★CACM 中国版★

计算机协会通讯

CACM.ACM.ORG

01/2014 第57卷 第01期

外科手术中的非接触式交互

先发表后评判

Unikernels

语音识别

硅通孔的硅应力

Association for Computing Machinery

ACM 通讯

精选文章中文译文

ACM中国理事会

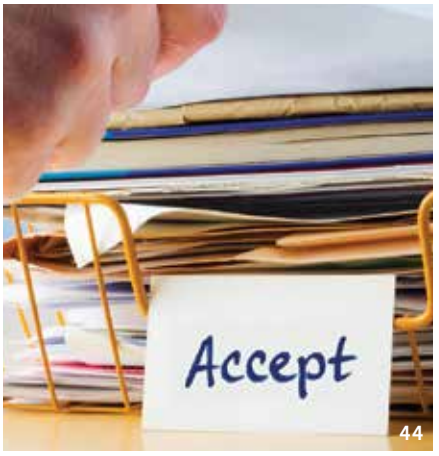
中国北京清华大学
东主楼 11-236 室
邮编：100084
电话：+86-10-62785025
电子邮件：acmchina@acm.org
联系人：辛爽

ACM通讯

(ISSN 0001-0782) 由计算机协会（2 Penn Plaza, Suite 701, New York, NY 10121-0701）按月发行。

Association for Computing Machinery

观点



实践

投稿文章

评论文章

研究亮点

关于封面：
尽管医疗成像技术比比皆是，但是医生仍然无法在不损及手术室无菌状态的情况下充分地与可视影像交互。本月的封面故事（第 70 页）探讨了让外科医生在无需接触的情况下即可控制和操纵医学图像的最新技术。封面插图照片：Kollected。

从左至右：图片由 SHUTTERSTOCK.COM CENTER 提供；插图由 KOLLECTED 提供

Association for Computing Machinery
Advancing Computing as a Science & Profession

DOI:10.1145/2542506

作者：Doug Terry

观点
先发表，后评判

本提案旨在解决会议投稿论文数量过多而审稿人没有足够时间仔细评估每一篇论文的问题。

计算领域的会议存在投稿论文数量庞大、审稿人工作过于繁重、过于挑剔以及录用率低的普遍问题。会议宣扬其低录用率，仿佛这是评估会议质量的主要指标。由于录用的论文数量有严格限制，因此会议程序委员会将面对选择顶尖论文这一艰巨的任务，即使是最好的委员会也会拒绝一些学术界可以从中受益的论文。被拒绝的论文会许多次地重新投稿给不同的会议，直至这些论文最终被录用或者作者沮丧地放弃。好的想法得不到发表或者其发表被滞后，这对于学术界很不利。较差的论文获得的关注很少，也得不到必要的建设性反馈意见，以便改善论文或研究工作。

由于审稿人在着手处理其工作时知道他们最终必须拒绝五分之四（甚至更多）的投稿论文，因此他们往往会将注意力集中于寻找拒绝论文的理由。一旦他们找到这样的理由，无论正确与否，他们便会较少着眼于论文的其余部分。他们不会充分考虑瑕疵是否可以通过适度的修订予以纠正，或者优点是否超过缺点。有可能产生长期影响的论文被拒绝，而结论易于评估、很难反驳的论文得以录用。程序委员会花费大量时间试图就评选出最好的20%的投稿论文达成一致，而不是为了大家的利益提供改善论文的意见。即使委员会能够完美地按照质量对投稿论文排序，实际上他们并不能做到，质量接近的论文也可能会因必须要在某处划分界线来决定录取而得到不同的结果。如果发明新技术的人的投稿论文被拒绝，而后来的一些他人的相关工作得以先发表，他们并不总是能得到应有的认可。

建议方案

我提议的解决方案很简单。会议应录用并发表所有质量尚好的投稿论文。据我所知，某些领域，如物理，在举办大型年度会议时，任何人都可以谈论几乎任何东西。我并不是建议我们的会议录用每一篇投稿论文。我认为计算领域的会议应执行一定程度的论文发表质量标准，但我们现行的标准实在过于严格了。我们可能会争论怎样才算是质量尚好的发表论文。但是请记住论文发表的主要

目的是教授他人，因此我的建议如下。

如果投稿论文包含新的内容（全新的想法、新的实验结果、对以前结果的验证、解释事物的新方式等等）、基于完善的方法体系、以足够清晰的方式解释了其新颖点以供他人从中学习并且将新的结果置于恰当的关联上下文中（即适度地将结果与以前的工作进行对比），则认为投稿论文“质量尚可”，因此可以发表。与其寻找拒绝论文的理由或者花费时间比较论文，会议审稿人的角色现在将是 (1) 评估每篇投稿论文按照此标准是否质量尚可；也许更为重要的是，(2) 提供具体的改进建议。任何符合此标准的论文都应录用发表，也许再加以指导以确保审稿人的建议得以正确遵从。

最终，论文将根据大家公认的文献计量数据（如被引用次数），以及更为重要的，根据其对领域和对行业的影响，由时间来公平地评判。已发表论文的重要性往往要在许多年之后才为人所知。应使用“10 年之后”或者“名人堂”之类的奖项作为赞誉最佳论文的方式。这些奖项应在ACM数字图书馆中注明。搜索引擎，以及协同筛选和公开推荐，可以将研究人员指引到高质量、相关度高的工作。

实际问题

如果录用的论文数量超出了在会议期间所能安排演讲的数量，该怎么办？在稳定的情况下，这不会是严重的问题，因为有大量的会议，而新论文并没有那么多。如果论文不再反复投稿给众多的会议，然后被拒绝，最终的投稿论文数量将会少得多。为了应对大量的论文，会议可能需要设立并行会场或缩短演讲时间，或者两者兼而有之。就个人而言，我更喜欢简短的演讲。作者应能够在 10 至 15 分钟的时间内呈现其工作背后的主要想法，然后让大家阅读论文了解更多细节。一些论文可能仅以海报方式呈现，但我个人不喜欢这种方式。我更希望看到所有录用的论文得到平等对待。让学术界来对论文进行评判。

作者如何决定向哪里提交其论文？会议仍然会有焦点主题。例如，我们仍然会有关于数据库、算法、系统、网络等的会议。一个额外的录用标准是论文是否符合会议的主题范围。一些论文可能符合多个会议的主题。例如，关于分布式存储系统的论文既是数据库论文，又是系统论文，即同时适合在 SIGMOD 和 SOSP 上讲演。在这种情况下，由于所有会议录用论文的标准都相同，因此将论文投稿给哪个会议并没有多大区别。在任一情况下，假定是 ACM 会议，论文最终都会收录在数字图书馆中。最可能的情况是，作者会将其论文投稿给吸引与其关系最紧密的社区的会议，例如其所属的特别兴趣组 (SIG) 主办的会议。低质量的会议将会逐渐消失，每个技术领域或者每个技术社区只留下一个顶尖的会议。对我来说，会议少一些是件好事情。

怎样防止大家投稿包含“最少可发表单元”的论文？当作者取得了他们希望与学术界分享的显著成果时，他们可以自行决定。让想法和成果得以快速发表是一件好事。没有理由要等到取得的成果足以写一整篇论文时才能发表其工作。论文的长度与其贡献相称。投稿大量贡献非常微小、内容简短的论文的人将要承担其声誉受损的风险，并且与提交更多重大成果的人相比，得到的“时间考验”奖项可能会更少。这可能足以劝阻那些太过于增量的投稿论文。

这对期刊会有怎样的影响？我觉得期刊投稿论文的数量将会增加，更多的重点还是放在期刊发表上。期刊将继续让权威的评审委员

活动日程

2 月 15 日至 19 日
计算机支持的协同工作
马里兰州巴尔的摩
主办方：SIGCHI
联系人：Susan R. Fussell
电子邮件：sfussell@cornell.edu
电话：607-255-1581

2 月 22 日至 26 日
ACM SIGPLAN 并行编程原理与实践会议
佛罗里达州奥兰多
主办方：SIGPLAN
联系人：Jose E. Moreira
电邮：jmoreira@us.ibm.com
电话：914-525-6267

2 月 23 日至 25 日
2014 年 ACM/SIGDA 现场可编程门阵列国际研讨会
加州蒙特雷
主办方：SIGDA
联系人：Vaugn Timothy Betz
电邮：vaughnbetz@gmail.com
电话：416-766-2197

2 月 24 日至 27 日
第 19 届智能用户界面国际会议

以色列海法
主办方：SIGART、SIGCHI
联系人：Tsvi Kuflik
电邮：tsvikak@is.haifa.ac.il

2 月 24 日至 28 日
第七届 ACM 网络搜索与数据挖掘国际会议
纽约州纽约市
主办方：SIGWEB、SIGIR、SIGKDD、SIGMOD
联系人：Ben Carterette
电邮：Carteret@cis.udel.edu
电话：302-31-3185

3 月 1 日至 2 日
第 10 届 ACM SIGPLAN/SIGOPS 虚拟执行环境国际会议

犹他州盐湖城
主办方：SIGPLAN、SIGOPS
联系人：Martin Johannes Hirzel
电邮：hirzel@gmail.com

3 月 1 日至 5 日
编程语言与操作系统架构支持
犹他州盐湖城
主 办 方：SIGPLAN、SIGOPS 和 SIGARCH，
联系人：RajeevBalasubramonian，
电邮：Rajeev@cs.utah.edu

会根据质量来接受和拒绝论文。因此，期刊发表将被视为比会议发表更有声望。包含早期成果、在会议上讲演的论文在今后有了更多实质性成果、细化想法或实际经验之后可能会成为期刊文章。来自多篇会议论文的成果可以合并成更为全面的期刊论文。这样可以使计算研究领域的论文发表实践更类似于其他的科学学科。

我当然不是第一个注意到我们目前的论文发表实践存在缺陷并建议改变的人。

其他提案

我当然不是第一个注意到我们目前的论文发表实践存在缺陷并建议改变的人。[5,6] 最近关于“计算研究领域论文发表文化”的 Dagstuhl 观点研讨会 (Dagstuhl Perspectives Workshop) 的与会者花了数天时间讨论可选的方案。正是该研讨会促成了本文的立场声明。也有人建议修改我们的论文发表流程，例如公开访问 [1] 和发表后同行评审 [3]，并且其中一些观点已经出现在《通讯》中。[2,4,7] 一些社区已部署了新的服务，例如 PubZone[a]，其旨在促进公开讨论数据库领域已发表的论文。这些做法和系统值得考虑，但与我提议的方案大体上是正交的。

业内已建立了公共网站，如计算研究知识库 (CoRR)，[b] 用以鼓励迅速传播新的想法。作者可以选择通过将其论文存放在此类知识库中来使其论文立即可供访问。这种方法解决了我提出的部分问题，但有三个根本性的区别。首先，作者体验不到在现场会议听众面前展示其工作的兴奋与经历。其次，存放的论文一般以后会投稿给更加权威的会议或期刊进行发表。因此，反复投稿及其对审稿人带来负担等问题仍然存在。第三，也是最为重要的，论文未经同行评审。我的提案保留了发表前的同行评审。因此，作者可以在论文发表之前获得建设性的反馈意见以便进行修改，从而获益良多；而读者知道该工作经过权威的程序委员会的审核，也会从中受益。

如何达成

实行新的论文发表政策并不简单。我不指望既有的会议一夜之间改变其做法。会议有着通过保持低录用率来维护其来之不易的声誉的既得利益。大学计算机科学系也已成功让晋升委员会重视会议论文，并且不愿意做出可能会损害其地位的改变。不过，我认为循序渐进的改变是可能的。有一个令人鼓舞的趋势，我知道一些最近的系统会议录用了比平常更多的论文，同时会议继续单轨举行。作为其中一次会议 (MobiSys 2012) 的程序委员会成员，我第一手观察到了让审稿人改变其思维方式以及录用甚至仅略多一点投稿论文的难度。

往前推进的一种方法是在现有的“低录用率”会议之外，举办新的“高录用率”会议。增加更多的会议并不是一个很好的长期解决方案，但可以敦促社区往正确的方向前行、提供实验数据并引发讨论。例如，去年 SIGOPS 在其备受推崇的操作系统原理研讨会 (SOSP) 之外还举办了一个新的会议——操作系统及时成果会议 (TRIOS)。这次实验性的会议录用了被 SOSP 拒绝但仍然有显著贡献的论文。从这次实验中学到的经验为 SIGOPS 社区对论文发表实践开展更广泛的讨论提供了素材。TRIOS 提供了相关的见解，例如社区是否会重视录用率限制较少的会议，以及作者是否会选择在此类会议上展示其工作，还是会等待在其简历上可能看起来更好的发表机会。

结论

我的主要提案是会议录用和发表任何为我们的知识库贡献新知识并以清晰和公正的方式表达其贡献的投稿论文。录用任何质量尚可的会议投稿论文并摒弃低录用率的好处很明显：

- 研究成果得以更及时地发表。
- 审稿人专注于提供建设性的反馈意见。
- 程序委员会不再浪费时间反复评审相同的投稿论文。
- 荣誉归于率先构想出某个想法的人或者同时形成类似想法的小组。
- 学术界通过论文的长期影响来评判论文。

然而，这确实需要研究界以及授职委员会和其他评审委员会在评估会议发表论文时作出根本性的转变。我认为某些形式的转变是必要的。 C

参考资料

1. Beaudouin-Lafon, M. Open access to scientific publications.*Commun.ACM 53*, 2 (Feb. 2012).
2. Meyer, B., Choppy, C., Staunstrup, J., and van Leeuwen, J. Research evaluation for computer science.*Commun. ACM 52*, 4 (Apr. 2009).
3. Neylon, C. Reforming peer review.What are the practical steps?” (Mar. 8, 2011); http://cameronneylon.net/blog/reforming-peer-review-what-are-the-practical-steps/.
4. Roman, D. Scholarly publishing model needs an update.*Commun.ACM 54* (Jan. 2011).
5. Rosenberger, J. Should computer scientists change how they publish?, *BLOG@CACM* (July 29, 2012).
6. Vardi, M.Y.Revisiting the publication culture in computing research.*Commun.ACM 53*, 3 (Mar. 2010).
7. Wallach, D.S.Rebooting the CS publication process. *Commun.ACM 54*, 10 (Oct. 2011).

Doug Terry (terry@microsoft.com) 是位于美国加州 Mountain View 的微软研究院硅谷实验室的首席研究员。

责任编委：谢涛

a PubZone科学论文发表论坛；http://pubzone.org/。

b CoRR：计算研究知识库；http://arxiv.org/corr/home。

DOI:10.1145/2541883.2541895

文章编写主导者：acmqueue
queue.acm.org

如果虚拟应用装置中所有软件层都在同一个安全的高级语言框架中编译会怎么样？

作者：ANIL MADHAVAPEDDY 与 DAVID J. SCOTT

Unikernels：虚拟库操作系统的崛起

对于将大型数据中心的计算资源租赁给多个（甚至可能具有竞争关系）租户的业务，云计算一直是这方面的先驱。云的支撑技术是操作系统虚拟化，如 Xen[1] 或 VMWare，其允许客户在物理机的共享集群上运行多个虚拟机 (VM)。每个虚拟机作为自包含的计算机运行：启动标准的操作系统内核，运行未修改的应用程序，而应用程序就如同在物理机上运行一样。

云计算最早的关键增长力来自*服务器整合*。现有应用程序通常安装在资源利用率低下的独立物理主机上，但虚拟化技术可将它们封装在更少的主机中，而不需要做任何修改，也不需要重新编译代码。虚拟机也通过软件 API（而不是物理接口操作）来管理。它们不仅可以集中备份，而且还可以在不中断服务的情况下在不同的物理主机之间迁移。如今，Amazon 和 Rackspace 等商业供应商都有庞大的数据中心，托管着数百万的虚拟机。这些云供应商为客户减轻了管理数据中心的负担，并实现规模经济，从而降低成本。

不可否认，操作系统虚拟化非常有用，但它在本来就高度层级化的软件堆栈上又增加了一层，现在这些层次包括：对传统物理协议的支持（例如 IDE 等 20 世纪 80 年代

制定的磁盘标准）；不相关的优化（例如 SSD 驱动器上的磁盘电梯算法）；向后兼容的接口（例如 Posix）；用户空间进程和线程（包括虚拟机管理程序上的虚拟机）；以及托管代码运行时（例如 OCaml、.NET 或 Java）。所有这些层都位于*应用程序代码*之下。难道我们注定每隔几年都要增加新的间接层和抽象层，使未来的程序员为了调试最简单的应用程序，都得像个考古学家一样去探究几百个软件模拟层吗？[5,18]

针对这个问题，剑桥大学计算机实验室（2003 年，Xen 虚拟机监控器诞生的地方）和 Xen Project（虚拟机管理程序的管理机构，现在通过 Amazon 和 Rackspace 等公司助力公有云）都提出了很多想法。名为 *MirageOS* 的解决方案所基于的研究概念虽然几十年前就已成型，但直到现在才随着云计算资源出现而变得可行并得以大规模部署。

MirageOS 的目标是将整个虚拟机（包括所有内核和用户空间代码）重构为模块化程度更高的组件。这些组件灵活、安全，能够以库操作系统的方式重用。应用装置中*所有*软件层都可以在同一个高级语言框架内编译，而不是每次启动时都要进行动态汇编的好处是什么？首先，我们来了解一下有关应用装置、库操作系统和类型安全编程语言的背景信息。

图 1.（传统）软件层次和一个独立内核 (MirageOS) 的编译。

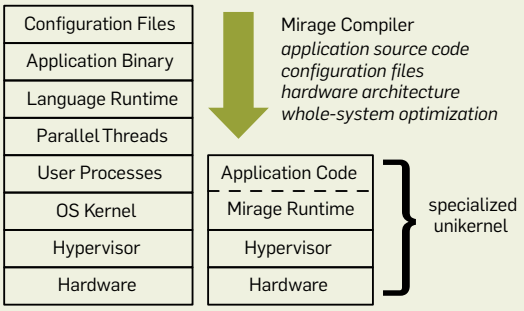
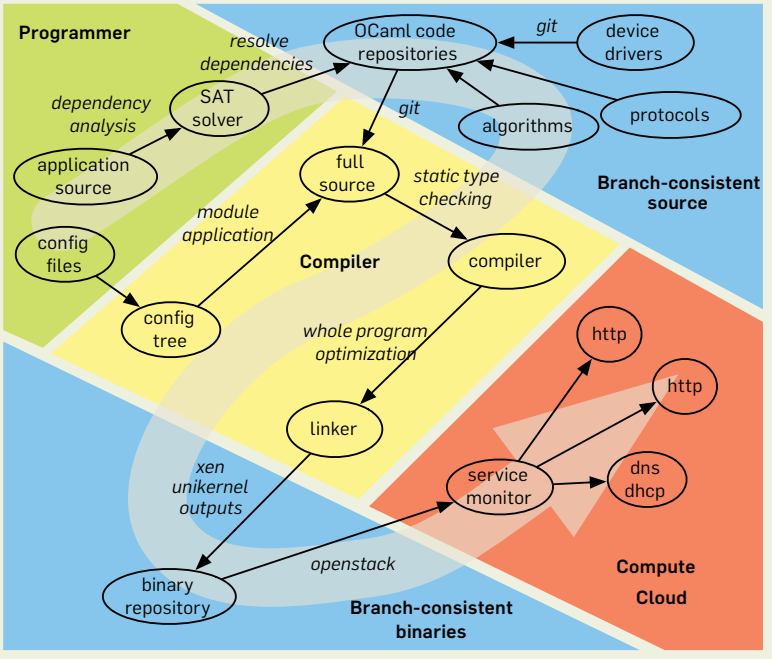


图 2. MirageOS 中的逻辑工作流。



向单一用途应用装置的迁移。目前，在云端运行的典型虚拟机包含完整的操作系统映像：如 Linux 或 Windows 内核，其托管在用户空间运行的主要应用程序（例如 MySQL 或 Apache）和并发运行的辅助服务（例如系统日志或 NTP）。虚拟机每次启动时都会通过从存储读取配置文件来初始化其通用软件。

大多数已部署的虚拟机虽然包含许多灵活的软件层，但最终都执行单一功能，例如作为数据库或 Web 服务器。向单一用途虚拟机的迁移反映了按需部署新虚拟计算机的便捷性。十年前，部署一个（物理）计算机实例甚至需要更多时间和成本，所以一台计算机需要运行多个最终用户应用程序，进而需要仔细配置计算机，以将成员服务和用户相互隔离。

构成虚拟机的软件层尚未赶上这一趋势，所以这提供了真正的优化机会，而优化空间不仅在于通过根据任务对应用装置做适应性修改来提升性能，而且还在于通过消除冗余功能和减少公有云上所运行服务的受攻击面来提高安全性。但由于现有操作系统的结构问题，通过静态方式来实现难度颇大。

当前操作系统的局限性。现代虚拟机监控器提供了可动态纵向扩展（通过添加内存和核心）和横向扩展（通过生成更多虚拟机）的资源抽象。许多应用程序和操作系统无法充分利用这一功能，因为它们是在现代虚拟机管理程序出现之前设计的（内存热插拔等物理上的类

似操作在商用硬件中并不常见）。人们通常将应用程序级的外部负载平衡器添加到虚拟机中运行的传统应用程序，以便在负载增加时生成新的虚拟机，从而使服务能够*灵活地*响应。但是，传统系统没有在大小或启动时间上做优化（例如，Windows 可能会在启动时应用多个补丁），因此负载平衡器必须随时保留空闲虚拟机来应对负载尖峰，以此来弥补优化上的不足，但这非常浪费资源和金钱。

为什么无法简单修复操作系统的这些问题呢？现代操作系统仍然是坚定不移地以通用性为导向，目的是解决大众问题。例如，运行 Linux 的平台非常多样，从低功耗的移动设备到驱动庞大数据中心的高端服务器。仅仅是为了帮助某一类用户提高应用程序性能而牺牲这种灵活性的做法是不可接受的。

另一方面，专用服务器应用装置不再需要操作系统充当资源复用器，因为虚拟机管理程序可以在更低的级别做这些事情。这种方法的一个明显问题是，现有的大多数代码都依赖于已经固化的庞大接口，例如 POSIX 或 Win32 API。另一个潜在问题是，传统操作系统提供 TCP/IP 堆栈等服务用于通信，提供文件系统接口用于存储持久性数据：在我们美好的新世界，这些从何而来？

称为 *unikernel* 的 MirageOS 架构如图 1 所示。Unikernel 是专用的操作系统内核，用高级语言编写，并作为独立的软件组件运行。完整的应用程序（或*应用装置*）由一组运行中的 unikernel 组成，这些 unikernel 像分布式系统一样协同工作。MirageOS 基于 OCaml (http://ocaml.org) 语言，可生成在 Xen 虚拟机管理程序上运行的 unikernel。为了解释其工作原理，让我们了解一下诞生于上世纪 90 年代，但相当前卫的基本操作系统架构。

MirageOS 的目标是将整个虚拟机（包括所有内核和用户空间代码）重构为模块化程度更高的组件。这些组件灵活、安全，能够以库操作系统的方式重用。

库操作系统。这已经不是人们第一次询问此类操作系统是否存在这一问题了。一些研究组提出了基于*库操作系统*（或 libOS）架构的操作系统设计。第一批此类系统是上世纪 90 年代末出现的 Exokernel[6] 和 Nemesis[10]。在 libOS 中，保护边界被推压到最低的硬件层，从而产生：一组用来实现某些*机制*（例如用来驱动硬件或表示网络协议的机制）的*库*；和一组用于在应用层实现访问控制和隔离的*策略*。

