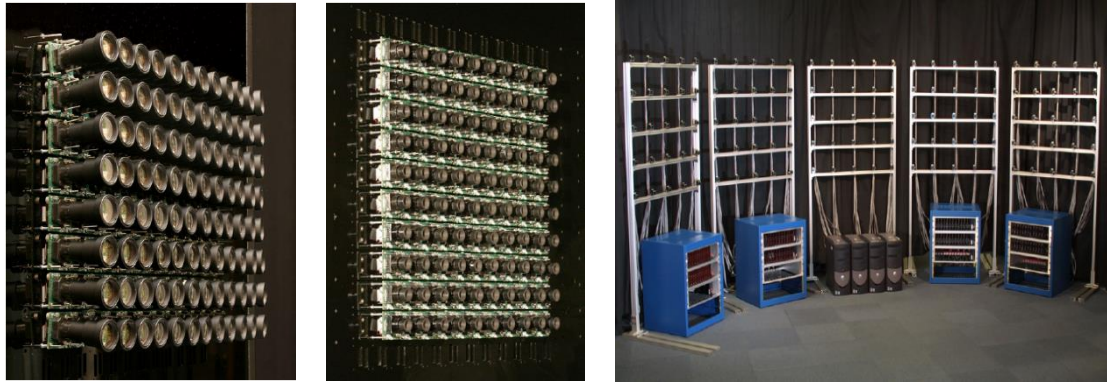


基于大型相机阵列的高性能成像



(a)

(b)

(c)

图 1: 相机阵列的不同格局。(a) 紧密排列并且呈张开状态的长焦相机阵列。这种排列是用于高分辨率成像 (4.1 节)。(b) 紧密排列的广角镜头相机阵列, 旨在用于共享相同的视野。这种排列是用于高速视频采集 (4.2 节) 和混合孔径成像 (6.2 节)。(c) 空间宽阔的相机阵列。还能看见每个装着相机的处理板的机柜和用来运行系统的四个 PC 主机。

摘要

廉价的数字图像传感器的出现和结合一系列传感图像的信息来创造照片的能力, 正在改变着我们对摄像的认识。在这篇论文中, 我们描述了一个已经搭建好了的一个独特的由 100 个定制的视频相机构成的相机阵列。我们总结了这个相机阵列在一系列成像应用中使用的经验。我们的目标是探索系统的能力, 这个系统在未来生产费用不贵。有了这个想法, 我们使用简单的相机, 镜头和配件, 并且我们假定, 对大量图像的处理最终会容易和便宜。我们已经探讨过的应用包括将常规的单中心投影视频摄像机和高性能结合起来, 比如分辨率, 动态范围, 帧速率, 大光圈。还包括使用多个相机来达到和大的合成孔径的相机接近的效果。这使得我们能够捕捉到一个可以应用时空视图插值算法的视频光场, 从而可以用数字模拟时间扩张和相机运动。还使得我们可以使用自定义的非均匀合成孔径的视频序列。

CR 类别: I.4.1 [Image Processing and Computer Vision]: Digitization and Image Capture.imaging geometry, sampling; C.3 [Computer Systems Organization]: Special Purpose and Application-Based Systems.real-time and embedded systems

关键词: 相机阵列, 时空采样, 合成孔径

1 简介

半导体产业的经济原理之一是在大批量销售的产品是便宜的，而低容量的产品更贵，几乎独立于零件的复杂性。对于计算机来说，这种关系已经改变了人们对于建设高端系统的思维方式。相比于建设一个定制的高端处理器，使用大量的常用处理器会更符合成本效益。

我们现在在数字成像方面看到了类似的趋势。随着数码相机的普及，低端成像设备的性能持续改进，而高端相机的成本保持相对恒定。此外，研究人员已经表明静态场景的多幅图像可以用来扩展这些相机的性能范围。例如，创建高分辨率【Szeliski 1994】或动态范围【S.Mann and R.W.Picard 1994; Debevec and Malik 1997】的图像。还有，Schechner和Nayar基于旋转摄像头使用空间变换滤波器创建高分辨率的全景图，也具有高动态范围或高光谱分辨率【Schechner and Nayar 2001】。多视图的另一个应用是视图插值，用来在静态或动态场景中创建平滑移动相机的假象【Levoy and Hanrahan 1996; Gortler et al. 1996; Rander et al. 1997; Matusik et al. 2000】。

这些成果大多是移动一个单一的高品质相机来观察静态场景。为了在冬天场景中取得类似的效果，我们需要使用多个摄像头。这就促使我们在 1999 年考虑设计一个灵活的包含大量廉价视频成像设备的阵列。最终这个多相机阵列由 100 个相机组成，每个相机都连接在自己的处理板上。处理板能够进行进行局部图像的计算，包括 MPEG2 压缩。

在第 2 节中，我们回顾了以前关于多相机系统的工作，虽然这些系统一般都是针对特定的应用程序，但是他们对灵活的采集系统的要求提出了宝贵的见解。在第 3 节中，给出了我们的多个摄像头阵列的概览，并且更深层解释了一些新特征，使之成为一个通用的研究工具。

本文的其余部分集中讲述了我们最近在不同成像应用中使用相机阵列的结果。我们开始通过探索如何使用多个相机来创建一个聚合的，性能超过单个相机能力的虚拟相机。由于这些应用程序本打算实现的效果要和只有一个单一投影中心的相机比较接近，因此通常使用密集的相机。特别是，第 4 节探讨了超高分辨率的相机的创建，这些相机通过调整保证有适度的重叠区域。然后我们努力向内调整相机的角度知道他们有完全重叠的区域，我们使用我们系统的精细定时控制来提供一个超高帧率的虚拟相机。在这两种应用中，大量的相机会有一些单一相机系统所没有的优势。对于虚拟的高分辨率成像设备，可以对每个相机进行单独的曝光测试，这对于空间亮度变化的场景，我们能够组成高动态范围的拼接图案。对于虚拟高速成像设备，可以集成每帧达到帧率比单相机高，从而获得在单位时间内比单台高速相机获得更多的图像。

第 5 节和第 6 节考虑的是相机分散开的应用，从而创建一个多视角相机。这类数据的一个重要应用是视图插值，其目标是在捕获的视点间平滑地移动虚拟观察者。在视频光场方面，这个问题就变成了一个时空差值的问题。第 5 节表明，解决这个问题的最佳采样模式使用的是交错的相机，不重合的，有触发时间的相机。它也描述了一个时空插值方法，这个方法使用了一种新的光流的变体在时间和虚拟的相机位置上来平滑阵列中的插值数据。

第 6 节我们考虑结合多个角度的图像来创建合成孔径图像序列。如果我们对齐，移动，均衡所有的相机的图像，那么我们就可以得到一个非常接近于有一个大的单孔径的相机。通过改变移动的量，我们可以聚焦不同深度的合成相机。使用每个相机处理板的处理能力，我们可以实时聚焦合成孔径相机，即视频捕获。另外，我们可以通过调整孔径的形状来符合特定的场景的特点。例如，我们通过在时空上调整孔径形状来冻结一个自然场景中的高速风扇。

2 早期的相机矩阵

早期的从多个角度捕捉场景的系统使用的是单个转换相机【Levoy 和 Hanrahan 1996】，而且局限于静态场景。Dayton Taylor 通过使用一个由静态相机组成线性阵列把这个想法扩展到了动态的场景。同时触发相机并且从一个相机图像跳到下一个，他通过冻住动态场景创造出虚拟相机的运动错觉。Manex Entertainment 使用空间间隔更宽的相机，并且在相机之间增加了一个可调整的触发延迟，从而使得捕获的图像与一个飞翔在场景中的虚拟高速相机捕获的图像一致。这些系统都是用静止的相机，所以只能在捕获一个特定的虚拟相机的时空轨迹，这个轨迹由相机的排列固定。

为了捕获更一般的数据集，研究人员转向了相机阵列。和静止的相机一样，摄像机必须同步，但是他们也提出了一个新的挑战：巨大的数据速率。多相机阵列的先驱设计是 1997 年 Rander 等人的 Virtualized Reality TM project 项目。他们的目标是捕捉一个场景的许多视图，用于视图插值。他们设计的系统的第一个版本使用多个 VCR 记录视频，这就使得他们拥有几乎无限的记录时间但是质量低。他们的第二个版本使用了 49 个摄像机捕获图像到计算机的主内存中。这个系统有了更好的质量（VGA 分辨率为每秒 30 帧），但仅限于九次捕捉时间。第三个摄像机是用来捕捉彩色视频的。为了解决相机的带宽问题，他们把每三个相机接在一台计算机上。

