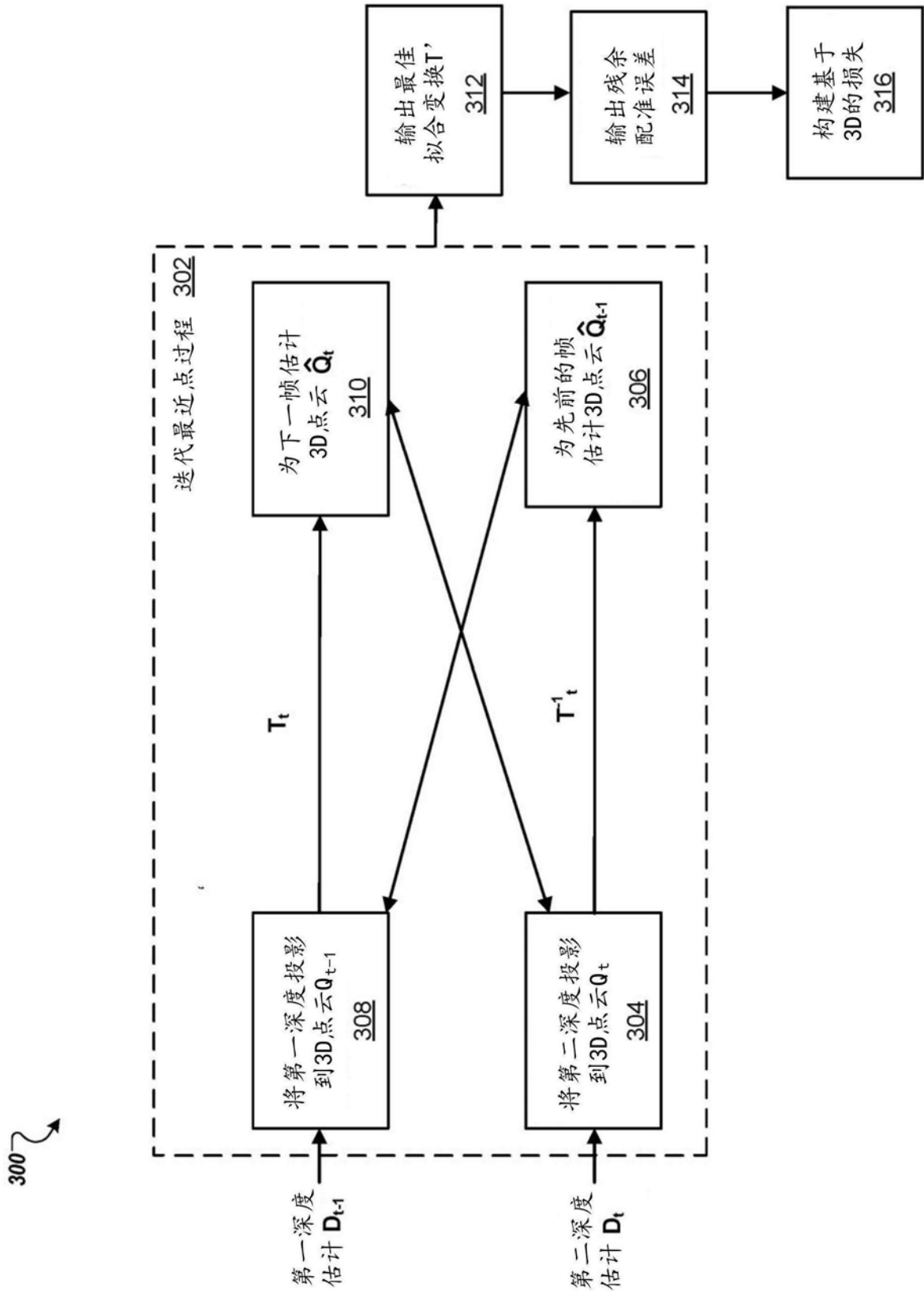


图3