相比较为传统的设计，LibOS 架构有几大优点。对于有性能要求（尤其是*可预测*性能）的应用程序，libOS 的优势是，允许应用程序直接访问硬件资源，而不必反复进行权限转换，以在用户空间与内核空间之间移动数据。libOS 没有中央网络服务来使高优先级网络数据包（例如来自视频会议呼叫的数据包）和低优先级数据包（例如来自后台文件下载的数据包）强制混合和干预。相反，libOS 应用程序具有完全独立的队列，数据包只有在到达网络设备时才会混合。

LibOS 架构有两大缺陷。首先，同时运行多个应用程序并使资源高度隔离颇有难度（虽然 Nemesis 在尽可能减少交互式应用程序之间的串扰方面做了大量工作）。其次，必须重写设备驱动程序，以适应新的模型。随着商用 PC 硬件的快速发展，无论有多少研究生奉命编写驱动程序，所有研究 libOS 原型注定要在短短几年内被废弃。这种方法只适用于硬件支持较窄的实时操作系统空间（例如 VxWorks）。

幸运的是，操作系统虚拟化在商用硬件上克服了这些缺点。现代虚拟机管理程序为虚拟机提供了 CPU 时间和高度隔离的虚拟设备，以满足在网络、块存储、USB 和 PCI 桥方面的需求。作为虚拟机运行的 LibOS 只需要实现这些虚拟

硬件设备的驱动程序，而可以依靠虚拟机管理程序来驱动真实的物理硬件。LibOS 应用程序之间的隔离能够以较低的成本实现，方法是：使用虚拟机管理程序为每个不同的应用程序生成一个新的虚拟机，同时使各虚拟机可以自由地被用来根据特定用途创建高度定制化的虚拟机。虚拟机管理程序层所使用的策略比传统操作系统更简单，粒度也更大，因为它只是提供由虚拟 CPU 和内存页组成的低级接口，而不是传统操作系统中面向进程和文件的架构。

尽管操作系统虚拟化使得 libOS 不需要编写设备驱动程序，但仍然需要*协议库*来代替传统操作系统的*服务*。现代内核都是用 C 写的，适合于低级程序（如设备驱动程序），但缺乏高级语言的抽象设施，并且要求对内存缓冲区等资源进行细致的手动跟踪。因此很多应用程序都有内存处理缺陷，经常表现为严重的安全漏洞。研究人员做了大量工作将 Windows 和 Linux 移植到 libOS 模型[16]，但对于我们而言，这提供了一个完美的理由来探索向后兼容性不那么好，但可以集成得更自然的高级语言模型。图 2 显示了 MirageOS 中的逻辑工作流。通过从源代码（局部和全局）和配置文件进行精确的依赖项跟踪，可以将已部署的内核二进制文件的来源详情记录在不可变的数据存储区，以便精确地按需求重新编译。

更强大的编程抽象。在通用应用程序开发领域，高级语言正在稳步发展，并越来越广泛地用来通过业务流程框架（例如 Puppet 和 Chef）将组件粘合在一起。不幸的是，整个逻辑通常分散在不同的软件组件中，并且通常用多种语言编写。因此，仅分析源代码很难对整个系统的行为进行静态逻辑分析。

MirageOS 的目的是用一个高级语言框将这些多样化的接口（内核和应用程序用户空间）进行统一。现代编程语言的一些好处包括：

尽管操作系统虚拟化使得 libOS 不需要编写设备驱动程序，但仍然需要*协议库*来代替传统操作系统的*服务*。

- *静态类型检查*。编译器可以将程序变量和函数划分为多个类型，如果变量的行为与变量所属类型不符，则会拒绝编译代码。静态类型检查在编译时（而不是运行时）捕获这些错误，并为系统程序员提供了灵活的方式来保护程序的不同部分，而不需要完全依赖于硬件机制（例如虚拟内存分页）。类型检查最明显的好处是它会报告内存不足的错误（例如缓冲区或整数溢出），在 CERT（计算机应急小组）漏洞数据库中，这类错误仍普遍存在。更高级的用途是功能式访问控制[19]，它可以完全实施在 ML 等静态类型系统中，但前提是代码完全运行于同一个语言运行时内。
- *自动内存管理*。运行时系统为程序员减轻了分配和释放内存的负担，同时仍允许手动管理缓冲区（例如，用于高效 I/O）。现代垃圾回收器还可通过增量回收和分代回收最大限度减少应用程序中断，从而使其可用于高性能系统构造。[7,11]
- *模块*。随着代码库的增长，模块将其划分为逻辑组件，并通过良好定义的接口将各组件连接起来。模块有助于软件开发的扩展，因为可以将内部实现细节抽象化，并且单次源代码更改的范围有限。有些模块系统（例如 OCaml 和 Standard ML 中）在编译时静态解析，很大程度上不受运行时成本的限制。目的是利用这些模块系统来构建整个系统，从而跨越程序中传统内核和用户空间的边界。
- *元编程*。如果编译器在编译时部分理解了系统的运行时配置，那么编译器能对程序所做的优化将大大增加。如果不知道运行时配置，编译器会非常谨慎，因为输出程序必须保持完全通用，以防万一。这里的目标是在编译时统一

配置和代码，以确保在部署到公有云之前消除无用代码。

这些功能大大简化了大型系统的构建：受管内存消除了许多资源泄漏；类型推断实现更简洁的源代码；静态类型检查在编译时（而不是执行时）检验代码是否符合某些抽象标准；模块系统允许以整个操作系统和应用程序堆栈所需要的尺度操作此代码。

OCaml 中的函数原型

我们从 2008 年开始构建 MirageOS 原型，目的是了解我们能够将库操作系统和云服务部署底层的编程模型统一到什么程度。第一个设计决策是采用*函数式编程*背后的原理来构建原型。函数式编程强调对抽象的支持，因为抽象更便于跟踪程序的可变性，并且以往的研究表明抽象不需要以性能为代价。[11]

但挑战是确定正确的模块化抽象，以支持整个操作系统和应用程序软件堆栈在单个可管理结构中的表达。MirageOS 从此发展成了一套由约 100 个开源库组成的成熟库集，实现了各种功能，并开始集成到商用产品，如 Citrix XenServer。[17]

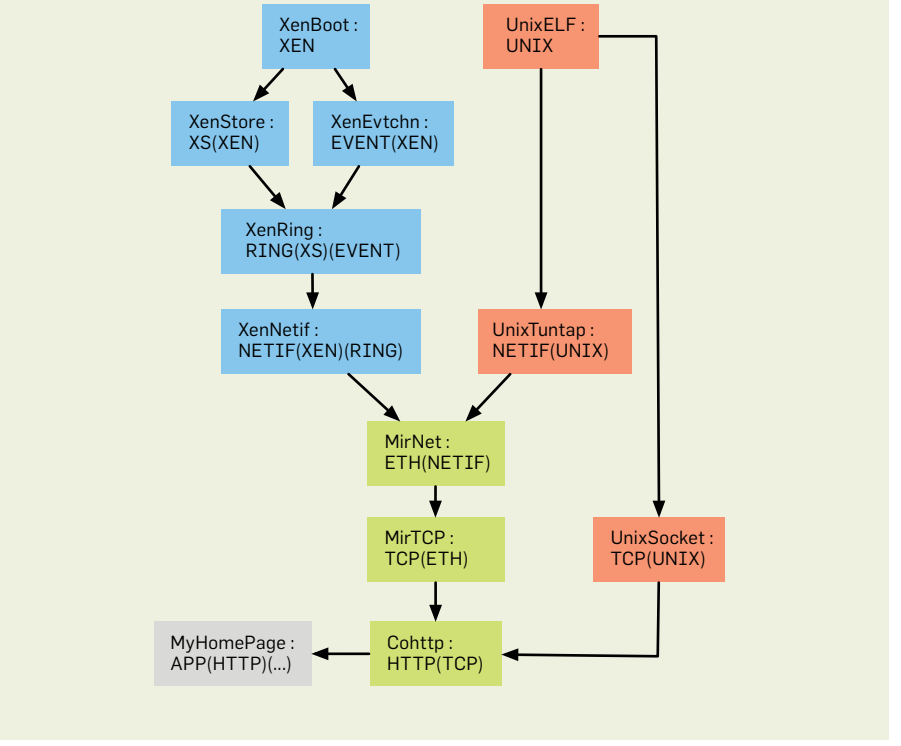
图 2 显示了 MirageOS 的设计。相比传统的云部署周期，它能让编译器更全面地了解源代码的依赖项：

- 输入应用程序的所有源代码依赖项都会被明确跟踪，包括实现内核功能所需的所有库。MirageOS 包括一个构建系统，其使用内部 SAT 解析器（使用 OPAM 包管理器和 Mancoosi 项目的解析器）从已发布的在线包集搜索兼容的模块实现。编译时，OCaml 的静态类型检查功能将捕获接口中所有不匹配的行为。
- 然后，编译器可以输出完全独立的内核，而不只是一个 Unix 可执行文件。这些 unikernel 是单一用途的 libOS 虚拟机，只执行其应用程序源代码和配置文件中定义的任务，并且依赖于虚拟机管理程序来提供资源复用和隔离。甚至是必须设置虚拟内存页表和初始化语言运行时的引导加载器也被写成了简单的库。每个应用程序都链接到其所需的一组特定库，并且能够以应用程序特定的方式将这些库粘合在一起。
- 这些专用 unikernel 部署在公有云。与传统虚拟化技术相比，unikernel 的受攻击面大大变小，并且在启动时间、二进制文件大小和运行时性能方面，资源利用率更高。

为什么使用 OCaml？ 将 OCaml 作为 MirageOS 唯一的基本语言有几个关键原因。它是一种成熟的系统编程语言，有灵活的编程模型、单个受 ML 启发的类型系统以支持函数式、命令式和面向对象的编程模式。它还提供可移植的单线程运行时，非常适用于移植到受限环境，如 Xen 准虚拟机。编译器非常重视静态类型检查，并且生成的二进制文件是快速本机代码，包含最少的运行时类型信息。主要类型推断功能允许安全地省略类型批注，并且在通用编程语言中，模块系统十分擅长允许灵活、安全的代码重用和重构。最后，OCaml 已经有多个在行业内[14]和 Xen 本身[17]的大规模使用案例，并且所取得的积极成果令人鼓舞，MirageOS 也正是在这种鼓舞之下开始了其长达数年的项目旅程。

模块化的操作系统库。 OCaml 支持定义*模块签名*（数据类型和函数声明的集合），其抽象了*模块结构*（具体数据类型和函数的定义）的实现。模块可以通过签名参数化，即创建定义其他数据类型操作的*仿函数*。（有关 OCaml 模块、仿函数和对象的详细信息，请参见 O'Reilly 发布的 *Real World OCaml*：https://realworldocaml.org。）我们利用 OCaml 模块系统将通常庞大的操作系统内核功能分解成离散单元。这样一来，程序员构建的

图 3. 静态 Web 服务器的局部模块图。



代码就可以在编写的过程中*逐步专用化*，即首先在熟悉的 Unix 环境中处理，最终得到在 Xen 中运行的专用云 unikernel。

举一个简单的例子。图 3 显示了一个静态 Web 服务器的局部模块图。库是抽象操作系统功能的模块图，而 OPAM 包管理器克服了对目标架构的限制。应用程序 `MyHomePage` 依赖于 `Cohttp` 库提供的 `HTTP` 签名。刚入行的开发人员想要通过 Unix 型开发环境相互查看代码。`Cohttp` 库需要 TCP 实现，以满足其模块签名（可由 `Unix-Socket` 库提供）。

程序员如果对其 HTTP 逻辑感到满意，即可重新编译，以便从使用 Unix 套接字转为使用 OCaml TCP/IP 堆栈（如图 3 中 `MirTCP` 所示）。这仍然需要 Unix 内核（但只是作为 shell）向 Web 服务器进程（现包含 OCaml TCP/IP 栈以作为应用程序的一部分）发送以太网帧。最后的编译策略完全放弃了对 Unix 的依赖，并重新编译 `MirNet` 模块以直接链接到 Xen 网络驱动程序，而驱动程序将获取其在 Xen 上启动所需的所有依赖项。累进式重编译对于 MirageOS 的可用性至关重要，因为我们可以从经过尝试和测试的 Linux 或 FreeBSD 功能逐步演进，但最终结果仍然是可以部署在公有云中的专用 unikernel。这一模块化操作系统结构使大量其他后端都可以通过类似方式实施到 Xen。MirageOS 现在有可在 NS3 中实现模拟器的实验后端（用于大规模功能测试）、FreeBSD 内核模块后端，以及甚至 JavaScript 目标（通过使用 `js_of_ocaml` 编译器）。这种模块化的自然结果是更便于编写可移植代码（精确定义在目标平台上的所需），而如果没有可替代 Posix（通过它，Linux、FreeBSD、Mac OS X 和 Windows 可以通过大量不兼容的 API 提供高性能服务）的现代技术，这会越来越困难。

图 4. MirageOS Xen unikernel 的虚拟地址空间。

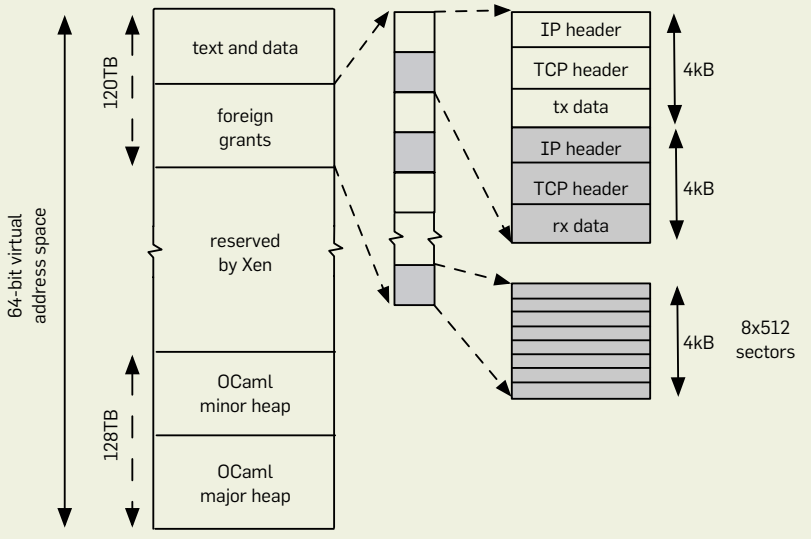
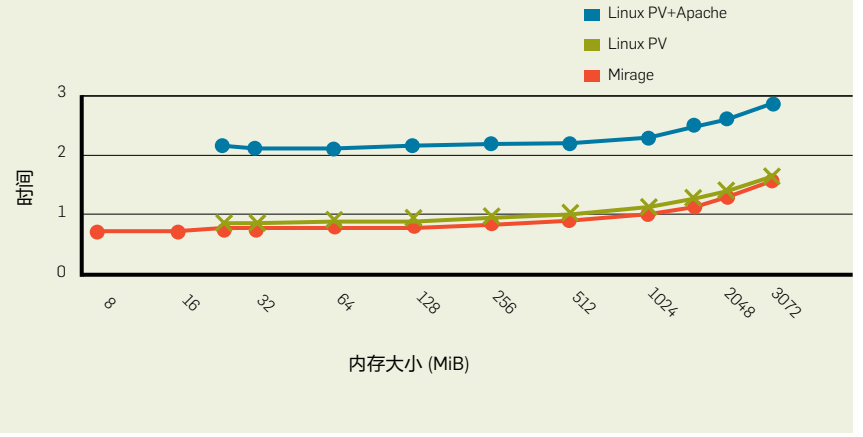


图 5. 启动时间。



配置和状态。在 MirageOS 中，库尽可能以函数式设计：它们是可重入程序，具有显式状态句柄，而这些句柄又是可序列化的，所以库能够以显式方式重建。应用程序由一组库和配置代码相互链接而成。配置是树结构，大致像一个文件系统，其中各库解析子目录以初始化各自的值（类似于 Plan 9 操作系统）。所有这一切都通过*元编程*来连接 — OCaml 程序生成越来越多已编译的 OCaml 代码，直到达到所需的目标。

元编程还延伸到了存储。如果应用程序只使用少量文件（这通常需要块设备和文件系统的所有资源），MirageOS 可以将其转换成满足文件系统模块签名的静态 OCaml 模块，从而使其不需要依赖外部存储。整个 MirageOS 主页 (http://openmirage.org) 就是以这种方式呈现的。

元编程的一个后果（故意为之）是，在输出的二进制文件中，大的功能块可能会完全丢失。这样就无法动态地重新配置专用化程度最高的目标应用，并且更改配置后需要重新链接 unikernel。MirageOS

Web 服务器所涉及的活动代码行数（即后配置）如表 1 所示，能体现出此类重新编译所需代码量之少。

链接 Xen unikernel。在传统操作系统中，应用程序源代码首先通过本机代码编译器编译成对象文件，然后提交给链接器生成可执行的二进制文件。编译后，动态链接器将可执行文件和所有共享库加载到具有自己的地址空间的*进程*。然后，进程可以在操作系统内核的协调下，通过系统调用与外界通信。在内核中，网络堆栈或虚拟内存系统等各种子系统处理系统调用并与硬件交互。

在 MirageOS 中，OCaml 编译器接收源代码并使其成为整个内核代码的一部分，然后将其链接到独立的本机代码对象文件。它根据可提供引导支持和垃圾回收器的最小运行时进行链接。没有抢占式线程，内核是事件驱动的，通过轮询 Xen 设备的 I/O 循环实现。

运行中的内核有一个虚拟地址空间，专用于运行 OCaml 运行时，这就是 Xen unikernel 编译的性能优势所在。MirageOS Xen unikernel 目标的虚拟地址空间如图 4 所示。由于所有配置信息都明确是编译的一部分，所以不再需要通常的动态链接支持，而这一支持要求在虚拟机启动后添加可执行文件映射。[13]

益处

思考下传统应用程序的生命周期。首先，将源代码编译为二进制文件。然后，将二进制文件加载到内存，并创建操作系统进程执行该文件。运行中的进程要做的第一件事是读取其配置文件，然后根据确定的环境将其自身专用化。许多不同的应用程序将运行完全相同的二进制文件（从相同二进制包获得，但具有不同的配置文件）。这些配置文件实际上是额外的程序代码，不同的是它们通常用专门的语言编写，并在运行时被解释而不是被编译。

unikernel 的一个缺点是需要调度更多改动更大的虚拟机，从而给云业务流程层施加沉重负担。

部署和管理。在管理大规模云托管服务的部署时，配置是极大的开销。使用 unikernel 编译技术，*已编译*（代码）和*已解释*（配置）之间的传统分离完全没有必要。应用程序配置是代码（也许作为一种嵌入式的领域特定语言），并且编译器可以分析和优化整个 unikernel。

在 MirageOS 中，数据库、Web 服务器等并不是必须由配置文件连接的独立应用程序，而是单个应用程序中的库，允许应用程序开发人员使用简单的库调用（适用于动态参数）或元编程工具（适用于静态参数）对它们进行配置。这可以使配置策略变得明确，并能够用一种主机语言来编程，而无需操作大量专门的文本文件，从而受益于静态分析工具和编译器的类型检查器。其结果是大大减少配置复杂多服务应用程序虚拟机所需的工作。

unikernel 的一个缺点是需要调度更多改动更大的虚拟机（由于每次重新配置都需要重新部署虚拟机），从而给云业务流程层施加沉重负担。近年来，常用的业务流程实现发展非常缓慢，并且其包含的许多分布式组件不仅难以管理，而且延迟和资源消耗都相对比较高。

MirageOS 最常见的生产用途之一是，使 XenServer[17] 中的 OCaml 代码往结构化 unikernel 理念演进，从而修复云管理堆栈。这将整个管理层变成一组更敏捷的可相互通信的虚拟机，这些虚拟机可以独立调度和重新启动。MirageOS 使构建这些单一用途的虚拟机易如反掌：首先将它们作为常规的 Unix 应用程序进行构建和测试，然后再根据 Xen 内核库重新链接 (http://openmirage.org/blog/xenstore-stub-domain)。如果将它们与 Xen 驱动程序管理域[3]结合使

用，可以极大地提高云管理堆栈的安全性和稳定性。

资源效率和定制。云是一个对所有资源使用都会加以计量和租用的环境。同时，由于多租户服务的负载千变万化，所以需要迅速对部署进行扩展：*向上*扩展以满足当前需求，或*向下*扩展以避免浪费。在MirageOS中，特定构建版本中不使用的功能不会包含在内，并且可在编译时（而不是部署时）通过整个系统优化技术来消除多余代码。在专用化程度最高的模式中，通过重新编译以重新配置服务，所有配置文件都是静态评估，从而可以广泛消除死代码。

由于unikernel的二进制文件很小（大约几百KB），因此可以更为流畅地通过互联网部署到远程数据中心。启动时间基本都在一秒以内，完全可以在收到传入网络数据包时再启动unikernel。

图5显示了MirageOS与Linux/Apache分发版中服务启动时间的比较。精简版Linux内核与MirageOS的启动时间差不多，但当Linux必须初始化用户空间应用程序时，效率立刻降低。MirageOS unikernel启动后可立即传输流量。

MLton[20]编译器率先采用了WPO（整个程序优化）技术，其中应用程序和所有库一同优化。在libOS中，整个程序实际上就是整个操作系统：无论是应用程序级别的代码还是低级的设备驱动程序，这种技术现在都可以对它们进行优化。传统系统为了动态链接而避免使用WPO技术，有时结合JIT（实时）编译技术，即动态分析程序，并实时生成经优化的代码。整个程序的编译时优化更适合关心资源效率和减少受攻击面的云应用程序。关于安全优势，请参见其他研究。[13]

表1. 典型的MirageOS unikernel运行Web服务器时所使用的库大小（近似值）。

库	C/kLOC	OCaml/kLOC
启动	18	0
OCaml运行时	20	0
线程	5	27
域间通信	跟踪	1
网络驱动程序	0	1
TCP/IP	执行踪迹	12
块驱动程序	0	1
HTTP	0	11
总计	**43**	**52**

表2. 其他unikernel实现。

Unikernel	语言	目标
Mirage[13]	OCaml	Xen、kFreeBSD、POSIX、WWW/js
Drawbridge[17]	C	Windows“picoprocess”
HalVM[8]	Haskell	Xen
ErlangOnXen	Erlang	Xen
OSv[2]	C/Java	Xen、KVM
GUK	Java	Xen
NetBSD“rump”[9]	C	Xen、Linux kernel、POSIX
ClickOS[14]	C++	Xen

最近一个有趣的趋势是操作系统容器，其中每个容器都由同一个操作系统内核管理，但有一个隔离的文件系统、网络和进程组。可以快速创建容器，因为无需启动新内核，并且它们与现有内核接口完全兼容。但是，这些优点是以更低的安全性和隔离性为代价；unikernel通过易于理解和审核的简易API，只共享最少量的虚拟机管理程序服务。Unikernel展示了将语言运行时分层到虚拟机管理程序完全可以替代轻量级容器。

可移植性的新领域。图3所示的MirageOS库结构明确地将库在其执行环境中的所需进行编码。虽然在传统意义上，这意味着类Posix内核和用户空间，但现在可以将OCaml编译到更多外部环境，包括FreeBSD内核模块、在浏览器中运行的JavaScript，或（像Scala语言一样）直接针对Java虚拟机(JVM)进行编译。