当 Virtualized Reality TM project 项目使用相对高质量的相机的时候，另外两组实验使用廉价相机组成的大阵列。2002 年，Yang 等人由常用网络摄像机构成的 8x8 的分布式光场相机阵列（Distributed Light Field Camera）渲染实时动态的光场。2004 年，Zhang 和 Chen 的自重构相机阵列使用 48 个常用的带电子水平平移控制的以太网相机来改善视图插值的效果。从每个相机成本上来说，尽管这些系统的设计使其和 Virtualized Reality TM 相比更便宜一些，但是为了使用这些常用的相机，他们做了重大的妥协。首先，相机阵列中的相机是不能同步的，这就会造成视图中的物体会重建。此外，因为他们是在寻找单一的应用，所以他们的系统没有解决建立一个大型通用的相机阵列时关于带宽的挑战。Yang 等人选择做的是一个有限视野（finite-view）的系统，也就意味着每个相机每一帧里面只发送足够的数据来重建一小部分光场视图。Zhang 和 Chen 的相机使用了 JPEG 压缩，但是他们选择以太网和一个单一的计算机来运行相机阵列，会限制他们只能获得每秒钟 15 到 20 帧的 320x240 分辨率的图像。

这些努力的结果有助于指导我们的系统设计。因为我们的目标是创建一个通用的系统，我们需要严格控制相机的时间和位置。我们还需要能够记录所有相机的数据，但是计算机的数量少于 Virtualized Reality TM 系统。我们针对这些目标设计的系统在下文描述。

3 多相机阵列



图 2: 我们的相机瓷片包含 Omnivision 8610 图像传感器，无源电子和一个嵌入的镜头。带状电缆传送视频数据，同步信号，控制信号，还有瓷片和处理板之间的能量。为了保持低成本，我们使用固定焦距，固定孔径的镜头。

当我们想用现成的计数构建相机阵列时，会发现没有一种商业相机可以满足我们的系统对定时和位置的灵活性的要求。因此，我们决定建立一个自定义的成像阵列，但我们利用现有的标准，尽可能的减少系统所需的定制硬件的数量。这个系统的初步版本的描述发表在 2002 年的 Wilburn 上。

3.1 硬件组件

我们的系统包括三个主要的子系统，相机，本地处理板和主机电脑。相机安装在小型印刷电路板上，可以根据自己的安排达到最大的灵活性。每个相机的瓷片是通过一个 2 米长的带状电缆连接到本地处理板上。这些处理板配置每个相机，在发送图像数据给计算机之前，可以在本地处理原始图像数据或者压缩的 MPEG2 视频流。系统由 4 个计算机组成，将收集到的数据存储到磁盘上，或者处理之后实时显示。

相机瓷片。相机阵列中的最重要的选择之一就是图像传感器和它们的光学系统。我们原以为计算的代价会越来越低，但是我们发现这个论断并不适合高品质的镜头。因此，我们选择使用廉价的镜头和光学元件，以及廉价的传感器。特别的，我们选择了 Bayer Mosaic 颜色

滤波矩阵的 CMOS 图像传感器，这个颜色滤波矩阵由 Bayer 于 1976 年提出。虽然他们比 CCD 成像设备有更多的图像噪声，但是 CMOS 传感器提供一个数字接口，而不是一个模拟接口。而且他们提供了一些方便的数字控制，比如增益，偏移，曝光时间。这使得系统更容易集成。

图 2 显示了一个相机瓷片。对于室内应用，通常需要一个大的工作空间和大的区域深度。由于这些原因，我们使用 Sunex 公司的 DSL841B 镜头，这个镜头有 6.1 毫米的焦距，F 数（相对孔径的倒数，用 F/#表示）是 2.6，还有较宽的 57 度视角。对于需要一个狭窄的视野（一般在室外），我们使用 Marshall Electronics 的 V-4350-2.5 镜头，这种镜头有 50 毫米的固定焦距，F 数是 2.5，并且从有 6 度的视角大小。这两套光学元件都包含红外滤光片。

相机瓷片每边测量 30 毫米，安装时支持三弹簧螺钉。这些螺钉不仅可以固定相机，还可以让我们在任意方向上粗略的调整 20 度。对于紧密堆积的相机排列。我们直接把瓷片安装在丙烯酸板子上。对于更宽阔的排列空间，我们设计了塑料适配器连接到 80/20（一个工业框架系统）组件上。

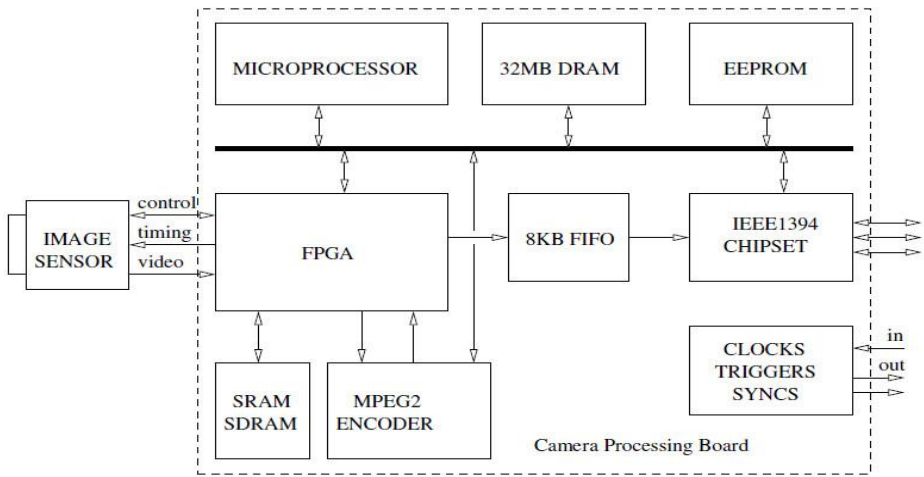


图 3：相机处理板框图

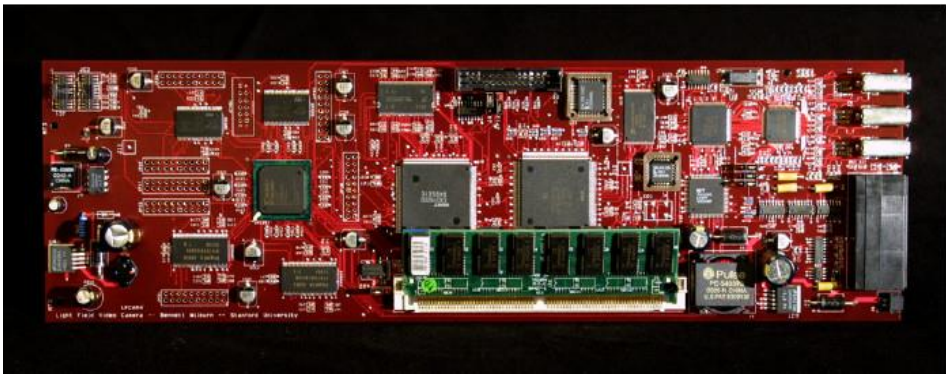


图 4：相机处理板

本地处理板。图 3 显示了一个完整的相机系统的框图，图 4 显示了相机处理板。处理板有五个主要的子系统：一个微控制器，一个存储器，一个 MPEG2 压缩器，IEEE1394 接口，时钟接口和一个座位主数据路由器和可编程图像计算单元的 FPGA 板子。通过选择既定的标

准，这些子系统都可以用现成的芯片组实现。

我们选择了 IEEE1394 高性能串行总线座位处理板和 PC 机之间的接口，IEEE1394 高性能串行总线架构由 Anderson 于 1999 年提出，也被称为 FireWire 和 i-Link。他的默认带宽保证了同步传输可达到 40Mb/s，即数据以恒定的速率发送。对于视频流来说这是完美的，而事实上许多数码摄像机都是通过 IEEE1394 传输到计算机上的。他也非常适合模块化、可扩展的设计，因为它允许多达 63 个设备在每个总线上，并且支持插件和播放。IEEE1394 的另一个优势是设备之间的电缆可以达到 4.5 米长，整个总线可以跨越 250 米。因此，基于这样一个系统的摄像机排列起来可以离得比较开，可能跨越一个建筑物的边缘部分。

即使有这样的高速接口，由 100 相机（640x480 像素，30 帧每秒，每个像素一个字节，Bayer Mosaic）组成的相机阵列将需要大约 25 个物理总线来传输原始数据，大约每秒 1GB，和相当数量的电脑来接受这些数据。我们决定通用 MPEG2 压缩视频然后发送给计算机，而不是限制图像的大小和帧率。SONY 编码器产生的默认 4Mb/s 的比特流转换成一个压缩率为 17.5:1，分辨率为 640x480，帧率为 30fps 的视频。为了确保压缩不会引入工件到我们的应用中，我们设计的相机阵列在流传输压缩视频的同时还可以在内存中同时存储多达 20 帧的原始数据。这让我们可以离线检查 MPEG2 压缩视频和原始视频的完整性。

嵌入式微处理器管理相机组件，并且通过 IEEE1394 与计算机通信。FPGA 是用来将图像数据正确的传送到目的地，通常是 IEEE1394 芯片或 MPEG2 压缩芯片。可以配置 FPGA 利用其本地 DRAM 存储临时变量和常量，也可以利用其 SRAM 作为帧缓冲，从而直接操作图像数据。一个小的引导 ROM 中的代码配置了 IEEE1394 接口，以至于计算机可以下载一个更复杂的可执行文件和配置代码到处理板中。

