

人机界面 操控指南

Kinect for Windows v1.6.0

© 2012 Microsoft Corporation。保留所有权利。

免责声明

您应确认本文档为“原样”提供，而且您应承担因使用本文档所带来的风险（本文档中所给出的信息和观点，其中包括 URL 及其它 Internet 网站引用，可能随时更改，恕不另行通知）。本文档未向您提供关于任何 Microsoft 产品知识产权的法律授权。本文档中的信息未涉及任何法律建议或其他专业建议，也不可被等同为由某位授权专家所提供的具体建议。如您未获得针对于当前具体情况和条件而给出的专业建议，则不应按照本文档中所提供的信息采取任何行为（或避免任何行为）。您可以出于内部参考目的而复制和使用本文档。在使用本文档中关于如何使用 Kinect 软件和 Kinect 传感器的说明信息时，您需承担因使用 Kinect 传感器和/或相关软件所导致的任何危害或损失（其中包括由 Kinect for Windows 应用程序的最终用户所导致的危害或损失）及由此带来的风险，另外，您自愿放弃向 Microsoft 及其分支机构进行索赔的全部权利（其中包括但不限于因 Kinect 传感器或 Kinect 软件自身缺陷所导致的索赔要求），同时，您还自愿主张 Microsoft 及其分支机构不会因此类索赔而产生损失。

Kinect for Windows

人机界面操控指南 v1.6.0

目录

引言.....	第 04 页
了解我们的设计原则，熟悉 Kinect 传感器，以及为您的应用程序搭建一个最佳匹配的运行环境。	
最佳人机交互设计.....	第 09 页
了解如何有效使用手势、语音和反馈，以设计出尽善尽美的 Kinect 人机交互体验。	
基本的人机交互操作.....	第 45 页
将手势和语音融入到一些强输入方法之中，并综合使用这些手段来实现定位、选择、平移、缩放和文本输入。	
依距离而变的人机交互操作.....	第 62 页
在人机交互操作及其界面的设计中，充分考虑用户与输入设备之间的距离变化，并随之做出调整。	
多模态.....	第 66 页
探讨异步与同步人机交互操作之间的区别，以便您充分利用多种全新的输入手段。	
多用户.....	第 68 页
允许多位用户共同使用手势和语音来操控您的应用程序。	
总结.....	第 70 页

引言

欢迎来到 Kinect 人机交互设计的神奇世界！要成功设计此类人机交互体验，并非难事，但是，如果能让您的用户从中获得享受，并让多年前因财力和技术限制而无法实现的解决方案变成现实，这将是一种莫大的收获。此份“人机交互指南”文档旨在向您分享我们现已掌握并使用的 Kinect 人机交互设计经验，让您不再遭遇我们所经历的一些重大难题，为您实现成功设计而铺平道路，另外，此文档将重点介绍我们现已采用的一些行之有效的设计手段，让您专注于利用 Kinect 来解决当前的特定问题。随着我们不断学习和进步，我们将对此文档做出进一步更新，以分享我们的发现和经验，希望我们的知识能够帮助您实现最完美的用户体验。

设计原则

如下设计原则是对本文档中一些重点经验的汇总，如果您要设计自己的人机交互体验，请务必牢记这些内容。

具有环境感知特性的用户体验才是最棒的

人机界面应能自动适应用户和 Kinect 之间的距离变化。

人机界面应能自动对用户数量和多用户参与做出响应。

控制器的摆放位置应能根据预期的用户位移或动作来进行设计。

每种输入方法都具有各自的优势和劣势

在某种既定的场景中，用户都应能选择总体上最省力的输入方法。

用户习惯于采用某种特定的输入方法，应尊重用户的习惯。

输入方法应可靠、一致、简便，否则，用户就会选用其它替代手段。

在当前场景中，输入方法之间的切换应该自然，或者，采用某种自然的过渡手段。

只有用户操控自如，才能从中获得享受

最重要的就是，让人机交互尽可能保持简单易学、轻松上手。

避免曲解用户意图。可采用综合型输入方法来增强用户信心，例如同时采用语音和手势。

提供即时反馈，让用户随时知道正在发生什么和即将发生什么。

经过用户测试的设计方案才是最稳健的

Kinect 提供众多全新的人机交互方法，但是，这也带来了前所未有的挑战。

最难以预料的就是什么即将发生和什么不会提前发生。

通常，极小的调整也可能让结果发生巨大变化。

用户测试要做到尽早和高频率，并在计划中为此留出时间，以做好对设计方案进行多次调整的准备。

透彻了解 Kinect 传感器

Kinect 可以看到什么

Kinect 传感器：它是应用程序的耳朵和眼睛。此处给出了该传感器的一些主要功能。

物理功能

Kinect 视角范围（色深和 RGB）

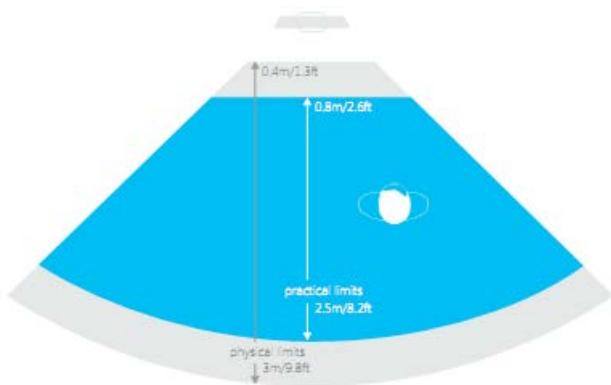
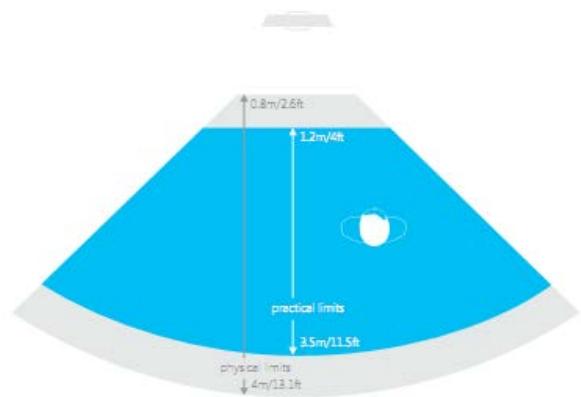
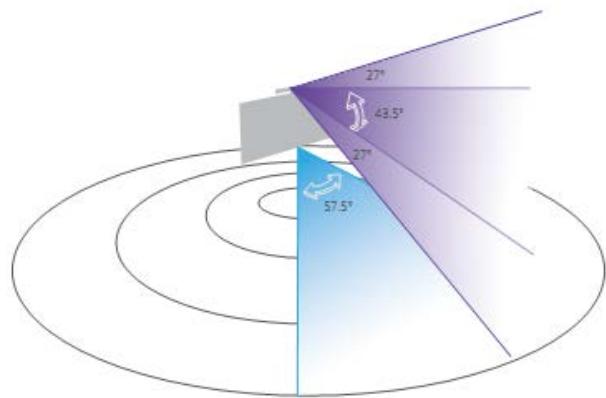
水平视角：57.5 度

垂直视角：43.5 度，上下倾斜角度为 -27 度到 +27 度之间

视距范围（默认模式）

物理范围：0.8 至 4 米

实际范围：1.2 至 3.5 米



视距范围（近景模式）

物理范围：0.4 至 3 米

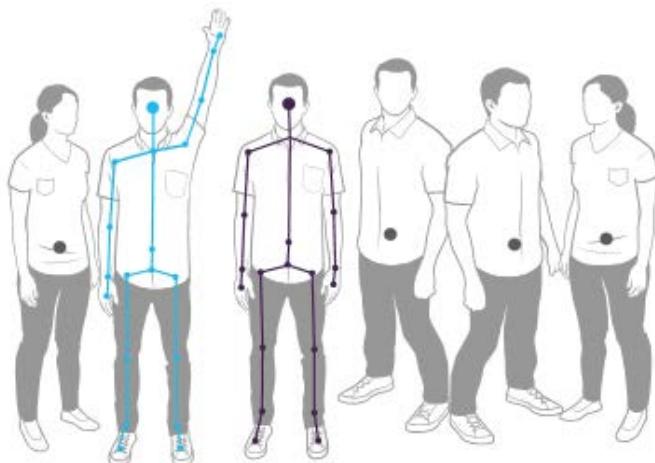
实际范围：0.8 至 2.5 米

应用程序可以根据当前场景的变化，在默认模式和近景模式之间切换。

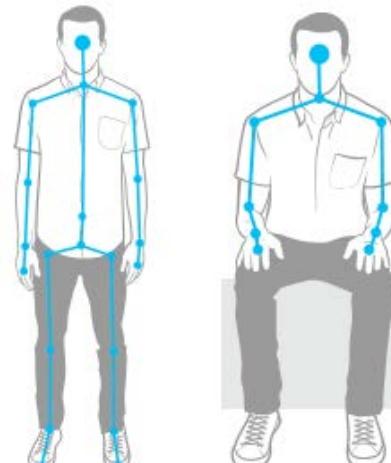
Kinect 最多可以跟踪 2 个骨架，其视野范围最多可以检测到 6 个人。

在默认的站立姿态下，Kinect 可以跟踪骨架的全部 20 个关节。

另外，在坐姿下，Kinect 还可以跟踪骨架的 10 个主要关节。



Kinect 可以跟踪 2 个骨架，并最多检测到 6 个人。



Kinect 可以跟踪默认站立姿态的骨架，并跟踪坐姿的骨架。

Kinect 可以听到什么

传感器前方 +50 度至 -50 度的音频输入。

在此范围之内，麦克风阵列的方向可按 10 度为单位进行调整。

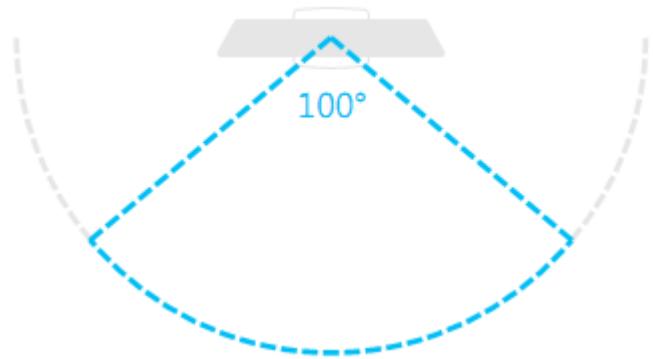
麦克风阵列可以屏蔽 20 dB 的环境噪声，从而提高音频保真度。环境噪声的大小约为普通交谈的音量。

SDK 支持单声道声音屏蔽，但不支持立体声屏蔽。

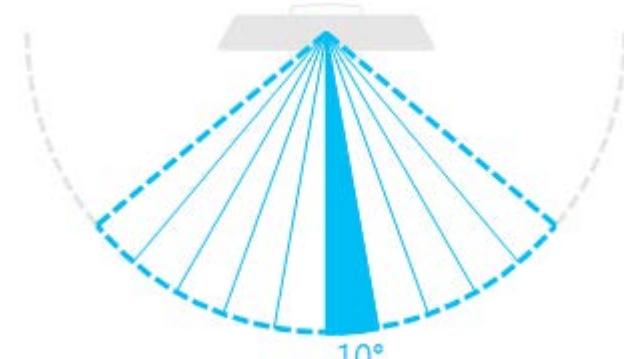
默认情况下，Kinect 将会跟踪最大音量的音频输入。

还可直接对麦克风阵列进行编程控制。例如：对准某个目标位置，或者，跟随当前正被跟踪某个骨架。

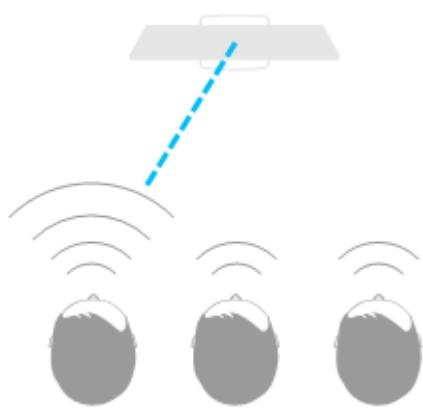
按照目前的麦克风接收范围设计，传感器后方达到 26dB 的环境噪音将被屏蔽。



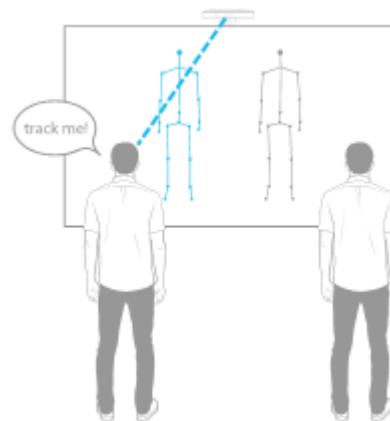
传感器前方 +50 度至 -50 度范围内的音频输入



在接收范围内，麦克风阵列的方向可按 10 度为单位进行调整



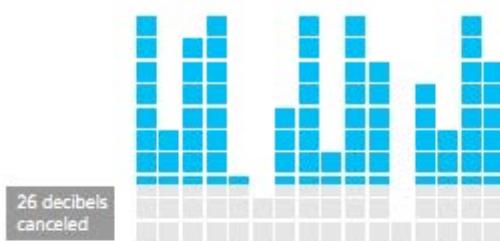
Kinect 跟踪最大音量的音频输入



可直接对麦克风阵列进行编程控制。例如，跟随当前正被跟踪的某个骨架



麦克风阵列可以屏蔽 20 dB 的环境噪声，从而提高音频保真度



传感器后方达到 26 dB 的环境噪音将被屏蔽

环境搭建及其注意事项

您在为自己的 Kinect 应用程序搭建环境时所采用的具体方案，将直接决定感官上的真实性和可操控性。通过对如下所述的各种影响因素尽可能全面地掌控，您所打造的用户体验就会美轮美奂，并减少人机操作过程中的各种不确定性和不连贯性。只要条件允许，应尽早在目标环境中对您的应用程序进行测试，并提高测试频率。

目标用户和随机人群

如果您正在设计的环境包含许多人，而且他们在传感器周围走动，请确保您采用正确的跟踪模式和空间设计，以避免其他人在目标用户和 Kinect 之间走动。

环境噪音

如果您采用语音作为输入方法，请务必注意，当前环境应尽量保持安静，以保持声音的保真度。如果您无法控制当前环境中的噪音大小，可以要求当前用户靠近音频接收设备。如果噪音过大，语音输入方法可能不太适用。

屏幕大小和分辨率

在选择显示器时，要考虑用户与其之间的距离。请记住，操作界面的大小取决于用户的所在位置和他们的理想视觉距离。

照明

避免环境中充斥大量自然光线，这会导致景深跟踪功能无法正常工作，景深摄影机无法处理直射日光。暗光照明非常适合于景深跟踪，但会降低 RGB 图像品质。

多余的物体或衣物

彻底改变人体外型的衣物或手持物品可能会干扰骨架跟踪系统正常工作。另外，具有反射质地的衣物或物体将会干扰红外线反射，进而干扰骨架跟踪系统正常工作。

传感器的摆放位置

在摆放传感器时，需要牢记的是，传感器可以看到地板，以实现定向。采用极端倾斜角度会降低跟踪系统的准确性。

最佳人机交互设计

手势和语音是 Kinect 为您所带来的两种全新输入方法。在接下来的部分中，将会介绍它们的工作原理、如何充分利用它们、可能遇到的挑战以及怎样为您的用户提供易于理解的反馈信息。

手势

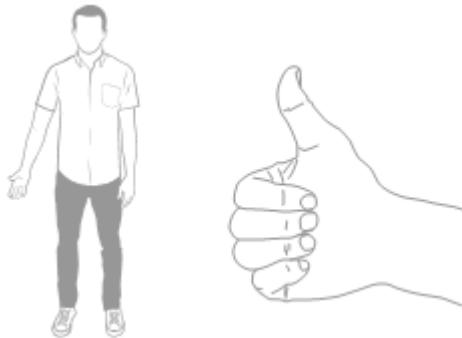
手势是我们即将探索的一种有趣的人机交互方法，同时，它也将带来一些巨大的难题，只有解决这些难题，才能让用户界面更加自然、易用并适用于应用程序的全部用户。本节将介绍一些基本的手势类型、如何就 Kinect 所看到的事物向用户给出反馈以及在设计手势时所应考虑到的一些注意事项。

基本的手势类型

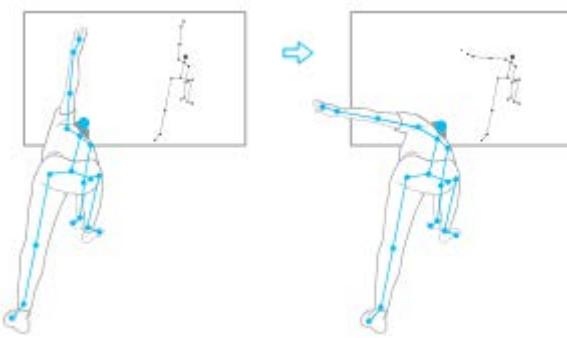
手势具有千变万化的特点，从使用您的手在屏幕上进行定位，到明确界定的一些具体手势，再到长距离的连续移动。如下是一些常用手势类型的具体示例及其使用说明，这将有助于您理解如何设计手势。

静态手势、连续手势和动态手势

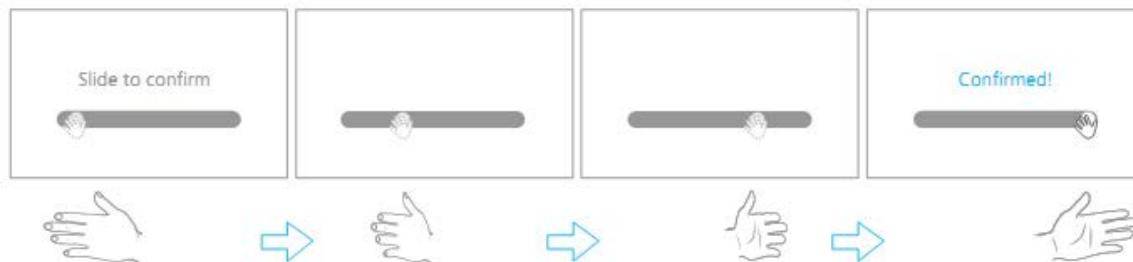
静态手势也被称为姿态或造型，用户摆出这种姿态并保持不动，直至已被识别为止。请注意，不同文化之间的差异对诸如“竖起大拇指”这样的静态手势可能具有不同的意义诠释。动态手势用于在用户做出某个明确的动作过程中或之后为其提供反馈。连续手势用于跟踪用户在 Kinect 前方所做的连续动作。



