

硕 士 研 究 生 读 书 报 告



题目 智能手机AR的运动深度

作者姓名 侯保蕾

作者学号 21960540

指导教师 李启雷

学科专业 计算机技术

所在学院 工程师学院

提交日期 二○一九 年 十二 月

Depth from Motion for Smartphone AR

A Dissertation Submitted to

Zhejiang University

in partial fulfillment of the requirements for

the degree of

Master of Engineering

Major Subject: Computer Technology

Advisor: Li Qilei

By

Baolei Hou

Zhejiang University, P.R. China

2019

摘要

智能手机的增强现实(Augmented Reality, AR)已经从早期只能在某些高端手机上使用，发展到了真正可供公众使用的技术。其中一项关键突破是仅使用现有硬件（相机和惯性传感器）就可以在手机上进行六自由度(6DoF)跟踪的低计算方法。六自由度跟踪是智能手机AR的基石，它可以将虚拟内容精确锁定在现实世界的顶部。但是，为了真正的给用户可信的AR印象，就需要移动深度。没有移动深度，就连最简单的效果例如虚拟对象被现实世界遮拦都是不可能的。但是，需要一个移动深度传感器会严重限制对此类功能的访问。在本文中，这里提供一种用于移动深度的新的管道。该管道可以支持多种手机，并且只需要使用现存的单眼彩色传感器。通过多种技术文献，这里提供了仅使用多种中高手机的单个CPU内核来计算低延迟密集深度图的能力。最后演示了该方法在高级AR应用程序包括实时导航和购物上的功能。

**关键词**：运动深度，运动立体声，运动结构

Abstract

Augmented reality (AR) for smartphones has matured from a technology for earlier adopters, available only on select high-end phones, to one that is truly available to the general public. One of the key breakthroughs has been in low-compute methods for six degree of freedom (6DoF) tracking on phones using only the existing hardware (camera and inertial sensors). 6DoF tracking is the cornerstone of smartphone AR allowing virtual content to be precisely locked on top of the real world. However, to really give users the impression of believable AR, one requires mobile depth. Without depth, even simple effects such as a virtual object being correctly occluded by the real-world is impossible. However, requiring a mobile depth sensor would severely restrict the access to such features. In this article, we provide a novel pipeline for mobile depth that supports a wide array of mobile phones, and uses only the existing monocular color sensor. Through several technical contributions, we provide the ability to compute low latency dense depth maps using only a single CPU core of a wide range of (medium-high) mobile phones. We demonstrate the capabilities of our approach on high-level AR applications including real-time navigation and shopping.

**Keywords：**depth from motion, structure from motion, motion stereo

1 引言

近年来，移动行业已在智能手机的增强现实（AR）方面进行了大量投资。 人们的手机现在也可以作为AR的取景器，在虚拟世界中，虚拟内容将实时呈现在彩色摄像头的顶部。现在，几乎所有新兴的中高端手机仅使用手机内部配置的传感器比如彩色相机和惯性测量单元（IMU）即可实现一些六自由度（6DoF）的跟踪功能。这种转变要求在视觉惯性里程计（VIO）和同时定位和制图（SLAM）方面取得许多突破。

这些进步促成了一些6DoF跟踪平台的发布，包括Apple公司的ARKit和Google公司的ARCore以及带有AR的应用程序，这些应用程序让用户产生了可以在所处环境中放置和锁定虚拟对象的错觉。但是，一旦出现视觉上的不一致性，比如真实对象与虚拟对象之间的错误遮挡，这种错觉就会消失