3.2 系统结构

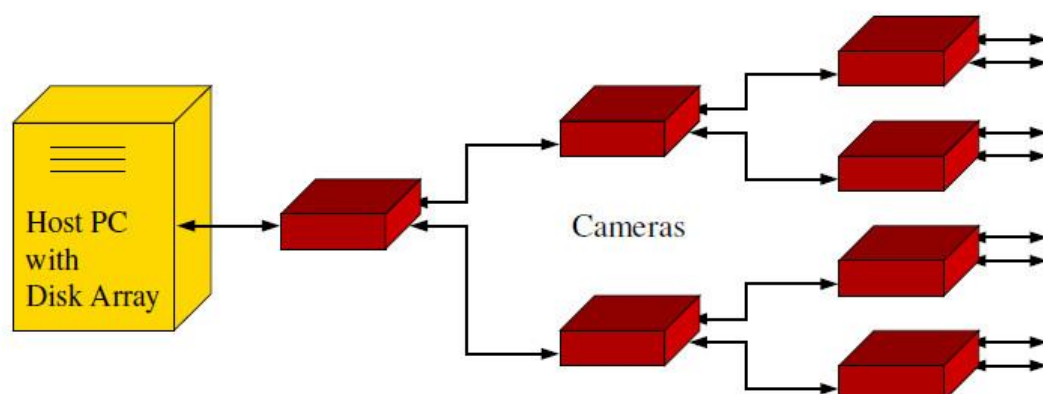


图 5：相机阵列结构

图 5 显示了我们系统的高层体系架构。我们的每一个摄像头都是一个独立的带 3 个端口的 IEEE1394 设备。这些相机都链接在一棵树上，一个端口链接父节点，另外两个端口连

接子节点。根节点的父节点连接的是计算机，计算机上有两个条纹的 IDE 硬盘可以捕获图像数据。对于大的相机阵列，我们使用多个计算机和 IEEE1394 总线。从理论上讲，IEEE1394 的 40Mb/s 流传输带宽应该容纳 62 个压缩的视频流，但是实施细节（总线仲裁还有无法获得总线准确的周期控制）只允许我们在每条总线上使用 30 个相机。我们运行一个网络摄像机控制应用程序，让我们从一台计算机上操作整个相机阵列。

对阵列的时间要求比使用 IEEE1394 通信更严格，特别是多台计算机的时候。为了实现所需的时间性，我们使用额外的 CAT5 电缆芯片给整个阵列设定了一个通用的时钟和触发型号。这些电缆大致符合 IEEE1394 布局。除非他们形成一个单一树，甚至使用了多个 IEEE1394 总线。阵列中的一个单一的主引导板自己产生频率为 27MHz 的时钟，通过 CAT5 电缆发送给他的两个子节点，然后缓冲该时钟并继续发送给后续的两个子节点，依次发送下去。主引导板也会产生一个触发，这个触发也会被缓冲，并且所有其他的板子也会被触发。这个触发在两个处理板之间同步相机，提供偏差不超过 200ns 的定时信号。从这个观点加以考虑，200ns 是我们的最小积分时间 205 μ s 的千分之一。

大多数系统使用触发器来同步所有的摄像机。事实上我们系统的早期圆形（2002 年 Wilburn 等人提出的）也是这样用的。最终的系统为每个相机提供了一个任意的，时间恒定的相位移动。因为图像传感器的定时信号由 FPGA 产生，这是通过向 FPGA 代码中添加可编程定时器复位值。因此，使用一个触发信号，我们可以重置所有的相机到任意相位偏移。

3.3 结果

我们的多相机阵列从 100 个相机捕获每秒 30 帧的 VGA 视频到 4 台电脑中。默认的 MPEG 比特率为 4Mb/s，但是我们可以自由的改变视频的比特率或者传输视频流。比特率为 4Mb/s 时，在我们的操作系统达到 2GB 文件大小限制之前我们最多可以捕获时长 2 分 30 秒的序列。我们还没有必要要延长这一限制。

4 改进的成像性能

通过结合相机阵列中所有相机的数据，我们可以创建一个聚合的虚拟相机，这会大大的提高性能。尽管可以设计光学系统，以确保所有的相机有一个共同的投影中心，但是这些系统会随着相机数量的增长变得昂贵又复杂。相反，我们把相机都尽可能的聚合在一起，达到和单投影中心相同的效果，并且从软件上补偿视差。在这里，我们讨论两种高性能应用：高分辨率、高动态范围视频捕获和高速视频捕获。

4.1 高动态范围高分辨率视频

如果我们把相机排列的很紧，并让它们有邻接或部分重叠区域，那么我们就创建了一个高分辨率的摄像机。使用此配置和现有的图像拼接技术的文献，我们可以记录并融合图像来创建一个单一的高分辨率图像。使用多相机的一个优点是，我们可以单独测量他们。假定每个相机的狭窄的视场中的动态范围足够小，那么和我们单一的相机能够记录的范围相比，这让我们能够捕捉到更大的动态范围的场景。有些场景，本地的动态范围超过我们的传感器的能力，我们通过增加相机视场范围的重叠度，使带不同曝光设置的多个相机能够观察到每哥视线。

为了证明这个想法，我们把相机排列在 12×8 的矩阵中，相机之间有约 50% 的重叠区域，如图 1 (a) 所示。每个相机都有一个长焦镜头，大约有 6 度的斜角。当相邻相机之间有 50% 的重叠时，场景中的大多数点都能被 4 个相机观察到，而且整个阵列能观测到水平方向 30 度和垂直方向 15 度的范围。

颜色校正。由于我们的阵列中的廉价传感器有不同的颜色相应，我们必须与他们的颜色匹配，以防止在图像拼接中受工件影响。在任何涉及多相机的应用中，颜色校正都是很重要的，但是在本次应用中至关重要，因为图像的不同部分由不同的相机记录。如果我们希望创建高动态范围的图像，我们还必须确定我们的相机的响应曲线。由于没有使用相机的伽马校正，除了在输出范围的高低两端，我们的传感器的响应曲线是合理的线性。我们已经设计了一个自动的颜色匹配的例程，通过迭代调整每个相机的每个颜色通道的偏移和增益，能让所有相机和所有的颜色通道的线性响应是一样的。我们的目标是确保一致性，并不完全精确的最终图像可以使用最后的变换转换到另一个颜色空间。

颜色校正例程的每一次迭代采用的是几个不同曝光水平下的白色目标的图像。目标离阵列足够的近，已填充所有相机的视场。曝光设置是传感器集成光的实际持续时间，是非常准确的。该例程计算传感器响应的斜率和偏移量，然后计算新的设置以匹配目标相应。我们选择最小响应 20 到最大响应 220 之间做一个线性映射，稳定在我们的传感器的线性范围之内。使用白色目标的图像为每个通道进行这样的操作也会对我们的传感器有一个白平衡的效果。整个过程需要不到一分钟的时间。

HDR 图像拼接组装。我们使用 Brown 和 Lowe 与 2003 年提出的 AutoSittch 来创建我们的拼接图像。AutoSittch 采用的是尺度不变特征检测器检测重叠图像中的对应的特征，用光束平差来估计将所有图像排列在一条线的全局最优单应矩阵，多频带混合算法结合并配准图像成为一个拼接的图像。相机不需要有精确的瞄准，因为 AutoSittch 可以找到合适的单应矩阵进行无缝拼接。鉴于我们的相机之间相隔 34 毫米，相机和场景之间相隔 120 米，我们可以容忍 ± 20 米的深度变化，在拼接的图像中少于 0.5 像素的视差。

在我们的应用中，我们对 AutoStitch 进行了两方面的修改。首先，我们使用我们的响应曲线和相机的曝光时间将像素值从相机转换成一个浮动的点，在混合前的相对幅度值。因此

混合输出的结果是一个浮点图像。我们的第二个修改是一个有信心的措施，调整了多频带混合的权重，在传感器响应的中间像素值调整权重高，在饱和的或曝光不足的像素调整权重低，在每个镜头的边缘像素的权重也调整为低。

结果。图 6 显示了由均匀独立曝光时间的相机捕获的 3800x2000 像素拼接图像的比对。均匀曝光会在明亮的小山和黑暗的前景树丢失细节。单独计量的相机捕捉更大范围的亮度，但是他们任然有饱和和曝光不足的像素，这些像素已经超出了动态范围了。可以利用相机在视场上的重叠区域来在每一个点根据不同的曝光时间成像，从而生成一个比较好的图像。图 7 (a) 显示了使用 4 种曝光时间 (0.2ms, 0.62ms, 1.4ms, 3.07ms) 获得的拼接图像。增加的本地动态范围可以在覆盖的人行道上看到，如插图 7 (c)



(a)



(b)