静态手势是指用户做出动作并在被识别之前始终保持该姿态



连续手势用于跟踪用户在 Kinect 前方所做的连续动作



动态手势用于在用户做出某个明确的动作过程中或之后为其提供反馈

先天手势与后天手势

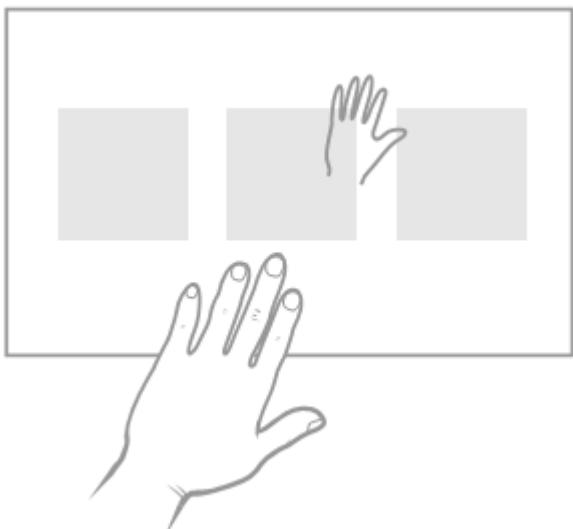
先天手势是用户出于本能或表达对周围环境认知结果所做的手势动作。

后天手势是用户被教授的用来同周围环境进行交流的手势动作。

例如 (先天) :

指向目标

抓取并拾起

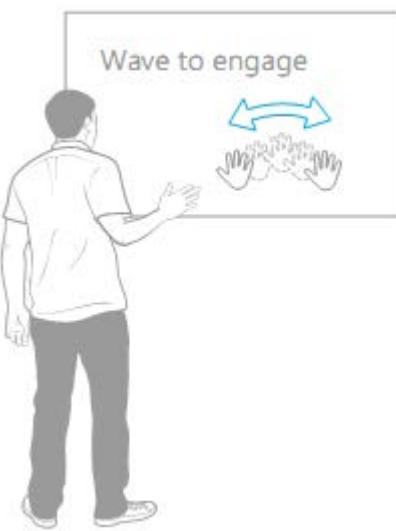


先天手势是用户出于本能的手势动作

例如 (后天) :

挥手以示参加

举起左手以示取消



后天手势是用户被教授的手势动作

视觉反馈

只有当前用户处于正确的视觉范围之内，用户的的手势才会被有效识别，因此，必须向用户给出反馈，以帮助其回答如下问题：

Kinect 传感器是否准备就绪？

我如何确定它看到的是我而不是别人？

Kinect 能够看到什么或能看到多大一部分？

Kinect 能够看到什么或能看到多大一部分？

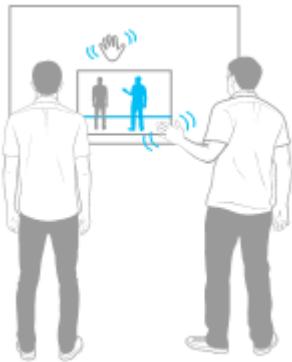
我是否已加入并处于监控之中？我现在是否可以挥手？

Kinect 能看到多少位用户？用户的头部是否在它的视野范围之内？

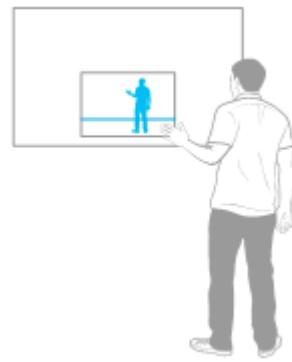
Kinect 的视域如何界定？

我如何判断做手势的时机和位置？

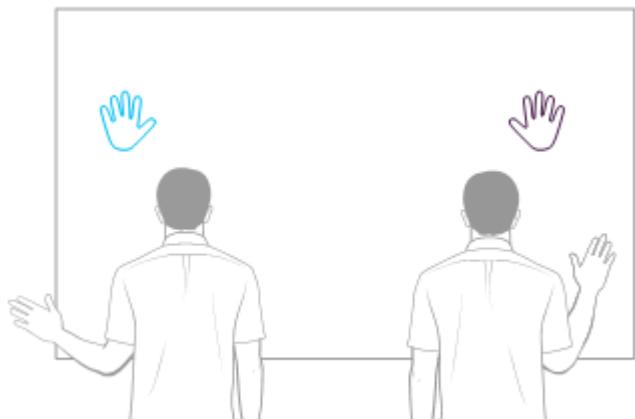
Kinect 能看到多少个人？



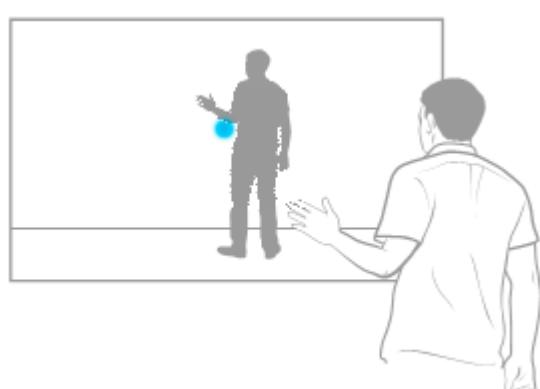
Kinect 能看到多少个人?
我如何确定它看到的是我而不是别人?



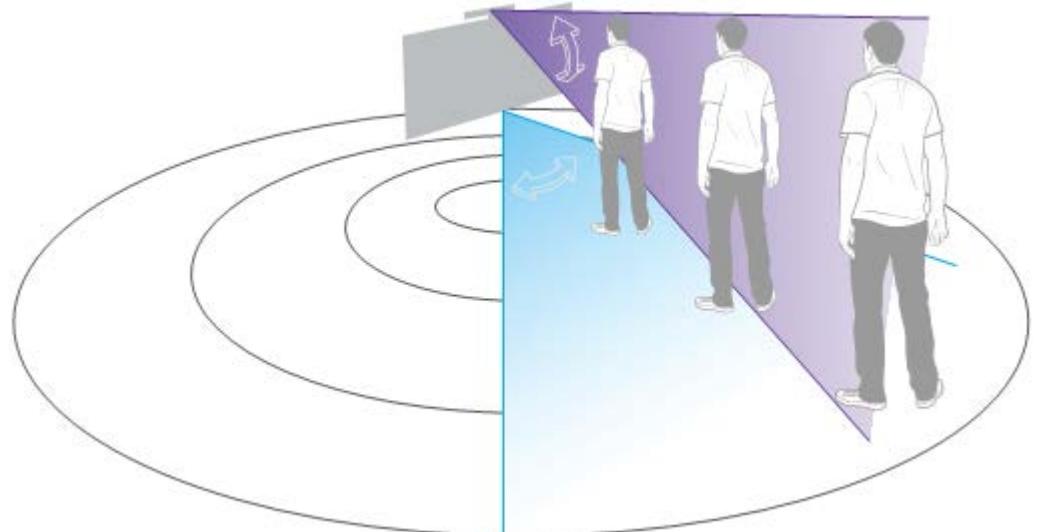
Kinect 能看到用户的多大一部分身体?
用户的头部是否在监控范围之内?



我如何判断做手势的时机和位置?



我是否已加入并处于监控之中?我现在是否可以挥手?



Kinect 的视域如何界定?

设计人机交互手势

虽然 Kinect 为我们带来了人机交互手势，并为我们正确理解它敞开了大门，但是，在设计上，也产生了新的问题。如下是我们用来让手势设计充满神奇魅力的一些重大发现和注意事项。

手势原则

当用户在您的应用程序中使用手势时，应遵循如下一些原则：

我能快速掌握全部基本的操控方法

当我做手势时，会感到非常舒服自然

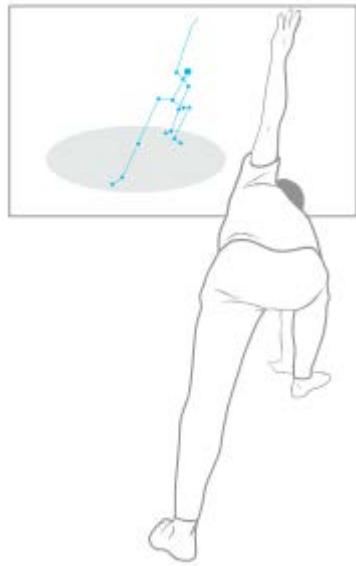
我只要学会一个手势，就可以快速而准确地加以使用

当我做手势时，系统会做出响应，并不断给出反馈

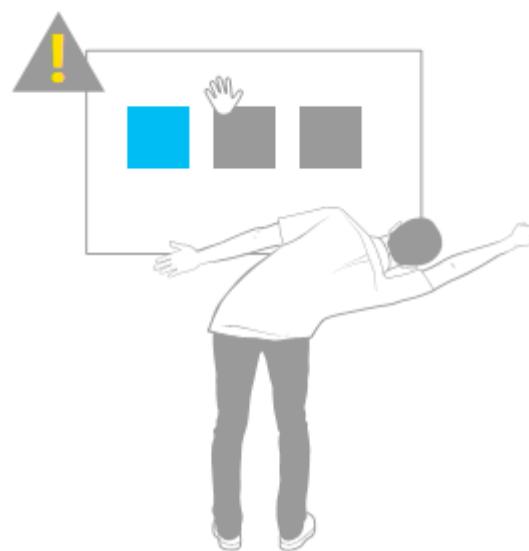
设计时应考虑用户的思维方式

人们仍经常将 Kinect 与游戏联系在一起，但是，当您在设计 Kinect for Windows 应用程序时，请记住：游戏的思维方式不等同于用户界面的思维方式。

游戏的思维方式：玩的就是难度！



用户界面的思维方式：难度让人产生反感情绪！



正确

- ✓ 如果用户置身于游戏的思维方式，而且无法做出任何手势，那么，就说明很难过关，用户也愿意再接再厉重新尝试

- ✓ 在游戏的思维方式下，看似愚蠢的手势可能是非常有趣或具有娱乐性

错误

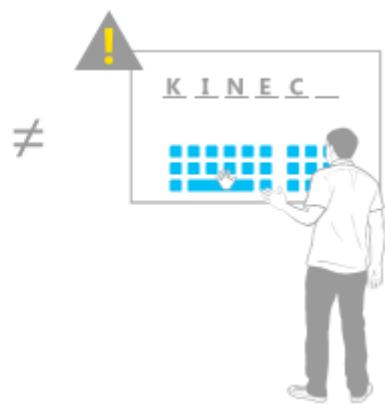
- ✗ 如果用户置身于用户界面的思维方式，而且无法做出手势，那么，他们会产生挫败感，而且几乎没有耐心去进行学习

- ✗ 在用户界面的思维方式下，看似愚蠢的手势确实会产生尴尬或不够专业的印象

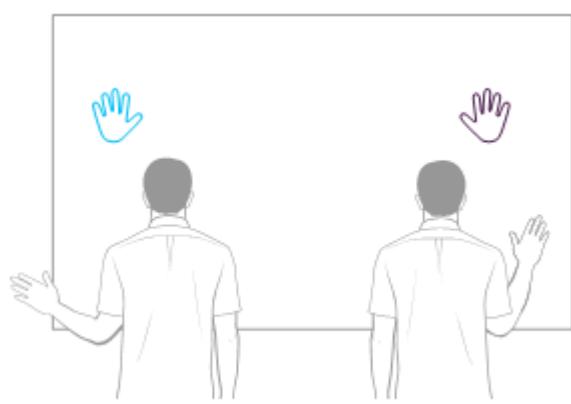
设计人性化的人机交互体验

请勿尝试在现有的 UI 上强行采用其它输入方式。例如，不要触摸界面，也不要将手势泛泛地映射到全部输入方法上。

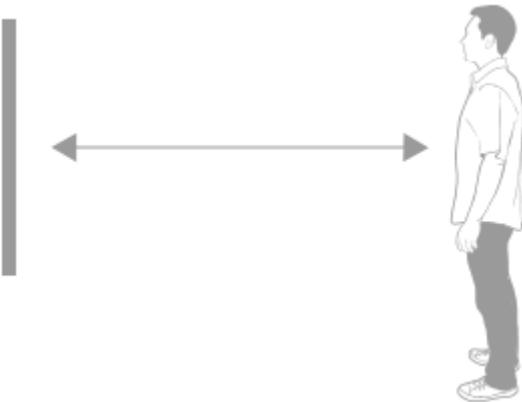
手势为您操控应用程序提供了一种炫酷的手段，但是，请记住，每个手势都应具有特定的意图。



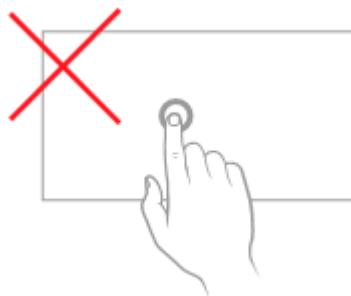
请记住，每个手势都应具有特定的意图



多位用户将会同时操控您的应用程序



您的设计方案需要为用户留出一定的人机交互距离



您的设计方案需要避免用户直接触摸到任何东西

正确

- ✓ 您的方案需要为用户留出一定的人机交互距离
- ✓ 手势可以提供其它输入设备所无法实现的互动体验
- ✓ 部分或全部人机交互体验均不使用任何中间介质，这样，自然而然的手势才显得“充满魔力”

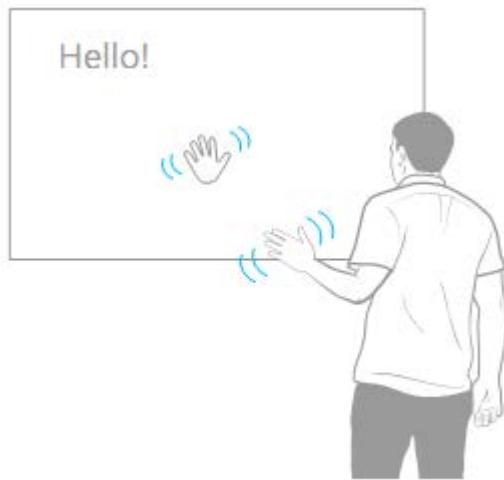
错误

- ✗ 全部手势均可被替换为另一种更快捷、更简便的输入方法
- ✗ 您的设计方案应专注于生产力、运行速度和精准度
- ✗ 当前用户界面直接继承自仅适用于另一种输入方法的用户界面

判断用户的意图和介入状态

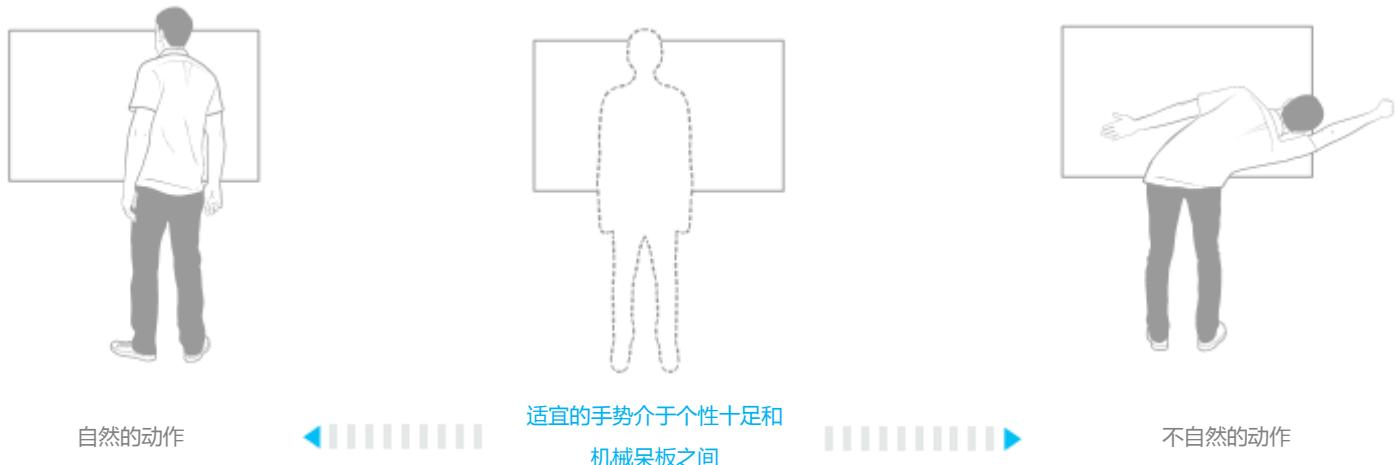
判断用户意图是一项艰巨的挑战，也是一个十分不易于解决的难题。与其它输入设备不同，您的用户“始终都处于活动状态”。

如果您的手指始终停留在触摸屏上，或者，您的手始终不离开鼠标，这该怎么办？

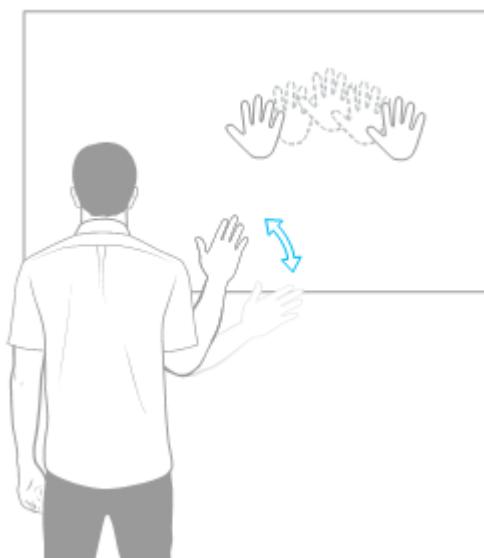


系统中负责判断用户意图的功能始终在判断用户何时打算加入系统。但是，对于某些方案来说，可能需要系统始终保持监听和监视状态，从而允许用户随时通过手势和语音来表明自己即将加入系统，这种机制有助于避免产生不必要的主动识别错误。如果当前用户必须主动触发系统对其介入行为进行识别，则必须显示出明确的触发手段，或者，需要在用户体验中给出某种明确提示。

正确	错误
<ul style="list-style-type: none"> ✓ 其它输入设备所无法识别的独特且意图明确的动作将被识别为手势/人机交互操作 	<ul style="list-style-type: none"> ✗ 习惯性的行为必须被改正，以避免由此产生人机交互
<ul style="list-style-type: none"> ✓ 用户可采用简单而明确的手段加入和退出 	<ul style="list-style-type: none"> ✗ 错过的激活：我做了动作，但系统没有识别出来
<ul style="list-style-type: none"> ✓ 习惯性的身体动作不会被识别为手势 	<ul style="list-style-type: none"> ✗ 错误的激活：习惯性的身体动作被误解为有某种意图的手势



要求用户挥手示意加入是我们所惯常采用的方法，而且该方法已被证明是一种用于判断用户加入意图的合理方法。用户需要来回挥手至少三次，并晃动时关节以便于被识别。挥手是自然而然的感官动作，通常不会被误解。这对于用户的另一层含义是，即将开始一次人机交互体验。



您还可以识别某位用户何时退出，以便显式地结束当前的人机交互，而不是等待超时或采用其它被动方法。例如：当某位用户没用正对屏幕时，将此行为确定作为他们退出当前人机交互的一种暗示。

设计时应考虑输入行为的非固定性

用户展现在系统面前的一些习惯性动作和操控意向将对他们的人机交互体验产生影响。请记住，对当前用户所做手势的解读可能完全不同于下一位用户。

预测用户意图具有相当巨大的挑战，这是因为并不存在诸如触摸屏或鼠标这样的物理介质可供用于检测用户意图。一旦用户已经加入到互动系统中，Kinect 将始终对其进行监控，以试图捕获同预定义手势相匹配的手势。