为了达到更高层次的沉浸感，需要在设备上实时提供密集的深度图。深度是实现更真实的AR的先决条件，包括正确处理真实对象对虚拟内容的遮挡，更好地放置虚拟内容以及实现真实和虚拟内容之间的交互例如物理碰撞。但是，尽管智能手机AR取得了许多突破，但这些设备当前都无法在不添加新的专用传感器的情况下提供实时密集深度图。尽管这里将朝着专用深度传感器普及的未来迈进，但目前，添加此类传感器意味着智能手机AR的很多普及性和吸引力都将丧失，并对成本，功耗和工业设计产生负面影响。因此，利用单眼彩色相机是将密集深度估计扩展到数百万个现有设备的最佳途径。尽管有关单眼深度估计的文献非常广泛，但当今尚不存在能够以较低的计算量在手机上提供密集且边缘保留的深度图的方法。

本文介绍了一种新颖的流水线，该流水线仅利用单个RGB相机和智能手机的一个CPU内核即可提供30Hz的密集和低延迟QVGA深度图。 图1说明了如何使用此深度图实时逼真地渲染虚拟对象，从而实现可以增强包括购物，街道导航和自我表达在内的各种移动应用程序的新效果。

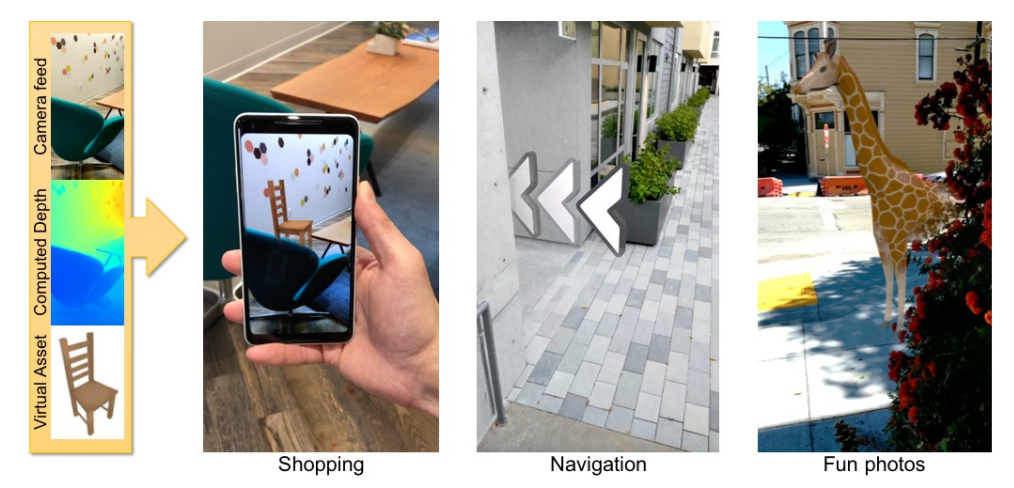


图 1：利用场景

**2 相关工作**

（1）移动深度

这里专注于使用现有的单眼传感器而不是专用硬件来评估手机深度等相关工作。对于有关被动深度估计技术的更一般性讨论，很多文献已经有所阐述，在那些文献里，几乎仅在3D重建的背景下研究了移动平台上的被动深度估计。这些类型的算法将随时间推移的稀疏深度图融合为基础的体积表示形式。有些研究人员在iPad Air 2上以24Hz描述了融合深度图的管道，但假定深度数据是由第三方提供的。不幸的是，由于功耗，散热考虑，因为GPU如果连续不断地最大化会引起散热问题以及需要此类GPU资源可用于渲染的目的，将移动GPU连续用于移动用途的使用被禁止。另一种方法在移动CPU上以15Hz的频率计算了320×240个半密集深度图，并从中提取了碰撞网格。但是，网格物体不会覆盖整个图像，并且对象边界也没有得到明确处理。因此，三角形可以跨越属于场景前景和背景的对象。