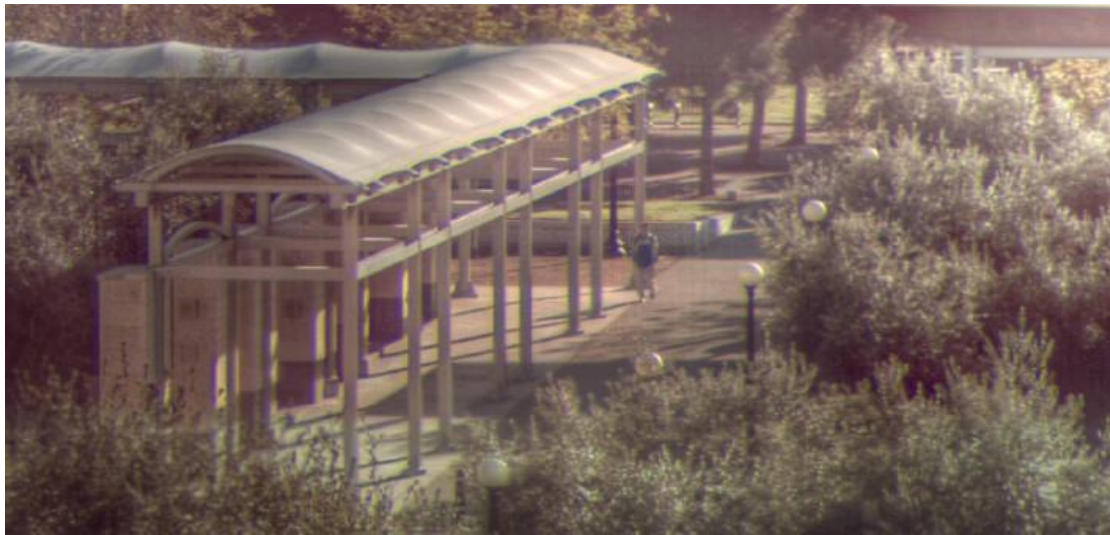
图 6: 高动态范围全景视频。通过单独测量相机，我们可以提高全景视频的总动态范围。

(a) 在这张图片中，所有的相机都设置为相同的曝光。注意在阳光下和阴凉区的饱和区域。

(b) 这张拼接图像，每一个相机的曝光都设置成传感器范围的中间部分的平均像素值的程度，以及由此产生的高动态范围图像定调之后显示（和打印）。还显示出来了更多的细节，包括雷达盘和地平线上的小山，还有前景树的黑色区域。覆盖的人行道的顶层已经超出了观测摄像头的范围。灰色的颜色是由于色调映射，我们并不知道顶层有多么明亮。全景图的左上方的天空也被曝光了。



(a)



(b)



(c)



(d)

图 7：使用佳能 20D 比较。(a) 设置曝光时间，所以每个像素由四个不同的曝光时间（0.20ms, 0.62ms, 1.4ms, 3.07ms）的相机观测。该方案和图 6 (a) 和图 6 (b) 相比增加了拼接图像的本地动态范围。插图 (b) 表明了我们现在有覆盖的人行道的有效数据。图 (a) 中全景图的边界的颜色变化是因为观看那部分场景时曝光时间比那四种曝光都少。这就导致我们没有有效数据的领域中的工件。(c) 佳能 20D 拍摄的相同场景的图像，其分辨率为 3504x2336 像素。(d) 佳能拍摄的覆盖人行道的查到，用做比较。我们的全景图像和佳能图像相比有更大的动态范围。然而，佳能的图像是清晰的，比全景图像的对比度高。后者是由于我们相对低质量的镜头的杂散光和像差。

对图像的整体质量的评估，我们使用 3504x2336 像素的佳能 20D 相机拍摄了一张照片，配置了几乎相同的视场，并和我们高分辨率视频的一帧进行比较，如图 7 (b)。结果是令人鼓舞的，当插图表明佳能图像比较优越的时候，有效的分辨率差异还不大。沿着两

个图像的边缘绘制像素强度图表明佳能的分辨率大约要高 1.5 倍。因为我们可以很容易的增加相机，或者减少重叠，以提高分辨率，这退化的分辨率并不是一个严重的限制。事实上，使用我们的相机测量分辨率图表明他们的有效分辨率是水平方向 400 像素，而不是 640，所以拼接图像的分辨率是不比我们看到的单一图像差的。

更令人惊讶的是，我们的图像拼接的对比明显比 D20 差。这是由于镜头中光的泄露和像差。总体而言，这些结果表明，使用大量廉价的相机建立一个高动态范围高分辨率的虚拟相机是可能的。在这个例子中，我们的重叠区域比较大，所以四个摄像头能够看到每一个像素。我们的阵列可以很容易改变配置，减少重叠区域来创造更大的拼接图像。例如，减少相机的重叠度到 10% 时使用相同数量的相机会产生非常大的拼接图像（大约 6900x3500 像素）。（记住，这些都是相机，还没有已知的相机有这么大的分辨率）。这种灵活性提出了如何最佳分配用于成像的相机视场的问题。这个反过来又取决于场景的动态范围和用于自适应设置曝光时间的算法。我们正在寻找相机阵列的自适应测光算法，以解决这个问题。

4.2 高速视频

以前的应用会利用我们的灵活的支撑系统和曝光控制，以提高分辨率和视频捕捉的动态范围。我们的阵列的时间精度提供了另一个创建一个高性能聚合相机的机会：高速视频捕捉。我们先前已经描述过 2004 年 Wilburn 等人提出的通过均匀交错调整相机的触发时间在 30Hz 帧时间将相机阵列配置成一个单一的，虚拟的，高速视频相机的方法。从使用 52 个紧密排列的相机面向全重叠区域来看，我们模拟了一个 1560fps 的视频相机。

在本应用中使用相机阵列的一个好处是帧率和相机的数量呈线性关系。此外，每个相机并行的压缩视频降低了瞬时数据速率，并允许我们用几分钟连续地传送到磁盘上。相比之下，典型的商业高速相机在本地内存中捕捉时间是有限的，往往很低只有几秒钟，并且在高速事件时需要一些手段来同步捕获。最后，不像一个单一的相机，每帧的曝光时间可以比高速帧率的倒数大一些的。换句话说，我们可以在相机之间让部分帧时间相同。在增加的运动模糊成本方面，这让我们可以收集更多的光，减少噪音。对捕获相机在时间上做逆卷积，我们可以恢复一些丢失的分辨率。Wilburn 等人与 2004 年提出，Shechtman 等人与 2002 年提出。

在图像拼接之前，我们必须考虑不同相机视图之间轻微的视差。我们假设一个相对较浅或较遥远的场景，使用平面的单应矩阵对齐所有相机的图像到所需的对象平面。这会导致对象工件不在假定的场景深度。在下一节，我们扩展了这个高速的方法来适应更宽阔的相机，在 5.2 节我们描述了一种技术，用于相机视图之间的插值。正如我们将看到的，这种技术在高速视频中还可以用来纠正偏差。

5 时空采样

我们现在转向了使用阵列的另一种方法：相机分开采样，得到一个非常宽的空间孔径。从这样的排列中捕获的数据，可用于合成孔径的摄影，视图插值，对场景结构和运动的分析。我们在第六部分探讨合成孔径摄影。对于其他的两个应用，一个主要的挑战是建立不同视图上的点之间的对应关系。一般而言，视图之间的运动最小的时候，计算点的对应关系的算法效果更好一些。在这一节，我们会展示如何通过错开相机的触发时间来减少动态场景的图像之间的运动。5.2 节介绍了一种新的基于光流的视图插值算法。

5.1 平面相机阵列

从数字上推理时间和空间上的采样，我们将考虑一个平面相机阵列，相机阵列的所有图像都对准一个共同的前端平行的基准面。这种排列是用于光场渲染，以及本文中的诸多应用中。我们也可以构建出三角形的相机，这种也适合分析。鉴于这个框架，我们提出了 2 个问题。首先，从不同位置和时间的一张图像的最大可能的运动是什么？其次，我们应该如何触发固定帧率的相机来减少图像的运动？

图 8 显示了在和场景的几何和速度相关的参考平面上的图像是如何运动的。我们假定参考平面上的场景有远近深度限制，分别用 Δz_{near} 和 Δz_{far} 表示。参考平面放置的最佳位置是深度 Z_0 ，由 Chai 等人于 2000 年首次提出 Z_0 的概念。相机的间距是 Δx ，对齐图像中距离参考平面 Δz_p 的一点 P 的视差是 Δp 。 $\Delta p = \Delta x \cdot \Delta z_p / (\Delta z_p + Z_0)$ ，如果我们定义该点的相对深度为 d ，则 $d = \Delta z_p / (\Delta z_p + Z_0)$ ，简化的结果就是 $\Delta p = \Delta x \cdot d$