关键是要在具有特定意图的人机交互与其它动作之间加以区分

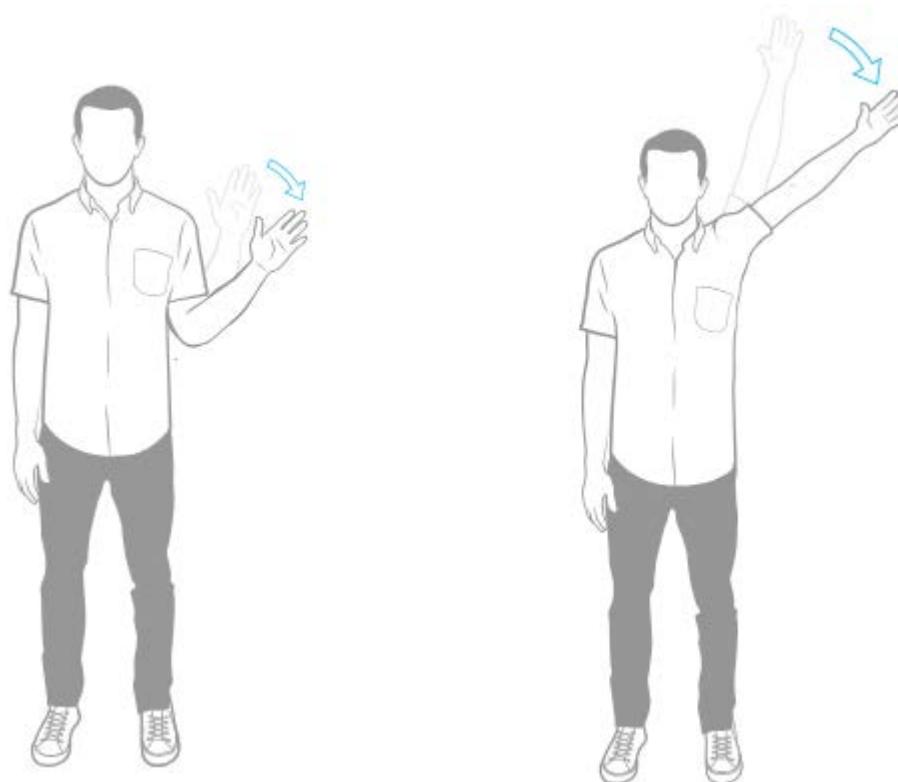
例如：摸脸、调整眼镜、喝水

为了增强用户信心，应始终在用户使用手势操控的同时给出反馈。

包括“抽象”阶段、“介入”阶段以及“完成”阶段。当某个手势已被取消或识别失败时，也必须给出反馈。

每个人都知道如何挥手，但每个人的挥手方式却有很大的不同。有些人用手腕挥手，有些人用手肘挥手，还有些人挥动整个胳膊。有些人张开手掌左右挥动，而有些人手指并拢上下摆动。仅仅是要求用户挥手并不能保证他们所做出相同动作。

这些图像展示了两种为大多数人所认可的挥手方式。



设计时应考虑真实性

真实性应该是您的第一要务。如果缺少这一特性，您的应用程序将会让人感觉反应迟钝且难以使用，您的用户也将受到挫败感的打击。

正确	错误
<ul style="list-style-type: none"> ✓ 用户知道如何有效地做手势 	<ul style="list-style-type: none"> ✗ 较差的真实性会导致用户选用其它的输入方式 / 人机交互手段
<ul style="list-style-type: none"> ✓ 用户可以示范或指导其他用户如何做手势 	
<ul style="list-style-type: none"> ✓ 用户可以快速学会如何做手势，以及当前手势在整套用户界面的类似上下文中发挥何种用途 	

尝试寻找一个良好的**真实性平衡点**：

手势的设计或定义将会影响真实性

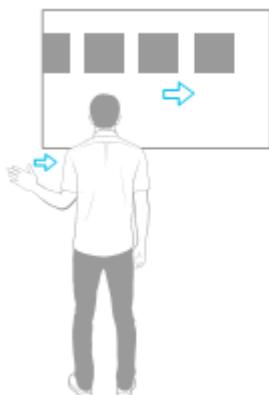
如果手势太过僵硬，将不会出现错误的激活，但是，可能会不易于做出这样的动作

如果手势太过灵活，将会易于做出动作，但是，可能会时常出现错误的激活或者与其它手势之间发生冲突

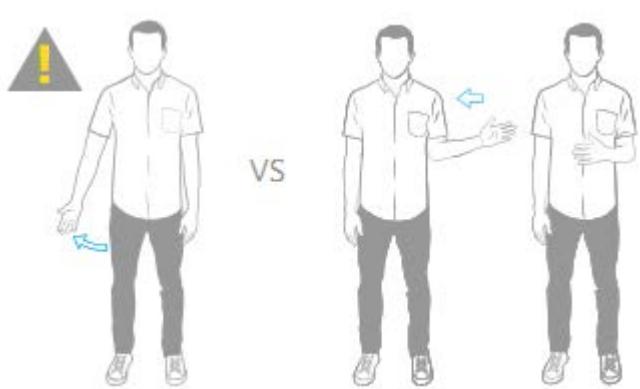
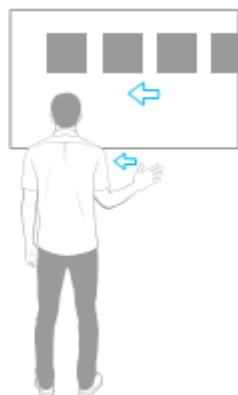
正确	错误
<ul style="list-style-type: none"> ✓ 错误激活的频率和次数将决定正确的平衡点 	<ul style="list-style-type: none"> ✗ 如果手势太过僵硬，用户的思维和行为就会过于拘谨

让手势适应用户的当前任务

恰当的手势可以表达某种意义，而且对应于用户界面任务或操作。系统的用户界面反馈应体现当前用户的身体动作。



向上抬手表示向左滚动内容



抽象的身体动作无法对应于当前任务，而且更不利于掌握和记忆

正确

- ✓ 向左滑动表示向左滚动或移动内容
- ✓ 恰当的手势更易于掌握和记忆

错误

- ✗ 向上抬手表示向左滚动内容
- ✗ 抽象的身体动作无法对应于当前任务，而且更不利于掌握和记忆

设计强手势集

要为您所添加的每种手势设计一个强手势集无疑会增加设计难度。因此，建议您尽可能压缩应用程序中所使用的手势数量，这样，用户既易于掌握和记忆，而且这些手势之间也容易相互区分，尽量避免混淆和冲突。尽可能同其它应用程序之间保持一致的手势风格，这样，有助于增强用户的亲切感，并从整体上减少手势的种类，从而缩短用户的学习和培训过程。

在定义手势集时，需注意如下一些事项：

确保相似的手势表达相同的意义。

例如：

如果您已定义向左滑动所代表的意义，那么，向右滑动自然就应代表向反方向。

使用明显的手势区别来从根本上做出区分，从而减少错误识别的几率，并避免存在重叠手势。

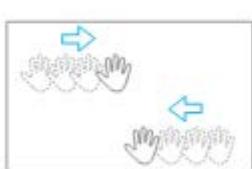
例如：

变向：如果两个手势的方向相同，它们就更有可能被重叠识别

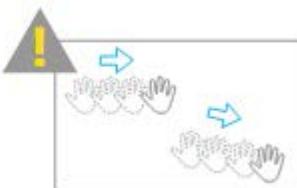
连续：特别是当两个手势趋于同向的情况下，应让这两个手势所经过的路径有所区别

起点和终点：起点和终点通常是被识别的重点信息，因此，尽量尝试多些变化

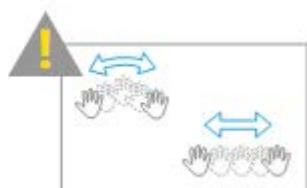
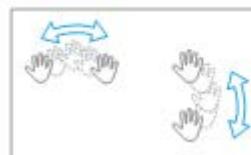
完成时间：某位用户完成一个手势所必须的时间长度可能很难控制，但是，还是应该表现出少许差异。



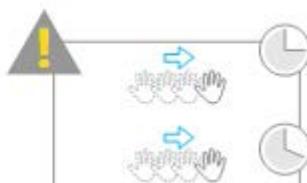
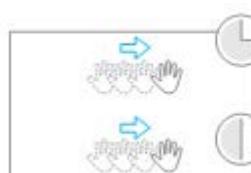
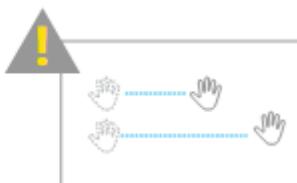
变向：如果两个手势的方向相同，它们就更有可能被重叠识别



连续：更改手势所经过的路径



起点和终点：通常是被识别的重点信息，应尽量尝试多些变化



完成时间：更改某位用户完成一个手势所必须的时间长度

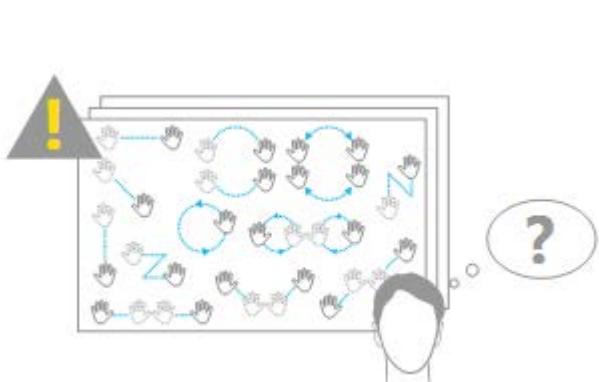
不要让用户记忆过多手势。让您的用户时刻感觉到操控方法并不费力。研究表明，用户最多只能记住 6 种手势。

从其它 Kinect 应用程序中的现有手势获取设计灵感

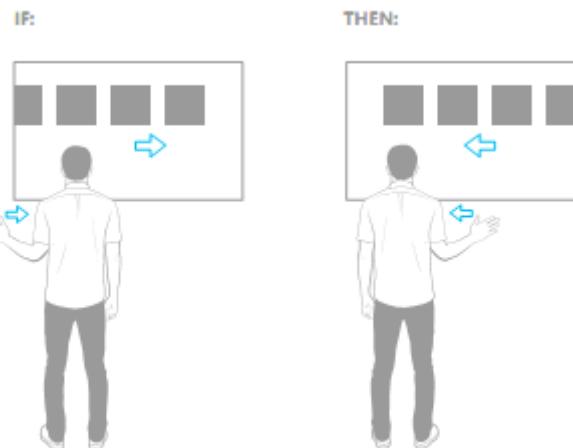
全面测试各种手势，以从中找出错误的主动触发手势

全面考量当前设计方案。当用户完成一个手势之后，会做什么动作？这个动作是否是一系列无意识动作的开始？您还可以在一个手势的结束时添加一些无响应时隙，在此期间的全部手势都将被忽略，从而允许用户在开始下一个动作之前还原到自然体位。

应用程序手势集中的每个手势都应具有感官上的关联性和衔接性。



全面考量当前设计方案..
当用户完成一个手势之后，会做什么动作？



确保使用类似的手势。如果您已定义向左滑动所代表的意义，向右滑动自然就应代表向反方向

正确	错误
<ul style="list-style-type: none"> ✓ 用户可以轻易反复使用手势 	<ul style="list-style-type: none"> ✗ 向左或向右滑动代表左移或右移内容，而圆圈动作则代表上移和下移
<ul style="list-style-type: none"> ✓ 用户可以举一反三，并轻易掌握各种手势 	
<ul style="list-style-type: none"> ✓ 向左滑动代表左移内容，同理，向右、向上、向下滑动就代表右移、上移和下移内容 	

从容处理重复手势

用户是否愿意或需要接连重复使用某个建议的手势？

例如：

多次滑动代表快速遍历多页/多幻灯片内容

警惕一些常见问题：

忽略某个重复手势中的“归位”动作

设计对称手势也要考虑到前一个手势中的“归位”动作

正确	错误
<ul style="list-style-type: none"> ✓ 一连串重复手势应连贯，不应夹杂突然的开始或停止 	<ul style="list-style-type: none"> ✗ 用户界面反馈会起到阻碍作用，而且会妨碍或限制当前的节奏感 例如：用户觉得他们需要等待动画结束才能重复该手势
<ul style="list-style-type: none"> ✓ 用户可以快速习惯动作的节奏 	<ul style="list-style-type: none"> ✗ 手势设计让重复动作效率低下

避免单手手势

单手手势是指仅可由某一只手来完成的手势。

例如：

只有左手才能完成手势 A。此类需求通常仅限于从技术角度提高性能。

我们有何最佳建议？让所有的手势都能通过任何一只手来执行。

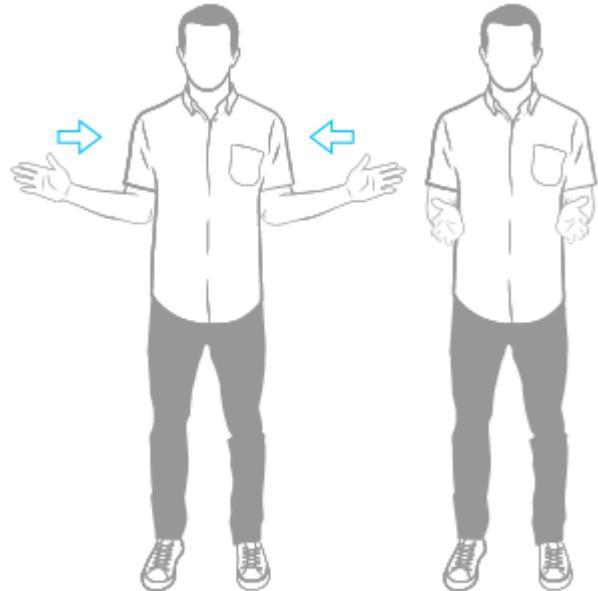
正确	错误
<ul style="list-style-type: none"> ✓ 用户可以交替使用两只手，以缓解疲劳 	<ul style="list-style-type: none"> ✗ 单手手势：降低可识别性和可轻松访问性
<ul style="list-style-type: none"> ✓ 同时满足习惯于使用左手和右手的用户 	

在单手手势与双手手势之间进行选择

单手手势是首选的手势，有利于提高效率和完善轻松访问特性。双手手势应采用对称设计，以便更容易执行和记忆。



单手手势更易于识别、掌握和记忆



双手手势应采用对称设计，以便更容易执行和记忆

正确

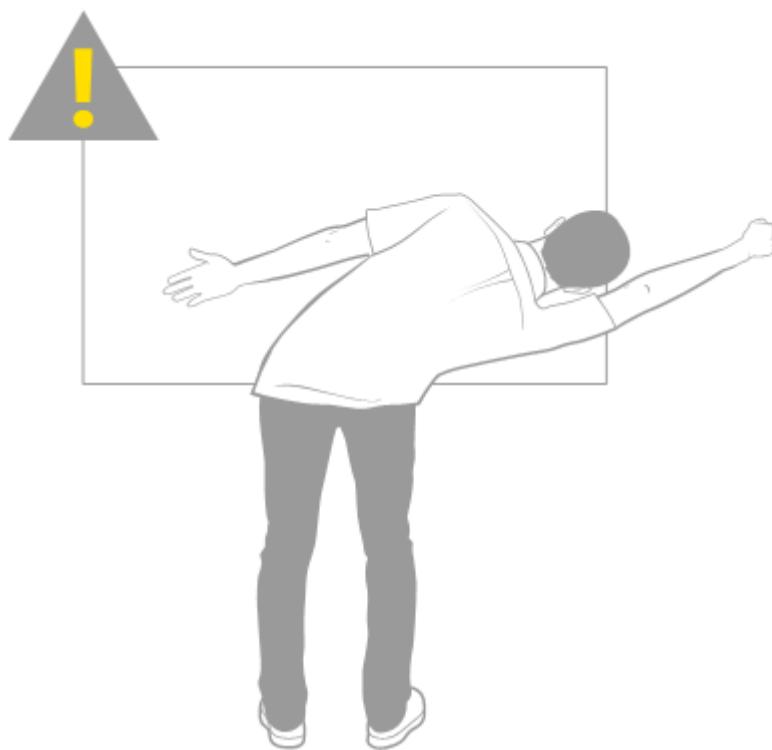
- ✓ 单手手势适用于全部关键路径任务
- ✓ 双手手势适用于高级用户、非关键任务

错误

- ✗ 双手手势适用于频发的关键任务

时刻谨记，疲劳是任何手势的天敌

如果您的用户因做某个手势感到疲劳，他们就会认为用户体验糟糕透顶，并很可能放弃使用。



疲劳增加思绪混乱 → 表现不佳 → 挫败感 → 糟糕的用户体验

正确

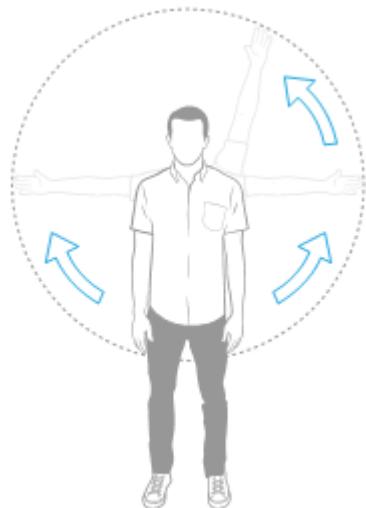
- ✓ 提供替换方案：列表导航（短距 / 长距跨度）、列表跳转、放大或缩小、列表删减或筛选

错误

- ✗ 单个手势的过度连续频繁重复，例如翻页（以遍历某个冗长的列表）
- ✗ 高举手臂，以示向上滚动一个冗长的垂直列表

考虑用户姿态和动作幅度

用户姿态可能会影响到手势的设计，因此，要考虑到您的用户将在何种场地中以及如何使用您的应用程序。例如，坐姿将会限制用户的动作。还要记住，人体的人机交互动作幅度应保持在正常和舒适的范围之内。



另外，如果您从侧面观察某位用户，还要注意真实性的损失程度。未包含在全景视图中的关节将被 Kinect 骨架跟踪系统放置在一些推算的位置上，因此，这些位置并非始终正确。

正确

- ✓ 在设计任何一种手势时，都应使其适应于全部可能的姿态

错误

- | |
|---|
| ✗ 1 个命令对应 2 种手势 坐立和站立姿态分别需要不同的手势 |
| ✗ 一种手势仅适用于一种姿态，而很难使用另一种姿态 |

提供有关手势及其可识别性的辅助指导

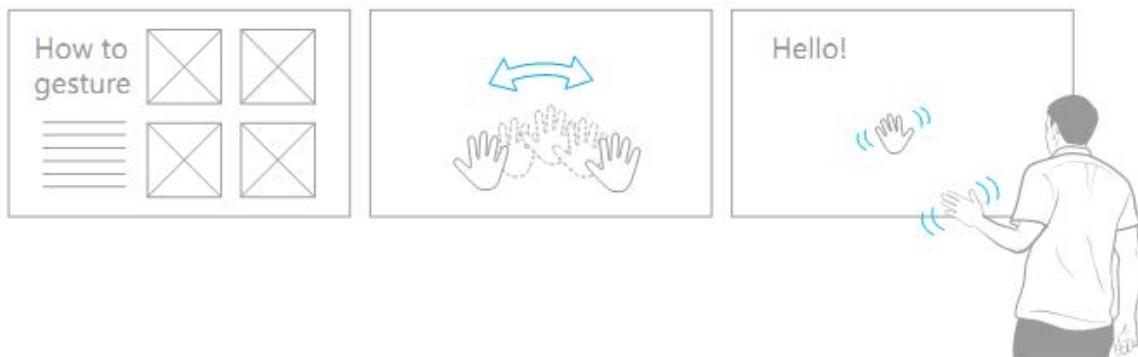
无论您提供何种手势来同系统进行互动，都应随时通过某种途径来告知用户其使用方法。您可以找到很多途径来达到这一目的，其中包括：