（2）后期渲染

有种方法可以减少感知的延迟，后期渲染或时间扭曲。该方法主要是在AR / VR渲染的背景下进行研究的，其中头部运动发生在渲染之后，并且需要在显示之前进行补偿以减少感知到的延迟。为了获得最佳的用户体验，此端到端延迟不应超过20ms。 常见的方法是通过应用单应性将旋转的视点转换为最终的显示视点，从而对异步重投影应用仅旋转校正。完全可感知位置的变形解决方案需要访问场景的完美深度，这对于渲染的内容是可行的，但在一般情况下更困难。在本文中，这里提出了一种新颖的屏幕空间技术来减少延迟，该技术不依赖于准确的深度估计或简化的场景假设。

（3）深度图致密化

大多数立体声或深度恢复算法都包含一个致密化步骤，其中将嘈杂，稀疏或不完整的深度观察结果转换为密集且平滑的深度图，通常使用“参考” RGB图像来鼓励深度边缘与颜色同时出现边缘。

变色绘画基于Total Variation（TV）的方法可以在平板电脑的GPU上实时运行。但是，这些系统需要高于手机的CPU内核的计算性能，同时还带来了与前面所说移动GPU相关问题。

快速双边求解器是另一种去噪和完整深度图的方法。通过解决“双边空间”中的全局优化问题（而不是图像中的所有像素），它可以产生高质量的结果，从而导致运行时间在很大程度上与图像分辨率无关。使其能够更有效地利用移动硬件上的并行化和矢量化功能。本文将基于这种方法。

（4）深度图时间过滤

为了获得时间上一致的深度图，许多作品将时间一致性直接纳入了立体声匹配过程。与这里的方法最相关的是使用了基于引导图像滤镜和双边网格的时空滤镜来平滑成本。其他方法使用时空滤波器对深度图进行后处理，以提供更一致的解决方案但是，这些方法在GPU上执行过滤。这里选择将时间一致性合并到双边求解器中，从而实现计算高效的方法。

**3 系统总览**

当用户手持智能手机在所处环境中时，流程将从使用ARCore [Google 2018]的现成VIO平台跟踪6DoF姿势开始。系统在此阶段也可以使用任何其他VIO或SLAM平台。

一旦6DoF跟踪管道初始化并获得了最新的可用摄像机图像（出于计算原因，一般使用灰度图像），计算深度图的第一步包括从过去的图像帧中确定适合执行立体匹配的关键帧 。接下来，关键帧和当前帧之间的相对6DoF姿势用于执行极性校正。

立体整流不是必需的步骤，但是可以帮助减少搜索直接用两个图像中的相同水平线，可以显着加快立体匹配。一个非常快速的CRF求解器用于计算对应关系。错误的估计值将通过基于机器学习的高效解决方案丢弃，从而导致几乎没有异常值的视差图。

根据视差，可以通过三角剖分估计稀疏的深度图。然后，将稀疏的深度图馈入快速双边求解器的一种改进方法中，该方法生成双边深度网格（与深度图相对）。双边网格可以按需转换为密集的时空平滑深度图。

然后通过使用最新的可用图像（而不是用于填充双边网格的框架）对网格进行切片，以确保生成的深度图的边缘与智能手机上当前显示的RGB图像对齐。

双边网格的生成和切片被分解并在单独的线程上运行。因此，可以以极低的延迟高帧速率生成深度图，有效地使其独立于CRF推断的运行时间，从而允许在不影响质量或有效帧速率的情况下将系统部署在中高端设备上。