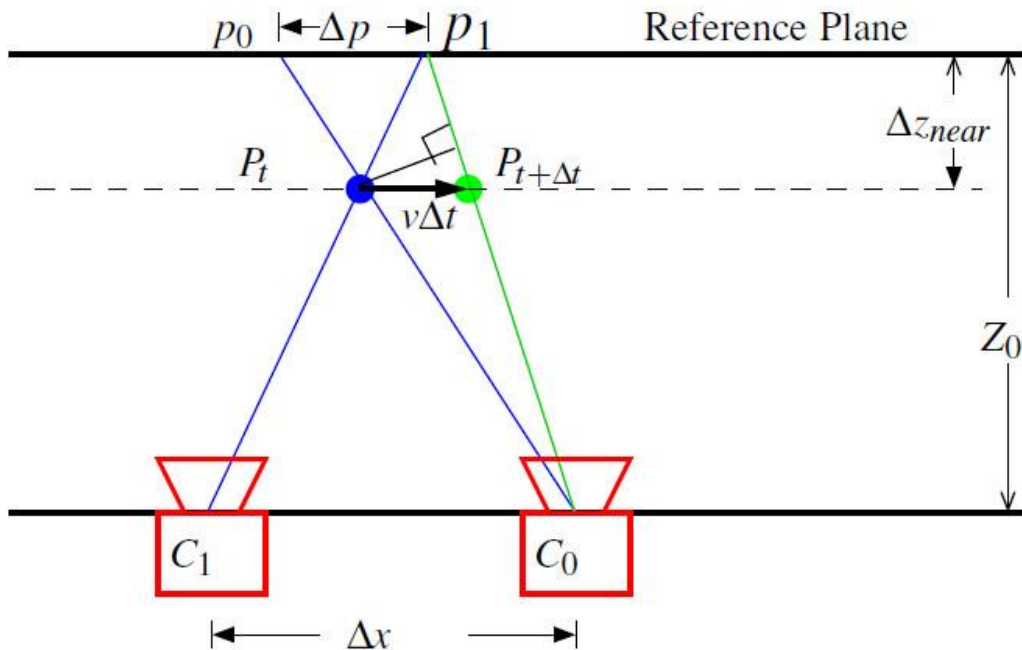


图 8: 与图像运动相关的时空轴。对于一个给定的场景配置, 我们可以确定一个时间步长 Δt , 时间样本之间的最大图像运动等于空间相邻视图的最大视差。如果我们按照步长 Δt 测量时间, 按照相机间距测量空间, 那么视图坐标的曼哈顿距离对应于视图中间的最大的可能的图像运动。

最坏情况下的视差出现在远平面和近平面。如果 P 在近平面的场景中以最大速度 v 运动, 最坏情况下的时间运动也会出现, 从而向量 $P_t P_{t+1}$ 正交于 $t+1$ 时刻从 C_0 发出的投影射线。如果我们从镜头中看一个狭窄的区域, 我们可以使用一个平行于焦平面的向量来近似表示, 用 $v\Delta t$ 表示, 如果 P 点的速度是 v , 在 C_0 点图像中最大的时间运动是 $v\Delta t Z_0 / (Z_0 + \Delta z_{near})$, 在 P 的邻近相机做上述同样的计算。

$$\Delta t = \frac{\Delta x \Delta z_{near}}{v Z_0}$$

(1)

这是在同一相机视图之间的最大图像运动的时间步长, 等于相邻视图之间的最大视差。如果我们用两个空间 (X ; Y) 坐标和一个时间坐标 t 表示一个视图, 测量相机之间的时间步进 Δt , 提供了一个标准化的轴集合来关联时空视图。由于视差和瞬时运动不是正交的, 所以测量的真正距离是欧式空间距离加上时间距离。最小化视图的距离测量减少了最大的图像运动。

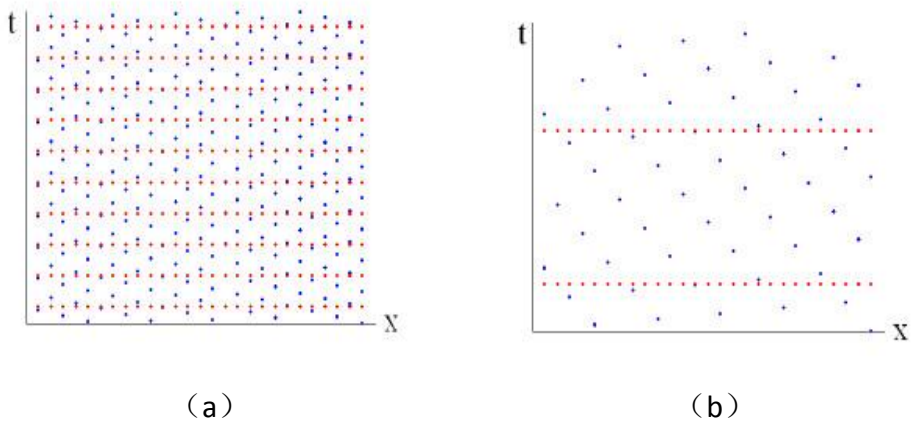


图 9: 途中显示的是 (x, t) 的视图坐标, 用于不同数量的物体运动, 以及不同的采样模式。这两个图都显示了一个统一的时间采样, 用红色表示, 还显示了最佳分布的采样, 用蓝色表示。(a) 对于大相机间距或缓慢运动的场景, 相机的时间移动差别不大。(b) 对于小相机间距或高速度的场景, 均匀采样创建了紧密行采样, 留下了大量未采样的区域。一个优化的采样模式是从 $Q * i \bmod N$ 点开始对应相机采样, 其中 i 是相机的下标, N 是相机的数量, 在那里我是相机的指数, 是相机的数量, Q 同样选取 N 的 $1/3$ 大小左右的最近的素数。

5.2 多基线时空光流

更好的时空采样，甚至可以提高最简单的视图插值算法，如混合，但是无重影图像混合时所需要的采样密度是望而却步。相反，我们创建了一个新的光流变体，用于从一个平面视频相机阵列产生新的视图。我们改进的时空光流算法有 2 个新的特点。光流通常计算反复弯曲的两张图片之间的流。我们的第一次修改是为了解决我们所需虚拟视图的 (X; Y; T) 位置处的流场。我们从 Kang 等人 2003 年的观点中收到启发开始计算新图像中像素的流。我们扩展的方法是在我们的标准的 (x;y;t) 视图空间中所需的视图计算流。我们修改了 Black 和 Anandan 于 1993 年提出的具有鲁棒性的光流统计算法，代码可以从作者的网站获取。我们反复地变换最近的四张图像到虚拟视图中，并减少成对鲁棒数据和平滑误差项的加权和。

只使用水平和垂直图像流的不同时空位置出的四张图像的运动并不能一致建模。我们算法的第二部分分别解释了视差和时间移动。光流场的标准强度恒定方程：

$$I(i, j, t) = I(i + u\Delta t, j + v\Delta t, t + \Delta t)$$

(2)

在这里 (i; j; t) 分别表示像素图像的坐标和时间，u 和 v 是一个图像点水平和垂直的运动。我们修改的强度恒定方程在虚拟视图和附近捕获的图像中间的原图像空间的一些偏移 (Δx ; Δy ; Δt) 表现出了稳定性。

$$I_{virtual}(i, j, x, y, t) = I_{source}(i + u\Delta t + d\Delta x, j + v\Delta t + d\Delta y, t + \Delta t)$$

(3)

流组件被分离到视差运动中，由四个因素决定，分别是：一个点的相对深度 d，视图间的空间距离，时间运动，视图和时间运动在图像平面的投影 (u; v) 之间的时间乘积。

对于每个虚拟视图，通过计算相机采样点的三维 Delaunay 三角剖分，还有从包围所需的 (x; y; t) 视图的四面体中选择视图，我们选择流算法的输入图像。这些图像在每次迭代中的算法的共同的虚拟视图中逐步扭曲。我们不可能为每一个扭曲图像测试强度恒定方程。相反，我们使用成对的鲁棒强度恒定错误评估的总和来最小化四个扭曲图像自身的误差。这产生了一个单一的流图，它可用于扭曲四个原图像到虚拟视图中。我们目前没有理由使用四面体的质心权重进行简单的遮挡和简单的融合扭曲的图像。

结果。对于我们的实验，我们配置了一个 12x8 的相机阵列，相机间距是 3 英寸。我们实验后确定了 30Hz 的帧时间的 9 个错开对于我们的场景就很充足了。所以我们创建了一个 3x3 的局部统一的，阵列之间可复制的网格布局的触发器。因为我们的应用是和相邻图像比较，所以局部均匀采样就足够了。我们校准我们的相机，以确定在相机平面上的相对位移，使用的是 Vaish 等人与 2004 年提出的平面加视差框架。

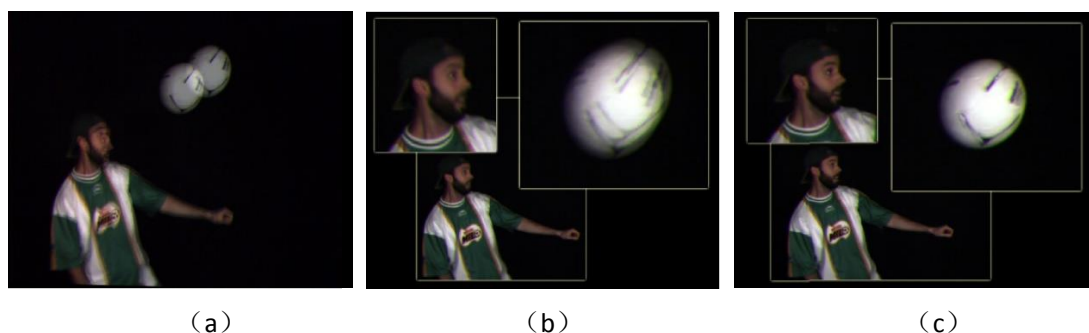


图 10: 更好的时空采样改善了视图插值。(a) 从 30Hz 相机获取的两帧之间的简单交叉溶解。(b) 在空间和时间上使用交错的触发器来同步相机会增加我们视图采样的密度。这个图中是由四张输入图像进行加权平均得到的，图中有更少的鬼影。(c) 更好的时空采样减少了图想间的运动，使光流更健壮。在这里，我们使用多基线时空光流算法扭曲四个原图像到所需的虚拟视图中。扭曲的图像按和中心图像同样的权重来混合。没有双图像显示出来，因为时差和球的运动都被正确地恢复了。