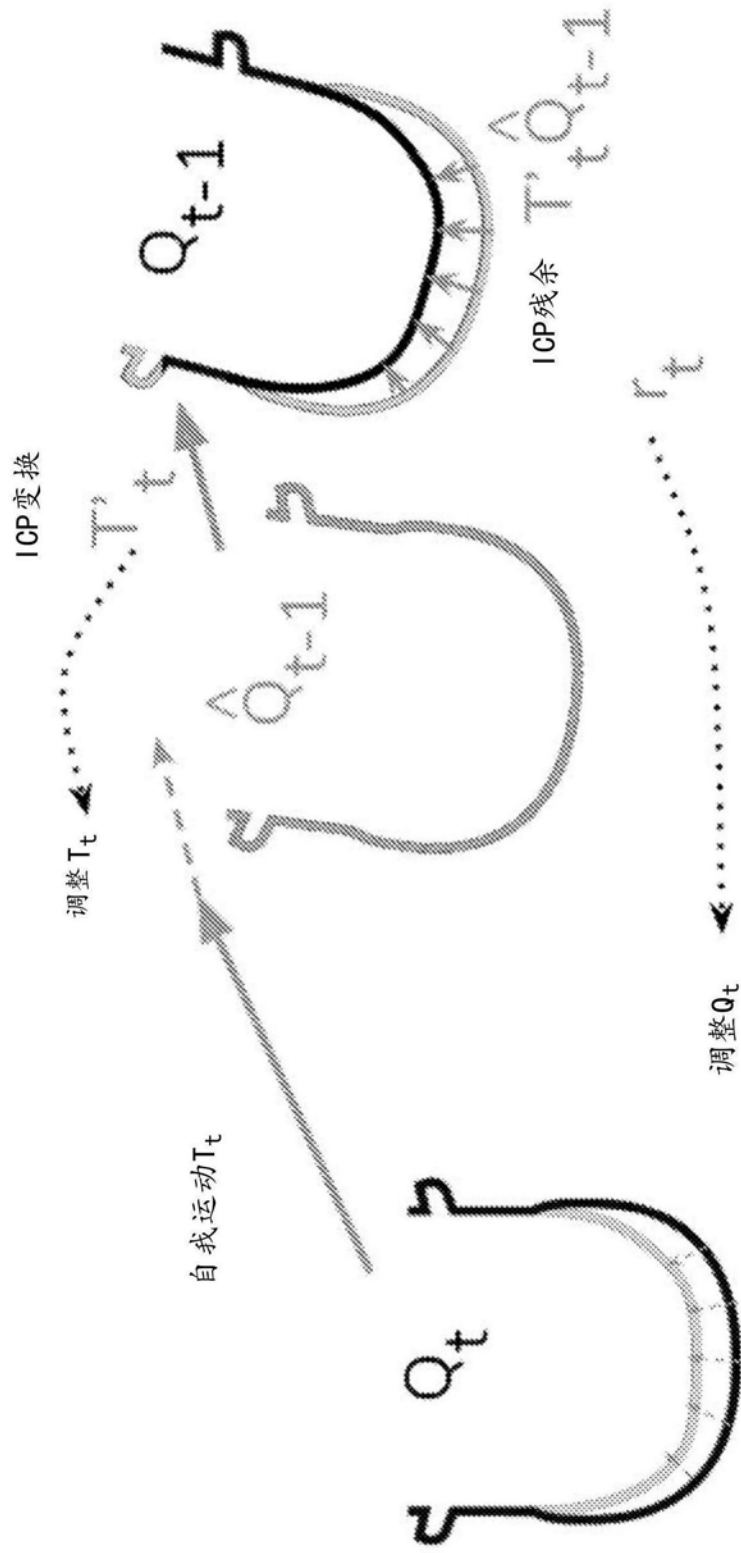


图4