最后，高级应用程序可以利用这种实时深度估计来实现诸如AR遮挡之类的效果。图2展示了上述管道的主要步骤。

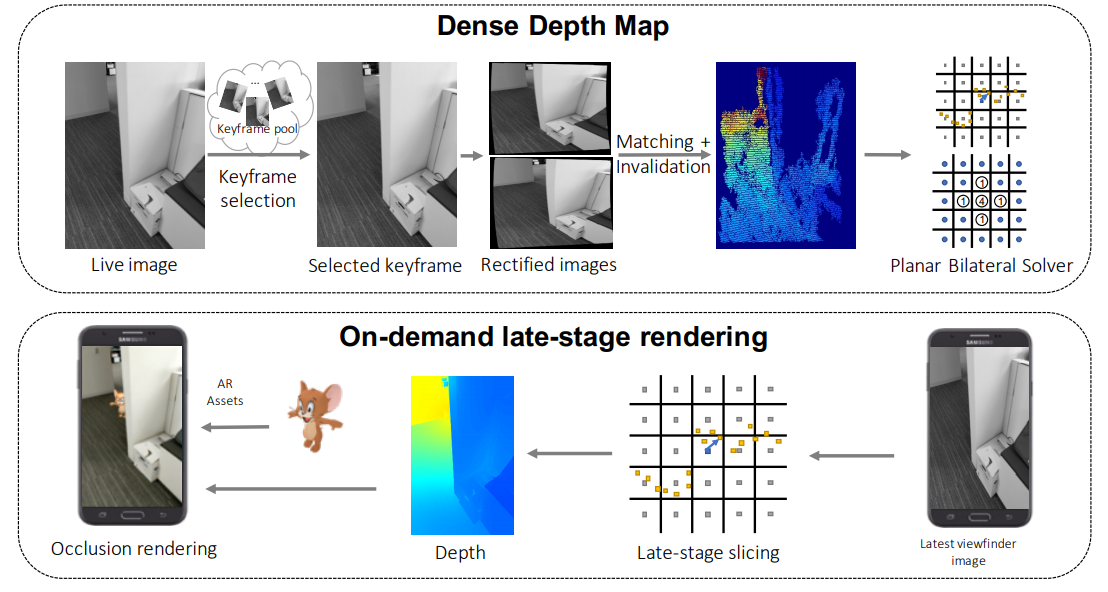


图 2：主要步骤

**4关键帧选择**

因为使用的深度估计方法是基于最新图像和过去关键帧之间的立体匹配。而关键帧的选择取决于几个不同的因素，所以每个因素都会影响候选关键帧的潜在匹配质量。例如，通过增加所选关键帧和当前位置之间的立体声基线可以获得更高的深度精度，但是此类帧的时间也会更晚，这可能会导致时间上的不一致。

有的方法是依赖于固定时间延迟的关键帧，当处理恒定运动时这个方法很有效，但是使用手机记录时难免会有些不均匀运动。其他有些依赖于特种或几何跟踪的方法可能会导致深度稀疏等问题。还有逐像素图选择的方法，但是计算量太大。

这里是定义了一个成本函数。使用多个参数为最新目标帧选择最佳关键帧。 同时会保留一个固定容量的潜在关键帧池，如果成功进行六自由度视觉惯性跟踪，则会将每个新捕获的帧添加到该池中，来替换过时的帧。

这里用于从池中选择最佳关键帧的参数有：

1. : 帧i和帧j之间的3D基线距离。这个值希望尽可能大。

2. : 帧i和帧j根绝他们的视锥度计算的分数重叠。范围在[0，1]。要求最大值。

3. : 帧i和帧j的姿势跟踪统计量的测量误差。越小越好。

使用公式计算关键帧k与最新参考图象r，从关键桢池K中产生最小的总成本

这里直接将设置为0.4，最小允许基线的严格限制为4cm。用加权处理，因为过大的重叠减少了将深度值 外推到目标图像其他部分。这里设定为0.5，确保关键桢和最新参考帧之间速度变化不超过，同时加速度偏差不超过 0.2m/。通过最大程度提高限速深度误差并最小化测试数据集中报告的无效像素的数量来选择参数。最终将深度精度提高了10%，这种方法来确定关键桢。

**5立体声校正**

给出算法在给定一对图像及其相对姿态的情况下，对这对图像进行转换，使两个图像对极线平行且相应的对极线具有相同的垂直坐标。而且，出于降低计算量的需求，通常期望对于左校正图像中的点（x，y），对应关系位于右校正图像中的（x′，y），并且x′＜x，并且τmin<x − x'<τmax，其中τmin是一个小常数，通常为0或1，τmax是最大视差，这里设置为40。 同时提出了一种将解决方案约束到已知视差区间的解决方案，该区间可以在估计对应关系时发挥很大作用。