图 10 显示了改进的时空采样和视图插值算法的结果。作为参考，我们显示了一个相机的两帧之间的一个交叉溶解的过程，来说明帧之间的时间运动。交叉溶解或者混合，是同步的相机阵列的一种最简单的插值方法，用来同步触发。在时空中，错开相机的触发时间来均匀采样甚至可以改进最简单的插值方法。图 10 (b) 显示了一个来自同一个相机阵列的四个视图错开触发时间后进行的加权混合。鬼影显著的减少了。最后，右边的图像显示了我们的多基线时空光流算法的结果。因为四个视图计算的流是一样的，当原图像被扭曲和混合时，球就显得尖锐了。

讨论。我们使用改进的采样方法来创建一个相对简单的插值方法，这个方法使用光流来解释视差运动和场景中真是物体的运动。这种方法是能够估算任何相机的图像，这些图像都在原始相机区域的时空扩展范围之内。如果我们在每一个触发时刻宝石虚拟视点稳定和合成新的视图，那么我们能产生一个一配准的高速视频。但是，由于时间扩张和虚拟相机运动，我们可以自由的改变虚拟视图的位置和位置（在数组的范围之内）。

虽然我们的时空光流算法在实践中效果很好，但是它偶尔会受到光流的普通工件影响，例如当图像运动很大时，大型运动就会掩盖小区域的运动和问题。因此，随着相机间距的增加，将需要更复杂的方法进行插值。许多用于同步相机的方法应该会受益于相机的最佳采样定时。例如，基于分割的立体方法最近已被证明非常有用，比如 2004 年 Zitnick 等人提出的空间视图插值，2001 年 Tao, Zhang 和 Kambhamettu 等人提出的动态场景中的结构和运动分析。因为这些方法匹配的是视图中的小图像区域，所以有人会期望他们能够从时空相邻的视图的图像运动减少中获益。

高分辨率视频捕捉应用将拼接图像的分辨率分成四份，来增加动态范围。相比之下，错开相机的触发增加了时间采样分辨率，而且几乎不需要成本。因此，我们相信视频相机阵列中错开定时总是有益的。如果场景速度比较小，时间偏移是无关紧要的。如果速度比较大，错开的相机能捕捉到一些会被忽略的事情，能最大限度的插值工件。

6 合成孔径成像

时空视图插值模拟了时空中一个狭窄的运动的孔径。如果不进行视图插值，我们将穿过孔径到平面上的图像对齐，并且平均一下之后我们就可以大致模拟了一个有大孔径的相机。移动对齐的图像来改变焦距。【Levoy 和 Hanrahan 1996; Isaksen 等人 2000; Vaish 等人 2004】。除了移动外还进行扭曲可以让焦平面倾斜。【Vaish 等人 2005】。在这些实验中，在视频压缩并被传送至计算机之前，我们在每个相机的对齐和移动视频的过程中加入了 FPGA 来加快计算。这给我们提供了一个实时合成的孔径相机系统。具体而言，当用户交互的调整物体焦距时，计算机可以以广播的方法传送需要进行的图像变化给各个相机。目前，我们的计算机的处理能力限制了我们只能在一台计算机上使用 15 个相机。

传统相机的孔径在时空上是一个圆柱体。符合曝光时间和交叉部分的高度处的形状和镜头孔径一样。合成的孔径摄像增加了孔径的范围。我们现在考虑两个由我们的阵列构成的独特的光圈形状。第一个，无光泽的合成孔径成像，调整孔径只能捕捉透过部分遮光器的光纤。第二个，创建了一个混合的时空孔径，在低光照条件下高深度和慢运动的图像会变得模糊。

6.1 非线性合成孔径成像

合成的孔径相机效果，通过模糊遮光板我们可以看到隐藏在部分遮光板后面的主体。然后，遮光板并不是可见的，合成孔径摄影会削弱感兴趣信号，即主体。假设有测量噪声的相近查看场景。要创建合成孔径图像，我们将所有相机的视图对齐到一个平面，然后平均一下。如果只有 K 个相机能够通过遮光板看到主体上的任意给定的点，那么合成孔径图像中的信号会被衰减因子 K/N 削弱，而测量噪声下降到 $1/\sqrt{N}$ ，因此，相对于单个图像的信噪比，信噪比已经下降了至少 K/\sqrt{N} 。因为遮光板并不完全均匀，它会增加一个额外的噪声成分。

我们知道，如果每个相机都能透过遮光板看到物体，那我们可以均衡那些未被遮挡的像素的影响。只是均衡 K 个未被遮挡的像素会单个图像的 SNR 增加 \sqrt{K} 倍，并且削弱信号时不会降低图像的对比度。在实践中，许多像素都是混合像素，包括前景和背景信息，因此 SNR 的增益会小于 \sqrt{K} 。

要实现这一点，我们为每个相机创建一个二进制的遮罩图像。遮罩是不会被遮光板阻挡的像素。尽管二进制遮罩图像会丢失信息，为了使用分数，即 α 值，我们还必须回复前景色。二进制遮罩是一个强大的，传统的解决方案。为了创造遮罩的合成孔径图像，我们按照每个像素上均衡的遮罩来分配均衡的遮罩的输入图像的数量。

有几种方式可以创造遮光板遮罩图像。我们已经实现了在遮光板和其后的物体之间识别所有像素的遮光板运动和相互反射的运动。这些像素能捕捉到一些随时间变化的部分物体，

因此不遮挡。我们根据输入视频的每一秒和阈值来计算每个像素的方差，从而识别这些像素。



(a)

(b)

(c)

图 11：遮罩的合成孔径成像。(a) 用于本实验的 90 个相机中的一个产生的样本图像。(b) 合成孔径成像聚焦于人物所在的平面，这个平面通过对齐均衡文中描述的 90 个相机得来。

(c) 抑制每个相机的静态像素的影响产生了遮光板背后的生动的图像。人和毛绒玩具更加清晰可见。

结果。图 11 显示了我们使用遮罩的合成孔径方法透过树叶拍摄任务的结果。通过改变孔径的形状来减少遮光板的影响。遮罩的合成孔径产生了隐藏物体的一个更加精确的图像。将像素混合来防止遮光板被完全消除，没有光线穿过的空间都是黑色的。我们将使用图像在时间上的方差产生的图像和我们通过遮光板后面的黑白背景成像创建的一个地面上真实的遮罩图像进行比较。我们发现了两种这招的微小区别。

讨论。正如我们在这节看到的，定制有助于合成孔径图像的光纤可以显著的改善效果。基于每个像素时间方差计算的遮罩适用于静态的遮光板。使用其他的遮罩技术处理运动的遮光板，这一技术我们也感兴趣。包括基于颜色阈值的均衡遮光板的遮罩在内的一些可能，和立体，焦点，主动测距三种方法都不同。

到目前为止，我们展示了如何形成空间中的孔径，但是毫无疑问，我们不能同时在时间和空间上形成孔径。例如，如果我们估计部分遮光板物体的运动，我们可以形成一个空间-时间合成孔径，遵循物体路径。这个孔径会产生更好的图像，在一些视图中的信息也可以添加到缺失它的视图中。这部分显示了为了特定问题定制孔径的方法。在下一个部分，我们将这个想法扩展到在时间和空间上形成孔径。

6.2 混合孔径成像

传统相机有两种收集光线的手段：增加曝光时间和增大镜头孔径直径。两者都有副作用。增加曝光时间会增大快速移动物体的移动模糊程度，并增加孔径直径会导致场景深度更小。因此，为了拍摄广泛的深度领域中的快速移动的对象，静止或缓慢移动的场景，更喜欢使用小孔径和短曝光时间。如果场景没有明亮的照明，可能会导致黑暗，嘈杂的图像。