仍然需要注意OCaml类型系统中不可抽象的执行属性。例如，作为内核模块运行时，通常会禁用浮点数；因此，如果使用浮点代码针对该硬件目标进行编译，被修改的编译器会产生类型错误。

其他第三方OCaml代码通常呈现一种类似的结构，这使得在MirageOS下工作更加容易。例如，Arakoon (http://arakoon.org)是一个分布式键-值存储，实现了高效的多Paxos一致性算法。用于在MirageOS下编译的源代码补丁只处理了两个文件，并只限于添加新的模块定义，用于将Arakoon后端存储映射到Xen块驱动程序接口。

现实中的Unikernel

当然，MirageOS并不是最近几年出现的唯一unikernel，虽然它也许是在探索全新的设计空间方面做得最彻底的。表2显示了其他一些可构建unikernel的系统。HalVM[8]

最接近于 MirageOS 理念，但它基于著名的纯函数式惰性语言 Haskell，而不是具有严格评估标准的 OCaml。相反，OSv[2] 和 rump kernel[9] 为现有应用程序提供兼容性层，并弱化编程模型的改进和类型安全，而类型安全正是 MirageOS 的准则。Drawbridge 项目[16] 将 Windows 转换成 libOS，虽然每个应用程序只报告了 16 MB 的开销，但为了实现这一效率，它需要暴露比 Xen 更高级别的接口（例如线程和 I/O 流）。

最终，公有云应将所有这些新兴项目作为一等公民来支持，就像今天的 Linux 和 Windows 一样。Xen 项目的目的是支持崭新的尘云世界：微小的一次性虚拟机以目前无法想像的高密度运行在虚拟机管理程序上，并通过不断调用云结构对资源需求进行自我扩展。MirageOS 底层是 libOS，所以它并不局限于在一个虚拟机管理平台上运行，而是其中许多库可以编译到各种规模的环境[12]，从 ARM 智能手机到裸机内核模块。为了理解这种灵活性的影响，我们一直在研究用例，从管理个人数据[4]、促进匿名通信[15] 到构建软件定义的数据中心基础设施。

鸣谢

MirageOS 是一个浩大的工程，它的成功离不开社会各界在知识和资金上的支持。Richard Mortier、Thomas Gazagnaire、Jonatham Ludlam、Haris Rotsos、Balraj Singh 以及 Vincent Bernardoff 组成的核心团队在我们构建全新的 OCaml 代码时给予了大力支持，同时 Jon Crowcroft、Steven Hand、Ian Leslie、Derek McAuley、Yaron Minsky、Andrew Moore、Simon Moore、Alan Mycroft、Peter G. Neumann 和 Robert N.M.Watson 也在不断地给我们提供支持和反馈。鉴于篇幅限制，请恕我们无法逐一向此项目的所有贡献者致谢。我们鼓励读者访问 http://queue.acm.org 获取我们的完整名单。

此项目主要由 RCUK 投资的 Horizon Digital Economy Research (EP/G065802/1) 支持。一个部分由 DARPA（美国国防部高级研究计划局）和 AFRL（美国空军研究实验室）根据 FA8750-11-C-0249 合同予以赞助。本报告中包含的观点、意见和 / 或发现均属于作者，不得解释为代表 DARPA 或美国国防部 (Department of Defense) 明示或暗示的官方观点或政策。

MirageOS 可在 http://openmirage.org 自由获得。欢迎向我们提供反馈、补丁和奇思妙想。 C

queue.acm.org 上的相关文章

Self-Healing in Modern Operating Systems（现代操作系统的自我修复）
Michael W. Shapiro
http://queue.acm.org/detail.cfm?id=1039537

Erlang for Concurrent Programming（Erlang 并发编程）
Jim Larson
http://queue.acm.org/detail.cfm?id=1454463

Passing a Language through the Eye of a Needle（穿过针孔的语言）
Roberto Ierusalimschy, Luiz Henrique de Figueiredo 和 *Waldemar Celes*
http://queue.acm.org/detail.cfm?id=1983083

OCaml for the Masses（为大规模而生的 OCaml）
Yaron Minsky
http://queue.acm.org/detail.cfm?id=2038036

参考资料

1. Barham, P. et al.Xen and the art of virtualization.In *Proceedings of the 19th ACM Symposium on Operating Systems Principles* (2003), 164–177.
2. Cloudius Systems.OSv; https://github.com/cloudius-systems/osv.
3. Colp, P. et al.A. Breaking up is hard to do:Security and functionality in a commodity hypervisor.In *Proceedings of the 23rd ACM Symposium on Operating Systems Principles* (2011), 189–202.
4. Crowcroft, J., Madhavapeddy, A., Schwarzkopf, M., Hong, T. and Mortier, R. Unclouded vision.In *Proceedings of the International Conference on Distributed Computing and Networking*, 29–40.
5. Eisenstadt, M. My hairiest bug war stories.*Commun. ACM 40*, 4 (Apr. 1997), 30–37.
6. Engler, D. R., Kaashoek, M. F. and O' Toole, Jr., J. Exokernel:An operating system architecture for application-level resource management.In *Proceedings of the 15th ACM Symposium on Operating Systems Principles*, (1995), 251–266.
7. Eriksen, M. Your server as a function.In *Proceedings of the 7th Workshop on Programming Languages and Operating Systems*, (2013), 5:1–5:7.
8. Galois Inc.The Haskell Lightweight Virtual Machine (HaLVM) source archive; https://github.com/GaloisInc/HaLVM.
9. Kantee, A. Flexible operating system internals:The design and implementation of the anykernel and rump kernels.Ph.D. thesis, Aalto University, Espoo, Finland, 2012.
10. Leslie, I.M. et al.The design and implementation of an operating system to support distributed multimedia applications.*IEEE Journal of Selected Areas in Communications 14*, 7 (1996), 1280–1297.
11. Madhavapeddy, A., Ho, A., Deegan, T., Scott, D. and Sohan, R. Melange:Creating a "functional" Internet. *SIGOPS Operating Systems Review 41*, 3 (2007), 101–114.
12. Madhavapeddy, A., Mortier, R., Crowcroft, J. and Hand, S. Multiscale not multicore:Efficient heterogeneous cloud computing.In *Proceedings of ACM-BCS Visions of Computer Science.Electronic Workshops in Computing*, (Edinburgh, U.K., 2010).
13. Madhavapeddy, A. et al.Unikernels:Library operating systems for the cloud.In *Proceedings of the 18th International Conference on Architectural Support for Programming Languages and Operating Systems*, (2013), 461–472.
14. Minsky, Y. OCaml for the masses.*Commun.ACM 54*, 11 (Nov. 2011), 53–58.
15. Mortier, R., Madhavapeddy, A., Hong, T., Murray, D. and Schwarzkopf, M. Using dust clouds to enhance anonymous communication.In *Proceedings of the 18th International Workshop on Security Protocols* (2010).
16. Porter, D.E., Boyd-Wickizer, S., Howell, J., Olinsky, R. and Hunt, G.C. Rethinking the library OS from the top down.In *Proceedings of the 16th International Conference on Architectural Support for Programming Languages and Operating Systems*, (2011), 291–304.
17. Scott, D., Sharp, R., Gazagnaire, T. and Madhavapeddy, A. Using functional programming within an industrial product group: perspectives and perceptions.In *Proceedings of the 15th ACM SIGPLAN International Conference on Functional Programming*, (2010), 87–92.
18. Vinge, V. *A Fire Upon the Deep*.Tor Books, New York, NY, 1992.
19. Watson, R.N.M.A decade of OS access-control extensibility.*Commun.ACM 56*, 2 (Feb. 2013), 52–63.
20. Weeks, S. Whole-program compilation in MLton.In *Proceedings of the 2006 Workshop on ML*.

Anil Madhavapeddy 是剑桥大学系统研究组的高级研究员。他曾是用 OCaml 开发 Xen 虚拟机管理程序和 XenServer 管理工具集的最早团队的成员之一。XenServer 已在数百万主机上部署，并为许多财富 500 强企业的关键基础设施提供支持。

Dave Scott 是思杰系统公司 (Citrix Systems) 的首席架构师，主攻 XenServer 虚拟化平台。他的工作重点是利用开源软件和高级语言的先进成果，提高 XenServer 的可靠性和性能。

责任编委：陈海波

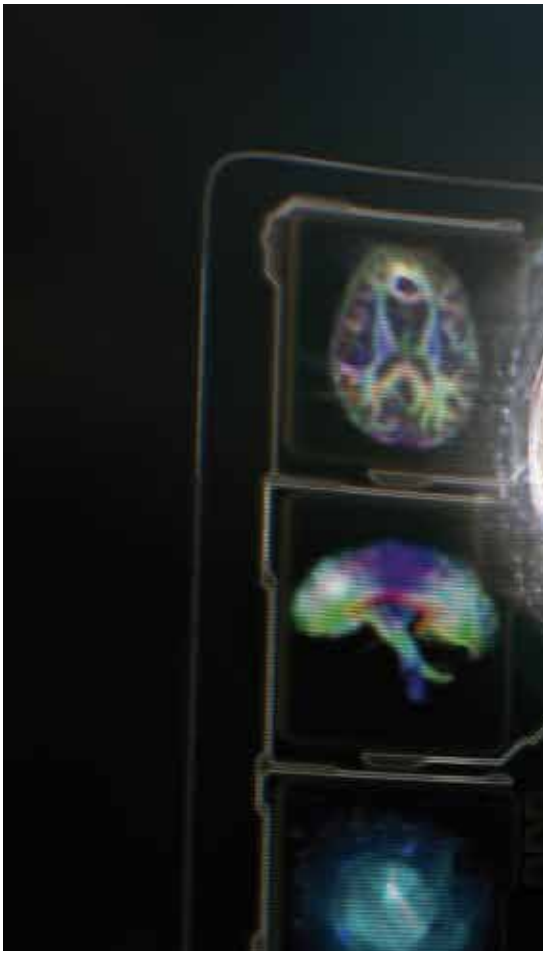
2014 ACM 0001-0782/14/01 $15.00

DOI:10.1145/2541883.2541899

与医学影像的非接触式交互让手术医生在手术过程保持无菌状态

作者：KENTON O'HARA、GERARDO GONZALEZ, ABIGAIL SELLEN、GRAEME PENNEY、ANDREAS VARNAVAS、HELENA MENTIS、ANTONIO CRIMINISI、ROBERT CORISH、MARK ROUNCEFIELD、NEVILLE DASTUR 与 TOM CARRELL

外科手术中的非接触式交互



对任何手术室扫一眼就可看到许多用于术前和术中影像评估的显示设备，包括计算机断层扫描 (CT)、磁共振成像 (MRI) 及荧光镜检查，以及各种专门针对手术的影像应用。这些显示设备能为诊断和手术计划提供支持，医生利用这些显示设备，可在外科手术期间观察到患者体内的情况。虽然手术医生们依靠采集、浏览和操作这些影像来获得信息，但他们也受到了传统交互方式 (如键盘和鼠标) 的限制。

这些限制的关键方面就是需要在无菌物体与有菌物体之间保持严格的界线。当手术医生清洗并带上手套后，除非中断无菌操作，否则他们就不能接触这些输入设备。如要绕开这一限制，目前可使用几种策略来与影像交互。尽管这些策略经常不是很理想，例如，手术医生们通常可请求手术小组其他成员 (如放射线技师及护士) 在他们的指示下操作影像。7,11 虽然这样做能成功，但也可能会带来额外的问题。小组成员并不总是空闲，尽管对于相对离散和简单的影像交互请求来说发布指令还可以接受，但会比较麻烦并且耗费时间。更为重要的是，对更多需要手术医生利用医学影像完成分析和解释的任务来说，对影像间接的操作并无助益。

» 重要见解

- **除证明非接触式交互系统的技术可行性外，还应将外科手术中的这类系统进行合理设计，让其能在手术室操作环境内工作。**
- **手势设计不仅应考虑与医学影像的个体交互，还应考虑如何在协作讨论环境下使用这些影像。**
- **与单手和双手相关的手势设计应能满足表达丰富度的要求，以及手术医生双手操作的要求，同时还要受到了手术小组成员靠近程度及无菌操作所产生的动作限制的限制。**

他们与这些影像交互，浏览并且选择性操作这些影像的方式与其临床知识及临床解读密切相关。

研究表明，手术医生们需要对影像数据进行直接控制，以便心智上“掌控”手术中正在发生的各种情况，[7] 这是通过代为操作者所难以实现的。为了能实施直接的亲手控制，一些临床医师会将其手术衣拉到其手上，隔着手术衣操作鼠标。[7] 这样手术衣未经灭菌的背面就接触到了鼠标 (也未经灭菌)，而手术衣及手部的正面是无菌的，仍然与有菌面相隔离 (见图 1)。这样的做法并非没有风险。对于无创手术，由于这些做法在节约时间和影像直接控制方面能带来临床好处，因此这些做法有其合理性。但对于创伤性较高的手术，这类做法就显得有些不妥。在手术医生需要对影像进行亲自控制的情况下，他们必须摘下手套并再次进行手臂消毒，这会花费宝贵的手术时间。如果手术时间较长，比如很可能涉及到多种需要与影像交互的情况，则就会因这样的操作而造成手术的严重延误，从而增加费用和临床风险。

让手术医生们在手术室内保持无菌状态的同时还能直接掌控影像的操作和导航是一项关键目标，[20] 这一课题激发了全世界研究团体和商业机构的想像力。对于某些医生，方法就是在手术医生的无菌手套与有菌交互设备之间增加一道屏障设

插图由 KOLLECTED 提供。

备 (如艾迪欧公司的袋内光学鼠标解决方案 [5]) 虽然这类解决方案简单而精巧，但在实际操作时仍有一些隐忧。另外，由于屏障设备可能会损坏，以屏障设备为核心的这类解决方案存在一些固有的风险。另一些方法则寻求在手术室内实现能避免与输入设备接触的交互技术。2005 年前后，此类兴趣的萌芽渐趋显现。这一时期，计算机视觉技术刚刚用于通过跟踪手术医生空中手势控制医学成像系统。Graetzel 等人 [4] 在一个早期的非接触式医学影像系统应用实例中，让手术医生通过被摄像头跟踪的手部动作进行标准的鼠标功能操作 (如光标移动和点击)。不久之后，在 Wachs 等人的 Gestix 系统中对外科影像技术使用了更复杂的空中手势 [21]。Gestix 系统不只是模拟鼠标功能，而且还使得执行定制性更强的手势控制功能 (如导航、缩放和旋转) 成为可能。

这些初步的系统为该类技术的发展铺筑出了一条重要道路，在近期，如 Ebert 等人 [1,2] Gallo、[3] Johnson 等人、[7] Kipshagen 等人 [9]、Mentis 等人、[11] Mithun 等人、[13] O’Hara 等人、[15] Ruppert 等人、[16] Stern 等人、[17] Strickland 等人、[18] 及 Tan 等人 [19] 所论述的那样，考虑在外科手术环境中使用医学影像非接触式控制技术的系统数量及研究力度已经出现明显增长，支持该增长的其中一项驱动要素就是 Kinect 传感器及软件开发套件 [12]，这些要件降低了进入的诸多障碍，包括降低了经济成本和开发复杂度，以及不再需要穿戴可跟踪标记器。Kinect 传感器基于激光器及可水平移动的红外 (IR) 相机。激光器将已知的图案投射到场景上。通过分析在 Kinect 公司的红外相机中图案的变形特征来估算景像上每个点的深度。

估算出景深后，就会基于机器学习技术的算法，自动将每个像素解读为属于背景还是属于操控人体的 31 个细分部分的某一部分。之后将此信息用来计算“人体骨架”（人类控制者的人偶图）的位置。Kinect 已经帮助克服了纯摄像机系统在全深度人体骨架捕捉领域所固有的一些挑战。由于这类系统的不断涌现，业界产生了共同的关注点，随之也产生了通过更多方式解决外科手术期间非接触式交互问题的机会。业界关注的不再是证明此类解决方案的技术可行性，而是如何最佳地设计和实现这类非接触式系统，让其在具有手术室操作特点的特定需求和情况下能发挥功效。通过反思此类解决方案，并且也由于对该技术愈加浓厚的兴趣，我们以关键项目为出发点，重点介绍所获得的一些经验以及与这些系统开发相关的各种问题和挑战。

首先要介绍的就是用于多伦多新宁医院多种外科手术的系统 [18]，在这家医院，Kinect 系统帮助利用一组简单的手势在预设定的磁共振影像或计算机断层扫描图片栈内实现前后翻页等浏览操作，并能与系统连接和脱离；而与系统的连接和脱离，是非常重要的一个问题，本文将在后文继续进行论述。

除非图片转换操作 (如旋转、缩放或其他参数调节) 被集成到预先定义的影像栈中，否则在系统中不能进行任何此类操作。新宁医院系统的简洁真正展现了设计上的精巧。有限数量的手势动作也带来了易用性和系统学习能力方面的益处。另外，这一系列的限定手势也能在可靠性方面带来一定的益处：能确保使用明显不同的手势，避免“手势冲突”。在“手势冲突”的情况下，手势集中的手势使用了一些相同的肢体运动，这样就很可能导致系统对这些手势的解读错误。考虑到该系统是现今已经实际部署和使用的少量系统之一，那么开发人员做出设计选择的过程中首先要考虑的就是系统的可靠性。同时请注意，还应考虑在手势库设计中采用双手动作，这一技术可带来一定的益处，同时也限制了该系统在外科手术环境中使用的方式，这一关键问题本文在后续内容中也会涉及。

虽然新宁医院系统表现出精巧和简约，但这一解决方案还必须解决其固有的一些限制。在外科手术环境中与医学影像进行交互经常不仅仅涉及到导航方面的操作，除旋转 / 摇移 / 缩放外，还需能执行丰富得多的影像操作，这些操作可能包括对各种影像参数 (如用来显示诸如骨骼、组织及血管等各个特征的密度函数及不透明度) 进行调

图 1. 用来避免带有无菌手套的手触摸未经灭菌的鼠标的外科手术衣。

节。甚至还可能包括在手术期间对影像进行标记或添加注解。而且，操作还可能适用于整个影像或临床医师指定的更特定的感兴趣区域。考虑到这些可能性，用 Kinect 技术与外科手术影像进行非接触交互的最近几个项目已经开发了更大的手势集来满足更多的功能需求，并接入符合医学数字图像与通信标准 (DICOM) 的标准化开源影像显示工具以及影像存档及传输 (PACS) 系统 (如医学影像开发包及 OsiriX 系统)。一些值得注意的系统还包括 Ebert 等人、[1,2] Gallo 等人、[3] Ruppert 等人、[16] 及 Tan 等人 [19] 开发的系统。

为系统扩充更丰富的功能的确很好，但也涉及一些严峻的挑战。其中一种挑战就涉及表达的丰富性这一概念，或者说如何将不断增加的功能 (经常涉及一个参数水平的连续调节) 映射到手势差异明显的手势库中。在这些系统中，已经应用了几种方法 (如使用状态来区分手势和输入通道，包括语音及双手组合动作)。例如，使用单手和双手手势跟踪不仅带来了双手交互的好处，还丰富了表达方式。在 Ebert 等人 [1,2] 及 Gallo 等人 [3] 的系统中，手势集都采用了单手和双手手势。这样，就可使用不同的手势组合 (如单手、双手组合在一起、双手分开) 表示特定的影像参数；可根据这些参数在 *x*、*y* 及 *z* 平面上的各自位置对他们进行调节。Ebert 等人 [1] 系统的较新版本还实现了更进一步的表达功能，这些功能利用了可识别手指的动作跟踪算法，其中双手摊开的动作是可以和手掌张开这样动作区分开来的。

由于如此表达能力可实现更大的手势集，开发者们随之就产生了对系统学习能力的关注，[14] 尤其当可能需要吸纳越来越多新的系统功能时。在 Ruppert 等人 [16] 及 Tan 等人 [19] 的系统中，我们已经看到

与其说是多人想同时控制影像，倒不如说是有时一个人必须能流畅地将控制转交给另外一个人。

应对这些问题的一些尝试，这些系统将以一种协调和可扩展的方式积累结合惯用手和非惯用手的组合手势。非惯用手用来选择特定的功能或模式，而惯用手在 *x*、*y* 及 *z* 平面内移动，用来对影像参数进行连续调节。通过这种方式，就可使用一些常用的手势执行各种不同的功能，从而提高系统的学习能力及可扩展性。

现在单手及双手手势的使用已经出现，这在非接触式医疗系统的设计和理解中，已经成为一个重要问题，本文后续内容将会再次对此问题进行论述。特别要指出的是，虽然从表面上看可以很明显地确定这些不同的解决方案是受到特定的控制语用学 (如对表达丰富度及学习能力的需要) 驱动而开发的，但不明确的是具体的设计决策是如何受到双手交互设计 [10] 原则驱动而产生的，甚至更重要的是当考虑在实际外科手术环境中可能如何使用这些系统时，是如何受更广泛社会技术问题的驱动而产生的。

如我们在 Ebart 等人 [2] 及我们自己的作品中看到的那样，另外一种可能就是使用语音识别技术。[15] 但是，在噪音较大的手术室内使用时，语音识别软件会涉及到特殊的挑战，因此如果单独使用，可能就不适合用于连续参数的操作。但在这些系统中使用语音最重要的一方面就是如何能将语音识别与手势模式结合在一起，实现控制的目的；对于离散的动作和功能 (如更改模式和功能)，语音控制可能带来重要的益处。

社会技术方面的考虑

除开发出非接触式控制机制来满足无菌要求外，还需解决一个最关键的问题。首先，它们需要适应手术小组实施手术操作的环境以及手术室的环境。这类环境和实际操作塑造和限定了系统设计的选择，这些

图 2. 用于在血管外科手术中操作 3D 覆盖层的手势系统。

选择涉及到比如跟踪算法、手势集及交互在不同输入通道 (包括语音和手势) 之间的分布。虽然本文讨论的系统中许多都是与临床合作机构合作开发，并且获其成功采用，但他们根据手术环境和工作实践所做的设计选择背后的理由仍然不清晰。随着这一领域的发展壮大，我们有必要仔细研究这些问题，并让这些问题更清晰。为了达到这个目的，我们利用了血管外科手术系统方面的开发经验，以及如何在现场观察之后将该系统的设计选择与特定的社会技术考虑因素关联在一起。我们重点关注自己的经验是为了便于说明，这样做的意图是为强调我们所讨论的更广范围的技术方面的经验教训。

我们开发的系统用于在英国伦敦盖圣托马斯医院 (GSTT) 进行影像引导式血管外科手术。在这类手术期间，手术医生通过手术床上方一排监视器的实时荧光镜检查及 X 射线影像获得连续的手术指导。在其中一台监视器上，在连续更新的 X 射线图片上重叠了 (依据术前 CT 数据) 主动脉的容积渲染影像，帮助手术医生观察到插入的导丝和支架相对于主动脉实际结构的位置。而对于这种组合重叠影像的操作，是通过该系统基于 Kinect 的手势及语音识别功能实现的 (见图 2 和图 3)。