估计图像翻转。某些相机配置可能会产生校正后的图像，这些图像的模数视差是相反的。翻转后导致x'>x。在世界坐标系下，当穿过光学中心和他们各自的图像中心的点击以及链接两个光学中心的矢量具有相反符号时，其中一张图象需要翻转。

估计图像交换和X位移。高效的立体声匹配假定所有对应关系都位于视差值[τmin，τmax]的预定范围内，并且τmin>0负视差或超过最大视差的视差可能会违反此假设。需要调整，使差异达到有效范围。首先，设定两个帧之间相对变换和深度值的预定范围[Dmin，Dmax]，将左图象或右图象为参考，则间隔分别对应于进行预测所需的视差范围。 然后，根据配置要求最小偏移以及预估视差符号，选择参考图像以及适合于预期视差范围[τmin，τmax]所需的水平偏移量。如果预期的视差范围太小无法容纳当前的图像对，则相应调整校正后图像的大小

改善水平分辨率以获得更高质量的立体声匹配结果。在带纹理的场景上进行操作时，使用PatchMatch Stereo等标准技术子像素精度约为0.2像素。子像素精度直接影响深度精度，调整校正图像的大小，以使[τmin; τmax与最大期望视差数（在这里的例子中为40）匹配，同时保持像素总数不变。

考虑一些姿势不确定性。两个图像之间的相对变换中不精确的结果会导致严重的校正问题。 在实践中，像ARCore这样的系统只会遭受轻微的姿势不精确性。将τmin设置在[5，10]而不是0或1之间可以有效地消除小姿态误差。 当子极位于图像中时，使位于子极周围的像素盘无效。这里的磁盘的半径为20像素。 在此阶段，无论智能手机遵循的轨迹如何，这里都有一对经过校正的图像。

**6 立体声匹配**

**6.1向量化推论**

在这里，主要介绍一种传播策略，该策略是针对依赖单指令多数据（SIMD）指令以实现寄存器内并行性和提高吞吐量的现代CPU指令集量身定制的。要在主要是移动设备的ARM架构上实现高性能，就需要有效利用这些向量寄存器。但向量寄存器的行为给立体算法带来了麻烦，立体算法通常在探索解决方案空间时对每个像素的发散偏移进行采样。

HashMatch允许并行更新每个像素，从而可以在GPU上以1000Hz进行深度图预测。与PatchMatch相比，HashMatch需要更多次迭代才能使信息进一步传播到图像中，但是每次迭代的成本都较低，并且可以独立执行。相比之下，PatchMatch本质上是完全顺序的：迭代地从一个像素移动到下一个像素，为该像素评估，并一直持续到图像结束为止。

出于每种方法的优势，这里提出了一种非常适合SIMD体系结构的改进方案。没有依次对八个方向执行多次独立的遍历，取而代之是执行了k次遍历，每个遍历旨在利用基础矢量体系结构的数据并行性。一般情况，k在2到4之间就足够了。在偶数遍中，每个像素考虑来自其上方三个邻居的假设（即（x，y）处的像素考虑来自（x − 1，y − 1），（x，y − 1），（ x + 1，y − 1））以及当前存储的假设。在奇数遍中，除了当前存储的假设外，每个像素还会考虑其下三个邻居的假设。从偶数遍图像的顶部开始，从奇数遍图像的底部开始，对行进行顺序处理。因此，给定行的所有像素都独立于同一行中的所有其他像素，从而允许并行处理。

另外推理步骤中的NEON加速是可能的。因为SIMD处理器的矢量单元沿X方向以块的形式加载数据，在对离散Y值的像素进行矢量化操作时效率更高。最后测试，总体来说所提出的方法被衡量为比HashMatch快4倍，比PatchMatch快10倍。