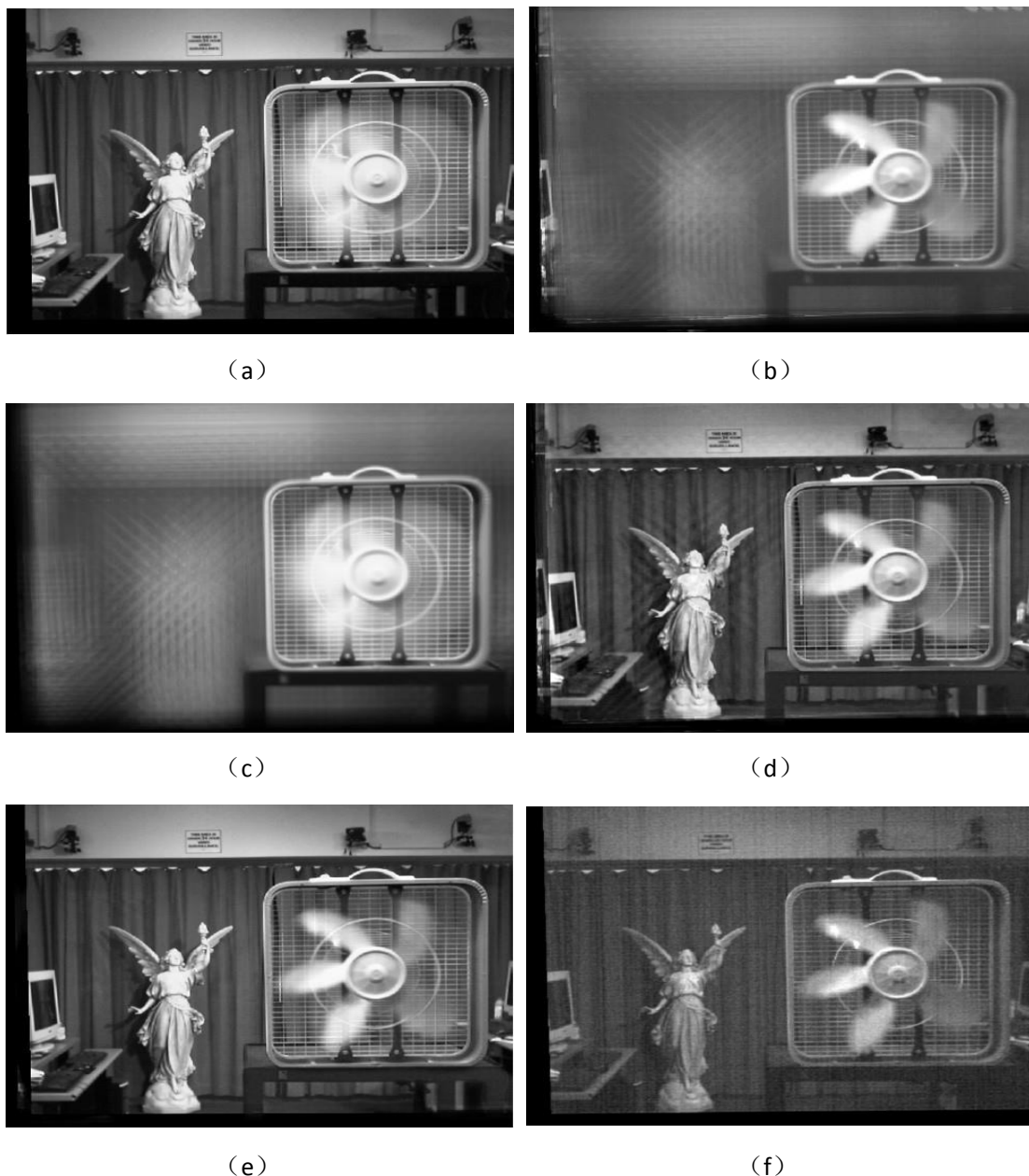


图 12: 混合合成孔径摄影技术, 用于高深度区域和低运动模糊。a 图像到 c 图像是通过三个不同的孔径同步拍摄的: 一个具有长曝光时间的单个相机 (a), 一个具有短曝光时间的大的合成孔径 (b), 一个具有长曝光时间的大的合成孔径 (c)。图像 a、图像 b、图像 c 进行 $a+b-c$ 的计算产生的图像 d, 其中已经出现物品混叠, 因为合成孔径是从稍微不同的位置稀疏采样的。在计算 $a+b-c$ 的差别之前, 掩蔽合成孔径图像中不在焦点处的像素可以消除混叠, 如图 e。相比之下, 图 f 是由一个空间和时间上都比较窄的孔径采集的。整个场景都被聚焦了, 风扇运动也被冻结了, 但是图像噪音很大。

正如前面所指出的, 我们的阵列不限于圆柱形的空间-时间孔径函数。我们可以将我们的阵列划分成子阵列, 从而使用多个不同孔径同时捕捉图像的场景。结合不同孔径捕捉到的图像, 我们有效地创建一个混合孔径, 使我们能够正确地拍摄这些场景。图 12 作为这个想法的一个例子, 我们考虑在一个深的房间里拍摄一个旋转的风扇的问题。要创建一个专门适

用这种场景的混合孔径，我们要通过以下三个孔径同时成像：

- 空间窄时间宽的孔径 s 。整个场景都会被聚焦，但是快速移动的物体还是会产生运动模糊。通过一个带有小孔径镜头和长曝光时间的单一相机产生的图像实现了这个孔径。
- 空间宽时间窄的孔径 t 。聚焦在物体上。时间窄的孔径消除了运动模糊，但是大空间孔径意味着一切不在物体深度处的都不会聚焦。我们使用由短曝光时间的相机组成的一个合成孔径捕捉到这些。
- 空间宽时间宽的孔径 w 。由于有限的景深和物体的运动模糊，这个图像会有两方面的散焦模糊。我们是通过由长曝光时间相机组成的交错阵列合成孔径获得了这个的。

图 12 显示了由孔径 s 、 t 和 w 捕捉的图像 I_s 、 I_t 和 I_w 。和带有时空都小的孔径相比这里的每个孔径都能收集更多的光。 I_s 对于快速运动的物体存在运动模糊，对于和物体不在同一深度的一切都有散焦模糊， I_w 中这两种都有。因为 I_w 聚焦在物体上，物体的运动模糊也被聚焦了，和 I_s 中的运动模糊是一模一样的。同样，由于两合成孔径的照片都集中在同一深度，对其余场景的散焦模糊在两幅图像中是等效的。因此，在规定每张图像的曝光之后，我们可以通过 $I_s + I_t - I_w$ 来计算出我们所需的图像，如图 12 (d) 所示。

合成孔径图像显示混叠失真，因为我们空间宽孔径的点采样。为了同时捕捉到两幅图像，我们分配了阵列中一半的相机到一个合成孔径中，剩下的分配到另一个合成孔径中，预留一个相机用做空间窄的孔径。我们把合成孔径相机的两个集合按照棋盘模式交错分布在我们的平面阵列中。但视图之间的轻微位移引起的合成孔径的混叠的轻微改变。从 I_t 中减去 I_w 之后混叠的差异还是会存在，还会造成伪影。

混叠只出现在 I_t 和 I_w 的散焦区域。再最终图像，我们希望散焦模糊可以取消。如果我们知道 I_t 和 I_w 中混叠出现的位置，那么就可以在构成最终图像之前，遮罩散焦区域。我们可以场景的一个深度图中构建这样一个遮罩。

要重建一个没有混叠的合成孔径图像，我们首先要把一个合适的重建过滤器用到孔径 w 的采样中。过滤器去除带有混叠部分的场景中的高频分量。通过计算每个像素的合成孔径采样的方差来过滤得到剩下的特征，对这些特征进行深度评估。假设场景中存在纹理物体，那么不在焦深处的物体的方差会比较高。通过对方差设置阈值来得到一个遮罩。在实践中，许多物体没有高频率的纹理，但是低频纹理没有创建混叠，所以对于我们的目标来说是足够鲁棒的。

图 12 (e) 是在计算 $I_s + I_t - I_w$ 之前 I_s 和 I_t 的遮罩结果。混叠伪影不见了，我们实现了场景深度的高深度和风扇的低速运动模糊。最后一张图片 12 (f) 是空间窄时间窄（例如，一个摄像头具有短曝光时间）的孔径拍摄的。风扇的运动冻结了，雕像聚焦了，但是结果比混合孔径图像的噪音更多。

把我们的方法和 Stewart 等人于 2003 年的方法相比较是很有趣的，Stewart 等人提出了一种光场渲染的混合重建滤波器用于减少鬼影。这个滤波器将捕捉物体细节的空间宽的孔径

和捕捉场景深度和相关反射率的空间窄的孔径结合起来。和他们一样，我们也使用混合重建过滤器，即由几个不同形状的过滤器组成。此外，包含对角线滤波器的两种混合在一个固定的深度。然而两种方法在很多方面也有区别。因为我们考虑到动态场景，我们的混合包括时间，而他们的方法并没有。因此，我们必须考虑信号-噪声问题，这些并不会出现在他们的静态场景中。其次，但更重要的是，Stewart 等人将两个过滤器应用到同一个光场。相反，我们对光场多次采样，每一个过滤器都有不同的采样策略。最后，他们的混合过滤器是线性的，而我们的是非线性的，这是由前面所述的合成步骤导致的。

7 讨论和总结

我们在 1999 年创建了一个系统，这允许我们使用大量廉价相机进行成像实验。由此产生的阵列相机，虽然并不完美，但是还是完成了目标。其主要设计了功能小的相机瓷片，可以灵活安装，成像设备定时控制精确，每个成像设备可以本地处理和压缩，可以完成各种成像任务。高采样密度可以让相机接近非凡的功能，即使我们使用的是廉价的成像设备。这个观点导致了高分辨率的视频拼接，重叠视图可以用来提高有效的动态范围或者帧率，相机可以被分配到同时完成这三个场景。