在系统设计过程中，我们必须解决重要的社会技术问题，这些问题对于如何考虑系统开发，包括协作及控制、连接及脱离以及单手、双手和无手影像检视等具有广泛意义。

协作与控制。在许多系统中，设计重点关注的是为手术室中的手术医生提供单个控制点。虽然这一任务仍然是很重要的目标，但外科手术需要涉及针对影像操作中重要的协作方面的内容 (如 Johnson 等人 [7] 及 Mentis 等人 [11] 的著作所述)。与其说是多人想同时控制影像，倒不如说是有时一个人必须能流畅地将控制转交给另外一个人，例如，如果手术医生正在忙于手术及患者管理，则其他临床支持人员可能就必须接管影像的控制权。而在其他时间，主导手术实施的临床医生可能会将某些责任转交给专科医生或实习生。另外一个重要的协作问题涉及到协作性临床解释和讨论，在这一过程中，手术小组各个成员会在所显示的影像周围用手指指点和做出各种手势。

在 GSTT 的系统中，我们通过跟踪多个小组成员的人体骨架，利用颜色编号方式为他们分配各不相同的光标对，这些光标对与他们的双手对应。通过对光标进行颜色编号，就允许协作人员在对某一正确操作过程进行讨论、解释和计划时对影像的不同部分进行指点并做出各种手势。任何时候，他们都能举起双手并发出语音指令，请求获得系统的控制权，因此，与本文论及的其他系统一样，这一系统也是任一时刻只允许影像有一位主要控制者。但是，即使在此模式下，如果手术有需要，小组其他成员也能通过可见的光标指点和做出各种手势，从而通过语音指令在任何时刻接管控制权。

与系统的连接和脱离在屏幕前做出手势并不总是为了系统控制的目的，除了在谈话过程中起辅助示意作用的手势，在手术环境下执行的其他操作，或手术医生试图在手势之间过渡都可能在屏幕前产生一些移动。上述这些动作提高了系统将其意外识别为系统控制手势的可能性。因此，这些系统设计中的关键就是需要有相关的机制支持在各系统状态、以及连接状态与脱离状态之间切换，同时需要有指示系统状态的相应反馈去加强用户对状态的理解。

不同系统会采用多种方法，每一种都有其本身的利弊，例如，在新宁医院系统中，[18] 开发人员有意地加入了在头部上方的非常规手势，来实现与系统的连接 / 脱离。这一手势在其他活动过程中不大可能发生，因此认为其在避免意外触发方面是有用的。我们在开发系统的过程中尝试了各种方法，获得了不同形式的收获。例如，为了与系

统连接以识别各种手势，我们初始使用了右手“挥手”的手势，此手势存在“手势过渡”的问题，由于存在此问题，为启动挥手手势而有必要执行的动作有时就被识别为另一个离散的手势。

这种错误的识别涉及到“手势点定位”的问题或通过低级运动知觉特征 (如加速度 [8]) 检测手势的起始点和终止点。虽然手势点定位技术正在改进，但对于系统开发人员来说其仍然是一个内在的艰巨挑战。而解决这一问题的一种方法就是利用非分类式技术，利用这类技术，连续的影像参数可与控制者手部的连续位置相对应。但由于多种原因 (如参数调节幅度超出了手臂自然动作在任一平面上所能达到的范围，以及屏幕某些特定区域用于额外的功能时)，这类方法易于发生手势过渡的问题。为了解决这类问题，我们加入了离合机制，在这一机制中，将手臂回缩靠近身体，以实现与系统的脱离，从而允许在系统不对对应影像进行操作的情况下实现动作过渡。

另外，我们还采用了一个基于时间的锁定程序，利用该程序，手术医生可将其手部保持在正确位置几秒钟。虽然这一设计在其他手势交互领域中获得了成功，但通过对手术医生进行评估时我们发现，他们有一种自发的倾向，即暂停下来并对影像进行检查，或将姿势保持不动，以便指向影像中的某一特定的特征。这些行为与暂停类锁定手势是相冲突的，导致我们对系统进行修改，这样我们就通过作为手势集补充的简单语音指令来实现连接与脱离控制，这一控制在需要对离散状态进行切换时效果良好。

其他开发人员还对自动确定手术小组人员连接系统和脱离系统意图的技术进行了探索，例如 Mithun 等人 [13] 的研究成果中，讨论了通过上下文线索 (如注视、手部位置、头部朝向及躯干朝向) 来判断手术医生是否有意图执行可被系统读懂的手势。这类方法在避免无意手势方面表现出了较好的前景，尽管根据这些线索确定人类意图仍然是一项挑战，例如，在协同讨论过程中 (如在想与系统交互时) 围绕影像进行交谈和做出手势时这些背景线索就很可能是相似的。

单手、双手及无手动作本文讨论的一些系统利用了单手及双手手势。除增加手势集的丰富度和探索交互过程中双手动作的重要特性外，在设计系统用于单手操作或双手操作时，也考虑了一些重要的临床因素，如当手术医生手拿一些医疗器械时有时需要与影像进行交互，由于存在这一情况，因此就产生了一个问题，即在某一特定时刻要执行某些手势操作时医生还有几只手可用。手势集的设计不仅仅是关于设计正确数量的指令与功能相匹配的问题，而且是关于如何反映临床使用环境的问题。

在 GSTT 医院应用的我方系统采用了各种不同的单手和双手手势。我们对手术医生的观察及采访表明，要对影像进行旋转和缩放操作，一般需要将器械和导管放下时才可完成。而要降低覆盖层的不透明度并给覆盖层加上标记注解 (以高亮显示底层荧光镜检查影像上的对应点)，手术医生可能会保持一只手抓住导管不放，这样就只能留出一只手空闲了。由于这些临床方面的原因，为实施旋转和缩放操作系统就需操作者执行双手手势，而要降低不透明度，则可使用空闲的那只手来执行手势。要让影像重叠，系统就将单手手势跟踪与语音指令结合在一起，利用这一功能，手术医生就能在握住导管的同时让指令得以执行。

但这并不是说只要是在医生使用其他器械时都可实现非接触式控制。事实上，在手术过程中的确存在影像操作可能干扰手头任务的许多情况。但是系统开发人员也可以考虑手势组合的可能，因此，必须根据临床意义定义好涉及到双手的手势集的规范。不同类型的外科手术明显会涉及到不同的限制，比如怎样及何时可将影像操作机会与外科手术器械的使用结合在一起，这

图 3. 用于手术室中血管外科手术的手势系统。

图 4. 与医学影像的交互：(a) 远离手术台以及 (b) 待在手术台旁边。

(a)

(b)

样就需要仔细考虑尤其是在两只手可能都握有器械时如何完成输入的问题。在这些情况下，也许可以利用语音指令进行无手操作 (只要这些操作是适应语音指令的离散特征的) 或将语音与其他输入通道 (如脚踏板、眼神注视输入和头部动作) 结合在一起。总体来说，在设计这些系统时，开发人员必须不仅仅要根据技术需要还要根据临床需要，即有一只手、两只手还是两只手都不能用于影像交互来选择所要采用的方法。

在手术台前，远离手术台这里给出另一个重要的设计考虑 – 即手术医生需要与不同影像系统交互时他们所处的位置 (见图 4)。除了在手术台上会使用到各种工具外，手术台还会是一个相当拥挤的环境，在这种环境中，手术医生经常与手术小组的其他成员靠得非常近。这不仅会影响到系统的手势跟踪方法，而且还对手势设计中可以用到的动作类型带来了各种限制 (如因紧挨其他人工作而带来的身体动作限制，以及因严格无菌操作而产生的一些限制)。在无菌操作中，必须将手部动作限定在手术医生臀部与胸部高度之间躯干部分正前方的范围内。而其肩膀、胸部以上、大腿及背部周围的范围内执行动作则被认为会增加破坏无菌状态的风险，因此必须避免在这些区域内执行动作 (及手势)。而且，手术台本身也掩盖了手术医生身体的下半部，而当离开手术台后，手术医生的身体就有更多部分暴露在跟踪系统下。Kinect 提供了两种跟踪模式：默认跟踪模式 (此模式进行了全身人体骨架跟踪方面的优化)，以及坐下模式，此模式进行了上半身 (头部、肩部及手臂) 跟踪方面的优化。虽然全身跟踪适合图 4a 中的情形，但上半身跟踪模式更适合图 4b 中的情形。

手术医生在手术台旁边的位置取决于手术的临床需要，因此其距离和朝向方面并不总在手势捕捉设备前面的理想区域内。在手势及跟踪功能的设计过程中，系统开发人员可能不得不考虑和满足这类变化。虽然开发人员可能想考虑其他取决于临床需要和手术环境需要的配置 (如手术医生可坐在 PACS 系统前面，如图 1 所示)，但本文的实例目的只是用来对更广泛的问题进行讲解。

结论

本文的目的不仅仅是证明非接触式控制在临床环境中的可行性；重要的设计挑战包括从手势集的设计、到输入方式的合理组合、以及特定的感知机制。我们已经展示了这些挑战是如何影响系统开发的，尤其是将系统用于实际临床环境中时如何进一步解决这些挑战。而这并不仅仅是直接请求临床医师通过手势来指定他们想要执行的功能的问题。

虽然在设计过程中必须让临床医师参与进来，但这不仅仅是将手势设计交给临床医师的问题，而是涉及到系统开发人员应理解临床团队的工作是如何根据手术需要及物

理环境的特定特征 (如临床团队在患者、同事及设备周围的位置和动作) 来组织的。

开发人员还不能简单地将这些系统看作是用于替代以前无菌影像交互方式，还必须理解临床医师影像交互操作需要实现的目的，以及这些操作是如何由手术在无菌方面的要求所决定的。通过将这一原则与对非接触式系统的技术特征的理解结合在一起，系统开发人员就能通过考虑如何在临床小组成员之间实现影像解释、交流及协调来推动设计。

与此相关的是需要在其投入实际应用时对其进行评估。这里的可用性并不是指改变外科手术小组工作方式，而应该是适应小组的工作，以及适应系统操作过程中的限制因素。此处一项重要考虑就是因长时间使用而导致的疲劳或“手臂僵硬”(这可能影响到系统使用) 以及外科手术实践的其他物理特征。

虽然本文重点关注的是对手术室内无菌要求所带来的限制的克服，但还存在一个更为广泛的问题，即医院环境中的感控问题，这涉及到多个设备、系统及应用 – 从大型显示器到平板计算机等移动装置 - 对于这类设备，非接触式交互机制可能不仅为医疗专业人员所用，而且会为患者所用。GestureNurse 系统[6]便是其中一个有意思的实例，该系统通过手势命令来控制一个机器人手术助理。

另外也可考虑在手术室中进行 3D 成像。通过传统的逐片显示和回放技术来解释扫描技术产生的巨量影像是非常麻烦的。通过对扫描数据进行识别，可将数据越来越多地显示为相关解剖结构的 3D 重构影像，而通过全 3D 交互技术，可以对这些影像进行更好的利用。虽然有许多系统允许对 3D 解剖模型进行操作，但这些系统倾向于利用传统鼠标输入方式可提供的标准二自由度来实现这一目的。在 3D 空间中的手部跟踪和手势动作为手术医生通过六自由度操作影像并与影像交互提供了更多可能。

而且，通过增加 3D 渲染图形的立体显示功能，系统开发人员可进一步解决如何让临床医师能执行新出现的交互操作类型 (如到达解剖模型的内部)。他们还可能考虑非接触式手势交互机制如何为隔着一段距离或超出可达范围 (如在墙壁大小的显示器上和从手术台无法接触到的显示器) 与对象和解剖结构进行交互提供新的可能。因此不仅有机会可实现与传统手术室及显示器进行交互，而且还有机会按新的方式构想未来手术室的整个设计和布局。 C

参考资料

1. Ebert, L., Hatch, G., Ampanozi, G., Thali, M., and Ross, S. Invisible touch:Control of a DICOM viewer with finger gestures using the Kinect depth camera.*Journal of Forensic Radiology and Imaging 1*, 1 (Jan. 2013), 10–14.
2. Ebert, L., Hatch, G., Ampanozi, G., Thali, M., and Ross, S. You can’t touch this:Touch-free navigation through radiological images.*Surgical Innovation 19*, 3 (Sept. 2012), 301–307.
3. Gallo, L., Placitelli, A.P., and Ciampi, M. Controller-free exploration of medical image data:Experiencing the Kinect.In *Proceedings of the 24th International Symposium on Computer-Based Medical Systems* (Bristol, England, June 27–30).IEEE Press, 2011, 1–6.
4. Graetzel, C., Fong, T., Grange, S., and Baur, C. A non-contact mouse for surgeon-computer interaction. *Technology and Health Care 12*, 3 (2004), 245–257.
5. Ionescu, A. A mouse in the O.R.*Ambidextrous, Stanford University Journal of Design 4* (June 2006), 30–32.
6. Jacob, M., Li, Y., Akingba, G., and Wachs, J.P. Gestonurse:A robotic surgical nurse for handling surgical instruments in the operating room.*Journal of Robotic Surgery 6*, 1 (Mar. 2012), 53–63.
7. Johnson, R., O’Hara, K., Sellen, A., Cousins, C., and Criminisi, A. Exploring the potential for touchless interaction in image-guided interventional radiology. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, Canada, May 7–12).ACM Press, New York, 2011.
8. Kang, H., Woo Lee, C., and Jung, K. Recognition-based gesture spotting in video games.*Pattern Recognition Letters 25*, 15 (Nov. 2004), 1701–1714.
9. Kipshagen, T., Graw, M., Tronnier, V., Bonsanto, M., and Hofmann, U. Touch- and marker-free interaction with medical software.In *Proceedings of World Congress on Medical Physics and Biomedical Engineering* (Munich, Sept. 7–12).Springer, Berlin, Heidelberg, 2009, 75–78.
10. Leganchuk, A., Zhai, S., and Buxton, W. Manual and cognitive benefits of two-handed input:An experimental study.*ACM Transactions on Computer-Human Interaction 5*, 4 (Dec. 1998), 326–359.
11. Mentis, H., O’Hara, K., Sellen, A., and Trivedi, R. Interaction proxemics and image use in neurosurgery. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, TX, May 5–10). ACM Press, New York, 2012, 927–936.
12. Microsoft Corp.*Communicate with computers naturally:Kinect for Windows*; http://www.microsoft.com/en-us/kinectforwindows/
13. Mithun, J., Cange, C., Packer, R., and Wachs, J.P. Intention, context, and gesture recognition for sterile MRI navigation in the operating room.In *Proceedings of CIARP 2012:Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Vol. 7441 LNCS* (Buenos Aires, Sept. 3–6, 2012), 220–227.
14. Norman, D. Natural user interfaces are not natural. *ACM interactions 17*, 3 (May–June 2010), 6–10.
15. O’Hara, K., Gonzalez, G., Mentis, H., Sellen, A., Corish, R., and Criminisi, A. Touchless Interaction in *Medical Imaging*.Microsoft Corp., June 2012; http://research.microsoft.com/en-us/projects/touchlessinteractionmedical/
16. Ruppert, G., Amorim, P., Moares, T., and Silva, J. Touchless gesture user interface for 3D visualization using Kinect platform and open-source frameworks.In *Proceedings of the Fifth International Conference on Advanced Research in Virtual and Rapid Prototyping* (Leiria, Portugal, Sept. 28–Oct. 1).Taylor and Francis Group, London, 2011, 215–219.
17. Stern, H., Wachs, J., and Edan.Y. Optimal consensus intuitive hand-gesture vocabulary design.In *Proceedings of the IEEE International Conference on Semantic Computing* (Santa Clara, CA, Aug. 4–7), IEEE Computer Society Press, 2008, 96–103.
18. Strickland, M., Tremaine, J., Brigley, G., and Law, C. Using a depth-sensing infrared camera system to access and manipulate medical imaging from within the sterile operating field.*Canadian Journal of Surgery 56*, 3 (June 2013), E1–6.
19. Tan, J., Chao, C., Zawaideh, M., Roberts, A., and Kinney, T. Informatics in radiology:Developing a touchless user interface for intraoperative image control during interventional radiology procedures.*Radiographics 33*, 2 (Mar.–Apr. 2013), E61–70.
20. Wachs, J., Kolsch, M., Stern, H., and Edan, Y. Vision-based hand-gesture applications.*Commun.ACM 54*, 2 (Feb. 2011), 60–71.
21. Wachs, J., Stern, H., Edan, Y., Gillam, M., Feied, C., Smith, M., and Handler, J. Real-time hand-gesture interface for browsing medical images.*International Journal of Intelligent Computing in Medical Sciences & Image Processing 2*, 1 (June 2008), 15–25.

Kenton O’Hara (keohar@microsoft.com) 是英国剑桥微软研究中心的研究员，兼英国布里斯托大学计算机科学系客座教授。

Gerardo Gonzalez (gerardo.gonzalez_garcia@kcl.ac.uk) 是英国伦敦国王学院生物医学工程系的研究员。

Abigail Sellen (asellen@microsoft.com) 是英国剑桥微软研究中心的首席研究员，兼英国诺丁汉大学计算机科学系名誉教授。

Graeme Penney (graeme.penney@kcl.ac.uk) 是英国伦敦国王学院生物医学工程系的高级讲师。

Andreas Varnavas (andreas.varnavas@kcl.ac.uk) 是英国伦敦国王学院生物医学工程系的研究员。

Helena Mentis (mentis@umbc.edu) 是位于马里兰州巴尔的摩的马里兰大学信息系统系助理教授。

Antonio Criminisi (antcrim@microsoft.com) 是英国剑桥微软研究中心的高级研究员。

Robert Corish (rcorish@microsoft.com) 是英国剑桥微软研究中心的设计研究员。

Mark Rouncefield (m.rouncefield@lancaster.ac.uk) 是位于英国兰卡斯特的兰卡斯特大学计算与通信学院高级讲师。

Neville Dastur (neville@clinsoftsolutions.com) 是位于英国弗雷姆勒的弗雷姆勒公园医院国家医疗服务系统信托基金会的血管手术顾问。

Tom Carrell (tom.carrell@kcl.ac.uk) 是位于英国伦敦的盖圣托马斯医院国家医疗服务系统信托基金会的顾问级血管手术医生，兼英国伦敦国王学院的荣誉高级讲师。

责任编委：田丰

DOI:10.1145/2500887

我们现在知道哪些 40 年前尚不知道的东西?

作者：黄学东、JAMES BAKER、RAJ REDDY

从历史视角看语音识别

随着苹果推出 Siri，谷歌和微软推出类似的语音搜索服务，人们自然想知道，为什么语音识别技术花了这么长时间才发展到这样的水平。同时，我们也想知道，该技术什么时候才有望达到较接近人类水平的性能。1976 年，作者之一 (Reddy) 写了一篇关于当时语音识别最高水平的综述文章。该领域的非专业人士阅读原文会有所收获。[34] 在这里，我们共同从历史视角来阐述语音识别领域的进步。由于篇幅限制，本文将不进行全面的技术评述，而是将范围限定为讨论 40 年前所没有的语音识别技术以及那些帮助解决了一些最棘手问题的进步。

» 主要见解

- 从卡耐基梅隆大学几代研发人员开始，本文对过去 40 年人们从语音识别技术进步所获得的启示进行了探讨。
- 这些年的一些主要成就已被证实可以实际用于苹果、微软等公司的领先行业语音识别系统。
- 语音识别将通过图灵测试，使星际迷航般的移动设备愿景成为现实。这将有助于消除人类与机器之间的隔阂。这将有利于促进和增强人们之间的自然会话。实现这个大胆的梦想需要解决的六项难题。

图片由 SHUTTERSTOCK.COM 提供。

语音识别多年来一直是科幻小说的常见场景，但是在 1976 年其实际水平与虚构世界中那些牵强附会的功能大相径庭。尽管如此，Reddy 大胆预测，未来 10 年内有望实现成本为 20000 美元的联网语音系统。虽然超出了预计时间，但研究人员最终不仅达到了目标，而且建立系统的成本低得多并继续大幅下降。今天，在很多智能手机里，业内提供了明显超出 Reddy 预测的免费语音识别服务。在大多数领域，科幻作家的想象力远远超过现实。语音识别技术是少有的例外之一。语音识别的独特性不仅仅是因为其成就：尽管已有成果斐然，但剩下的难题和目前已克服的一样令人生畏。

1995 年，Windows 95 上首次搭载微软 SAPI，它使应用程序开发者能够在 Windows 上创建语音程序。1999 年，支持电话 IVR 的 VoiceXML 论坛成立。尽管语音电话 IVR 在商业上获得了成功，但事实表明，“语音输入”和“屏幕输出”的多模态隐喻对信息消费更自然。2001 年，比尔盖茨在美国消费电子展 (CES) 上展示了一台代号为 MiPad 的原型机。[16]Mipad 展现了语音多模态移动设备的愿景。随着最近苹果、谷歌和微软在产品中采用语音识别技术，我们正见证着设备处理相对无约束的多模态对话的能力不断提高。尽管仍面临许多困难，我们还是看到了这几十年来研究与开发的成果。我们认为，语音界正在向前迈进，争取未来 40 年通过图灵测试，最终目标是在日常

场景中媲美并超过人类的语音识别能力。

在本文中，我们重点介绍那些实际运用情况良好的主要语音识别技术，并总结了对于将语音识别从移动设备上当前提供的服务推动到下一阶段至关重要的六大困难领域。过去十年内发表的众多技术论文中有更全面的技术讨论，其中包括《*IEEE Transactions on Audio*》（有关音频的汇刊）、《*Speech and Language Processing*》（语音与语言处理）和《*Computer Speech and Language*》（计算机语音与语言），以及 ICASSP、Interspeech 和 IEEE ASRU 研讨会的论文。同时也有大量文章和书籍介绍了过去四十年内研发的各种系统和技术。[9,14,15,19,25,33,36,43]

基础语音识别

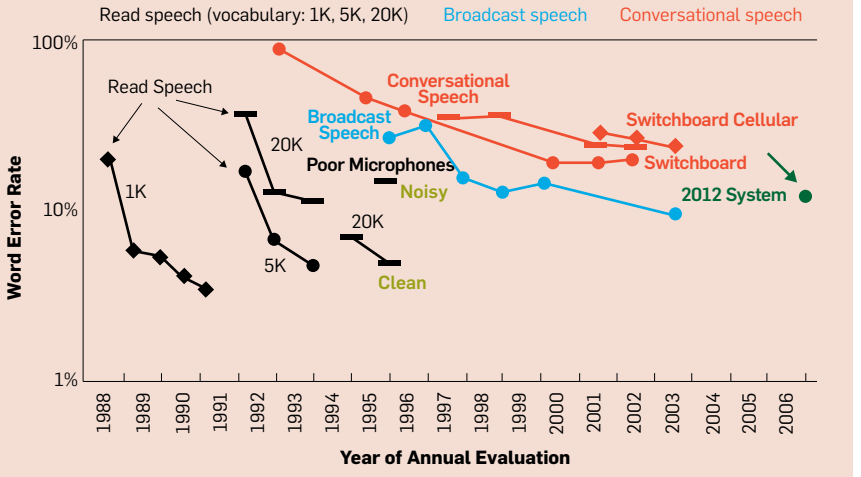
1971 年，由 Allen Newell 领导的一个语音识别研究小组建议引入更多知识来源来解决此问题。报告讨论了六个层次的知识：声学、参量、音素、词汇、语句和语义。Klatt[23] 综述了 ARPA 资助的各种语音理解系统的性能，这些系统是为了实现 NEwell 报告的目标。