面向新用户的快速上手指南

一副静态图像

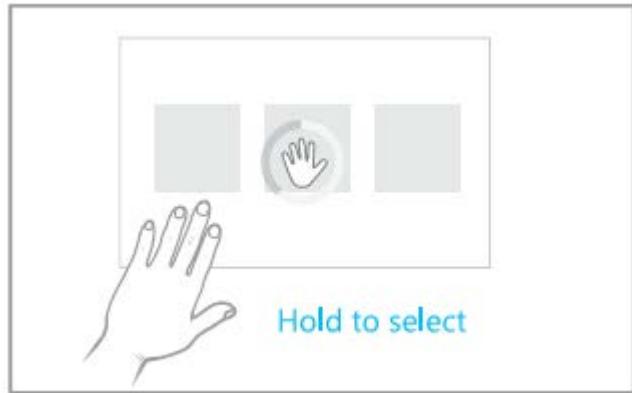
当用户首次加入时，或者，在整个人机交互体验的过程中，
给出视觉指引

一个动画

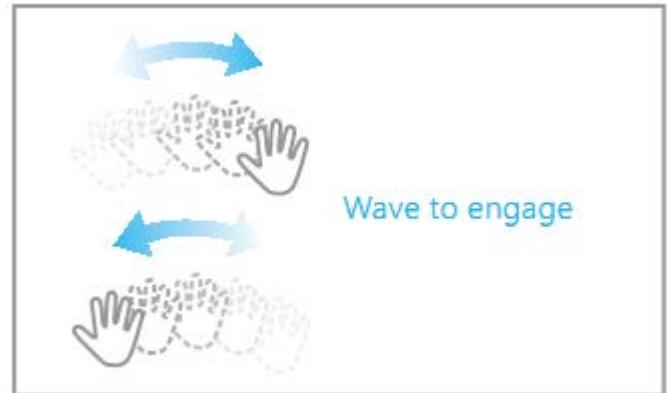


针对您要教授的手势，给出相应的指导方法。如果手势是静态的，就使用一副静态图像。如果手势是动态的，就使用动态方法或动画来向您的用户给出演示说明。

静态



动态

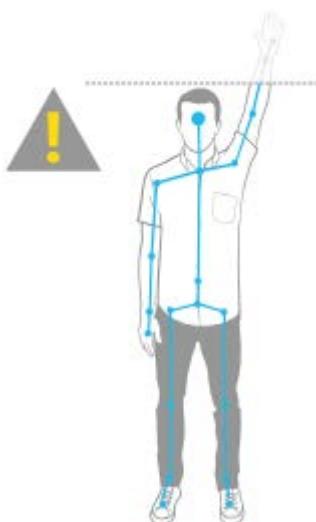


警惕技术障碍

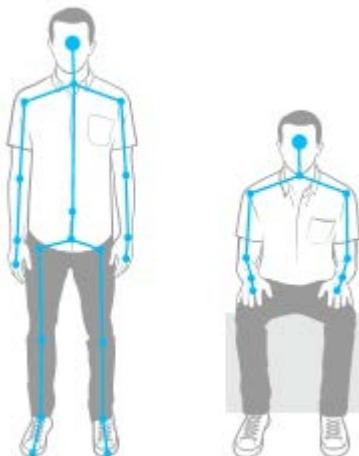
如果您在使用骨架数据来定义手势，应留意这些数据的局限性以及在何种条件下不可采纳。



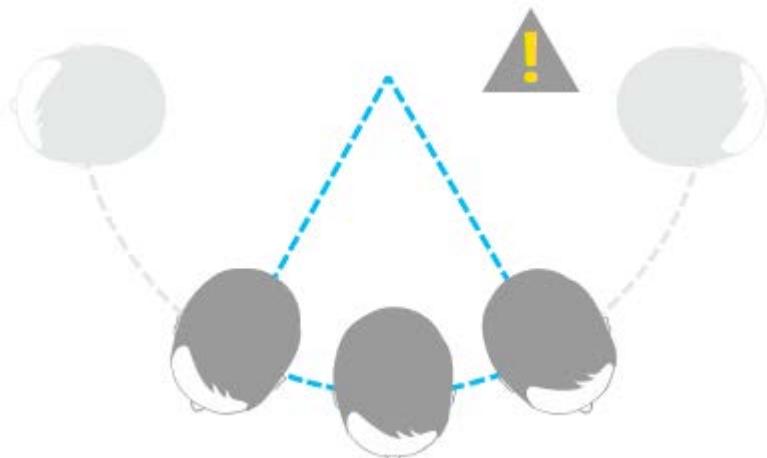
在做手势时，应保持胳膊和手部动作出现在身体两侧，以便于被追踪，反之，如果手部在身体前方运动，就可能无法被正确识别



将手伸到头部上方可能会导致关节点丢失，如果 Kinect 视域未经专门设置，就无法看到用户头部以上



在默认姿态下，可跟踪 20 个关节；在坐姿下，可跟踪 10 个关节；但是，在这两种姿态下，均无法跟踪手指。



当用户面对传感器时，骨架跟踪运行最稳定



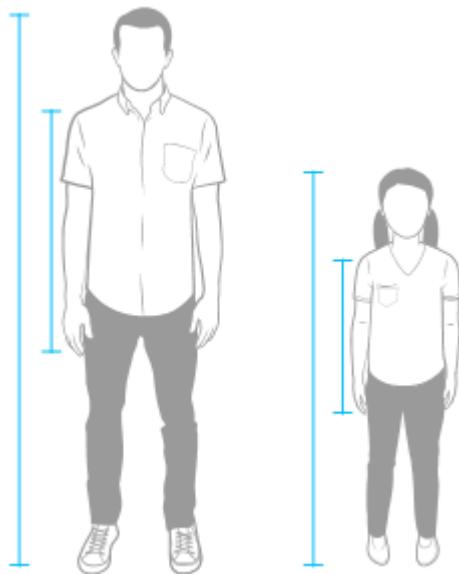
对于运动速度非常快的手势而言，应考虑到骨架跟踪系统每秒的速度和帧数限制。

考虑目标受众的要求

无论您如何定义手势，都要时刻考虑到目标受众的具体情况，以便让手势适应用户的身高范围和认知能力。需要考虑的因素包括用户可能采用的最长距离、用户可能采用的体位角度以及您打算要支持的各种身高尺寸。

例如，如果您的应用程序将同时被成人和儿童使用，您就需要考虑到各种可能的身高和臂长。与成人相比，儿童的身体柔韧性和自控力均有所不同，因而，在执行相同的操作时，他们也会表现出十分不同的动作。

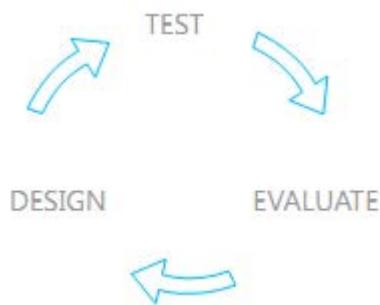
在做相同的手势时，儿童的动作会趋向于速度更快、幅度更大或姿态更夸张。



参数化与迭代

要让手势的动作比较舒适，可能需要经过许多次重复测试和调整。

将您能想到的和用户经常需要测试的各个方面全部实现参数化。



正确	错误
<ul style="list-style-type: none">✓ 手势应尽可能适应于整个目标用户群	<ul style="list-style-type: none">✗ 手势仅适用于您自己，而非其他任何人
<ul style="list-style-type: none">✓ 手势应尽可能自然和易于识别	<ul style="list-style-type: none">✗ 您的应用程序不够灵活，因此，很难做出调整

语音

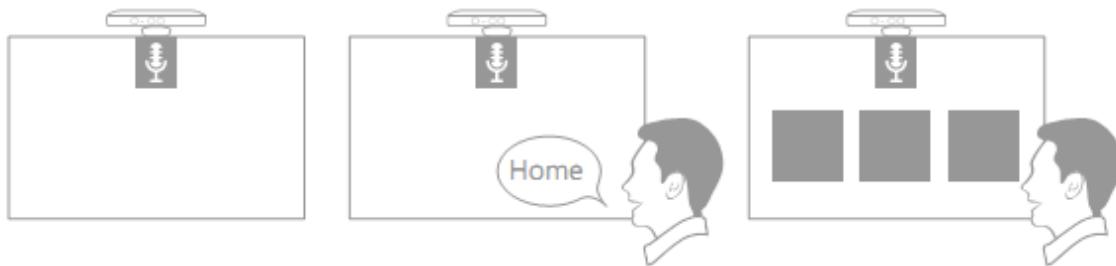
语音是另一种输入方法，可以开启全新的人性化操控体验，但是，目前仍存在一些技术上的局限性。通过在您的应用程序中使用语音，就可以允许您对一些特定的词语或短语进行监听，并将其作为触发器使用。口述的词语或短语只是交代命令，而非与之对话，因此，可能并不算是一种人性化的互动手段，但是，语音输入手段已得到良好设计和集成，完全可以让用户体验变得快捷，并增强您对用户意图的理解能力。

介入模式

在 Kinect 中使用语音时，可采用 2 种主要的介入模式：

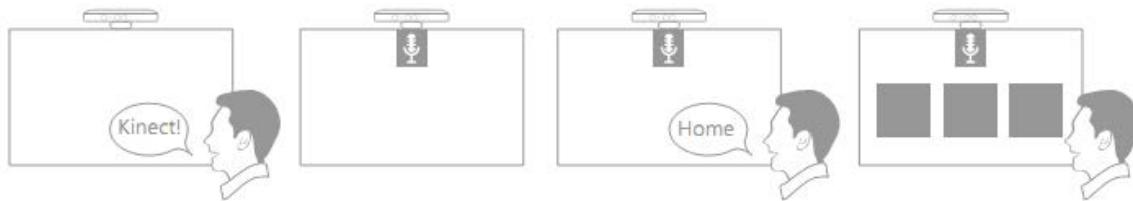
始终保持监听状态

Kinect 将始终监听您所定义的词语或短语。如果词语或短语相互区别较大而且数量较少，这种方法运转良好；但是，随着数量越来越多，很可能会出现错误识别的现象。这也取决于在应用程序的运行过程中您打算为用户提供多少语音输入的机会，这在很大程度上取决于您的具体环境和设计方案。



关键词 / 触发器

Kinect 只监听单个关键词或触发器词语。一旦它听到该词语，就会继续监听后续的其它指定词语或短语。这是减少错误识别的最佳方法。您所选的关键词必须非常独特，以防被误解。例如，在 Xbox360 中，“Xbox” 是关键词。与 “Xbox” 发音相似的词不多，因此“Xbox” 是一个代表性很强的关键词。



选择词语和短语

在选择使用哪些词语或短语时，需要谨记几点重要事项：

尽可能地选用简单常见的词语，以便获得更为自然的使用体验并且更易于记忆

保持短语短小精悍（1-5个词语）



DIAPHANOUS
VERDANT
ZENITH

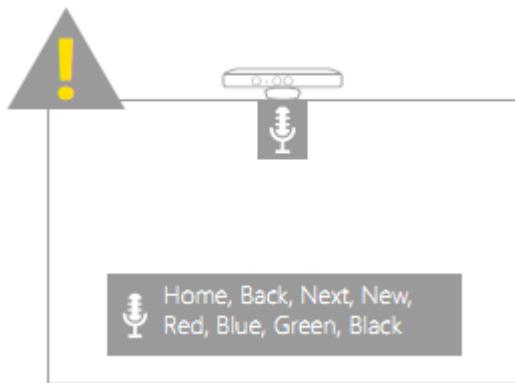
LIGHT
GREEN
TOP



"SHOW ME THE ONE ON THE LEFT"
"PUT THIS IN THE SHOPPING CART"

保持每屏的短语或词语数相对较少

因为单音节词较易重叠，请格外当心



CAT
DOG
FUN
TOP

避免头韵、押韵的词语、共同的音节长度、共同的元音，不同词语中出现相同的词语



Alliteration

CAR
KITE
CORE
COOL
KEY

Rhyming

MAKE
SHAKE
RAKE
TAKE
FAKE

Common Syllables

FERRY
BABY
WATER
COOKIE
CARSEAT

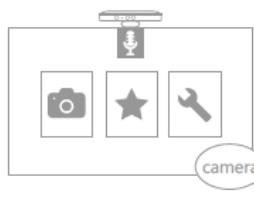
Common Vowels

MARE
CARE
FARE
PEAR
RARE

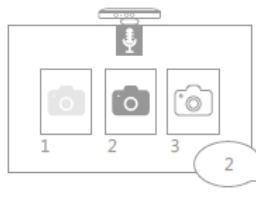
Same words, different phrases

ONE MORE TIME
SHOW ME MORE
MORE RED
NO MORE RED
PLAY MORE LIKE THIS

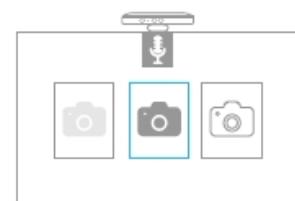
如果任务过于复杂或者基于非文字内容，请考虑使用数字来映射屏幕上的选项。



说：“Camera、Star 或 Wrench”

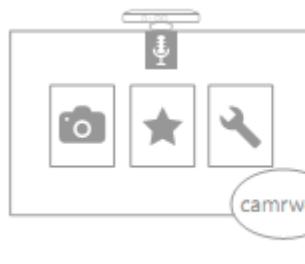


说：“Camera”，然后，说“2”

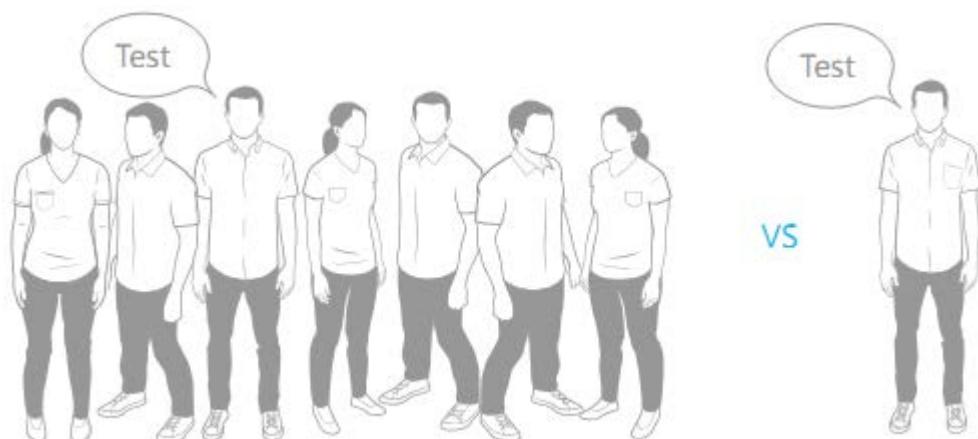


对于某些容易导致误识别的命令而言，提供某些操作提示，将有助于正确识别命令。

例如：“did you mean ‘camera’？”

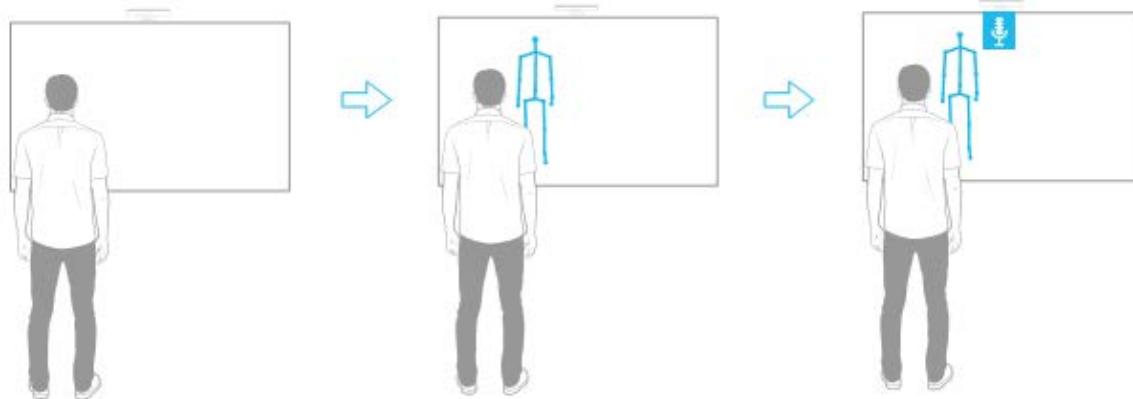


如果您打算在某个指定的视听环境中使用当前应用程序，就需要在该环境中对词语和短语进行测试，以确保其可被正确识别。



尝试使用某个触发器或事件来让当前系统启动监听。

例如：仅监听某个特定区域内所检测到的骨架



在降低错误激活概率的前提下，适当增强用户信心。这可能会导致较低的识别率，但是，却可以减少用户的挫败感。

反复测试和灵活调整。如果某个特定词语总是失败或被错误地识别，应尝试使用新的方法来对其做出说明

设计语音用户界面

虽然可以尝试并强制用户记住某些词语或自然而然地发现它们，但是，我们一般并不建议这样操作。下面列举了一些用户界面的最佳实现方法，这将帮助用户理解他们应如何使用语音以及可供使用的词语或短语。



基本原则：

对于任何一个语音命令而言，用户应能在屏幕上清楚地看到有关该命令使用方法的文字说明，只有按照说明操作，才能确保命令被正确识别。

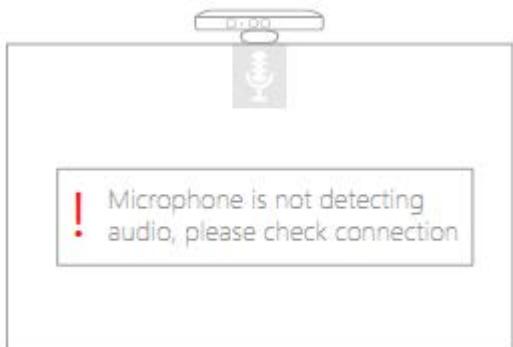


在画面上表明当前应用程序正在“监听”命令。

例如：Xbox360 使用一个麦克风图标来表明当前应用程序正在监听命令



如果您使用某个关键词，请在当前屏幕上显示该关键词，或者，在一开始就引导用户学习某个教程



如果当前界面已启用语音，但麦克风连接存在问题，那么，就应显示一个错误图标或消息，以便当前用户得知该问题并加以解决

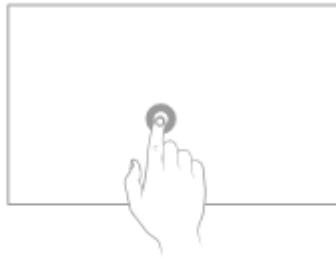


如果因为某种原因，当前用户无法面对屏幕，那么，就应向用户大声朗读当前可用的一系列短语

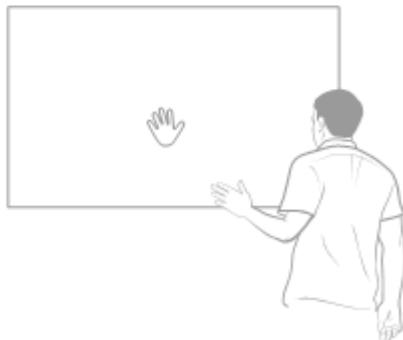
IF:



THEN:



可供用户用来操控应用程序的方法不应仅有语音一种。在语音无法正常工作或失去保真度的情况下，应向用户提供另一种输入方法



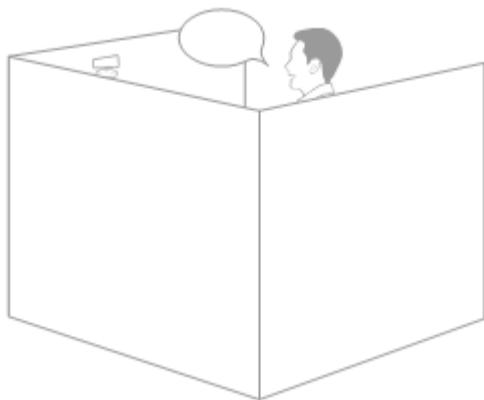
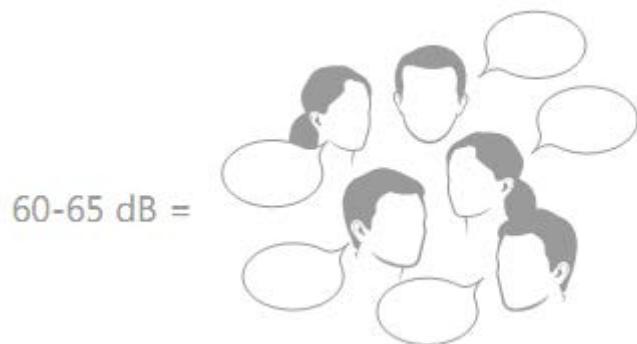
选择恰当的语音环境

许多环境因素都可能会对您是否可以使用语音操控应用程序产生巨大影响。

环境噪音：

Kinect 将集中监听音量最大的声源，并尝试抵消其它环境噪音。这意味着，如果当前房间内有其他对话（通常为 60-65 dB 左右），那么，这必将降低语音识别的准确性。

将环境噪音提高到商场或餐厅的声级，您就可以想象，在这种环境中识别命令会更为困难，即便是简单命令也不例外。在一定程度上，环境噪音是不可避免的，但是，如果您的应用程序必须在嘈杂的环境中运行，那么，语音可能并不是最佳的人机交互手段。理想情况下，只有在下述情况下才可使用语音：



当前环境较为安静并相对封闭，例如办公室。



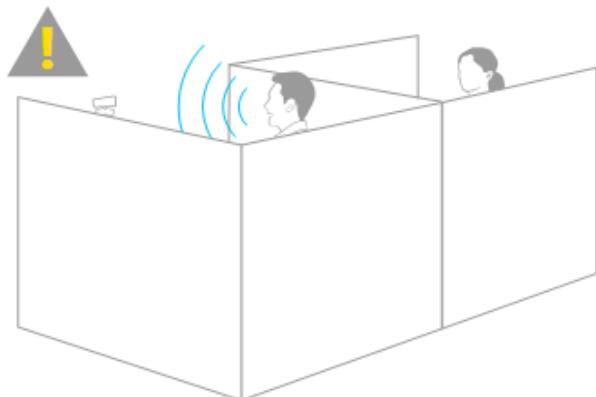
不会有多人同时发言。

系统噪音及其屏蔽方法

虽然 Kinect 传感器有能力屏蔽更为嘈杂的噪音，但是，如果某位开发人员打算亲自开发这种支持，Kinect for Windows 的内置功能仅能屏蔽单声道声音，例如系统提示音，而无法屏蔽立体声。也就是说，即便您知道您的应用程序将播放某段特定音乐或这段音乐将在室内播放，您也不能发出任何声音。另一方面，如果您打算使用单音调提示音来向您的用户告知一些事情，这是可以接收的。

社交注意事项

在使用您的应用程序时，请时刻谨记用户需要大声说出命令而产生的社会影响。请确保命令符合当前场合，这样，您也就不会强迫您的用户说出某些可能会使他们感到尴尬的事情，此外，请确保用户说出命令的音量适合当前场合。例如，在一个办公室隔间内大声说出命令可能会分散他人注意力，而且不适合这种场合。

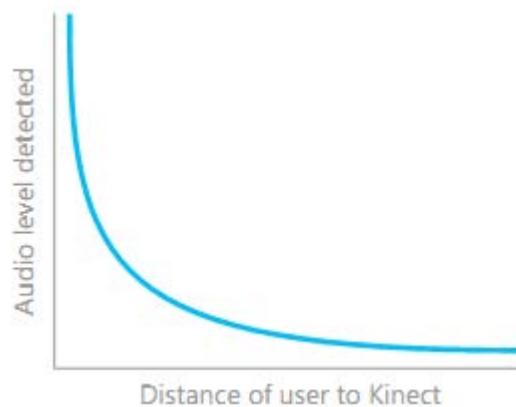


在一个办公室隔间内大声说出命令可能会分散他人注意力，而且不适合这种场合。

用户与 Kinect 之间的距离

当用户逐渐远离 Kinect 时，用户的语音音量也会成平方倍降低。也就是说，当用户非常接近传感器时，音量将会非常高，但是，随着用户逐渐远离传感器，音量也会迅速下降并变得几乎听不到，从而就会导致识别失真或需要用户提高说话音量。当用户逐渐远离传感器时，环境噪音也会影响 Kinect 听清用户命令。您可能需要略作调整，以便找到可清晰识别正常语音音量的设置方式与既定环境之间的“折中点”。

在环境噪音较低和 PC 噪音较柔和的环境中，无论用户距离远近，都应能以普通或较低音量（49-55 dB）自然地使用语音。



当用户逐渐远离 Kinect 时，用户的语音音量也会成平方倍降低。

反馈

无论是使用手势或语音，或者两者兼而用之，提供良好的反馈是让用户感受到一切尽在掌控的关键，同时，也有助于他们了解正在发生的一切。如果用户所在的位置距离 Kinect 较远而且不易直接接触界面，那么，您在为用户显示他们的操作与应用程序之间的映射关系时就要格外小心。本节内容将介绍一些用于帮助您尽力强调反馈信息的手段。

基本的视觉反馈

清晰界定可操作项及其操作方式



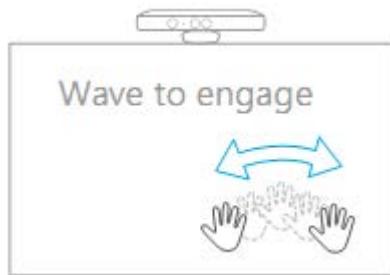
使用图示、颜色或教程来向用户明示如何区分可操作的控件与说明性的文本和内容



使用图示或教程来向用户明示当前可用的输入方法



告知用户哪些命令可采用语音来进行操控



告知用户哪几种手势可供使用。

如果 Kinect 出现状态或连接错误，系统会显示错误消息或状态，并给出解决问题的建议。



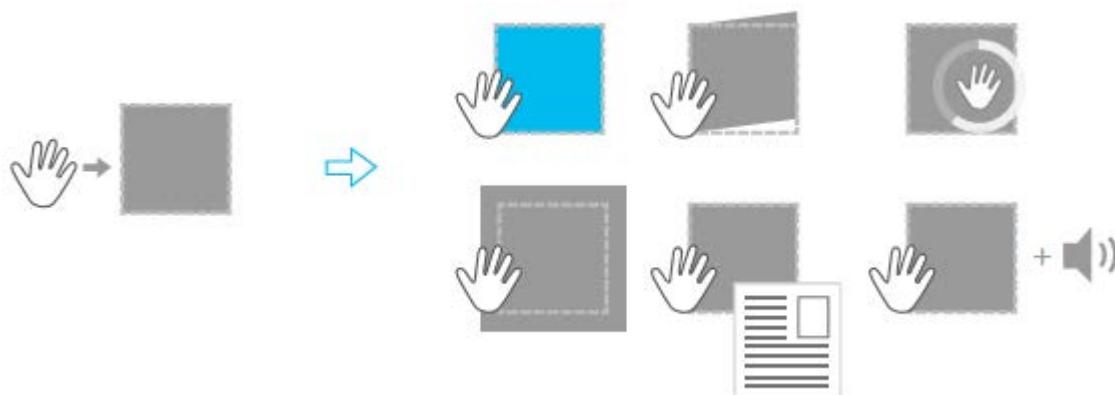
如果您正在跟踪用户的的手部，应在画面中显示游标

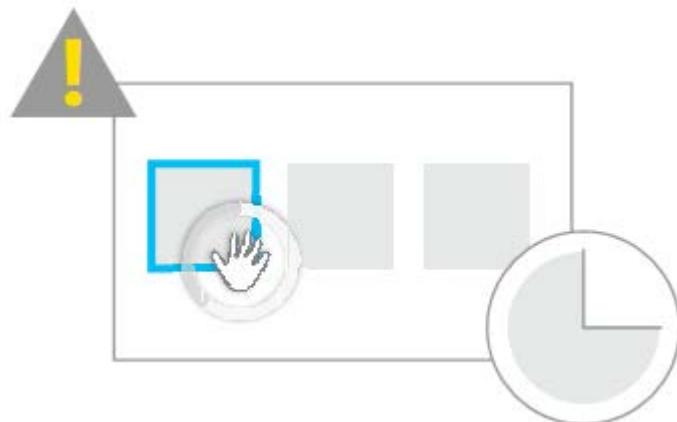
如果用户因 Kinect 的错误状态而被阻止操作，应通知用户并切换当前画面。请务必切换用户界面。我们建议用户切换到备选的输入方法。

悬停状态和选择状态

对于无法导航的控件而言，例如复选框和切换按钮，应使用明显的外观变化以示状态更新

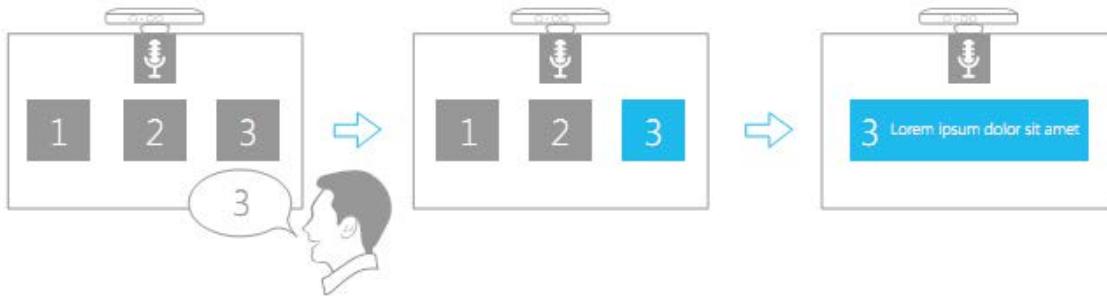
在悬停和选择状态下，使用颜色、倾斜、方向、大小或内容更改，以便就哪些项目是可操作项以及用户正处于何种状态来向用户给出反馈





在某些情况下，也可以伴随选择状态显示进度提示，但是，一定不要因强制拖延人机交互的时间而导致用户烦躁，因此，合理界定一个时间长度极为重要

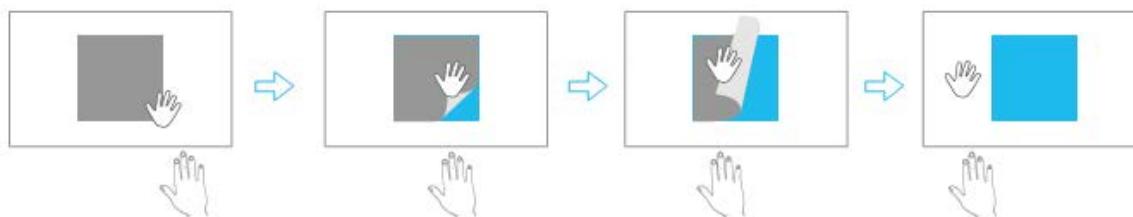
如果当前用户使用语音或手势来执行选择操作，应在更改该应用程序状态之前对识别进度的画面提示进行更新



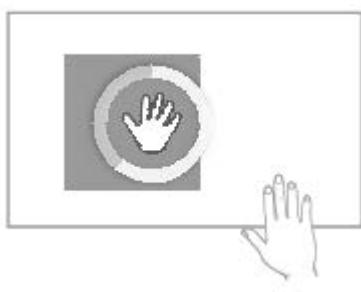
直接操控

如果您打算允许用户采用直接操控方式来执行操控任务，应在用户做出动作与该动作发挥作用之间显示进度提示

例如，如果当前用户正在使用手势翻页，应随着用户水平移动手部的过程，显示页面掀起和翻转的过程

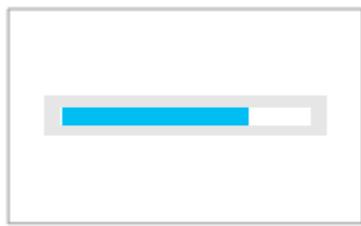


进度



在界定手势动作的执行进度是否超时方面，应考虑到用户保持某种姿势多长时间较为舒适，以及用户愿意重复这一人机交互操作的频率。请勿强制用户等待过长时间。

如果您打算使用进度、定时器或倒计时，应使用清晰的画面来显示整体进度



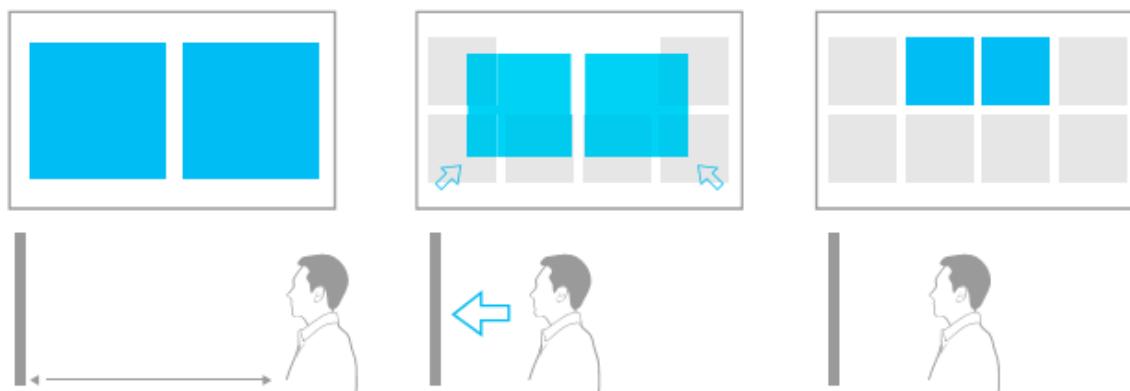
如果当前进度中需要用户做出一项重要选择，应确保此举实属必要，并将选项放置在用户的视觉中心

如果您预计应用程序的执行过程中会出现延迟或等待时间，应考虑使用用于提示正在加载或当前进度的用户界面，以避免应用程序给人以无法响应的假象

动画

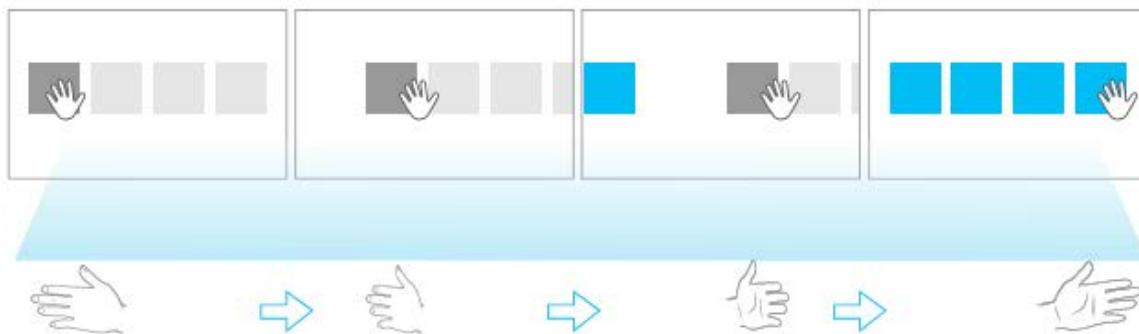
动画是一种用于帮助用户建立前后情景关联的绝佳手段

如果您打算根据当前用户无意触发的某些操作（例如距离变化）来更改当前用户界面，应尝试使用动画来提示用户刚才还在看的内容位于当前新画面布局中的哪个位置

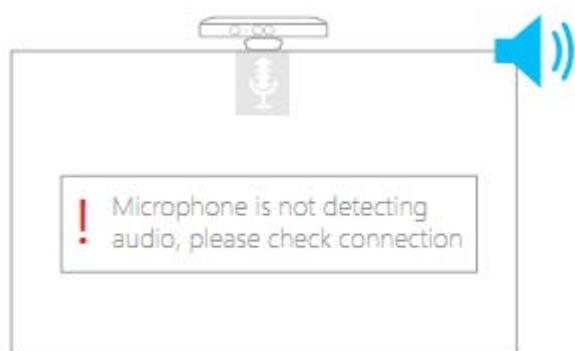


如果您的用户正在导航，应使用动画来帮助用户理解当前画面的布局

例如，用户的水平手势动作对应于应用程序画面布局空间中的水平移动。这种映射关系也应体现在您所使用的其它全部手势上。



音频反馈



抓住用户的注意力

如果用户需要获得通知，音频绝对是抓住用户注意力理想方式
例如：警告或提醒

预定义的声音模式或提示可被用于传达一个简单的消息或提醒
用户注意

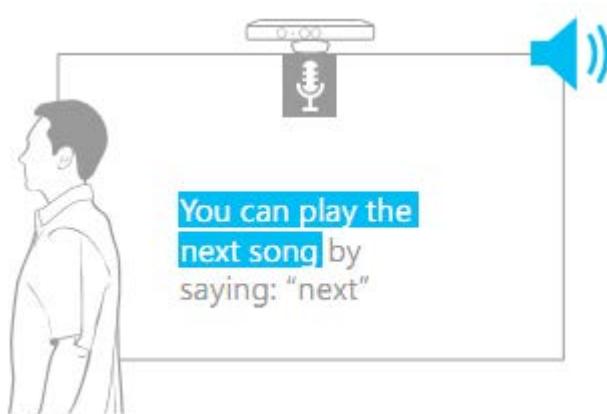


悬停状态和选择状态

声音可以很好地提示用户某些事情已经改变或发生

尽量在用户执行某项操作时给出相应的音频反馈，以便让用户感觉到一切都不陌生而且尽在掌控之中

例如：在输入文本时，给出敲击键盘的声音



按用户所面对的方向给出音频反馈

如果您的用户没有面对屏幕或者远离屏幕，应考虑使用音频来发出提示

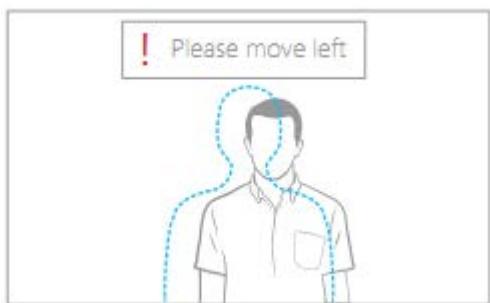
例如：当前可用的选项、方向或操作说明、提醒

骨骼跟踪系统的反馈

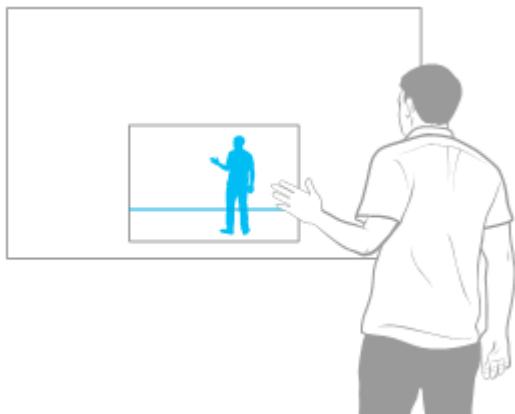
确保用户知道 Kinect 是否可以看到他们

如果您需要跟踪某位用户，但又无法“看到”或者说跟踪他们，应告知用户正确的站立位置，以便您对其进行跟踪。

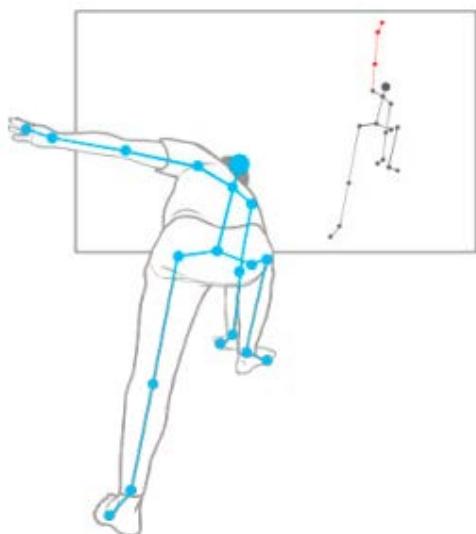
如果您在人机交互中途失去跟踪目标，应首先暂停，并引导用户站回可跟踪到他们的位置上。