**6.2 无效化**

MAP推断会为图像中的每个像素产生一个视差值。但是，当场景缺少纹理或包含重复的图案时，相应像素的MAP解决方案可能是错误的。通常优先使这些像素无效，通常使用对解决方案的一元成本进行阈值化来执行失效，或使用左右一致性检查还可以使用连通分量分析，或以上各项的组合。执行左右一致性检查会产生良好的效果，但涉及为每个图像计算视差图，这会增加大量计算成本。

通过对比过往策略，这里采用决策树的方式进行判断无效化。简要描述决策树的训练过程。决策树由拆分节点和叶子组成。每个拆分节点n存储一个由功能参数θn和标量阈值τn参数化的“弱学习者”。要针对像素p在树上进行推理，从树的根部开始并评估：

如果的值为0，则在节点n的左子节点上继续进行，否则在右子节点上继续进行。 重复此过程，直到到达叶子为止，该叶子包含在预测空间上的二进制概率分布，在这种情况下是无效的。 按照惯例，选择f作为位于p周围的两个像素索引值之间的点积。 用贪心法优化和的值以最大化信息增益：

E表示香农熵，表示节点n的左右子元素。 最后，每个叶子都存储像素有效或无效的概率。在测试时，针对每个像素执行对该树的推断，从而可以确定哪些像素应无效。

**6.3 深度差距**

没有异常值的视差图，可以推断出深度。当使用平面校正时，给定视差d，基线b和焦距f时，深度Z可以简单地计算为Z = 。 但是这个公式不能用于极性整流的情况。

**7双边求解器扩展**

前面的部分描述了如何获得极少误报的深度图。但是，这些深度图是稀疏的，在时间上不一致，并且与图像的边缘不对齐。在本节中使用双边求解器的扩展来有效地生成密集，时间稳定且边缘对齐的深度图，并且延迟时间短。使用预处理的优化来解决稀疏线性系统，双边求解器能够从嘈杂或不完整的输入中生成边缘感知的平滑深度图，并且可以生成在移动CPU上运行的实时结果。在以下各节中，这里描述了求解器的许多改进。首先，这里对求解器进行泛化，以产生在共面意义上平滑的输出，而不是在正平行意义上平滑的输出。可显着提高包含缩短平面（墙壁，地板等）的场景的输出，这在摄影和AR环境中很常见。其次，提出了一种简单而廉价的方法来在求解器中引入实时时间一致性，以及一种热启动求解器的多个实例的方法，该方法可以提高实时/视频处理上下文中的收敛速度。第三，证明了求解器可以用于后期切片，其中这里使用从较早的立体声输入中计算出的双边网格，从最新的取景器帧中生成边缘感知的深度图。结果是仍然具有边缘感知能力的极低延迟的输出。

**7.1平面双边求解器**

双边求解器生成每个像素的标签，标签与数据项一起让在空间上相邻且具有相似颜色或相似灰度强度的像素之间的平方距离最小。因此，忽略数据项求解器的输出是包含单个常数值的图像，在立体空间里，求解器容易偏向于生成平行于平面的深度图。这种偏差是有问题的，实际环境中经常包含平滑或平坦但并非平行的表面，例如地板，墙壁和天花板。确保可以执行全局和边缘感知的平面拟合。这使得深度图不会被缩短的表面所迷惑，因为它可以解释这样的表面，即仅仅是倾斜的平面。

这种方法可以叫做“平面双边求解器”，它在移动最小二乘的情况下将平面拟合到图像中的每个像素，其中每个像素的最小二乘拟合中的插值器是双边求解器的输出。标准的双边求解器可以看作是平面化的双边求解器，已对其进行了严格的规范化处理，以产生最大的正面平行输出-进一步证明了标准双边求解器的正面平行偏置。