国防部高级研究计划署 (DARPA) 赞助了为期多年的语音理解研究 (SUR) 项目，意在探索 Newell 报告中的创意。得到资助的研究小组不多，Reddy 1976 年在卡耐基梅隆大学领导的小组是其中之一。该小组开发了一系列的语音识别系统：Hearsay、Dragon、Harpy 和 Sphinx I/II。经过四十年时间，Reddy 和同事们创造了一些历史性的口语系统演示。例如，机器人的语音控制，大词汇量的联网语音识别，说话者无关的语音识别和无限制词汇听写。Hearsay-I 是首批有能力进行连续语音识别的系统之一。Dragon 系统是首批将语音建模为隐随机过程的系统之一。Harpy 系统引入了定向搜索 (Beam Search) 概念。几十年来，定向搜索一直是高效搜索和匹配中运用最广泛的技术。1987 年研发的 Sphinx-I 是最早演示说话者无关的语音识别系统。1992 年研发的 Sphinx-II 在同年 DARPA 资助的语音基准评测中获得了最高的识别准确度，这主要得益于其在高斯混合和马尔可夫状态层次上用栓连参数平衡了可训练性和高效性。

根据 DARPA 资助的多次语音评测，语音识别词错误率已经是评估进步的主要指标。如图 1 所示，历史性的进步也引导业内人士致力于解决更困难的语音识别任务。在最新的电话总机任务中，通过采用多伦多大学和微软的研究人员率先提出的深度学习框架，[5,14] 微软和 IBM 的研究人员分别将词错误率降低到了一个新的里程碑。[4,22,37]

上世纪 70 年代初，人们预计产生更高层次的知识来源需要人工智能方面有重大突破。按照 Hear-

图 1. 在难度不断提高的任务中，语音识别的词错误率取得了历史性进步。[10] 标有绿点的是用于电话总机任务的最新系统。



1976 年我们不知如何解决的问题。

统计建模和机器学习： 设计隐马尔可夫模型 (HMM)、语境相关的音素建模、统计平滑和回退策略、DNN、半监督学习、最大互信息估计 (MMIE) 和 MPE 等判别性训练

训练数据和计算资源： 随着分布式 CPU 和 RAM 资源的稳步增长，语音规模（数千小时）和文本数据（数万亿字）增加了几个数量级。

应对嘈杂环境的信号处理： DNN 习得特征、适合高斯混合模型的 MFCC、适用于 DNN 的滤波器组等低级原始特征、倒谱均值相减法、第一阶和第二阶 Δ 特征、在线环境的适应以及降噪麦克风 / 麦克风阵列

词汇量和不流利的语音： n 元和 RNN 语言模型支持的数千至数百万单词、显式垃圾模型、灵活添加带字素形式的新单词

说话者无关和自适应语音识别： 混合分布、不同方言和人群的说话者训练、声道归一化、最大后验概率 (MAP)、最大似然线性回归 (MLLR) 和无监督说话者自适应学习

高效的解码器： 时间同步的 Viterbi 搜索和具有复杂剪枝技术的 A 星堆栈解码器 *，支持基于服务器的大规模运行时解码器的分布式实施

口语理解与对话： 基于格框架的稳健分析器、半马尔可夫条件随机场 (CRF)、提升决策树、基于规则或者马尔可夫决策过程的对话管理、用于理解句子的递归神经网络

say 系统的体系结构设计，许多半自治模块既能在一项语音识别任务中相互沟通和合作，也能分别专注于自己的专业领域。相比之下，Dragon、Harpy 和 Sphinx I/II 系统全都是基于单一且相对简单的联合全局优化建模原则。Newell 报告中的每一个层次都由一个称为“隐马尔可夫过程”的随机过程表示。从概念上讲，连续层次就像嵌套分程序一样嵌套，所以组合过程同样也是一个（非常大的）隐马尔可夫过程。[2]

寻找最佳匹配单词序列 W 以匹配输入语音 X 的解码过程远不是一个简单的模式识别问题，因为它面临着搜索数量近乎天文数字的单词模式。上述解码过程是寻找一个单词序列，其对应的声学模型和语言模型最匹配输入特征向量序列。因此，用经过训练的声学模型和语言模型进行解码的过程通常被称为搜索过程。图搜索算法在人工智能、运筹学和博弈论领域得到了广泛研究，它也是语音识别中的搜索问题的基础。

解码过程的重要性在 Dragon NaturallySpeaking 中得到了最好的诠释。该产品是在作者之一 (Baker) 领导下历时 15 年开发完成的。被 Nuance 收购后，它历经一代又一代的计算机技术变革，存活了 15 年。Dragon Systems 的成功并非由于发明了性能优越的全新算法。Dragon NaturallySpeaking 的技术发展类似于本文回顾的同期总体发展。最显著的差别不是错误率更低的算法，而是着重于更好地平衡成本与性能的简化算法。从成立开始，Dragon Systems 的长期目标就是开发一款实时、大词汇量的连续语音听写系统。为此，Dragon 制定了持续数十年的一贯企业使命，达到最终目标需要这一使命，但每个时间段都会体现为适当的短期目标和中期目标：开发最好的语音识别系统，使其能够实时运行在当代桌面电脑上。

1976 年我们所不知道的

Reddy 最初的综述文章中阐述的每个组件都取得了巨大进步。我们不打算一一列举出过去几十年内发明的各种系统和方法。表 1 列出了经证实在行业领先的语音识别系统中行之有效的主要成就。如今，我们能够使用 HTK、Sphinx、Kaldi、CMU LM 工具包和 SRILM 等开放性的研究工具来搭建一个可运行的系统。然而，行业中的竞争优势主要源自使用云端提供的大量数据来不断更新和改进声学模型和语言模型。本文讨论了催生手机语音搜索的技术进步，比如图 2 所示的苹果、谷歌和微软语音搜索。

依托强大的计算基础设施和大量训练数据建立的统计机器学习框架，构成了促进语音识别发展的最主要力量。这使机器学习能统一处理音素、单词、语法和语义知识表示。例如，语音字符串的显式分割和标记不再必不可少。语音匹配和单词检验与单词序列生成得到了统一，后者依赖于通常使用语境相关的语音声学模型得到的最高综合评分。

统计机器学习。早期的语音识别方法的目标是从一组离散的标签中找到最接近的匹配声音标签。在非概率模型中，根据对两个声音相似性的估计来设定声音标签之间的估计“距离”。在一种形式中，概率模型以正确的标签是假设标签的概率（也被称为“混淆”概率）为条件，使用观察特定声音标签作为最佳匹配标签的条件概率估计。相比估计高斯分布的平均值（另一种常见表示），估计每个可能的声音与每个可能的标签发生混淆的概率所需的训练数据多得多。该方法对应 Reddy 1976 年综述文章中所描述的“分割与标记”中的“标记”部分，无论是否伴随分割，都是 1980 年代时基于非概率的模型常采用的做法。这个距离可能仅仅是需要最小化的得分。

Reddy 发表前述综述文章时，语音识别中的知识表示才刚刚开始迎来重要转变。这一变化的例子是将语音表示为隐马尔可夫过程。我们通常用首字母缩写 HMM 指代“隐马尔可夫模型”，这有点用词不当。因为隐的是过程而

图 2。必应和谷歌等现代搜索引擎都提供了易于访问的话筒按钮（红色标记）以使用语音搜索网页。苹果 iPhone 的 Siri 虽然不是搜索引擎（其网络搜索现在由 Bing 提供），但它有一个大得多的麦克风按钮用于进行多模态语音对话。

不是模型。[2] 从数学上看，隐马尔可夫过程的模型有一个名为期望最大化 (EM) 算法的学习算法，它具有广泛适用的收敛定理。[3,8] 在隐马尔可夫过程的特定情况下，通过 Forward-Backward 算法可以得到一种非常高效的实现。1980 年代末以来，人们还在最大互信息或相关最小错误准则的基础上发明了统计判别训练技术。[1,13,21]

2010 年以前，基于 HMM 的高斯模型混合通常是最先进的语音识别系统采用的技术。这些模型采用的特征通常是梅尔频率倒谱系数 (MFCC)。[6] 尽管人们开展了许多工作创建模仿人类听觉过程的特征，我们要强调通过引入深度神经网络 (DNN) 提供习得特征表示这一重要发展。DNN 解决了用高斯混合模型进行数据表示的低效问题，能够直接取代高斯混合模型。[14] 深度学习还能用于为传统 HMM 语音识别系统学习强大的判别性特征。[37] 该混合系统的优势是，能够直接使用语音识别研究人员几十年来研发的各种语音识别技术。相较于早期的一些工作，[29,40]DNN 和 HMM 相结合大大减少了错误 [4,14,22,37]。在新系统中，DNN 的语音类通常由捆绑 HMM 状态表示一这是一种直接继承了早期语音系统的技术。[18]

使用马尔可夫模型表示语言知识存在争议。语言学家确信，自然语言无法用上下文无关语法表示，更不用说用有限状态文法表示。同样，人工智能专家更加怀疑马尔可夫过程这样简单的模型能否用来表示 Newell 报告提到的更高层次的知识来源。

然而，假设语言本身是马尔可夫过程和将语言建模成隐马尔可夫过程的概率函数有着根本区别。后一模型是一种近似方法，它并不对语言做出假设，而是为设计者选择在隐过程中要表示什么提供一种解决方案。马尔可夫过程的确切属性是，给定当前状态时，未来事件的概率独立于该过程中过往的其他额外信息。此属性意味着，如果有任何关于被观察过程历史的信息（如观察到的单词和子词单元），则设计者应该在隐过程中以不同的状态为该信息编码。事实证明，Newell 层次结构的每一层都可以以合理的近似程度表示为一个隐马尔可夫过程的概率函数。

对于如今最先进的语言建模，大多数系统仍然使用统计 *N* 元语言模型及其变体，并用基本计数技术或 EM 类技术加以训练。经证明，这些模型非常强大且富有弹性。然而，*N* 元是实际人类语言的高度简化模型。与深度学习大大提高声学建模质量相似，递归神经网络也明显改善了 *N* 元语言模型。[27] 值得一提的是，对于大多数真实的语音应用，比适配应用领域的大规模文本语料库更重要的了。

训练数据和计算资源。由于语音数据和文本数据增多，计算能力提高，语音识别研究人员得以为规模足够大的任务开发和评估复杂算法。用于语音训练、开发和评估的常用语音语料库对创建功能不断增强的复杂系统起到了关键作用。因为语音是高变异性信号且需要许多参数描述，所以对于建立足够好的模型使自动化系统达到熟练程度，大型语料库显得至关重要。多年来，这些语料库已由美国国家标准和技术研究院 (NIST)、美国语言数据联盟 (LDC)、欧洲语言资源协会 (ELRA) 和其他组织创造、注释并分发给全球业内人士。录音的特点已经从有限的约束语音素材发展到大量日益真实自发的语音。

摩尔定律预测，给定成本的计算量每 12-18 个月会翻一倍，内存价格也会下降一半。摩尔定律使得语音识别能够利用到性能大大提升的计算基础设施。云语音识别技术使得积累超大规模语音数据比 1976 年所能想象到的更加方便。谷歌和 Bing 都编制了整个网络的索引。网络搜索引擎每个月会收到数十亿次用户查询。如此庞大的查询点击数据使得为语音搜索应用程序创建更强大的语言模型成为可能。

信号和特征处理。每个声学特征矢量通常每 10 毫秒计算一次。每一帧都会有选取一个短暂的语音数据窗口。通常每个窗口选取 25 毫秒的语音，所以语音窗口在时间上是有重叠的。1976 年，声学特征通常是测量每个时间窗口内各个频率的幅值，通常用快速傅里叶变换或者滤波器组来计算。幅值是频率的函数，叫做短暂语音时间窗口的“频谱”，发音时间内的此类频谱序列能够被可视化为声谱图。[31]

过去的 30 年左右，尽管修改声谱图造成了原始语音信息的损失，但也大大提升了基于高斯混合模型的 HMM 系统的性能。深度学习技术正是以最大限度地减少这些信息损失为目标，并旨在从原始数据中搜索更强大的、由深度学习驱动的语音表示。由于深度学习的成功，语音识别研究人员重新开始使用更多基础语音特征（比如声谱图和滤波器组）进行深度学习，[11] 这使得机器学习能够利用深度神经网络技术本身自动发现更多有用的表示方式。[37,39]

词汇量。从 1976 年以来，大型语音识别系统的最大词汇量已经大幅增加。事实上，1990 年代末实时自然语言听写系统的词汇量基本已经达到无限。也就是说，用户并不知道系统的词典中相对罕见的单词哪些有，哪些没有。

系统尝试识别听写的每一个单词，并将所有未识别的单词算作错误，即使这个单词不在词典里。

这种观点迫使这些系统不停学习新单词，以便系统每次再碰到同样的单词不会继续犯错。学习特定用户的口述中重复出现的人名和地名尤其重要。从单个或少数示例中学习的统计学习技术取得了显著进步。技术人员使这个过程对交互用户显得尽可能无缝。然而，这个问题仍然是个挑战，因为从模型的角度看，小样本模型与大数据模型完全不同，为新单词建模仍然远远未达到无缝的程度。

说话者无关的自适应系统。尽管采用统计机器学习的概率模型为多种语音信号变异来源的建模和学习提供了一种方式，单个说话者、说话者相关模型和针对多样化人口的说话者无关模型之间仍然有明显的性能差距。Sphinx 引入了大词汇量、说话者无关的连续语音识别。[24] 关键是使用来自大量说话者的更多语音数据训练基于 HMM 的系统。

适应性学习也被用于适应说话者差异和广泛的通道、噪音和领域的变化条件。[24] 有效的适应技术使我们能够进行快速的应用程序集成，并且也是成功进行语音识别商业部署的关键。

解码技术。从架构上看，知识表示的最重要发展是可搜索的统一图表示。它使得多种知识来源能够汇集到一个共同概率框架中。Reddy 1976 年的论文中总结的诸多系统已经演化出多种解码或搜索策略，比如堆栈解码（A 星搜索）、[20] 时间同步定向搜索 [26] 和加权有限状态传感器 (WFST) 解码器。[28] 这些实用的解码算法使得大规模的连续语音识别成为可能。

非组合法包括在 ROVER [12] 以及增加约束的多路系统等假设层次上结合的多语音流、多概率估计量，多识别系统。

语音识别的独特性不仅仅是因为其成就：尽管取得了所有这些成就，但剩下的难题和目前已克服的一样令人生畏。

口语理解。获得识别结果后，从识别结果中提取“意思”同样重要。1970 年代，口语理解 (SLU) 主要依靠表示语义概念集的格语法。DRAPA 资助的航空旅行信息系统 (ATIS) 研究计划是将格语法用于 SLU 的一个好例子。[32,41] 在这项任务中，用户可以随意语音查询航班信息。口语理解需要从给定的、基于框架的语义表示中提取出特定任务的参数，其中，框架可以是“出发时间”、“航班”等。这些格框架中的槽是涉及的领域特有的。从语音识别结果寻找属性值的过程必须稳健，能处理内在识别错误以及表示同一概念的多种不同表达方式。

人们使用了许多技术来填充训练数据中的应用领域的框架槽。[30,35,41] 与声学建模和语言建模类似，基于递归神经网络的深度学习也能够明显改进语言理解的槽填充。[38]

六大主要难题

语音识别技术远不完美。事实上，技术难题比比皆是。根据过去 40 年的经验，我们现在探讨实现语音识别梦想必须应对的六个最困难领域。

数据太多好比无数据。现在，我们有一些非常令人兴奋的机会来收集大量数据，从而产生了“数据洪流”。很大程度由于的互联网功劳，现在可以轻易获得大量日常语音，反映以往无法获得的各种材料和环境。最近兴起的手机语音搜索提供了丰富的语音数据来源，由于对手机用户操作的记录，这些数据可视为部分“标记”了的。苹果 Siri（Nuance 提供支持）、谷歌和微软都已经通过其产品的语音系统积累了大量用户数据。

一些基于 Web 的新工具可以用来以可控的成本收集、标注并处理许多语言的海量语音。在网络上感兴趣的人士齐力协助下，可以非常有效、廉价地生成大量语言资源。对于为资源“稀缺”的语言创造显著的新功能，这尤其弥足珍贵。

日益增加的数据量对于提高语音识别技术的最新水平既是机遇又是挑战，如图 3 所示，我们的微软同事 Li Deng 和 Eric Horvitz 使用了许多发表的论文中的数据来证明这一重要论点。即便我们尽最大努力从分散在近 10 年的数据得出一幅结构严谨的图，图 3 中的数字依然并不精确。

在抽样人们经常经历的多种语音、环境和信道方面，我们才仅仅进行了肤浅的研究。事实上，我们目前提供给自动系统的资料，与我们人类用来学习语言的资料相比，只占其很小的比例。若要使我们的系统更加强大并理解语音的本质，我们需要更充分地利用语音并标注更多的语音资料。标签完善的语音语料库已经成为当前语音系统发展和进化的基石。但是，大多数的海量数据都没有标签或标注不善，而准确地标注它们成本不菲。

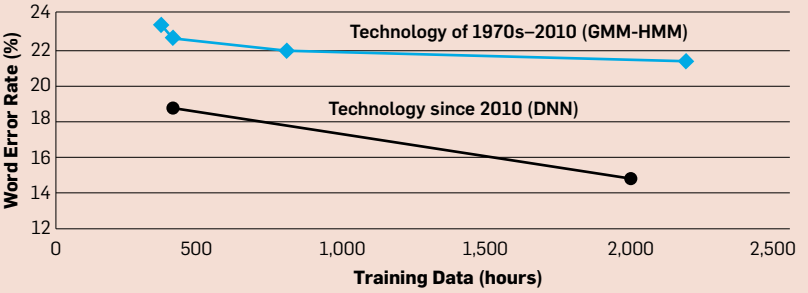
计算基础设施。GPU[5,14] 的使用是近年来一个显著的进步，它使中等规模的深层网络训练成为现实。GPU 方法的一个已知局限是，当模型与 GPU 内存（通常小于 6 GB）不匹配时，训练速度提升较小。据最近报道，分布式优化方法可以大幅提高深度学习速度并可训练更大的模型。[7] 大规模分布式计算机集群已被用于训练规模适中的语音深度神经网络 (DNN)，相比 GPU 实现方式，其速度提高了 10 余倍。

几十年来，摩尔定律一直是计算系统的计算能力和存储能力提高的一个可靠指标。这对语音识别和理解系统产生了巨大的影响，包括允许使用越来越大的训练数据库和识别系统，并整合更精细的口语模型。鉴于采用分布式计算机系统训练大规模 DNN 的最新进展，这似乎证明，未来的许多研究方向和应用隐式依赖于计算能力的不断提高。如图 3 所示，随着训练数据不断增加，即使用大规模分布式计算集群，训练一个新型语音系统预计也需要数周或数月。

英特尔和其他人最近指出，微处理器的功率密度提高到了极点，再提高时钟频率将会使硅开始熔化。因此，行业发展目前专注于实现多核微处理器。半导体行业的新路线图反映了这一趋势，未来的加速将更多地来自并行计算而不是单个更快的计算元件。

在大多数情况下，语音系统的算法设计者都忽略了对并行计算的研究，部分原因是可扩展性的进步一直非常可靠。未来的研究方向和应用程序将需要多得多的计算资源用于创建模型，因此研究人员将需要在其设计中考虑大规模分布式并行计算。这将是现状的一个显著变化。特别是，对于解码等任务，人们已经开发了极其聪明的方案来提高单处理器的性能，这些任务将需要完全重新思考算法。显式利用并行计算的新搜索方法应该是一个重要的研究方向。

图 3. 数据太多好比无数据。识别词错误率对比训练小时数量（仅供说明）。此图说明了增加训练数据可增强现代语音识别系统。



无监督学习已被成功用于训练一个比先前报道大 30 倍的深度网络。[7] 通过监督微调获得标签，基于 DNN 的系统在 ImageNet 这项非常困难的视觉对象识别任务中取得了最高性能。对于语音识别，用云端的大量用户交互数据（如网络搜索引擎中的点击数据）开发高品质的无监督或半监督技术也有实际需要。

语音搜索的成功开发使得利用未标记或部分标记的数据训练基本声学和语言模型变得可行。我们可以自动（并“主动”）按效用最大化的方式选择部分未标记的部分数据进行人工标记。采用无监督学习的一个重要原因是，和他们的人类“基准”一样，系统将不得不接受“终身学习”，适应不断发展的词汇、通道、语言运用等等。有必要在所有层次上学习应对不断变化的环境、说话者、发音、方言、口音、词语、意义和话题。与人类一样，系统将进行自动模式发现、主动学习和适应。

我们必须解决新模型的学习以及将此类模型集成到现有系统中这两大问题。因此，学习的一个重要方面就是，要能辨别何时已学会一

些知识以及如何运用学习的结果。从多个并发的模态学习也可能是必要的。例如，语音识别系统可能会在其输入语音中遇到新的专有名词，而且可能需要检查文本语境正确地确定名称的拼写。多模态无监督学习研究领域的成功将延长已部署系统的使用寿命，通过创建一个随着时间推移自动适应并改进的系统，直接提高我们无需大量昂贵的人工标记数据的情况下开发适用于新语言和新领域的语音系统的能力。

可移植性和泛化能力。学习的一个重要方面是泛化。当只有少量测试数据可用来调节语音识别器时，我们称之为泛化适应。适应能力和泛化能力使得快速语音识别应用的集成得以实现。如果可以获得训练数据，也有人尝试使用部分可观察的马尔可夫决策过程改进对话管理。[42] 对于许多新语言或新任务，往往难以获得这套语言资源。事实上，获得与该领域严密匹配的大量训练数据也许是使语音系统得到实际运用唯一最可靠的方法。

过去三十年中，语音界开发和完善了有助于促进语音技术稳步改进的实验方法。该行之有效的方法是开发共享语料库、软件工具和指南，它们可用于将实验设置之间的差异归结于算法，从而使量化根本改进变得更容易。通常情况下，这些语料都专注于特定任务。不幸的是，目前的语言模型不容易移植到不同的任务，因为它们缺乏语言学“头脑”，无法准确区别有意义的句子和无意义的句子。另外，它们也未考虑篇章结构，仅仅只涉及局部词语搭配。

这种策略与人类经验完全不同。我们一生中要从不受控制的环境、说话者和话题（也就是日常对话）中接触各种语音数据。尽管我们自己的个人训练数据如此多变，但我们能够创建非常善于应对语音变化的内部语音和语言模型。这种泛化能力是人类语音处理的一个关键方面，而现代语音系统目前尚未找到实现这种能力的办法。关于这一主题的研究活动发明的技术应该能在新环境下更有效地运行，并且能更好地从较少的数据进行泛化。另一个研究领域则可以探索如何将来自资源丰富的语言和 / 或领域的信息更好的推广到资源匮乏的语言和领域。

此处的难题是发明可迅速移植的口语技术。为了快速开发此类口语系统，我们需要新的范式来研究比特定于某种语言的音素更具语言普适性的语音和声学单元。有三个具体的研究问题必须解决：面向新目标语言的语音和声学单元的跨语种声学建模；针对新语言单词发音的跨语种词汇建模，以及跨语种语言建模。探索新语言和经过充分研究的语言之间的相关性将有助于提高快速移植和泛化能力。从少量标记话语建立初步系统，用其以无监督的方式标记更多的话语样本，迭代改进系统，直到其达到与如今高准确度系统相当的性能水平，在此过程中，自举技术是关键。