Kinect Sports 就是一个典型的示例，在 Kinect Sports 这款游戏中，系统会在一开始就告诉您站位是否过近，并给出最适合于跟踪的站位。



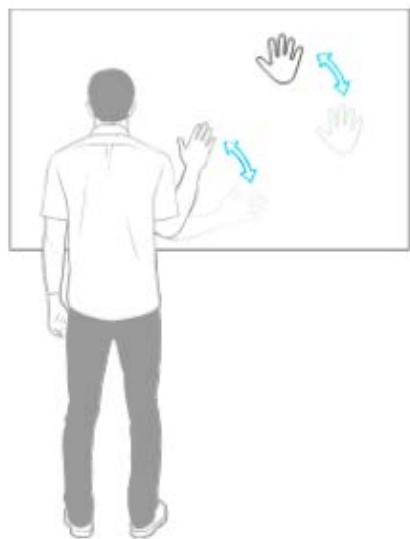
实现这一目的的另一种方法就是显示一个小型的景深跟踪窗口，或者，直接在画面上精确显示 Kinect 当前所看到的用户轮廓。这也许可以帮助用户理解如何保持恰当的距离，以及当他们的站位超出范围时系统无法正常工作。



动作示范

如果您打算让用户照做某个特定运动或动作，可以通过播放一个虚拟示范或动画的方式来实现上述目的，当然，也可以在开始跟踪用户动作或跟踪过程中给出这样的动作示范。如果您在用户动作过程中播放动画，应尽可能直观地指出他们所做动作是否正确。

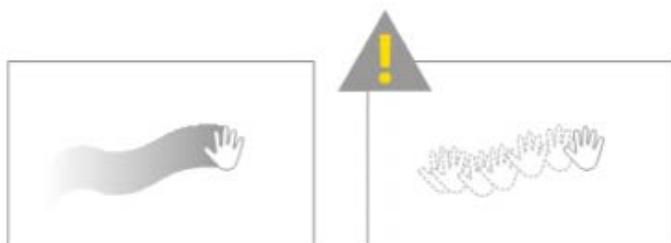
Dance Central 就是一个典型的示例，在 Dance Central 中，屏幕上的舞者做出正确的舞姿，并同当前用户相互独立，但是，当用户出错时，肢体就会以红色突出显示。



动作镜像

如果您打算提供一个同用户具有镜像关系的虚拟人物，应时刻谨记如下原则：

确保动作实时显示，以便让用户感觉到一切尽在掌控



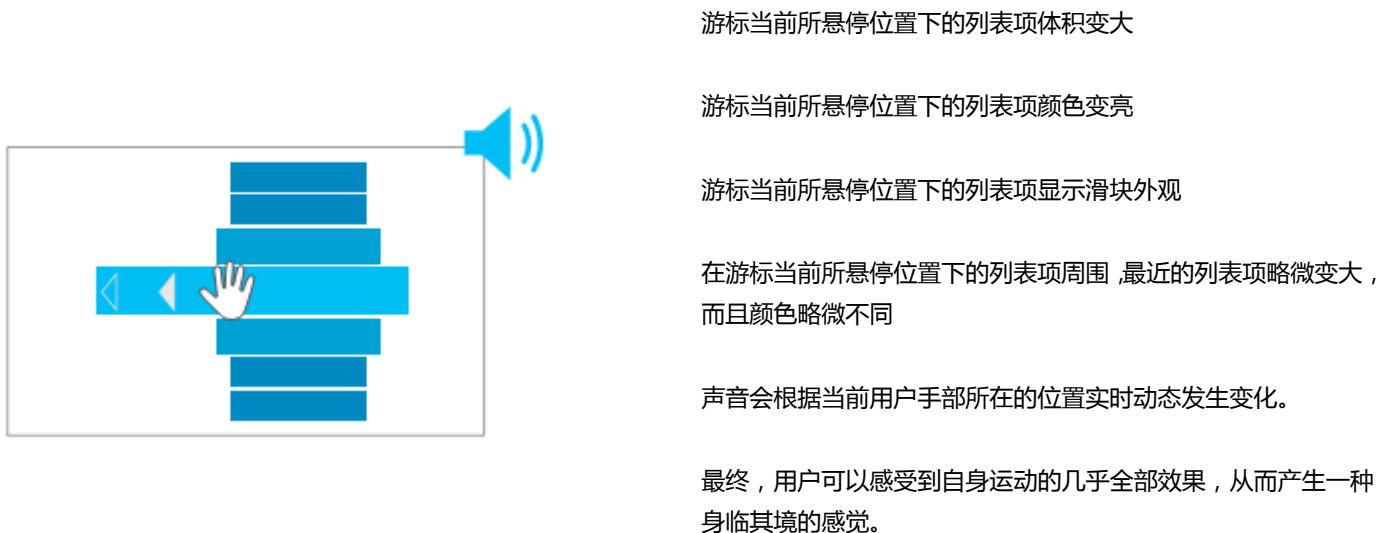
舒展的动作可以避免紧张或不自然的姿态和运动方式

让当前场景中的其它物体在同骨架或虚拟人物进行碰撞时产生正确的碰撞效果

综合反馈

在现实世界中，我们会用到自身所具有的全部感官来度量我们自己的动作所产生的效果。通过将各种不同类型的反馈结合在一起，通常就可以在虚拟世界中创造出一种更加真实的体验。

通过将视觉和音频输入方法结合在一起，就可以产生更强烈、更真实的反馈。最具代表性的示例就是 Dance Central 中的菜单。随着用户向上和向下滚动菜单，就会看到如下变化：



基本的人机交互操作

即使是最神奇的 Kinect 体验，也需要由最基本的人机交互操作来构成，而实际上，这些基本的人机交互操作也存在于鼠标、键盘和触摸设备上，例如定位并选择按钮、输入文字、缩放和平移某个界面以及让系统随时得知有用户打算加入。

在接下来的内容中，我们将介绍目前已知的人机交互知识，并针对如何设计适用于手势操控的人机交互体验给出一些建议。

设计强输入方法

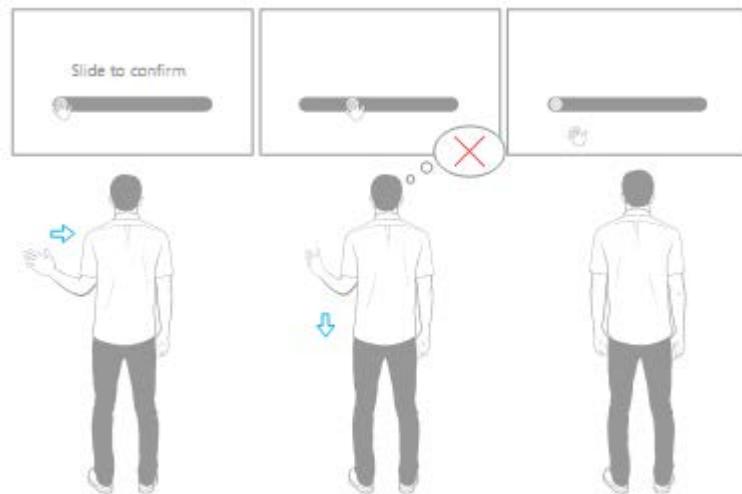
为了提供友好的用户体验并避免让用户产生挫败感，采用语音和手势作为输入方法的人机交互体验应具有如下特点：

降低操作难度：

准确	如果误启动某个操作，可以快捷退出，完全不必等到完成该操作方可退出
真实	仅显示适宜数量和类型的内容
快速	充分考虑社会影响因素。用户在自己的环境中使用这种输入方法时应感觉毫无拘束
省力	易于掌握，按需记忆
直观并拥有简单的心理映射	错误启动时可以轻松地退出，而不是必须完成操作才可取消
在各种距离范围内均有效。切换输入并不是最佳选择	



适用于各种距离。
切换输入方法并非最佳选择

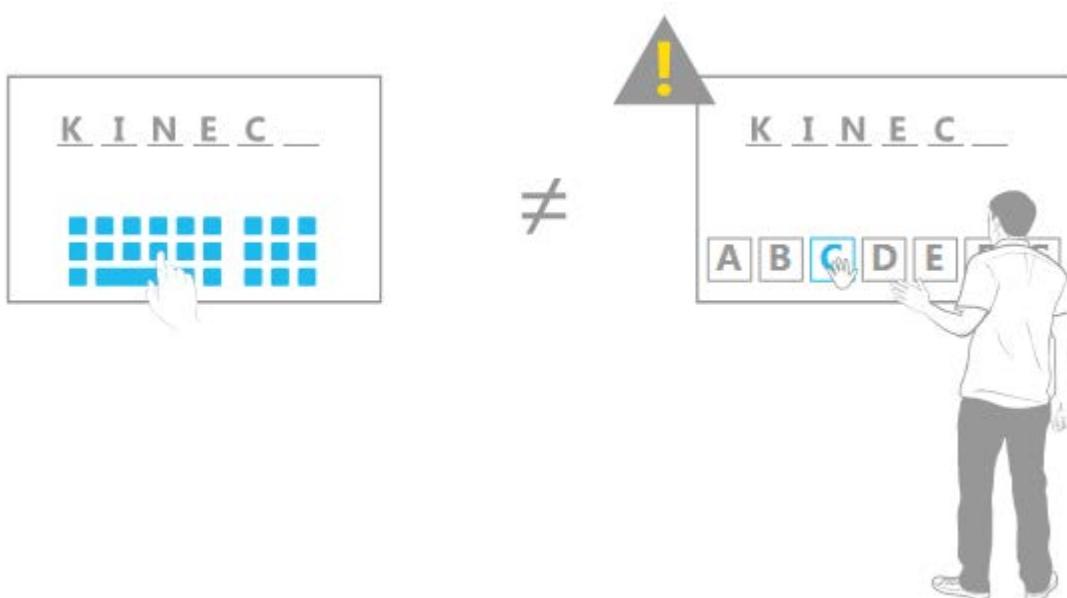


如果误启动某个操作，可以快捷退出，完全不必等到完成该操作方可退出

选择适宜目标任务的输入方法

您在设计人机交互体验时，请记住，您可以使用的全部输入方法均有各自的利弊。如果某中输入方法尤其适用于既定任务，应在此类任务中尽量使用这种输入方法，而不要将全部人机交互操作均限定于某一种输入方法。

例如：对于文本输入而言，如果您拥有键盘，就可以要求用户使用键盘输入文本，而不是使用手势输入文本。



正确

错误

✓ 用户可以快速、直观地完成任务

✗ 所选用的输入方法让人感觉到压迫感、不自然、笨拙或单调

✓ 根据目标任务，选用最为恰当的输入方法

✗ 用户在执行某些特定任务时，如果感觉到输入方法不适宜使用和被强迫感，只不过是因为他们已经习惯于在当前这种环境中执行其它任务而已

✓ 在切换输入法时，应充分考虑用户的环境和站位，仅在有助于提高便捷性和生产力的前提下进行切换

目标定位

在本节内容中，我们将重点介绍目标定位，即用户通过操控当前系统来跳转到他们所希望进入或选择的内容。对于绝大多数计算机用户而言，使用鼠标并非最人性化的目标定位手段，但是，在您设计 Kinect for Windows 应用程序时，您可能会希望为那些采用不同站位距离和拥有不同硬件配置的用户提供这一能力。如下信息将帮助您了解目标定位的基础知识、您必须根据具体设计方案所做出的一些折中以及让 Kinect 目标定位系统尽可能实现用户友好特性的一些选项。

目标定位的基础知识

要实现成功的目标定位，最好遵循如下一些原则：

让可操作项一目了然

让操控方式轻松自如

确保视觉反馈符合用户意图

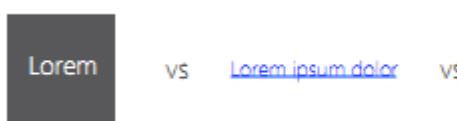


让操控方式轻松自如



确保视觉反馈符合用户意图

正确	错误
✓ 让按钮的外观有别于超链接和静态的正文内容	✗ 用户必须在一个安静的环境中大声喊出命令
✓ 定位操作应让用户自己的身体和周围其他人的视听都感觉到轻松惬意	✗ 用户必须使用别扭的姿势来定位用户界面中的某些部件
✓ 让用户的动作真实地映射到操作屏幕上	✗ 游标在移动过程中频繁抖动而且偏离方向
✓ 只要控件的间距、位置和大小均恰当合理，就很容易对其进行定位	✗ 用户的手部移动与屏幕上的游标移动之间存在延迟，从而导致用户必须拉长和缩短自己的动作，方可定位目标



Lorem
ipsum dolor
sit amet,
consectetur
adipiscing
elit. Duis

Lorem
ipsum dolor
sit amet,
consectetur
adipiscing
elit. Duis

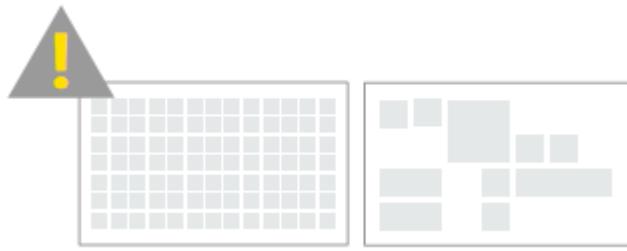
vs

Lorem

Lorem ipsum dolor vs [Lorem ipsum dolor](#)

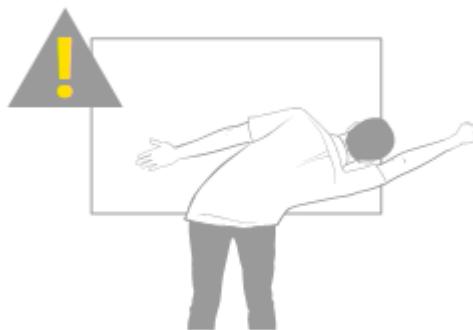
正确：让按钮的外观有别于超链接和静态的正文内容

正确：让可操作项一目了然



正确：只要控件的间距、位置和大小均恰当合理，就很容易对其进行定位

错误：用户必须在一个安静的环境中大声喊出命令



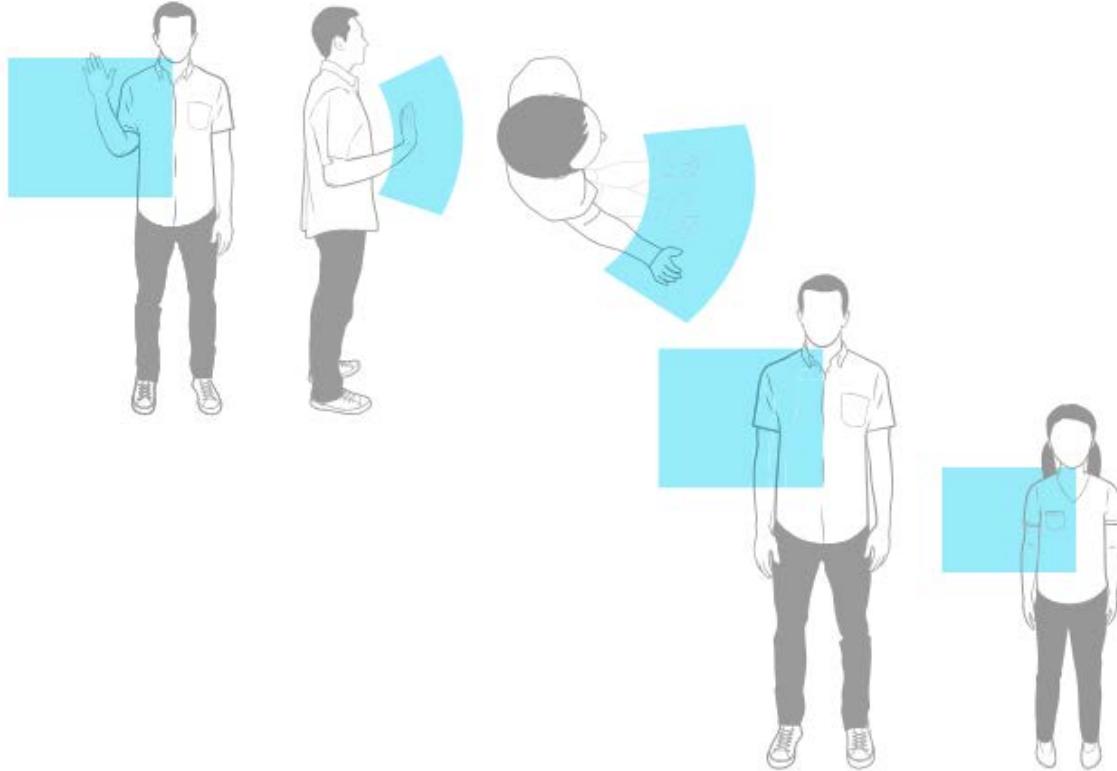
错误：用户必须使用别扭的姿势来定位用户界面中的某些部件



错误：用户的的手部移动与屏幕上的游标移动之间存在延迟

物理交互区域

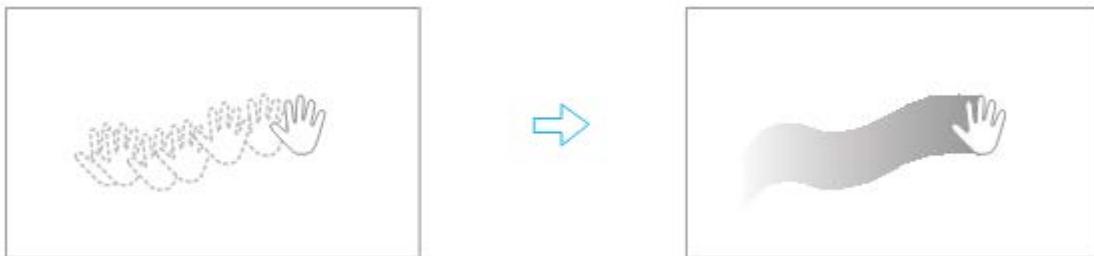
物理交互区域是用户面前的一个区域，用户手部在物理空间中的移动与游标在屏幕空间上的移动之间的空间映射关系将在这一区域中建立。当用户在物理交互区域的边界以内移动自己的手部时，游标也会在屏幕边界以内相应移动。物理交互空间的范围包括从头部上方到肚脐之间的区域，主要集中在身体左侧和右侧的手部移动空间范围内。物理交互区域的深度呈弧形延展，这是因为手部和手臂的移动范围呈弧形覆盖。物理交互区域的大小和方向是相对于当前用户而言的。