尽管我们已经提出的许多技术可以应用到高品质的相机中，易扩展他们的性能，甚至，我们特别感兴趣的是，探索使用大型便宜相机阵列成像的限制。一个公开的问题是，是否使用多个摄像头和智能的处理方法，我们可以克服较差的成像特性，超过单个高品质相机。例如，我们的高分辨率视频采集系统的分辨率会随着相机的数量线性增加，但是高分辨率图像传感器的制造产量随着像素分辨率的增加呈指数下降，因此阵列的方法似乎比较优越。另一方面，我们的系统将会天文学制造一个性能低下的相机，它要求非常低的噪声，因为噪声会随着相机的数量增加呈对数下降。这些理由都表明通常情况下，高品质相机比较优越，但是相机阵列在某些情况下可以表现地更好。

除了提高成像性能，我们的系统也可以用来创建一些普通相机无法捕捉的图像。一些应用使用张开更广的相机阵列来创建宽的合成孔径。这个更宽广基线的一个关键问题是如何沿两个空间维度和一个时间维度来分配摄像头。我们表明，对于有紧密空间相机或者快速运动的场景，同时触发所有的相机是很差劲的采样策略。相反，在帧时间内分散相机的触发时间，可以更均匀的采样 $(x; y; t)$ 视图体。我们利用所产生的减少的图像运动的光流变体，可以解释视差运动和物体运动。这让我们可以插值出时空体中丢失的点，从时间和空间中的新位置创建虚拟相机视图。这些技术可以用于在后处理中创建矩阵式子弹时间效果。

基于我们对非线性合成孔径和混合合成孔径的经验，我们相信大型相机阵列最有趣的应用就是不尝试近似传统相机。特别是，我们已经表明，通过形成合成孔径以避免没有照射在所需物体上的光线，或者通过在空间-时间创建非圆柱形的孔径，相机阵列允许创建一个之前不可能看到的图像。我们已经探讨了可能的应用的一小部分，每一个都提出了问题，这些

问题也就提出了新的机会。

展望未来，我们希望设计下一代的相机阵列。我们的系统的一个简单的改进就是增加对相机的处理。我们几乎是在 FPGA 容量做相对简单的图像处理任务操作。在未来的设计中，我们也不会使用带电子滚动快门的成像设备。滚动快门相当于扫描图像的机械狭缝快门，使低端的一排在升到顶端时会曝光。这种采样模式对许多应用都不方便。

对于实时应用，未来的系统应该支持更灵活的通信，以及提高处理能力。目前，我们相机的所有视频都是直接流到主机电脑的。实时合成孔径的视频，我们只演示了一个适度的摄像头数量，如果每个摄像头可以减少接受到的视频，并且在传输到上游相机之前从下游相机中增加图像到自己的扭曲输入中，将会更容易。我们可以在当前架构中加入这个功能，我们在每个相机中使用多个 IEEE1394 接口来支持设备之间的点对点通信，但是其他的应用可能有更复杂的通信需求。因此在设计一个新的架构之前，我们应该调查潜在的实时阵列应用以及他们如何映射到只能相机阵列。

最后，许多应用都将受益于系统中混合的有效技术。例如，我们设想使用范围传感器或者投影仪用于合成孔径成像中有效的遮罩技术。在非常非结构化的环境中校准可以用激光来辅助，可以投影几何校准目标到我们的场景中。投影仪是特别有吸引力的，因为目前 CMOS 传感器价格下滑，投影仪的价格也随着下滑。许多大阵列投影仪的面临的挑战和相机遇到的是相同的：带宽，控制和灵活性。随着类似投影仪和范围传感器的技术变得更加经济实惠，我们可以到预见创建大的混合的相机阵列。不仅能被动地观察，而且还能积极和他们的环境互动。

8 致谢

作为本文的作者，我要非常感谢 Harry Shum 在项目初期提供的非常宝贵的建议。也非常感谢 Michal Smulski, Hsiao-Heng Kelin Lee, Monica Goyal, Katherine Chou, Guillaume Poncin, Georg Petschnigg 和 Benjamin Jasen Levoy 他们 7 人对于相机阵列基础设备的贡献。相机阵列的构建得到了 Intel, Sony 和 Interval Research 的赞助。本文所记录的应用作品 IIS-0219856-001 合同下的 NSF 和 NBCH-1030009 合同下的 DARPA 支持。

参考

ANDERSON, D. 1999. *FireWire System Architecture, Second Edition*. Mindshare, Inc.

- BAYER, B., 1976. Color imaging array. U.S. Patent 3,971,065.
- BLACK, M., AND ANANDAN, P. 1993. A framework for the robust estimation of optical flow. In *Proc. ICCV 1993*, 231-236.
- BROWN, M., AND LOWE, D. 2003. Recognizing panoramas. In *Proc. ICCV*, 1218-1225.
- CHAI, J.-X., TONG, X., CHAN, S.-C., AND SHUM, H.-Y. 2000. Plenoptic sampling. In *Proc. SIGGRAPH 2000*, 307-318.
- DEBEVEC, P. E., AND MALIK, J. 1997. Recovering high dynamic range radiance maps from photographs. In *Proc. SIGGRAPH 1997*, 369-378.
- GORTLER, S., GRZESZCZUK, R., SZELISKI, R., AND COHEN, M. 1996. The lumigraph. In *Proc. SIGGRAPH 1996*, 43-54.
- ISAKSEN, A., MCMILLAN, L., AND GORTLER, S. 2000. Dynamically reparametrized light fields. In *Proc. SIGGRAPH 2000*, 297-306.
- KANG, S., UYTENDAELE, M., WINDER, S., AND SZELISKI, R. 2003. High dynamic range video. In *Proc. SIGGRAPH 2003*, 319-325.
- LEVOY, M., AND HANRAHAN, P. 1996. Light field rendering. In *Proc. SIGGRAPH 1996*, 31-42.
- MATUSIK, W., BUEHLER, C., RASKAR, R., GORTLER, S., AND MCMILLAN, L. 2000. Image-based visual hulls. In *Proc. SIGGRAPH 2000*, 369-374.
- RANDER, P., NARAYANAN, P., AND KANADE, T. 1997. Virtualized reality: Constructing time-varying virtual worlds from real events. In *Proceedings of IEEE Visualization*, 277-283.
- SCHECHNER, Y., AND NAYAR, S. 2001. Generalized mosaicing. In *Proc. ICCV 2001*, 17-24.
- SHECHTMAN, E., CASPI, Y., AND IRANI, M. 2002. Increasing space-time resolution in video sequences. In *Proc. ECCV 2002*, 753-768.
- S.MANN, AND R.W.PICARD. 1994. Being 'undigital' with digital cameras: Extending dynamic range by combining differently exposed pictures. Tech. Rep. 323, M.I.T. Media Lab Perceptual Computing Section, Boston, Massachusetts. Also appears, IS&T's 48th annual conference, Cambridge, Massachusetts, May 1995.
- STEWART, J., YU, J., GORTLER, S., AND MCMILLAN, L. 2003. A new reconstruction

- filter for undersampled light fields. In *Eurographics Symposium on Rendering (EGSR)*, 150–156.
- SZELISKI, R. 1994. Image mosaicing for tele-reality applications. In *WACV 1994*, 44–53.
- TAO, H., SAWHNEY, H., AND KUMAR, R. 2001. A global matching framework for stereo computation. In *Proc. ICCV 2001*, 532–539.
- TAYLOR, D. 1996. Virtual camera movement: The way of the future? *American Cinematographer* 77, 9 (September), 93–100.
- VAISH, V., WILBURN, B., JOSHI, N., AND LEVOY, M. 2004. Using plane + parallax for calibrating dense camera arrays. In *Proc. CVPR 2004*, 2–9.
- VAISH, V., GARG, G., TALVALA, E., ANTUNEZ, E., WILBURN, B., HOROWITZ, M., AND LEVOY, M. 2005. Synthetic aperture focusing using a shear-warp factorization of the viewing transform. In *Proc. A3DISS 2005*.
- WILBURN, B., SMULSKI, M., LEE, H., AND HOROWITZ, M. 2002. The light field video camera. In *Media Processors 2002*, vol. 4674 of *Proc. SPIE*, 29–36.
- WILBURN, B., JOSHI, N., VAISH, V., LEVOY, M., AND HOROWITZ, M. 2004. High speed video using a dense array of cameras. In *Proc. CVPR 2004*, 294–301.
- YANG, J., EVERETT, M., BUEHLER, C., AND MCMILLAN, L. 2002. A real-time distributed light field camera. In *Eurographics Workshop on Rendering*, 1–10.
- ZHANG, C., AND CHEN, T. 2004. A self-reconfigurable camera array. In *Eurographics Symposium on Rendering*, 243–254.
- ZHANG, C., AND CHEN, T. 2004. View-dependent non-uniform sampling for image-based rendering. In *Proc. ICIP 2004*, 2471–2474.
- ZHANG, Y., AND KAMBHAMETTU, C. 2001. On 3d scene flow and structure estimation. In *Proc. CVPR 2001*, 778–785.
- ZITNICK, C., KANG, S., UYTENDAELE, M., WINDER, S., AND SZELISKI, R. 2004. High-quality video view interpolation using a layered representation. In *Proc. SIGGRAPH 2004*, 600–608.