不确定性的处理。已经考验的统计 DNN-HMM 学习框架需要大量数据来处理不确定性。如何识别和处理多种变化因素是建立成功的语音识别系统的关键。尽管过去几十年中的进步令人印象深刻，但即便是遇到人类听者认为难度很低或毫无难度的细微偏差，现在的语音识别系统的性能仍会大幅下降。语音识别的鲁棒性仍然是一个重大的研究难题。我们希望不仅算法有所突破，而且在日益增多的无监督训练数据的使用方面有所突破。现在可以用以往不可行的方式获取这种数据。

语音信号中的一种普遍存在的变化因素是声学环境。这包括背景噪声、室内混响、语音的获取通道（例如蜂窝网络、蓝牙、固定电话和 VoIP）、重叠语音、Lombard 语音或超清晰语音。对于导致系统性能急剧下降的有害变异，采集语音的声学环境和传输语音信号的通信信道是重要原因。现有技术能够减少因加性噪声或线性失真导致的变异，并补偿缓慢变化的线性通道。然而，较复杂的通道失真，例如混响或快速变化的噪声，以及 Lombard 效应构成了重大挑战。尽管深度学习使得自动编码可以创造更强大的特征，但我们期望在学习有用的特征方面有更多的突破。这种学习可能模仿也可能并不模仿人类听觉系统。

人们深入研究的另一种常见语音变异是由于不同讲话人的特点造成的。众所周知，由于讲话者的生理机能、风格和口音（地方口音和非母语口音）等多种因素，不同讲话者的语音特征差异巨大。目前开发更健壮的语音识别系统所采用的主要方法是在训练中包括范围广泛的讲话者（和讲话风格），以便能够处理讲话人特征上的差异性。此外，目前的语音识别系统采用的发音词典建模的是某种语言的母语讲话者，并用不同母语讲话者的大量语音数据进行训练。人们已在探讨为带口音的语音建模，包括带口音的语音的显式建模，不大成功的母语声学模型适应，例如在苏格兰部署英式英语语音系统最初就遇到一些困难。发音变体也已收录进词典，但收效甚微。同样，检测语速变化方面也进展缓慢。

拥有苏格拉底的智慧。与大多数古希腊人一样，语音识别系统缺乏苏格拉底的智慧。这里的难题是搭建能可靠地检测其何时

不认识（正确）某个词语的系统。发生此类错误事件的迹象是纯感觉信号（如无约束电话识别）分析与单词或短语级假设的不匹配。其中，前者由先验知识支配，后者则基于更高级别的知识，并通常以语言模型编码。这项研究的一个关键组成部分是基于感官证据与先验信念之间的差异开发新颖的置信度量和精确的不确定性模型。检测此类事件后自然是用音标记录这些事件（当系统确信其单词假设不可靠时）并制订纠错方案。

当前系统难以处理不经常出现的（因而往往信息最丰富）词项。对于包含感叹词、外来词或词汇表以外的词的语音以及只有相对较少的数据用来建立系统的词汇词典和发音词典的语言，这尤其成问题。这种情况的常见结果是高价值术语被过度自信地误识为其他常见单词或发音相似的单词。然而，此类口语事件对于口语术语检测和从语音中提取信息之类的任务非常重要。因此，准确检测它们至关重要。

在大多数情况下，语音系统的算法设计者都忽略了对并行计算的研究，部分原因是可扩展性的进步已经非常可靠。

结论

过去四十年，语音识别技术迎来了许多突破，为以前不可能完成的任务提供了解决方案。在这里，我们将总结研究和产品开发的不断进步所带来的启示。

1976 年，计算能力只够执行对有较少分支因素（疑难）的高度受限任务进行语音识别。如今，我们能够处理分支因素多得多且近乎无限的词汇。1976 年，用于常规语音研究的最快的计算机是一台 4MB 内存的专用 PDP- 10。现在的系统能够获得一百万倍多的计算能力用于训练模型。数千颗处理器和云端几乎无限的总内存容量得到了日常应用。这些系统可以使用从开放性人群中的数百万人收集的以百万小时计的语音数据。这些系统的力量主要源自其收集、处理庞大数据集并从中学习的能力。

基本的学习和解码算法 40 年内并未发生重大变化。当然，人们也提出了许多算法改进，比如如何在深度学习任务中使用分布式算法。出人意料的是，尽管 iPhone 之类的智能手机可能有足够的计算能力和内存，但语音识别目前似乎是在远程服务器上完成，iPhone 几百毫秒内就能获得结果。这样的机制却使得有潜力将错误率降低一半的说话人和环境自适应技术变得难以实施。

处理以前未知文字对大多数系统问题仍成问题。以基于 Web 的分析为基础收集海量词汇使得用户很有可能会使用其中一个已知单词。现在的网络搜索引擎商存储了 5 亿多实体条目，它们可大幅扩大词汇量，对于语音识别而言，词汇量通常小得多。用于网络搜索引擎的社交图谱也可用于大幅减少所需的搜索空间。最后一点是，混合语种的语音，其中来自两种或更多语言的短语混合使用，使得新单词的问题变得更加困难。[17] 这种情况通常出现于许多英语夹杂母语的国家。

检错和纠错的相关问题导致了复杂的用户界面选择，在这方面，“Dragon Naturally Speaking”和后续系统已采用足够好的解决方案。我们认为，如 MiPad 演示 [16] 和类似苹果 Siri 的服务所示，多模态交互式隐喻将成为占主导地位的隐喻。对于系统此前未知的新单词，我们仍然缺少类似人类为弄清其含义而进行的对话。

另一个相关问题是识别高度易混淆的单词。此类系统需要使用更强大的辨别学习。在大多数依赖以大数据为基础的统计技术

的系统中，也没有类似人类经常进行的动态稀疏数据学习。

未来 40 年，语音识别将通过图灵测试。这将真正使星际迷航般的移动设备愿景成为现实。我们预期语音识别技术可帮助缩小消除我们与机器之间的隔阂。正如 Rick Rashid 展示的《*纽约时报*》新闻 [a] 英汉语音翻译演示 [b] C一样，不管是地理位置障碍还是语言障碍，它都将是促进和增强人们之间自然对话的强大工具。

a http://nyti.ms/190won1
b https://www.youtube.com/watch?v=Nu-nlQqFCKg

参考文献

1. Bahl, L. et al.Maximum mutual information estimation of HMM parameters.In *Proceedings of ICASSP* (1986), 49–52.
2. Baker, J. Stochastic modeling for ASR. *Speech Recognition.*D.R.Reddy, ed. Academic Press, 1975.
3. Baum, L. Statistical Estimation for Probabilistic Functions of a Markov Process.*Inequalities* III, (1972), 1–8.
4. Chen, X., et al.Pipelined back-propagation for context-dependent deep neural networks.In *Proceedings of Interspeech*, 2012.
5. Dahl, G., et al.Context-dependent pre-trained deep neural networks for LVSR. In *IEEE Trans.ASLP 20*, 1 (2012), 30–42.
6. Davis, S. et al.Comparison of parametric representations.*IEEE Trans ASSP 28*, 4 (1980), 357–366.
7. Dean, J. et al.Large scale distributed deep networks.In *Proceedings of NIPS* (Lake Tahoe, NV, 2012).
8. Dempster, et al.Maximum likelihood from incomplete data via the EM algorithm.*JRSS 39*, 1 (1977), 1–38.
9. De Mori, R. Spoken Dialogue with Computers.Academic Press, 1998.
10. Deng, L. and Huang, X.(2004). Challenges in adopting speech recognition.*Commun.ACM 47*, 1 (Jan. 2004), 69–75.
11. Deng, L. et al.Binary coding of speech spectrograms using a deep auto-encoder.In *Proceedings of Interspeech*, 2010.
12. Fiscus, J. Recognizer output voting error reduction (ROVER).In *Proceedings of IEEE ASRU Workshop* (1997), 347–354.
13. He, X., et al.Discriminative learning in sequential pattern recognition.*IEEE Signal Processing 25*, 5 (2008), 14–36.
14. Hinton, G., et al.Deep neural networks for acoustic modeling in SR.*IEEE Signal Processing 29*, 11 (2012).
15. Huang, X., Acero, A., and Hon, H. *Spoken Language Processing.*Prentice Hall, Upper Saddle River, NJ, 2001.
16. Huang, X. et al.MiPad:A multimodal interaction prototype.In *Proceedings of ICASSP* (Salt Lake City, UT, 2001).
17. Huang, J. et al.Cross-language knowledge transfer using multilingual DNN.In *Proceedings of ICASSP* (2013), 7304–7308.
18. Hwang, M., and Huang, X. Shared-distribution HMMs for speech.*IEEE Trans S&AP 1*, 4 (1993), 414–420.
19. Jelinek, F. *Statistical Methods for Speech Recognition.*MIT Press, Cambridge, MA, 1997.
20. Jelinek, F. Continuous speech recognition by statistical methods.In *Proceedings of the IEEE 64*, 4 (1976), 532–557.
21. Katagiri, S. et al.Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method.In *Proceedings of the IEEE 86*, 11 (1998), 2345–2373.
22. Kingsbury, B. et al.Scalable minimum Bayes risk training of deep neural network acoustic models.In *Proceedings of Interspeech* 2012.
23. Klatt, D.H.Review of the ARPA speech understanding project.*JASA 62*, 6 (1977), 1345–1366.
24. Lee, C. and Huo, Q. On adaptive decision rules and decision parameters adaption for ASR.In *Proceedings of the IEEE 88*, 8 (2000), 1241–1269.
25. Lee, K. *ASR:The Development of the Sphinx Recognition System.* Springer-Verlag, 1988.
26. Lowerre, B. The Harpy Speech Recognition System.(2004).Carnegie Mellon University.
27. Mikolov, T. et al.Extensions of recurrent neural network language model.In *Proceedings of ICASSP* (2011), 5528–5531.
28. Mohri, M. et al.Weighted finite state transducers in speech recognition. *Computer Speech & Language 16* (2002), 69–88.
29. Morgan, N. et al.Continuous speech recognition using mulitlayer perceptions with Hidden Markov Models.In *Proceedings of ICASSP* (1990).
30. Pieraccini R. et al.A speech understanding system based on statistical representation.In *Proceedings of ICASSP* (1992), 193–196.
31. Potter, R., Kopp, G. and Green, H. *Visible Speech.* Van Nostrand, New York, NY, 1947.
32. Price, P. Evaluation of spoken language systems:The ATIS domain.In *Proceedings of the DARPA Workshop*, (Hidden Valley, PA, 1990).
33. Rabiner L. and Juang, B. *Fundamentals of Speech Recognition.*Prentice Hall, Englewood Cliffs, NJ, 1993.
34. Reddy, R. Speech recognition by machine:A review.In *Proceedings of the IEEE 64*, 4 (1976), 501–531; http://www.rr.cs.cmu.edu/sr.pdf.
35. Seneff S. Tina:A NL system for spoken language application.*Computational Linguistics 18*, 1 (1992), 61–86.
36. Tur, G., and De Mori, R. *SLU:Systems for Extracting Semantic Information from Speech.* Wiley, U.K., 2011.
37. Yan, Z., Huo, Q., and Xu, J. A scalable approach to using DNN-derived features in GMM-HMM based acoustic modeling for LVCSR.In *Proceedings of Interspeech* (2013).
38. Yao, K. et al.Recurrent neural networks for language understanding. In *Proceedings of Interspeech* (2013), 104–108.
39. Yu, D. et al.Feature learning in DNN—Studies on speech recognition tasks. *ICLR* (2013).
40. Waibel, A. Phone recognition using time-delay neural networks.*IEEE Trans. on ASSP 37*, 3 (1989), 328–339.
41. Ward, W. et al.Recent improvements in the CMU SUS.In *Proceedings of ARPA Human Language Technology* (1994), 213–216.
42. Williams, J. and Young, S. Partially observable Markov decision processes for spoken dialog systems.*Computer Speech and Language 21*, 2 (2007), 393–422.
43. Zue, V. The use of speech knowledge in speech recognition.In *Proceedings of the IEEE 73*, 11 (1985), 1602–1615.

黄学东是华盛顿州雷德蒙市微软 Bing Core Search 团队的杰出工程师。他于 1993 年在那里成立了微软的语音技术小组。他曾任教于卡耐基梅隆大学。

James Baker 是马萨诸塞州牛顿市 Dragon Systems 公司的前董事长、首席执行官兼联合创始人。他在卡内基梅隆大学取得了博士学位。

Raj Reddy 是卡内基梅隆大学（位于宾夕法尼亚州匹兹堡市）计算机科学与机器人学 Moza Bint Nasser 大学教授。他于 1969 年加入卡内基梅隆大学。

责任编委：山世光

第106页

技术视角

硅应力

作者：Subramanian S. Iyer

第107页

考虑硅通孔（TSV）应力的三维集成电路（3D IC）全芯片机械可靠性分析及优化方法

作者：Moongon Jung、Joydeep Mitra、潘志刚与 Sung Kyu Lim

DOI:10.1145/2548017

技术视角

硅应力

作者：Subramanian S. Iyer

摩尔定律预测晶体管密度每两年左右增加一倍。1949 年 Shockley 等人发明晶体管，19 世纪 50 年代 Kilby 将其应用于集成电路。自此以后，半导体电子产品变得无所不在，移动革命改善了人们的生活，而摩尔定律在此过程中充当了重要基础。

过去几十年里，我们一直在按照 Dennard 提出的恒定电场缩小规则不懈地降低器件特征尺寸，并成功地使我们的芯片变得速度更快，体积更小，价格更低。这种演进隐含的假设是，我们可以经济地印制这些电路。但这个假设现在面临质疑，因为我们达到的芯片特征尺寸已经明显低于用于印制它们的激光的分辨率。虽然我们采用了各种技术来印制这些亚波长器件，但付出的代价是，降低单位电路成本的这一预期可能要落空。事实上，有些人预测每个功能的成本将增加，这自然会引出这样一个问题："为什么要进一步缩小尺寸？"

三维集成 (3Di) 减轻了一些这方面的压力。有必要指出，3Di 本身不能制造更快或更廉价的晶体管，但为整合多种成熟技术提供了可能，从而有效提高晶体管的空间有效密度，降低芯片到芯片的时延，并通过整合因分别优化而使复杂度降低的多种技术实现高度组件化。这种方法有望使摩尔定律的预期至少再延续几代。

硅通孔允许信号和电源通过整个硅层，这也许是三维堆叠最显著的特点。

3Di 的许多具体实例要么需要堆叠部分功能性芯片，要么甚至需要堆叠晶片。所有这些不同实例的一个共同特点是*硅通孔* (TSV)。顾名思义，硅通孔允许信号和电源通过整个硅层，这也许是 3D 堆叠最显著的特点。相比芯片中的其他器件，硅通孔往往相当大。此外，它们都带有厚电介质衬里并填充了热膨胀系数高于硅的导电材料。引入这些大尺寸不同器件可能会使芯片中产生很大的应力，从而导致芯片出现结构缺陷乃至故障。另外，也可能改变硅器件的电学性能，尽管这种后果可能没那么严重。

Jung 等人撰写的以下论著全面分析了在硅中引入硅通孔可能产生的应力。作者提出了一种相当全面而简便的方法，运用线性叠加估算这些 TSV 可能带来的热机械应力。该分析还可以用于估算冯·米塞斯应力，此应力是衡量机械稳定性的简便指标。此方法可以用于设计稳定可靠的硅通孔，并有望成为设计三维芯片的重要工具。 C

Subramanian S. Iyer (ssiyer@us.ibm.com) 是 IBM 院士 (IBM Fellow) 和 Systems and Technology 集团 System Scaling Technology, Microelectronics 部门总监。该集团位于纽约州 Hopewell Junction 市。

译校：姚海龙；责任编辑：陈文光

DOI:10.1145/2494536

考虑硅通孔（TSV）应力的三维集成电路（3D IC）全芯片机械可靠性分析及优化方法

作者：Moongon Jung、Joydeep Mitra、潘志刚与 Sung Kyu Lim

摘要

业界认为，与传统的二维集成电路（2D IC）相比，硅通孔（TSV）三维集成电路（3D IC）在效率、功耗、性能及形状因子等方面的优势都达到新的高度。然而，相对传统的二维集成电路而言，三维集成电路采用了颠覆性的制造技术。硅通孔会产生显著的热机械应力，可能严重影响电路的性能、漏电和可靠性。在本文中，我们讨论了高效、精确的全芯片热应力与可靠性分析工具和设计优化方法，以减轻三维集成电路的机械可靠性问题。首先，我们结合不同的关联结构（如着陆垫和介电衬垫），详细分析了硅通孔引起的热机械应力情况。其次，我们探讨和验证应力张量的线性叠加原理，并对照详细的有限元分析（FEA）模拟证明了该方法的准确性。接下来，我们将该线性叠加方法应用于全芯片应力模拟和一个称为“冯米塞斯屈服判据”的可靠性指标。最后，本文提出了一个设计优化方法，以缓解三维集成电路（3D IC）的机械可靠性问题。

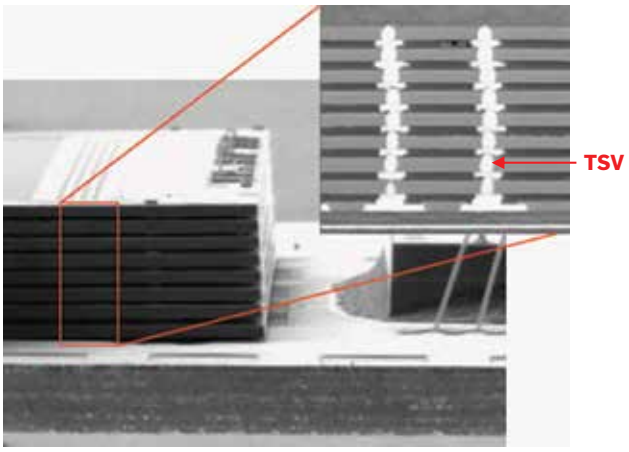
1. 为何引入三维集成电路?

过去四、五十年间，半导体产业的重点一直是用先进的光刻图形技术使集成电路微型化（目前已达 22 纳米节点）。虽然国际半导体技术蓝图（ITRS）预测 CMOS 仍有可微缩的空间，举例而言，在 2020 年微缩至约 7 纳米节点[7]，但是这样的微缩程度将达到基础物理的极限，甚至在达到物理极限之前，微缩的经济性将需要为“延伸摩尔定律”和“超摩尔定律”电路集成寻找其他生产手段。

鉴于工艺超越 32-22 纳米后功耗、性能和经济瓶颈皆会日趋明显，因此业界已开始寻找替代解决方案。在这样的背景下，人们开始积极地研究、开发和部署更薄的堆叠三维集成电路。这样的产品最初通过引线键合（wire-bond），后来采用倒装芯片（flip-chip），近来则利用硅通孔（TSV）加以实现。[18]

如图 1 所描述，硅通孔是三维集成电路的关键实现技术。此类硅通孔在堆叠的芯片间提供垂直的信号、电源和散热通路。凭借采用硅通孔的三维集成技术，可通过将集成电路组件放置在不同的芯片上大幅减少它们之间的平均距离和最大距离，继而显著降低时延、功耗和面积。而且，该技术能将异构设备（如 28 纳米高速逻辑电路和 130 纳米模拟电路）集成在一起，使得整个系统更加紧凑和高效。

图 1. 带硅通孔的三星 16Gb NAND 堆叠结构（8 个 2GB NAND）。[20]



学术界近期展示了含堆叠内存的 64 个并行处理器核[12]，以及大型三维单芯片多处理器（3D CMP）及其所采用的基于集群的近阈值计算架构[4]。此外，异构 3D FPGA（Xilinx 公司的 Virtex-7 FPGA）已开始量产。[22] 然而，这种新设计元素，即硅通孔（TSV），带来了一些难题。由硅通孔产生的应力所造成的热机械可靠性问题是三维集成电路面临的最大难题之一。

2. 三维集成电路的热机械应力

鉴于硅通孔填充材料（如铜（= 17 ppm/K）和硅衬底（2.3 ppm/K）之间的热膨胀系数存在重大失配，因此会于三维集成电路制造过程和硅通孔结构的热循环期间积聚热机械应力。由于铜（= Cu）的退火温度远高于工作温度，因此在冷却成室温后会对硅材料产生拉伸应力。这种热机械应力会同时影响芯片的性能与可靠性。

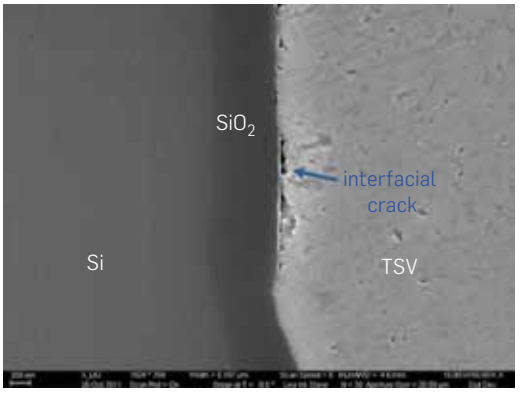
在半导体中，由应变产生的原子间距变化会影响带隙，从而使电子更易（或更难，取于材料和应变）

本著作的先前版本曾在 *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*（*有关集成电路和系统计算机辅助设计的 IEEE 汇刊*）（*31*，8 (2012)，1194-1207）上发表。

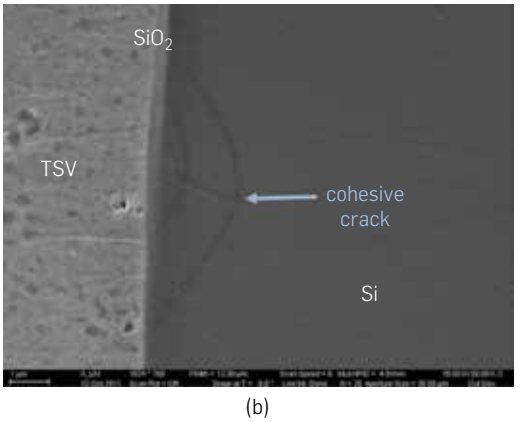
抬升进入导带。这会导致半导体电阻率的变化，继而会转换成迁移率的变化。[8] 在 100 纳米以下节点中，应变硅技术已广泛用于提高晶体管沟道中的载流子迁移率。硅通孔产生的应力会影响应变硅上方的载流子迁移率，并充当一个额外的偏差来源。实际上，硅通孔产生的拉伸应力会逆向影响电子和空穴迁移率。因此，如果设计师在芯片设计阶段未考虑这一迁移率偏差，则无法保证预期的芯片性能。以往著作 [2,23] 曾讨论硅通孔产生的应力对单个器件性能的影响，以及对全芯片时序的影响。

同时，人们的主要担忧一直是硅通孔结构的热机械可靠性。如图 2 所示，如果硅通孔周边存在诸如空隙等小瑕疵，硅通孔产生的应力会在介电衬垫和硅衬底之间导致接合部产生裂纹，或在绝缘衬垫和硅衬底之间导致粘结裂纹。[15] 这些裂纹会损坏附近的晶体管，在硅通孔之间创建通路（= 短路），在最糟糕的情况下会导致整个芯片无法运行。以往著作研究了裂纹在硅通孔应力下的扩散行为。[9,14,19] 不过，大多数以往的著作都侧重于就单个孤立硅通孔建立热机械应力与可靠性模型。业界曾用有限元分析（FEA）方法进行这些模拟，但该方法对于全芯片规模的分析来说在计算上过于昂贵或不具可行性。