游标的实现方法

最常见的目标定位方式，即 Xbox 360 上的 Kinect 所使用的目标定位方式，就是通过手部的移动来控制游标的显示。游标的最简单实现途径就是定义一个物理交互区域，并将当前该区域内水平方向和垂直方向的手部移动直接映射到屏幕上。这就是 Xbox360 游标的实现原理。

有些人可能会通过添加过滤器的方法来减少可能由骨架数据所返回的抖动和跳动现象，进而实现更完美的游标。这种过滤器的一个典型示例就是采用一系列帧的平均或中间位置作为取值，以此来剔除多余的或不自然的大幅度移动。

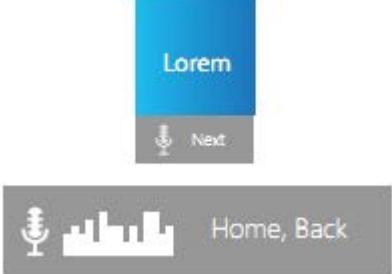


使用筛选器的副作用就是经常会增加延迟。延迟会让用户体验大打折扣，甚至会让用户感觉到缓慢至极，以至于无法响应。要在流畅和延迟之间寻找一个微妙平衡点是极其困难的，而且可能还需要额外的时间来执行迭代算法。



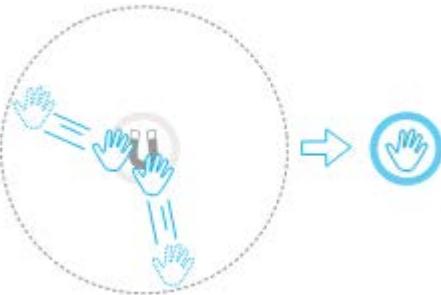
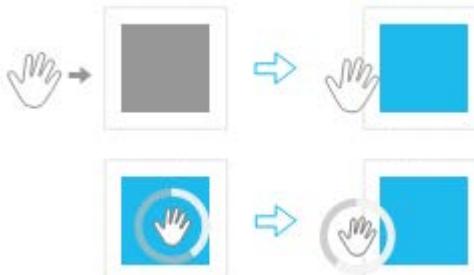
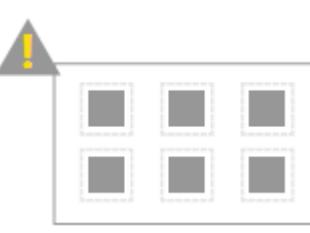
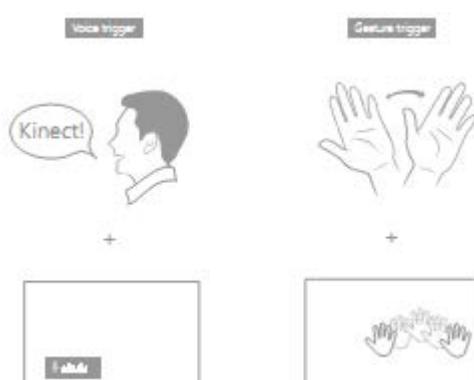
游标设计

虽然游标的设计方法多种多样，但是，此处所给出的一些游标已经过我们的筛选和测试，并且是我们所知道的较易于用户理解的游标。

	说明	示例图片
游标 符号	静态的符号表示，仅适用于将当前用户的动作意图映射到 2D 空间。即简单地将用户的意图移动映射到 2D 空间。如果使用手形符号，应考虑根据手部状态或用户所使用是的左手或右手来进行切换。	
景深 3D 表示	当前用户手部的景深表示，用于显示当前用户的手指、手掌以及（某些情况下）下臂位置正被如何映射到用户界面上的相应位置。不仅可以向当前用户给出方向反馈，还可以给出 Z 轴向上的深度反馈。	
VUI 语音用户界面	用户界面中的文字内容，用于表示用户可以使用的口语命令。VUI 应具有明显的差异化特征，以帮助用户理解。	

其它定位交互选项

要定位远距离目标十分困难，这是因为准确度和直接控制力的程度都会随之降低。此处的方法可以帮助您增强用户界面的容错能力和清晰度。

	说明	备注
磁力吸引	<p>如果游标处于某个特定的磁场控制半径内，游标就会被磁场的控制力所吸附。确保某个可操作区大于该区域的图形表示，就可以让用户更省力地实现高精确度。</p> 	<p>* 该方法仅适用于控件数量相对较少且相互间距相对较大的情况。如果屏幕上存在过多磁力点，游标就会在这些磁力点的作用力下无法遵循用户的移动轨迹，从而造成混乱，而且用户也无法精准操控游标。</p> 
扩大接触面积	<p>将控件的可响应范围扩大到其可视范围以外。例如，如果游标接近当前按钮但又尚未触及它，游标就可以显示为悬停状态。还可以在某个定时器启动时开启游标的悬停状态，以便当前用户的手部可以在不关闭该定时器的前提下，略微游移到该按钮以外。</p> 	<p>* 过于频繁使用该方法或控件间距过于紧凑，就会引起混淆。当用户在用户界面上移动游标时，不应让他们时常感觉到自己在触发非目标控件。</p> 
触发器 + 模式	<p>使用语音或单个手势来进入“目标定位模式”</p> 	<p>* 在目标定位模式中仅显示游标。</p> <p>* 明确指明如何进入该模式。</p> <p>* 如果您目前未处于目标定位模式，屏幕上的控件不应显示为可定位。</p>

选择

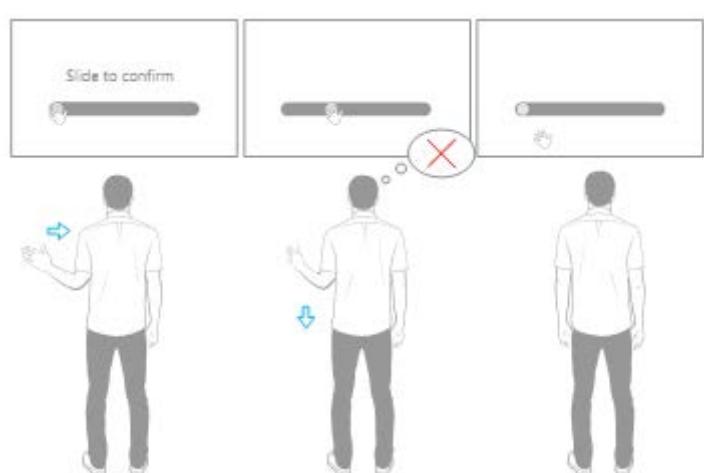
选择是指用户经过深思熟虑之后决定要执行某一操作，例如单击鼠标左键。由于选择的结果往往会影响重大更改，因此，必须首先弄清楚用户的真正意图。例如，如果屏幕上有一个“立即购买”按钮，您肯定不会希望用户很容易就误选这个按钮。手势为选择带来许多不确定性，而这些都是人际交互设计所必须要应对和解决的问题。

选择的基础知识

选择应是一种在个人自控力的作用下经过思考后的确认行为。



用户应明确知道哪个项目即将被选择



在最终敲定选择结果之前，用户可以取消误选结果

正确

在触发选择之前，需要正确理解用户意图

用户应明确知道哪个项目即将被选择

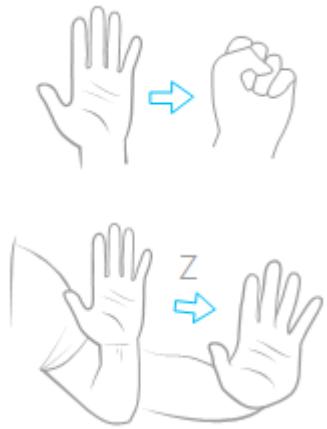
在最终敲定选择结果之前，用户可以取消误选结果

错误

在用户与系统之间的交互过程中，控件经常会被误选

选择用户界面设计

您选择使用的选择用户界面多数都是根据您使用的目 标定位用户界面而定，但是，我们仍为您提供了一些建议使用选项。

	说明	示例图片
进度游标	进度用户界面用于游标悬停在某个目标上一段既定时间之内填满进度。一方面，要通过延长超时时间来减少无意识选择，另一方面，要通过减缓人际交互速度来避免用户产生疲劳感和挫败感，重点是要在这两者之间找到一个微妙的平衡点。通过更改进度动画的播放速度或开始时间，就可以从根本上影响可感知的性能，同时，又不会影响实际的选择时间。	
改变手型的方向或 Z 轴向位置	握拳、改变手掌的方向或向前推手均表示选择。请注意，这些人机交互方法很难做到逼真，这是因为，当用户沿 Z 轴方向做按压动作时，难免产生 X 轴向和 Y 轴向上的移动。应尽量提供反馈，特别是有关 Z 轴向上选择操作的反馈，这一点极其重要。	
VUI	明确区分可作为命令读出的文本。	

更多选择交互选项

使用手势来选择可能会十分困难，主要是因为这种方法要求用户在一段既定时间内始终保持手部姿态不变。如下选项有助于让选择操作更倾向于一种具有明确意图的行为。

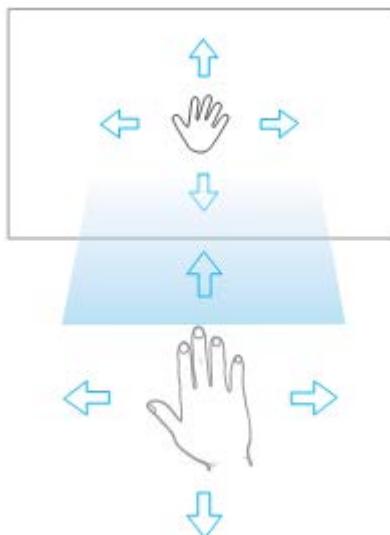
	说明	备注	
动态手势	随着用户做出选择动作的同时，给出直接映射进度。	<ul style="list-style-type: none"> ✿ 此类选择所要求的时间长度和动作允许用户从已错误启动的选择过程中跳出，并对自己所要做出的选择动作定义更明确的意图。确保给出清晰明确的反馈，以便用户了解如何在必要时跳出选择过程。 ✿ 如果当前的选择手势所在轴向与当前的定位轴向不同，就可采用这一选项。例如，如果某位用户正在一个垂直控件列表中进行导航，则选择手势应为水平方向。 	 
扩大接触面积	在使用手势定位目标之后，使用语音命令触发选择。 例如： 当游标悬停在某个按钮上之后，说“Select”。		<ul style="list-style-type: none"> ✿ 这种方法适合更快地做出选择，其原因是不存在定时器，而且您可以毫无顾虑地确定用户意图。这一方法并不适用于较为嘈杂的环境或者不可大声喧哗的公共场合。 

平移

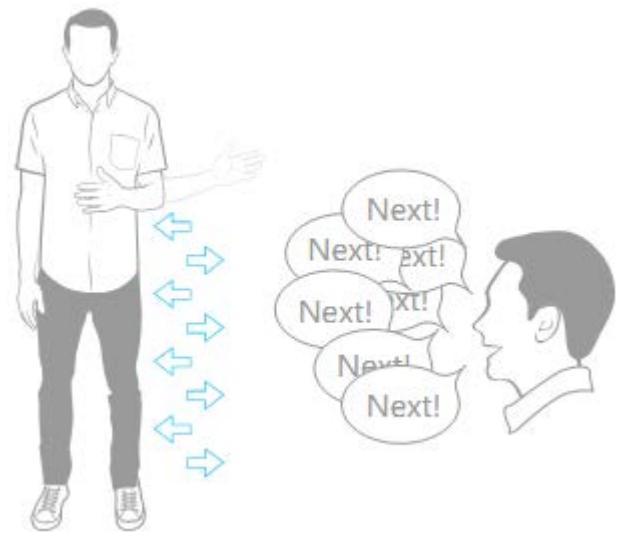
平移允许用户在画布内向上、向下、向左、向右导航，有时甚至可以在整个画布内自由导航，例如在触摸屏上拖动您的手指。平移可被用于在任何界面或页面上执行导航，或者，也可被用于在多个连续的屏幕或视图之间执行导航。语音显然不适用于提供功能强大的平移解决方案，但是，手势却可以针对一定距离之内的平移操作提供一些有趣的选项。如下示例将演示如何在您的应用程序中启用平移功能。

平移的基础知识

平移可以改变当前用户在用户界面中的所处位置。应用程序的画布或结构越大，用户就越需要使用平移功能。另外，由于平移操作通常要求直接操控，因此，响应效率和准确度都非常重要。请在设计过程中谨记这些提示，这样，您的用户就可以快速、准确、省力地跳转到目标位置。



平移手势动作从开始到结束均保持相同的语义和方向



即使要平移较长距离，也不要让用户必须执行许多重复手势

正确	错误
<ul style="list-style-type: none"> ✓ 用户可以在大幅画布上平移大段距离，同时，不会产生过多重复操作或疲累感。 	<ul style="list-style-type: none"> ✗ 要平移较长距离，用户必须做出大量重复的手势或说出大量重复的语音命令。
<ul style="list-style-type: none"> ✓ 用户可以完全控制他们正在平移的区域，而且可以轻易定位指定区域。 	<ul style="list-style-type: none"> ✗ 用户界面不停来回跳转，会让用户失去方向感，从而不易于找到特定内容。
<ul style="list-style-type: none"> ✓ 平移手势动作从开始到结束均保持相同的语义和方向。 	

平移用户界面设计

符合常规控制或手势识别要求的预期动作将导致当前用户界面的视图发生改变，而平移反馈就来自于这种视图切换。如下是用户所熟悉的一些简单的用户界面选项。如上所述，语音虽是一种选项，但它可能会带来一种单调乏味的用户体验。

	说明	示例图片
目标区域及其滑动	定位屏幕上的某个区域，然后，滑动屏幕。这就是 Xbox360 中的操作原理。	
使用类似滚动条的用户界面 定位并选择，或者，定位并保持	可被定位和选择的方向箭头和/或滚动条。	
VUI	允许用户平移的语音命令选项，例如“向前翻页”，“向后翻页”，“向右移动”，“向左移动”。	

更多平移交互选项

此处将给出一些可供您用来帮助用户更快速、更准确地跳转到目标位置的方法。

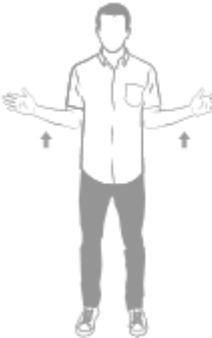
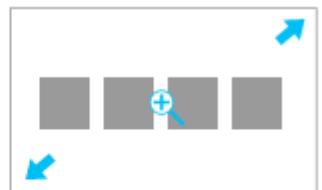
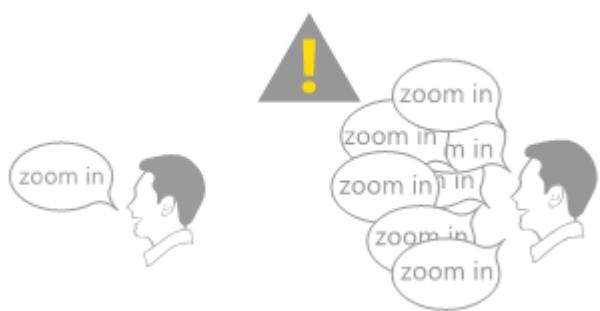
	说明	备注
加速平移	当用户将手部停留在某个区域上或正在滚动的用户界面上时，滚动速度就会提高。或者，根据用户手部所经过的距离长短来控制移动的速度。	<ul style="list-style-type: none"> * 提高加速度非常适于长距离滚动，但是，却不易于非常精确地定位滚动目标或平移到某个特定位置 * 根据手部所经过的距离来控制移动的速度，这样，就允许用户在不停止滚动的前提下减慢移动速度。
画布 定位、选择、定位	使用某一种手势或其它某一个命令专门用于表示画布在移动过程中始终位于用户手部之下。	<ul style="list-style-type: none"> * 在这种方法下，必须避免错误识别。如果画布被误选，就会严重干扰定向。 * 应尽量保持动作连贯，以便在选择动作被释放之前让画布保持连续移动，这样，用户才能感觉到反馈更加连贯、顺畅。
设置增量	设计一个简单的单向手势来触发某一方向上的平移，同时，为每种手势均设置相同的平移/滚动距离增量。	<ul style="list-style-type: none"> * 长距离滚动单调乏味，而且会让用户产生挫败感 * 这种方法非常适用于应用程序内的导航操作，这是因为增量大小始终如一（例如，下一页和前一页）

缩放 (沿 Z 轴向平移)

缩放与平移非常类似，但二者之间的区别是，缩放不涉及垂直或水平移动，而是对当前画布的远近进行操作。许多用户已习惯于使用鼠标和键盘的缩放操控按键或者触摸屏上的抓起动作来实现缩放。要在 Kinect 中实现缩放，将会遇到巨大挑战，这是因为用户已习惯于以手触的方式来控制缩放幅度，如果您无法直接触摸操作界面，就更难于精确定位起点、终点以及它们之间的距离。如下是一些简单的缩放解决方案，绝大多数用户已对此相当熟悉。