**7.2时间平滑度，热启动初始化和后期切片**

为了产生引人注目的用户体验，深度图必须随时间推移保持平滑。时间一致性的目标与这里对响应性低延迟输出（与当前取景器边框的边缘紧密对齐）的需求有些矛盾。 在本节中，为双边求解器和平面双边求解器提供一种简单有效的方法，以产生高质量的，时间上一致的实时结果。

使用类似于因果IIR的方法来实现时间平滑，在该方法中，这里跟踪估计深度的单个双边网格，并使用每个传入帧上的单图像双边求解器实例的输出重复更新此双边网格。这种方法可以实时运行，并允许这里使用后期切片来生成极低延迟的边缘感知输出，这是增强现实应用程序中的一项关键功能。

在这里的时间一致解中，这里跟踪深度为y的双边网格的指数移动平均值，并将其初始化为0。跟踪较小的双边深度网格而不是较大的每像素深度图具有明显的速度优势，因为每帧的更新都需要更少的计算量。以这种方式跟踪和模糊双边深度网格而不是跟踪深度图意味着时间平滑方法对于小型相机或场景运动是不变的，无需显式估计帧之间的每像素运动。同样，在双边空间中跟踪深度使这里可以通过称为“后期切片”的过程来生成极低延迟的边缘感知深度。由于切片比求解速度快得多，因此这种方法能够生成与当前取景器帧完全对齐的每像素深度图，从而实现引人注目的和响应迅速的视觉效果。

这种后期切片方法还允许算法在低端移动设备上正常降级运行，因为在低端移动设备上，每帧深度估计可能要花费17或33毫秒更长的时间。时间平滑方法可实现另一种加速，其中通过使用前一帧的解决方案通过梯度下降初始化要求解的双边网格来“热启动”每帧的双边求解器实例。由于相邻帧具有相似的图像内容（除了存在极端运动时），所以这种方法可以显着改善收敛性，并且除了第一个帧外，所有其他帧中的梯度下降迭代次数都更少。

这里时间一致性方法直接将常规双边求解器推广到平面双边求解器。将时间一致性应用于描述每个像素平面在LDL求解步骤之前的拟合的线性系统，这比将时间滤波器应用于每个帧的估计深度产生了更好的结果。后期切片还包括后期的每像素LDL解决方案，它可以稍微增加运行时间，但不足以阻止实时性能。

**8局限性与总结**

首先，当前帧和所选关键帧之间的相对姿势不精确时，许多组件都会受到影响。比如可能会导致对应关系分开几行，从而大大增加了立体声匹配的难度。另一个限制来自硬件限制，比如可能产生运动模糊。最后，当低纹理区域特别模糊时进行推断通常会导致错误的结果。

文章总共介绍了几项技术成果，这些成果首次可以预测手机的单个CPU内核上的低延迟，边缘对齐和密集深度图。并最终证明了这种方法是有效的，可以产生优秀的虚拟现实效果。未来改进的方向包括使用深度强化学习来直接预测遮挡蒙版，并努力减少此类架构的计算需求。

参考文献

[1] Julien V,Ivan D,Joao A,et al. Depth from motion for smartphone AR[J]. ACM Transactions on Graphics 37,2018(6):1-19.

[2] Robert A, David G, Jonathan T B,et al. Jump: Virtual Reality Video[J]. SIGGRAPH Asia 35, 2016 (6):198.

[3] Fayao L, Chunhua S, Guosheng L, et al. Learning depth from single monocular images using deep convolutional neural fields[J]. IEEE transactions on pattern analysis and machine intelligence 38, 2016(10), 2024–2039.

[4] Enliang Z, Enrique D, Vladimir J, et al.Patchmatch based joint view selection and depthmap estimation[J]. IEEE Conference on Computer Vision and Pattern Recognition.2016, 1510–1517.

[5] Johannes L S, Enliang Z, Jan-Michael F, et al s. Pixelwise view selection for unstructured multi-view stereo[J]. European Conference on Computer Vision. Springer, 2016,501–518