图 2. 由于热机械应力而致的裂纹增长。[15](a) 介电衬垫和硅衬底之间的接合裂纹；(b) 硅衬底中的粘聚裂纹。



(a)



(b)

在本文中，我们提出了一个全芯片硅通孔热机械应力和可靠性分析流程，可克服有限元分析方法的局限性。此外，我们引入了一个设计优化方法，以减少基于硅通孔的三维集成电路所面临的机械可靠性问题。为获取真实的芯片应力分布情况，我们首先建立了细致且符合实际的硅通孔结构模型。这正是许多未考虑设计场景的以往著作所缺少的。然后，我们参照有限元分析模拟，验证了应力张量的线性叠加原理，并用此方法来产生全芯片级的应力分布图和可靠性指标图。此外，我们提出了通过优化诸如衬垫厚度和硅通孔布局，降低三维集成电路全芯片冯·米塞斯应力的设计方法。冯·米塞斯应力是一项机械可靠性指标，用于识别机械不稳定点，如易出现裂纹的位置。

3. 基准建模

3.1. 现有著作的局限性

Yang 等人的著作将二维径向应力分析模型（名为 *Lamé* 应力解）用于研究硅通孔热机械应力对器件性能的作用。[23] 这个二维平面解决方案假设将无限长的硅通孔嵌入无限的硅衬底中，得出硅衬底区域的应力分布情况。分布情况如下所示 [16]：

$$\sigma_{rr}^{Si} = -\sigma_{\theta\theta}^{Si} = -\frac{E\Delta\alpha\Delta T}{2}\left(\frac{D_{TSV}}{2r}\right)^2$$
$$\sigma_{zz}^{Si} = \sigma_{rz}^{Si} = \sigma_{\theta z}^{Si} = \sigma_{r\theta}^{Si} = 0 \qquad (1)$$

其中，σ^{Si} 是指硅衬底中的应力，E 是杨氏模量（= 弹性材料刚性的量度），$\Delta\alpha$ 是热膨胀系数的失配量，ΔT 是热负荷差，r 是与硅通孔中心的距离，以及 D_{TSV} 是硅通孔的直径。

尽管这一闭合形式的公式易于操作，但是此二维解决方案仅适用于仅有硅通孔和衬底的结构，因此不适合于带有着陆垫和衬垫的实际硅通孔结构。同样，它还忽略了器件所在位置，即硅通孔周围晶片表面附近应力场的三维性质。此外，晶片表面附近的硅通孔 / 衬底接合区已知是一个机械可靠性问题较为严重的区域。[19] 在我们的研究中，晶片表面是指衬底（硅）/ 介电层（二氧化硅）接合区正下方的硅表面。

尽管 Ryu 等人 [19] 的著作提出了一种三维应力半解析模型，但该模型仅适用于具有高纵横比的硅通孔。同时，他们的硅通孔结构仅包括硅通孔和硅衬底。鉴于含着陆垫和介电衬垫的硅通孔的边界条件有所不同，因此我们无法采用他们的模型。此外，由于他们的模型仅适用于单个孤立硅通孔，因此无法直接用于评估全芯片规模的机械可靠性问题。

3.2. 经我们改进的结构

因为尚无适合实际硅通孔结构的已知应力分析模型，因此我们创建了针对硅通孔的三维有限元分析模型，以研究晶片表面附近的应力分布情况。为切合实际地研究硅通孔产生的热机械应力，我们的硅通孔基准模拟结构基于自行编造的数据和公开发表的数据（如图 3 所示）[3,14]。

我们构建了两个硅通孔单元，即硅通孔$_A$和硅通孔$_B$；两者分别占用了三个和四个标准单元行（以北卡州立大学 45 纳米技术制造）[6]。我们分别规定了以硅通孔边缘为基线起 1.205 微米和 2.44 微米为排除区（KOZ），即不得在此区域内放置硅通孔$_A$和硅通孔$_B$单元。我们的基准硅通孔直径、高度、铜扩散阻挡层厚度、衬垫厚度和着陆垫面积分别为 5 微米，30 微米，50 纳米，125 纳米和 6 微米，接近 der Plas 等人所著论文中的数据（除非另有阐述）。[3] 我们还分别将二氧化硅和钛用作基线衬垫和铜扩散阻挡层材料。本实验所用材料的属性在表 1 中列出。我们用商用有限元分析模拟工具 ABAQUS 开展实验，并假设所有材料都为线性弹性和各向同性。此外，假设所有材料的接合面都有完美的附着力。[17]

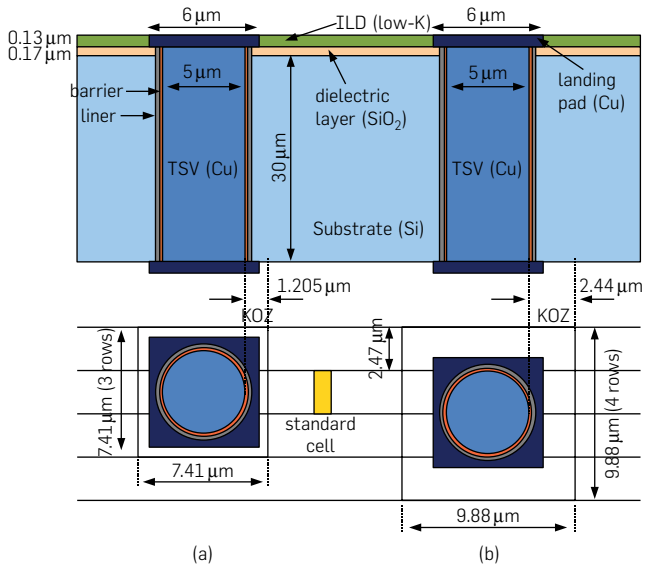
3.3. 应力张量

在详细讨论应力建模结果之前，我们要引入应力张量这个概念。对象中某个点上的应力可用九分量应力张量加以定义：

$$\sigma = \sigma_{ij} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}$$

其中，第一系数 i 表示应力作用于与 i 轴垂直的平面，而第二个系数 j 则表示应力作用的方向。如果系数 i 和 j 相等，则我们将之称为正常应力，否则则称为剪切应力。由于我们在针对圆柱形硅通孔的建模中采用了圆柱形坐标系，因此系数 1、2 和 3 分别代表 r、q 和 z。

图 3. 基准硅通孔结构。(a) 占用三条标准单元行的硅通孔$_A$单元（排除区 = 1.205 微米）。(b) 占用四条标准单元行的硅通孔$_B$单元（排除区 = 2.44 微米）。



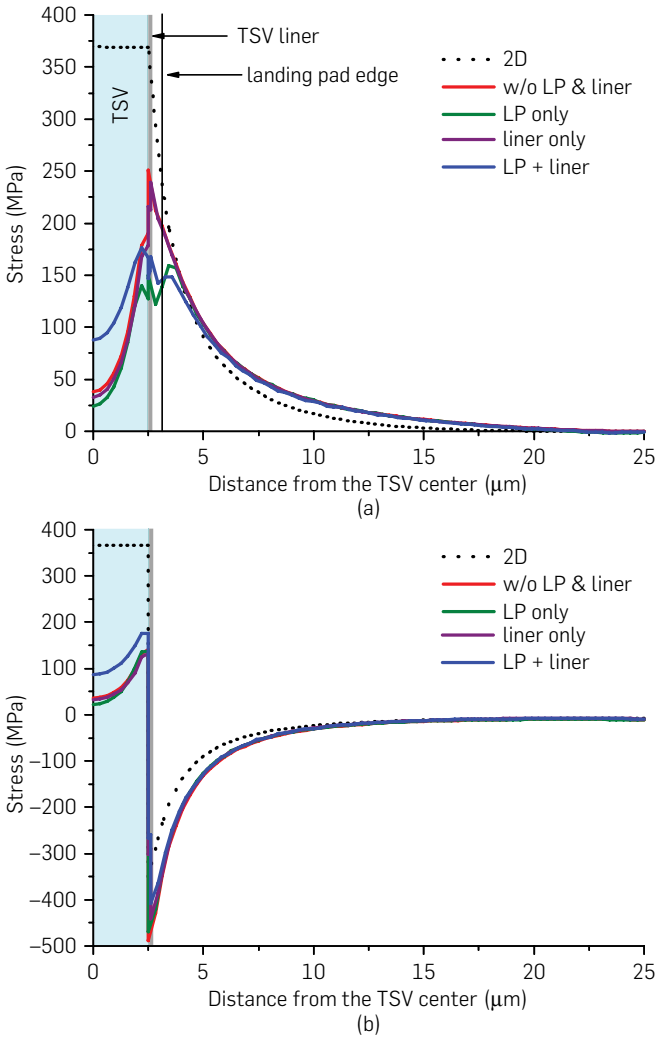
3.4. 应力等值线图

图 4 显示了当热负荷的 ΔT = -250° C 时，晶片表面上以硅通孔中心为圆心的任意径向线上正常应力组分 σ_{rr} 和 $\sigma_{\theta\theta}$ 的有限元分析模拟结果。也就是说，我们假设硅通孔结构的退火温度为 275° C，冷却至 25° C，以模仿制造过程。[11, 16, 19] 我们还假设整个硅通孔结构在退火温度时不存在应力。

表 1. 材料属性。

材料	CTE (ppm/K)	杨氏模量（GPa）	泊松比
铜	17	110	0.35
硅	2.3	130	0.28
二氧化硅	0.5	71	0.16
低介电材料	20	9.5	0.3
BCB	40	3	0.34
钛	8.6	116	0.32
钽	6.8	186	0.34

图 4. 硅通孔结构对正常应力分量的影响。(a) σ_{rr} 应力；(b) $\sigma_{\theta\theta}$ 应力。



在三维有限元分析模拟中，我们还考虑了硅通孔周围结构，如介电衬垫和着陆垫，而二维模型只考虑在 z 方向上无限延长的硅通孔和衬底。鉴于这一结构性差异，我们观察到二维解决方案和三维应力模拟结果在硅通孔边缘处存在巨大差异。众所周知，大部分机械可靠性问题都发生在不同材料之间的接合处，因此就可靠性而言，硅通孔边缘是关键区域。因此，二维解决方案并不能准确地预测硅通孔的机械问题。另外，与没有着陆垫和衬垫的情况相比，作为应力缓冲层的二氧化硅衬垫将硅通孔边缘处的 σ_{rr} 应力减少了 35 兆帕。着陆垫也有助于减少硅通孔边缘的应力。

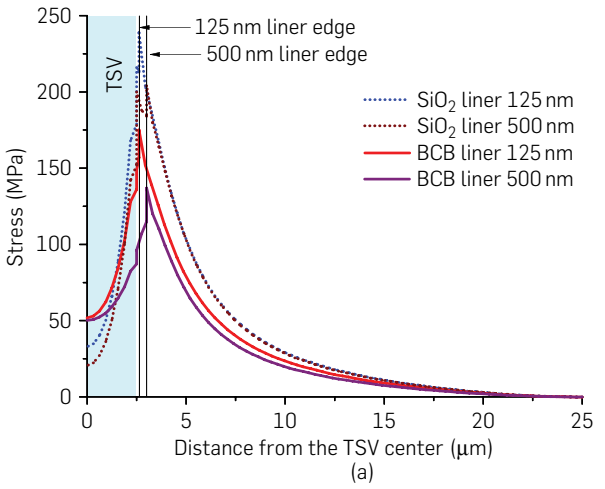
我们还将聚合物介电材料苯并环丁烯（BCB）用作硅通孔衬垫的替换材料。[16, 19] 由于杨氏模量（BCB 弹性材料的刚度值）较铜、硅和二氧化硅低得多，因此 BCB 衬垫能有效地吸收热膨胀系数失配引起的应力。图 5 说明了衬垫材料及其厚度对 σ_{rr} 应力分量的影响。随着衬垫厚度的增加，硅通孔边缘处的应力会显著减少，采用 BCB 衬垫的用例尤为如此。

从这些模拟可以看出，在就应力分布情况建模时考虑诸如衬垫和着陆垫等周围结构，对于更准确地分析硅通孔及周围的热机械应力十分重要。我们通过改变硅通孔直径 / 高度，着陆垫尺寸及衬垫材料 / 厚度构建一个应力数据库，从而实现对不同硅通孔结构的全芯片热机械应力和可靠性分析。

4. 全芯片可靠性分析

针对多个硅通孔结构的热机械应力有限元分析模拟需要海量的计算资源和时间，因此不适合全芯片分析。在本节中，我们将提出一个全芯片热机械应力和可靠性分析流程。为实现全芯片应力分析，我们首先要探讨来自单个硅通孔的线性应力张量叠加原理。基于线性叠加方法，我们构建了全芯片应力图，然后计算了冯·米塞斯屈服指标，以预测硅通孔三维集成电路的机械可靠性问题。

图 5. 衬垫材料 / 厚度对 σ_{rr} 应力的影响。



4.1. 全芯片分析流程概述

在本节中，我们将简要介绍我们的全芯片热机械应力和可靠性分析流程。我们首先进行了深入细致的单硅通孔有限元分析模拟，并将以硅通孔中心为圆心的径向线上的应力张量作为输入数据提供给我们的模拟引擎。我们还为模拟引擎提供硅通孔在三维集成电路布局中的位置以及热感图。凭借这些输入数据，我们找出了每个硅通孔的应力影响区域。然后，我们将影响区域中的点与施加影响的硅通孔相关联。接下来，对于每个要研究的模拟点，我们查出在关联步骤中找到的硅通孔应力张量，并用坐标转换矩阵获得应力张量在直角坐标系中的位置。我们着眼影响该模拟点的单个硅通孔，并累加他们所贡献的应力。一旦我们完成一个点的应力计算，我们就计算冯·米塞斯应力值。该算法的复杂性为 $O(n)$，其中 n 是模拟点的数目。

4.2. 机械可靠性指标

为评估计算所得的应力是否意味着潜在的可靠性问题，则必须针对潜在的机械问题选择一个临界值。冯米塞斯屈服判据是已知应用最广泛的机械可靠性量度之一。[5, 21, 24] 如果冯·米塞斯应力超过屈服强度，则材料开始屈服。在达到屈服强度之前，材料会弹性变形，并会于所施加的应力消除后恢复其原始形状。但是如果冯·米塞斯应力超过屈服点，则某些变形将在所施加的应力消失后仍是永久性和不可逆的。

参考文献中铜的屈服强度差异很大，从 225 兆帕到 600 兆帕。据报告，这取决于厚度、晶粒大小和温度。[24] 我们将 600 兆帕用作自身实验的铜屈服强度。硅的屈服强度为 7000 兆帕，对于冯·米塞斯屈服判据来说不会构成可靠性问题。

冯·米塞斯应力是某点的标量值，可用应力张量的组分计算而得。通过评估硅通孔和介电衬垫之间接合处的冯米塞斯应力（冯米塞斯应力最高值出现的位置），我们可以预测硅通孔中的机械问题。

4.3. 多硅通孔应力分析

我们观察发现由于硅通孔为圆柱形，因此单个孤立硅通孔的应力场沿径向对称分布。据此，我们根据以硅通孔中心为圆心的任意径向线上的应力张量集，按圆柱形坐标系求得应力在硅通孔中或周围的分布情况。要估计某个受多个硅通孔影响的点上的应力张量，则需要将应力张量换算成直角坐标系统。这是因为我们要从硅通孔提取应力张量，而硅通孔的中心正是圆柱形坐标系的原点；因此，对于中心点各不相同的各硅通孔，我们无法求出它们对某一点的应力张量的矢量和。这就是为什么在这种情况下我们需要一个通用坐标系，即直角坐标系。

然后，我们通过累加各硅通孔产生的应力张量，计算感兴趣点的应力张量。对于 5 微米直径的硅通孔，我们以其中心为圆心、25 微米为半径划定一个硅通孔应力影响区。根据有限元分析模拟，超过此距离的应力分量可忽略不计。

将直角坐标系和圆柱形坐标系中的应力张量分别设成 S_{xyz} 和 $S_{r\theta z}$。

$$S_{xyz}=\begin{bmatrix}\sigma_{xx} & \sigma_{xy} & \sigma_{xz}\\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz}\\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz}\end{bmatrix},\ S_{r\theta z}=\begin{bmatrix}\sigma_{rr} & \sigma_{r\theta} & \sigma_{rz}\\ \sigma_{\theta r} & \sigma_{\theta\theta} & \sigma_{\theta z}\\ \sigma_{zr} & \sigma_{z\theta} & \sigma_{zz}\end{bmatrix}$$

转换矩阵 Q 的形式为：

$$Q=\begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}$$

其中，θ 是 X 轴与硅通孔中心到模拟点之间连线所呈的角度。用圆柱形坐标系表示的应力张量可用转换矩阵转换成用直角坐标系表示：$S_{xyz}=QS_{r\theta z}Q^T$。

4.4. 线性叠加法

就线弹性结构分析而言，一项实用的原则就是对叠加情形进行分析。该原则指出，如果弹性体各点的位移与产生位移的力成正比，则该弹性体为线弹性。同时施加于此等弹性体的数个力的作用，即应力和位移，是分别施加的各个力的作用之和。我们通过按以下公式累加各硅通孔在某点上产生的单个应力张量，将此原则应用于计算该点所受的应力：

$$S=\sum_{i=1}^{n}S_i$$

其中，S 是该研究点处的总应力，而 S_i 是第 i 个硅通孔对该点施加的单个应力张量。

我们通过改变硅通孔的数量和其排布，对照有限元分析模拟验证了应力张量的线性叠加。我们将所有测试用例的最小硅通孔间距都设定为 10 微米。针对单硅通孔结构，以 0.1 微米为间距通过有限元分析模拟取得以硅通孔中心为圆心的半径上的应力张量。在我们的线性叠加方法中，模拟区域划分为均匀阵列式网格（间距为 0.05 微米）。如果无法直接从应力张量列表取得所研究栅格点的应力张量，我们将采用线性插值法根据列表中相邻的应力张量计算出该点的应力张量。

表 2 显示了我们所做的一些比较。首先，我们观察到用线性叠加方法后运行时间大幅减小。请注意，我们用 4 个 CPU 执行有限元分析模拟，但其中只有一个 CPU 用于线性叠加方法。即便线性叠加方法是对晶片表面的二维平面进行应力分析，而有限元分析模拟是在整个三维结构中进行的，但是我们可在必要时以类似的方式分析其它平面。此外，我们线性叠加方法的运行时间显示出对模拟点数量的线性依赖性，这与所研究的硅通孔的数量密切相关。因此，我们线性叠加方法的可扩展性高，从而适用于全芯片规模的应力模拟。

最重要的是，有限元分析模拟和线性叠加方法之间的误差几乎可以忽略不计。结果表明，我们的线性叠加方法高估了硅通孔内的应力大小。不过，虽然 10 个硅通孔用例中的最大硅通孔内误差 (%) 是 13.6%，有限元分析和我们方法之间的应力大小的差异仅为 5.0 兆帕。此外，由于大多数机械问题出现在不同材料之间的接合处，因此这个硅通孔内误差对于我们的可靠性分析并不构成严重影响。图 6 显示了含 10 个硅通孔的测试用例之一的冯米塞斯应力图。该图清楚地表明我们的线性叠加法与有限元分析模拟结果完全相符。

表 2. 有限元分析模拟和线性叠加方法之间的冯米塞斯应力比较。

硅通孔数	有限元分析 # 节点	有限元分析 运行时间	线性叠加 模拟点数	线性叠加 运行时间	最大误差 (%) 硅通孔内	最大误差 (%) 硅通孔外
1	153K	21m35s	1.0M	20.63	1.0	−0.4
2	282K	58m11s	1.2M	26.21	3.3	−0.8
3	358K	1h28m24s	1.44M	36.43	4.8	−1.3
5	546K	1h59m05s	1.68M	56.02	12.7	-1.9
10	1124K	4h34m14s	2.24M	65.32	13.6	−2.0

5. 全芯片模拟结果

我们用 C++/STL 实现了一个考虑硅通孔的全芯片应力和可靠性分析流程。如表 3 所列，我们分析了一个工业电路的四个不同变异实例（改变硅通孔的布局样式和硅通孔单元大小）。在所有测试用例中，硅通孔和逻辑门的数量均分别是 1472 和 370K。这些电路采用 45 nm 的物理单元库，用 Synopsys 公司的 Design Compiler 工具合成[6]，并且用 Cadence 公司的 SoC Encounter 工具得到最终布局图。所有电路都设计成双芯片堆叠三维集成电路。

我们采用供内部使用的三维布局程序来完成硅通孔和单元布局；有关硅通孔和单元的布局算法，详见 Kim 等人的著作。[13] 在规则硅通孔布局方案中，我们统一地在每个芯片上预布局硅通孔，然后完成单元布局，但在不规则硅通孔布局方案中可同时进行硅通孔和单元布局。不规则硅通孔布局可以求得比规则布局更优的线长结果。[13]

5.1. 总体影响研究

在本节中，我们将讨论硅通孔结构，硅通孔布局样式以及排除区大小对三维集成电路热机械可靠性的影响。我们根据用不同硅通孔结构得出的应力建模结果，对我们的基准电路进行了全芯片应力和可靠性分析。

图 7 显示了我们基准电路中的最大冯米塞斯应力。我们首先观察到，与含规则硅通孔布局的设计相比，含不规则硅通孔布局的设计表现出较差的最大冯·米塞斯应力。这主要是因为在不规则硅通孔布局方案中，硅通孔可以紧挨在一起，以尽量减少连线长度。图 8 显示了不规则 $_A$ 和规则 $_A$ 电路的部分冯·米塞斯应力图。

图 6. 有限元分析模拟和线性叠加方法之间的冯·米塞斯应力样本比较。 (a) 有限元分析结果； (b) 我们的结果；(c) 沿 (a) 图中的白线比较有限元分析结果和我们的结果。

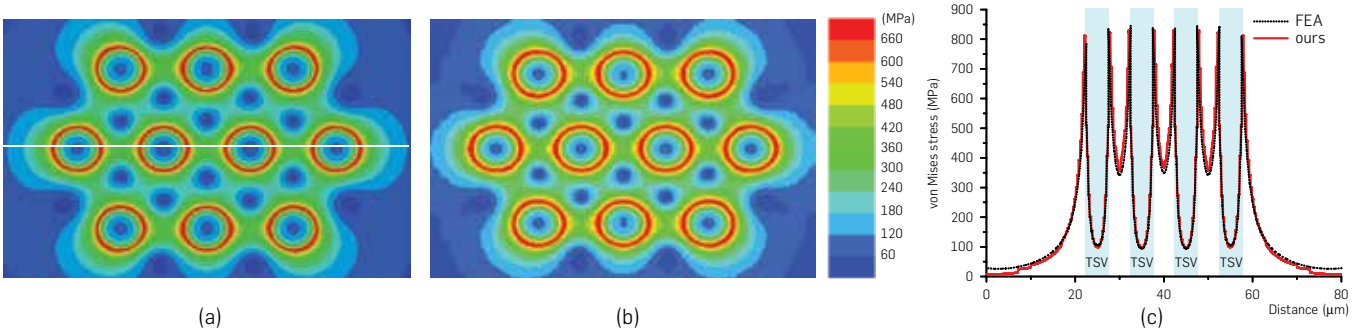


表 3. 基准电路。

电路	硅通孔布局	硅通孔单元大小（微米 × 微米）	线长（微米）	面积（微米 × 微米）
不规则 $_A$	不规则	7.41 × 7.41	9060	960 × 960
规则 $_A$	规则	7.41 × 7.41	9547	960 × 960
不规则 $_B$	不规则	9.88 × 9.88	8884	1000 × 1000
规则 $_B$	规则	9.88 × 9.88	9648	1000 × 1000

我们可以看到，规则 $_A$ 电路中的大部分硅通孔都超过铜屈服强度（600 兆帕）。

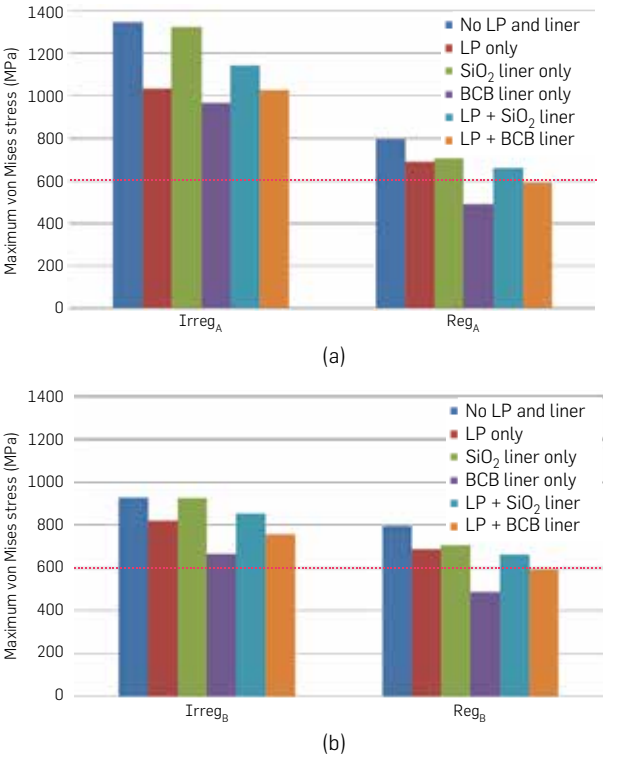
其次，随着排除区面积不断变大，不规则硅通孔用例的应力大小显著降低。通过扩大排除区面积，即在设计流程中增加硅通孔单元的尺寸，硅通孔间距也相应增加。这继而减少邻近硅通孔之间的应力干扰，从而减少硅通孔的冯·米塞斯应力大小。然而，对于规则硅通孔布局来说，由于规则 $_A$（23.5 微米）和规则 $_B$（25 微米）的硅通孔间距相似，并且按照这样的距离，来自邻近硅通孔的干扰可忽略不计，因此最大冯·米塞斯应力无显著差异。

最后，这些结果表明使用准确的硅通孔应力模型来评估三维集成电路机械可靠性集成电路的重要性。硅通孔周围存在的结构，如着陆垫或衬垫，对冯·米塞斯应力有显著影响。使用未考虑着陆垫或衬垫的简单硅通孔应力模型，可能会让我们高估可靠性问题。然而，大多数测试用例均大于适合铜硅通孔的冯·米塞斯屈服判据。第 5.4 节说明硅通孔衬垫如何能有助于减少大于此判据的情况。

5.2. 硅通孔间距的影响

硅通孔间距是决定硅通孔之间衬底区域内应力大小的关键因素。在本节中，我们将探讨硅通孔间距对冯·米塞斯应力的影响。我们整齐地把硅通孔放在 1 X 1 毫米 ^{2} 的芯片上。我们采用间距分别为 25、20、15 和 10 微米的 1600、2500、4356 和 10000 个硅通孔。我们得到两个数据集；一个没有着陆垫、衬垫和阻挡层；另一个有 6×6 微米 ^{2} 着陆垫、125 纳米厚 BCB 衬垫和 50 纳米厚钛阻挡层。

图 7. 硅通孔结构、硅通孔布局样式以及排除区大小对最大冯·米塞斯应力的影响。 (a) 用硅通孔 $_A$ 单元（排除区 =1.205 微米）的设计以及用硅通孔 $_B$ 单元（排除区 =2.44 微米）的设计。



我们首先观察到，冯·米塞斯应力随间距的增加而减小，并如图 9 所示在约 15 微米间距时开始饱和。这是可以理解的，因为相似间距条件下，单个硅通孔产生的应力变得可以忽略。此外，采用具有着陆垫和 BCB 衬垫的硅通孔的布局显示出类似的趋势，并且冯米塞斯应力较没有这些结构的布局低。

5.3. 硅通孔尺寸的影响

为研究硅通孔大小的影响，我们使用三种不同尺寸但高与直径比率相同（6）的硅通孔；小硅通孔（*H*/*D*=15/2.5 微米和排除区 1.22 微米），中硅通孔（*H*/

D=30/5 微米和排除区 1.202 微米）和大硅通孔（H/D=60/10 微米和排除区 1.175 微米），其中 H/D 是硅通孔的高度 / 直径。请注意，这些硅通孔单元分别占据两个、三个、五个标准单元行，旨在尽量减少它们之间在排除区大小上的差异。通过使排除区大小相似，我们可以专注于硅通孔尺寸的影响。此外为能公平比较各测试用例，我们将着陆垫宽度设定成比对应硅通孔的直径大 1 微米，并使用 125 纳米厚的二氧化硅衬垫和 50 纳米厚钛阻挡层。

表 4 显示了最大冯·米塞斯应力。不规则和规则硅通孔布局方案均显著受益于直径较小的硅通孔。这主要是由于正常应力分量的大小会随 $(D/2r)^2$ 减弱，其中 r 是距硅通孔中心的距离。

图 8. 布局和冯·米塞斯应力图近距照片：(a) 不规则 $_A$，(b) 规则 $_A$，(c) 不规则 $_A$ 的冯·米塞斯应力图，以及 (d) 规则 $_A$ 的冯·米塞斯应力图。

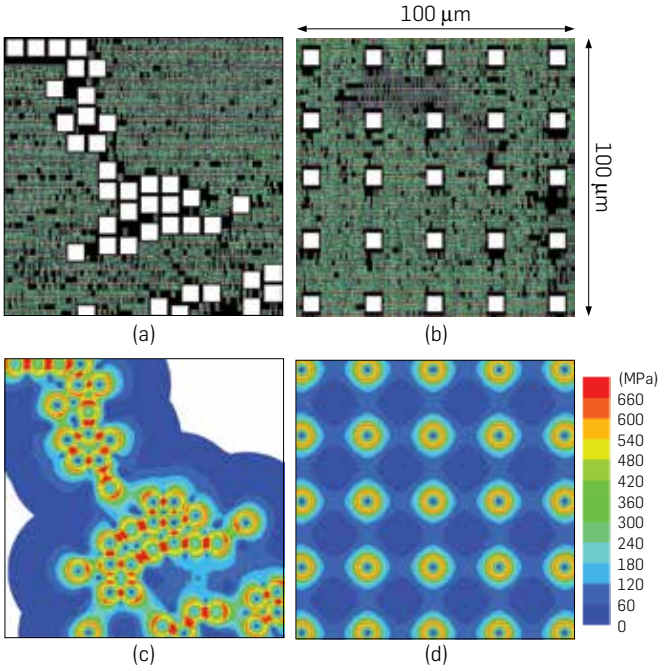
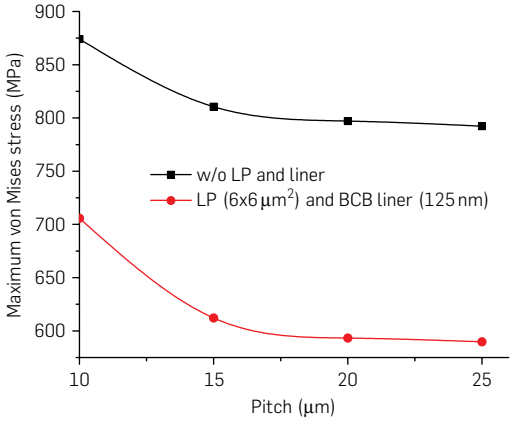


图 9. 硅通孔间距对最大冯·米塞斯应力的影响。



5.4. 衬垫厚度的影响

在本节中，我们将研究衬垫厚度对冯·米塞斯应力的影响。我们采用带硅通孔 $_A$ 单元和硅通孔 $_B$ 单元的设计，并将着陆垫的大小分别设为 6×6 微米 ^{2} 和 8×8 微米 ^{2}。我们还在所有测试用例中使用了 50 纳米厚的钛阻挡层。图 10 显示了衬垫厚度为 125 纳米、250 纳米和 500 纳米时的最大冯·米塞斯应力结果。

我们观察到，衬垫厚度对冯·米塞斯应力的大小存在巨大影响，因为较厚的衬垫可有效地吸收硅通孔 / 衬垫接合处的热机械应力。尤其是，与二氧化硅衬垫相比，BCB 衬垫因其极低的杨氏模量（如表 1 所示）而显著降低了冯·米塞斯应力的最大值。例如，500 纳米厚 BCB 衬垫可将不规则 $_A$ 所承受的最大冯·米塞斯应力降低 29%，并让所有含规则硅通孔布局的电路都不超过冯·米塞斯屈服判据。

表 5 显示了超过冯·米塞斯判据之硅通孔的数量。尽管不规则 $_A$ 电路仍有许多硅通孔不超过冯·米塞斯判据，但是如果我们在布局阶段置入硅通孔时能仔细考虑此可靠性指标，则可以减少冯·米塞斯应力。

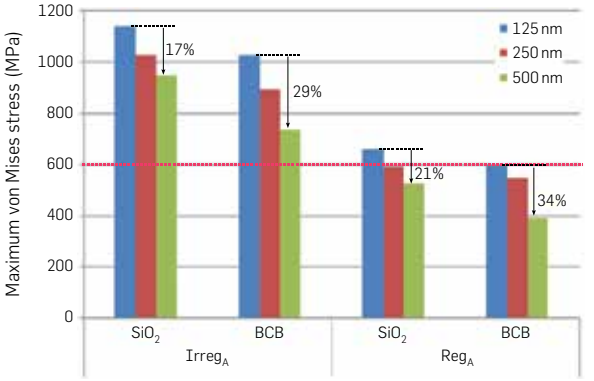
5.5. 硅通孔布局优化的影响

在本节中，我们将手动优化硅通孔的位置，同时尽量减少对布局的变动，以证明对硅通孔可靠性加以考虑的布局优化的潜在益处。我们采用了表现出最差冯·米塞斯应力的不规则 $_A$ 电路，并为此次实验采用 500 纳米厚 BCB 衬垫。我们涉及此类 BCB 衬垫对最大冯·米塞斯应力的影响以及硅通孔之间距对最大冯·米塞斯应力的影响的相关研究表明，10 微米间距是在考虑一

表 4. 硅通孔大小对最大冯·米塞斯应力的影响。括号中的数字是与大硅通孔用例相比的减少幅度 %。

	最大冯·米塞斯应力（兆帕）		
硅通孔布局	大硅通孔	中硅通孔	小硅通孔
不规则	1224.6	1126.4 (8% ↓)	902.7 (26% ↓)
规则	749.3	654.6 (13% ↓)	449.3 (40% ↓)

图 10. 衬垫厚度对含硅通孔 $_A$ 单元电路的最大冯·米塞斯应力的影响。



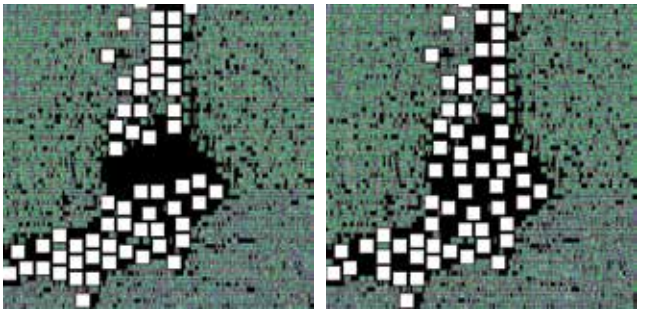
定安全边际后，降低冯·米塞斯应力的合理选择。如图 11 所示，我们将密集布局的硅通孔重新放置到附近的空白区（如果存在），以降低冯·米塞斯应力。

表 6 显示了高于 480 兆帕的冯·米塞斯应力在整个芯片上的分布，线长，以及硅通孔重新放置前后的最长路径延迟。我们用带有硅通孔寄生参数信息的 Synopsys PrimeTime 进行了三维静态时序分析。在重新放置硅通孔后，我们看到高冯·米塞斯应力区缩小了。通过微调硅通孔的位置，我们可以减少冯·米塞斯应力水平，将高于判据的硅通孔数量从 329 个减少到 261 个，这相当于在分别仅增加 0.23% 线长和 0.81% 最长路径延迟（最长路径延迟决定最大芯片运行频率）的情况下，获得 21% 的改善。这个小规模的测试用例显示了有可能在不大幅降低性能的情况下优化布局。

表 5. 衬垫厚度对大于冯·米塞斯判据之硅通孔数量的影响。括号中的数字是与 125 纳米厚衬垫用例相比的减少幅度 %。

		不合格硅通孔		
电路	**衬垫材料**	**125 纳米**	**250 纳米**	**500 纳米**
不规则 $_A$	二氧化硅	1462	1426 (2% ↓)	1281 (12% ↓)
	BCB	1389	1147 (17% ↓)	329 (76% ↓)
规则 $_A$	二氧化硅	1472	0 (100% ↓)	0 (100% ↓)
	BCB	0	0 (–)	0 (–)
不规则 $_B$	二氧化硅	1472	1236 (16% ↓)	64 (96% ↓)
	BCB	974	502 (48% ↓)	0 (100% ↓)
规则 $_B$	二氧化硅	1472	0 (100% ↓)	0 (100% ↓)
	BCB	0	0 (–)	0 (–)

图 11. 硅通孔重新排布以降低冯·米塞斯应力。硅通孔着陆垫和白色矩形。(a) 原布局；(b) 硅通孔重新排布之后。



(a) (b)

表 6. 硅通孔布局优化对冯·米塞斯应力分布、线长和最长路径延迟的影响。

	冯·米塞斯应力 (兆帕)				线长 (微米)	LPD (纳秒)
	480–540	**540–600**	**600–660**	**>660**		
原始	0.100%	0.041%	0.011%	0.002%	9060	3.607
优化	0.092%	0.036%	0.009%	0.0%	9081	3.636

6. 结束语

我们基于应力张量的线性叠加原理，提出了一个准确、快速的热机械应力和可靠性分析流程，从而克服了有限元分析工具的局限性，即耗费巨大的计算资源和时间。因此，我们的方法适用于对基于硅通孔的三维集成电路进行大规模机械可靠性分析。设计人员可以用我们的工具来评估三维集成电路设计中的机械可靠性问题，以及研究设计时如何在占位面积、性能和可靠性之间取得最佳平衡。

我们还开展了一些与硅通孔三维集成电路的热机械可靠性问题相关的后续研究。在 Jung 等人的著作 [9] 中，我们研究了机械应力与硅通孔接合处裂纹生长之间的关系。我们使用有限元分析模拟计算出所谓的能量释放率（ERR）指标，以衡量硅通孔中特定初始裂缝进一步发展的可能性。我们的研究表明，线性叠加在全芯片设计的能量释放率计算方面并不成立。然后，我们采用了响应面模型（RSM）方法，基于我们的基准有限元分析模拟，求得高度精确的全芯片能量释放率图。在 Jung 等人的著作 [10] 中，我们研究了诸如微凸块和封装凸点等片外元素对三维堆叠中芯片的机械可靠性的影响。我们的基准有限元分析结构经扩大可包括这些片外元件。相关结果表明，封装凸点对堆叠结构中的所有芯片造成显著的背景压缩应力，从而导致应力等值线向下移动。我们开发了所谓的横向和纵向线性叠加（LVLS）方法来处理各层片外元件的应力分量，以便得出全芯片应力图。

另一个相关的研究探讨了这些应力因素（包括片内和片外元件）如何影响邻近器件的迁移率以及三维集成电路的全芯片时序。[23] 然后，这个对应力情况有所考虑的时序信息将用于指导全芯片布局和优化。[1] 表 1 显示了硅通孔和三维集成电路所用各种材料的属性。但是，取决于所采用的工艺技术，各个值在硅通孔之间以及单个硅通孔的晶粒间也不尽相同。我们目前正在研究这些材料属性的变化会如何影响机械应力张量的分布、器件迁移率的变化以及全芯片的时序和可靠性。最后，这些热机械应力问题与三维集成电路的电气可靠性密切相关。在 Zhao 等人的著作 [25] 中，我们研究了电源 / 地硅通孔应力对的电迁移的影响，以及三维集成电路中配电网络（PDN）的长期可靠性。

三维集成电路中的这些热电子机械可靠性需要全面的多物理学方法才能实现更有效的设计方案。此外，业界需要设计师和制造商之间的紧密合作，才能更好地处理硅通孔和三维集成电路方面的这些迫切问题，加快主流用户对这种技术的接受。

鸣谢

此研究由美国国家科学基金会的第 CCF-1018216 号、第 CCF-1018750 号拨款、IBM 学院奖和英特尔公司提供部分支持。

参考资料

1. Athikulwongse, K., Chakraborty, A., Yang, J.S., Pan, D.Z., Lim, S.K.Stress-driven 3D-IC placement with TSV keep-out zone and regularity study. In *Proceedings of IEEE International Conference on Computer-Aided Design* (2010).
2. Athikulwongse, K., Yang, J.S., Pan, D.Z., Lim, S.K.Impact of mechanical stress on the full chip timing for TSV-based 3D ICs.*IEEE Trans.Comput.-Aided Design Integr.Circuits Syst.*(2013).
3. der Plas, G.V. et al.Design issues and considerations for low-cost 3D TSV IC technology.In *IEEE International Solid-State Circuits Conference Digest Technical Papers* (2010).
4. Fick, D., Dreslinski, R., Giridhar, B., Kim, G., Seo, S., Fojtik, M., Satpathy, S., Lee, Y., Kim, D., Liu, N., Wieckowski, M., Chen, G., Mudge, T., Blaauw, D., Sylvester, S. Centip3De:A cluster-based NTC architecture with 64 ARM Cortex-M3 cores in 3D stacked 130 nm CMOS.*IEEE J. Solid-State Circuits 48* (2013).
5. Franssila, S. Introduction to Microfabrication, John Wiley and Sons, 2004.
6. FreePDK45. http://www.eda.ncsu.edu/wiki/FreePDK.
7. International Technology Roadmap for Semiconductors (2012 Update). http://www.itrs.net.
8. Jaeger, R.C., Suhling, J.C., Ramani, R., Bradley, A.T., Xu, J. CMOS stress sensors on (100) silicon.*IEEE J. Solid-State Circuits 35* (2000).
9. Jung, M., Liu, X., Sitaraman, S., Pan, D.Z., Lim, S.K.Full-chip through-silicon-via interfacial crack analysis and optimization for 3D IC. In *Proceedings of IEEE International Conference on Computer-Aided Design* (2011).
10. Jung, M., Pan, D., Lim, S.K.Chip/package co-analysis of thermo-mechanical stress and reliability in TSV-based 3D ICs.In *Proceedings of ACM Design Automation Conference* (2012).
11. Karmarkar, A.P., Xu, X., Moroz, V. Performance and reliability analysis of 3D-integration structures employing through silicon via (TSV).In *IEEE International Reliability Physics Symposium* (2009).
12. Kim, D.H., Athikulwongse, K., Healy, M.B., Hossain, M.M., Jung, M., Khorosh, I., Kumar, G., Lee, Y.J., Lewis, D.L., Lin, T.W., Liu, C., Panth, S., Pathak, M., Ren, M., Shen, G., Song, T., Woo, D.H., Zhao, X., Kim, J., Choi, H., Loh, G.H., Lee, H.H.S., Lim, S.K.3D-MAPS:3D massively parallel processor with stacked memory.In *IEEE International Solid-State Circuits Conference Digest of Technical Papers* (2012).
13. Kim, D.H., Athikulwongse, K., Lim, S. K.A study of through-silicon-via impact on the 3D stacked IC layout. In *Proceedings of IEEE International Conference on Computer-Aided Design* (2009).
14. Liu, X., Chen, Q., Dixit, P., Chatterjee, R., Tummala, R.R., Sitaraman, S.K.Failure mechanisms and optimum design for electroplated copper through-silicon vias (TSV).In *IEEE Electronic Components and Technology Conference* (2009).
15. Liu, X., Chen, Q., Sundaram, V., Tummala, R.R., Sitaraman, S.K.Failure analysis of through-silicon vias in free-standing wafer under thermal-shock test. *Microelectronics Reliab. 5* (2013).
16. Lu, K.H., Zhang, X., Ryu, S.K., Im, J., Huang, R., Ho, P.S.Thermo-mechanical reliability of 3-D ICs containing through silicon vias. In *IEEE Electronic Components and Technology Conference* (2009).
17. Ong, J.M.G., Tay, A.A.O., Zhang, X., Kripesh, V., Lim, Y.K., Yeo, D., Chen, K.C., Tan, J.B., Hsia, L.C., Sohn, D.K.Optimization of the thermomechanical reliability of a 65 nm Cu/low-k large-die flip chip package.*IEEE Trans.Compon.Packag. Tech.32* (2009).
18. Pan, D.Z., Lim, S.K., Athikulwongse, K., Jung, M., Mitra, J., Pak, J., Pathak, M., Seok Yang, J. Design for manufacturability and reliability for TSV-based 3D ICs.In *Proceedings of Asia and South Pacific Design Automation Conference*, (2012).
19. Ryu, S.K., Lu, K.H., Zhang, X., Im, J. H., Ho, P.S., Huang, R. Impact of near-surface thermal stresses on interfacial reliability of through-silicon-vias for 3-D interconnects. In *IEEE Transactions on Device and Material Reliability* (2010).
20. Samsung.16 Gb NAND wafer-level stack with TSV. http://www.samsung.com.
21. Xiang, Y., Chen, X., Vlassak, J.J.The mechanical properties of electroplated Cu thin films measured by means of the bulge test technique. In *Proceedings of Material Research Society Symposium* (2002).
22. Xilinx.Virtex-7 FPGA. http://www.xilinx.com/products/silicon-devices/3dic/index.htm.
23. Yang, J.S., Athikulwongse, K., Lee, Y.J., Lim, S.K., Pan, D.Z.TSV stress aware timing analysis with applications to 3D-IC layout optimization. In *Proceedings of ACM Design Automation Conference* (2010).
24. Zhang, J., Bloomfield, M.O., Lu, J.Q., Gutmann, R.J., Cale, T.S.Modeling thermal stresses in 3-D IC interwafer interconnects.In *IEEE Trans. Semicond.Manuf.*(2006).
25. Zhao, X., Scheuermann, M., Lim, S. K.Analysis of DC current crowding in through-silicon-vias and its impact on power integrity in 3D ICs.In *Proceedings of ACM Design Automation Conference* (2012).

Moongon Jung（moongon@gatech.edu），佐治亚理工学院，佐治亚州。

Joydeep Mitra 和潘志刚（David Z. Pan）（{joydeep, dpan}@ece.utexas.edu），德克萨斯大学，奥斯汀，德克萨斯州。

Sung Kyu Lim (limsk@ece.gatech.edu)，佐治亚理工学院，佐治亚州。

译校：姚海龙；责任编辑：陈文光