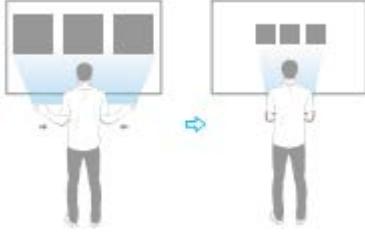
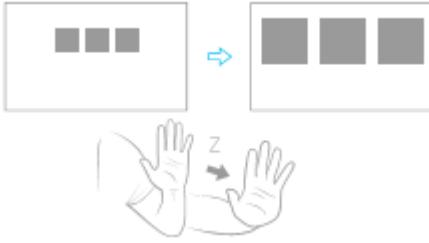
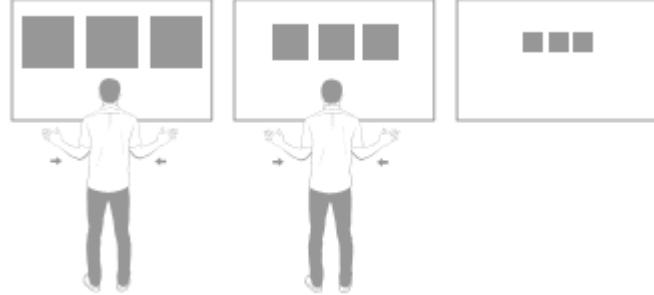
缩放用户界面设计

符合常规控制或手势识别要求的预期动作将导致当前用户界面的视图发生改变，而平移反馈就来自于这种视图切换。如下是用户所熟悉的一些简单的用户界面选项。如上所述，语音虽是一种选项，但它可能会带来一种单调乏味的用户体验。

	说明	示例图片
触发缩放模式	如果缩放模式已被触发（例如，通过用户保持双手向上的姿态来触发），用户界面就会指示缩放模式已被开启。作为操作提示，当前用户界面可以显示双手、更抽象的视觉表示或符号化的表示，例如箭头。	 
缩放控制界面 (定位并选择，然后，移动或定位并保持)	缩放用户界面非常类似于取值范围从 0% 至 100% 的滚动条。用户很容易就可以抓取并拖动滑块，也可以按下和按住滚动条的任意一端。	
VUI	语音命令允许用户按照多种不同的缩放增量来切换缩放比例，例如“zoom 100 percent”、“zoom in”或“zoom out”。请勿强制用户说出过多的重复语音命令。	

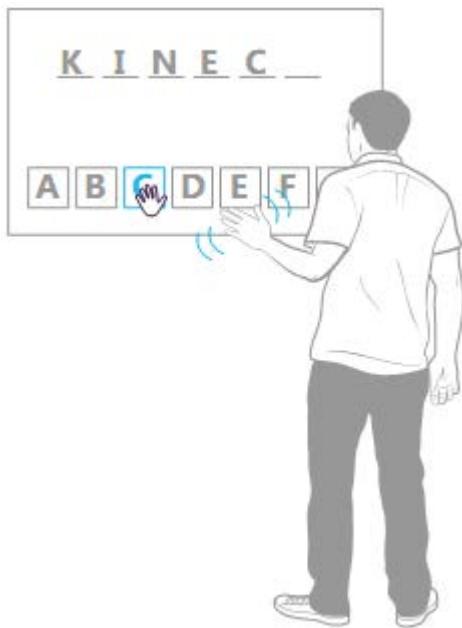
更多缩放交互选项

缩放的目标是操控屏幕上的某个对象并查看反馈结果。如下几种方法均可将用户动作映射到缩放操作。按照直接明确的操控原则，应尽量避免延迟，并让用户界面尽量保持高效响应能力。

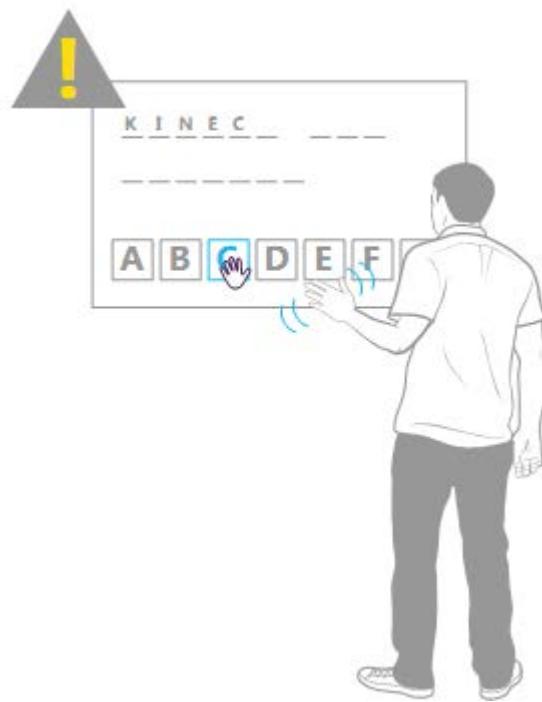
	说明	注解
双手缩放	<p>检测用户举起双手的动作，然后，按照双手之间距离扩大或缩小的百分比来决定缩放幅度的百分比。</p> 	<ul style="list-style-type: none"> 由于这种方法类似于用户所熟悉的触控交互方式，因此，使用起来更加直观、易于上手。但是，要实现这种操控方法，却具有很大挑战。 您需要准确地定义手势的起点和终点，以便让缩放幅度对应于预期站位。
Z 轴向空间缩放	<p>通过将手部沿 Z 轴向移动，缩放幅度直接取值于用户手部与目标对象之间的距离。随着用户手部拉回身体位置，增加缩放比例；随着用户手部远离身体位置，减少缩放比例。</p> 	<ul style="list-style-type: none"> Z 轴向的空间感应更不易于做到精确，因此，这可能更具挑战，但挑战的另一面就是极大的乐趣。
触发器 + 缩放手势	<p>允许用户使用语音或某种特殊手势来触发缩放，然后，再使用另外一种手势执行缩放。</p> 	<ul style="list-style-type: none"> 在使用这种方法时，应尽量避免错误识别，一旦误触发缩放，就会很难控制方向。
设置增量	<p>允许用户使用语音或某种特殊手势来触发缩放，然后，再使用另外一种手势进行缩放。</p> 	<ul style="list-style-type: none"> 这种方法并不适用于大幅度缩放操作，但是，可能比较适合采用语义来控制缩放。

文字输入的基础知识

使用手势或语音书写内容非常困难、缓慢、容易失真，因此，应尽量避免。如果用户必须选择一些字母，可以使用搜索或筛选来顺利实现这一目的。请注意，采用手势来实现的文字输入功能通常会涉及到一系列定位和选择操作，如果必须连续多次执行文字输入，就会让用户倍感厌烦和疲惫。



正确：如果仅需要输入很少几个字母，就可以采用在一小组字母中搜索或筛选的方式来实现文字输入



错误：口述较长的语句或频繁采用语音输入文字

正确	错误
<ul style="list-style-type: none"> ✓ 仅在必须输入文字且字母数量很少的前提下，可以采用在一小组字母中搜索或筛选的方式来实现文字输入 	<ul style="list-style-type: none"> ✗ 要输入较长文本内容，可模拟键盘输入
<ul style="list-style-type: none"> ✓ 仅在只有少量几个单词需要被识别的前提下，采用语音文字输入 	<ul style="list-style-type: none"> ✗ 采用语音文字输入功能来输入字母（发音太过相似：“E”、“P”、“G”）
	<ul style="list-style-type: none"> ✗ 口述较长的语句或频繁采用语音输入文字

文字输入用户界面设计

	说明	示例图片
虚拟键盘	这一解决方案允许用户从字母列表中定位并选择字母的方式来实现文字输入。	

依距离而变的人机交互操作

Kinect 将带来一种全新的距离互动体验，这将为应用程序开发人员和用户开辟一个充满新奇的探索旅程，用户再也不必守在自己的电脑跟前来对其进行操控。当然，这也将为 人机交互操作 及其界面的设计带来一些挑战和思考。本节内容将帮助您具体了解如何让您所设计的用户界面在任何距离下均具有更完美的可操控性。

人机交互的距离范围

触控距离 (0.0-0.4 米) : 可触距离是指用户可以使用鼠标、键盘或触摸屏的距离，按照这样的定义，相应的距离范围应该不超过单臂的长度。由于当前用户接近于屏幕，因此，这种距离范围可以最精细地操控画面及其上面的控件。

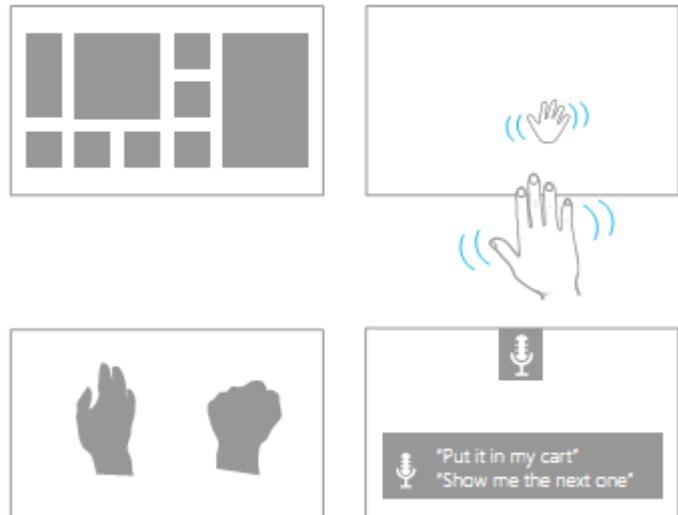
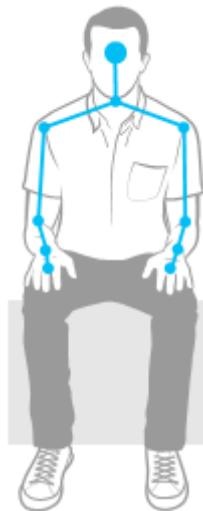
如果您的用户处于这种距离范围之内，鼠标、键盘和触摸屏可能更适合于作为输入方法，而语音和手势就不太适合。景深数据仅限于 40 厘米距离以外有效，也就是默认为 80 厘米的近距范围，因此，在触控距离范围内，景深数据或骨架数据无效。



近距范围 (0.8-2.0 米):

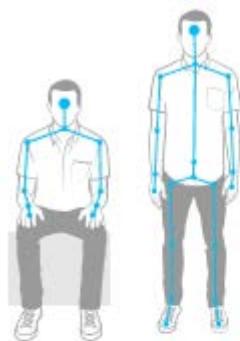
在近距范围内 ,整个骨架中的某些部分可能会被遮挡或屏蔽 ,但是 ,对于坐姿 ,这是比较恰当的距离。

由于用户始终保持在近距范围内 ,显示画面始终可以保持较高的细节程度 ,因此 ,在演讲过程中 ,可以使用更长的语句 ,更加细腻的手势和景深数据也可被使用。这种距离范围也适用于您进行物体识别或符号识别。

**远距范围 (2.0-4.0 米):**

在远距范围内 ,整个骨架均为可见。坐姿也同样适用。

在这种距离范围内 ,可以完美识别更为粗略的手势和更为简短的命令。必须采用足够大的用户界面 ,以便远距离观看。

**超出范围 (>4 米):**

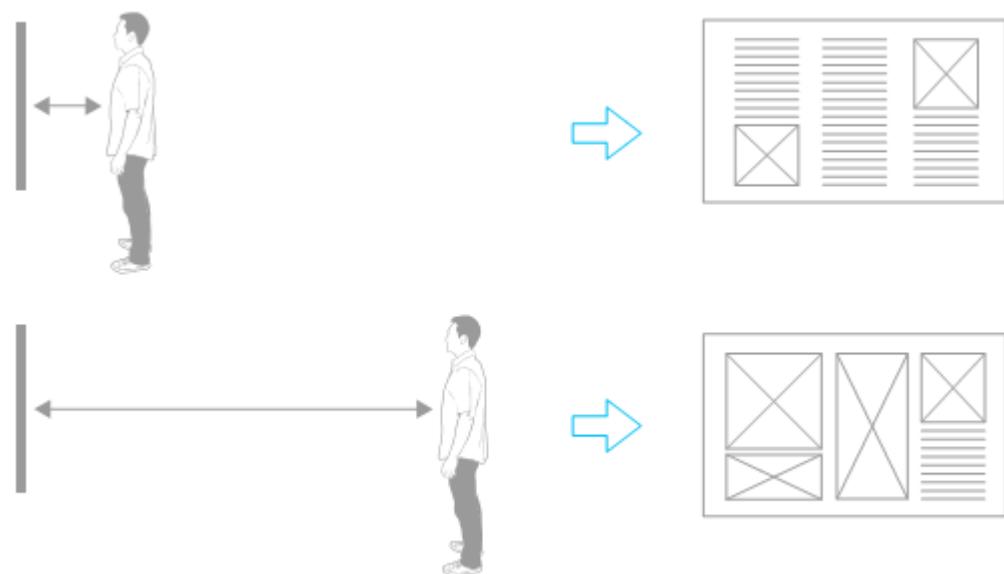
在这种距离下 ,大多数 Kinect 人机交互操作均无法正常执行。您的用户界面应该随时告知用户接下来的操作非常有趣 ,并诱使他们靠近用户界面。可视内容必须始终保持大尺寸、易识别的特点 ,另外 ,在某些情况下 ,可采用音频手段来吸引用户的注意力。



内容、文字和目标大小

当用户远离当前屏幕时，文字和目标的大小应该做出相应调整，以符合用户的视觉要求，另外，在手部和手势跟踪上，应该适当降低精确度和细节度。

随着距离增加，可以考虑成比例增加图形内容与文字内容的大小，以保持用户的互动能力，并减少视觉疲劳感。随着用户界面按照用户的距离而调整，可以使用切换效果来帮助他们始终跟随情境的变化，并了解他们的运动如何对所看到的内容产生影响。

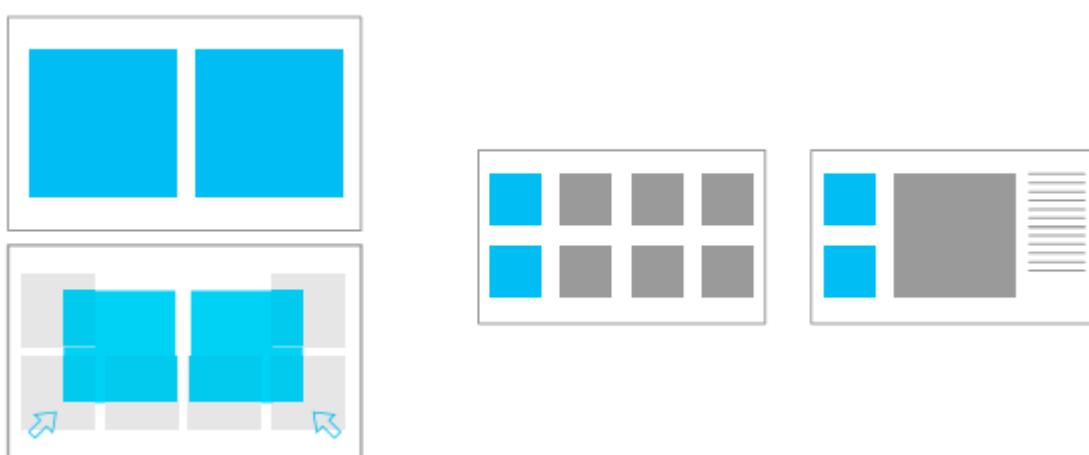


切换效果的设计原则：

平滑顺畅

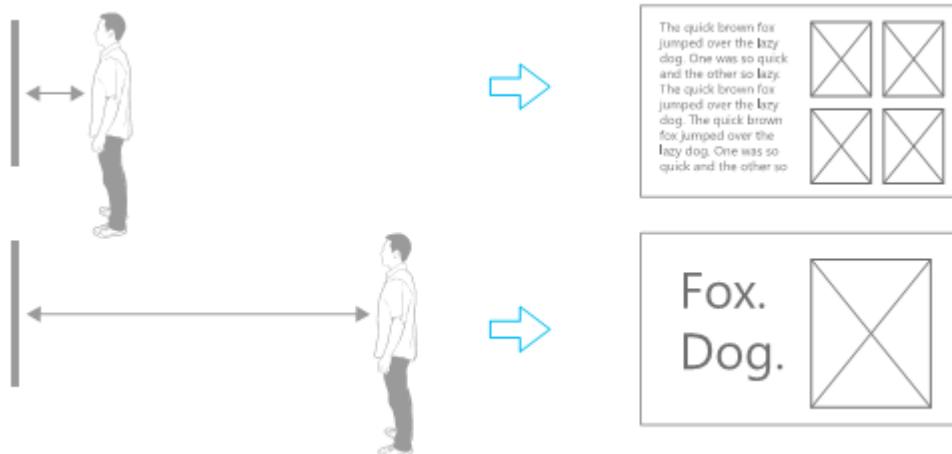
考虑使用锚点或固定节来帮助用户掌握方向和实现风格统一

让可操作项的摆放位置始终保持统一



可操作项的摆放位置和数量

在使用手势时，距离和屏幕大小应共同用于决定当前屏幕上显示多少个可操作项，以及这些可操作项的摆放位置。请注意如下一些原则：



距离越远，文字就要越大，手势就要越粗略，也就是说，在远距范围内，最好尽可能少地使用手势和语音命令。



通常情况下，成组的可操作项应显示在易于触及的区域中，以帮助用户提高人机交互操作的效率。

多模态

在人与人之间的沟通过程中，我们经常使用说话和手势来彼此交流，通常，如果要同某个应用程序之间进行互动，可能会用到多种输入方法，这同上述的人际互动极为相似。当然，如果您将许多选项及其组合投入使用，将会带来一种含混不清的用户体验。本节内容就将探讨各种异步和同步的人机交互操作，以及如何让这些操作变得更有条理和易于使用。

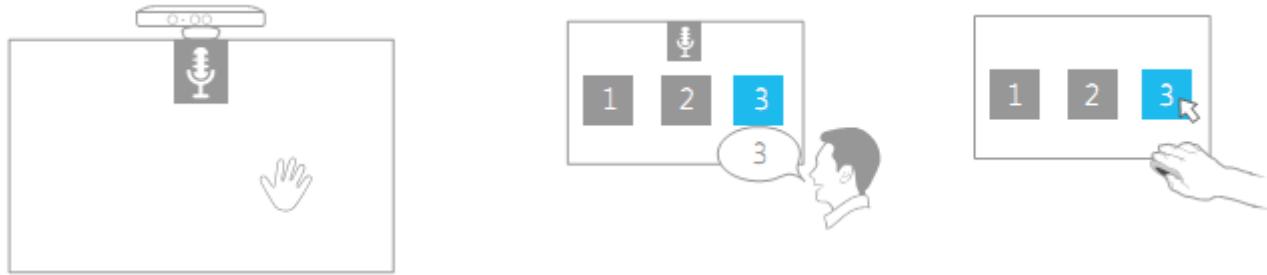
异步人机交互操作

异步人机交互操作是指在可供选用的多种输入方法中仅为每次操作选用其中一种输入方法。例如，您的应用程序可能允许用户使用语音来触发某些命令，并使用手势来执行导航操作，或者，它也可能使用键盘来以手触方式输入文字内容，并在一定距离范围内，以骨架跟踪和手势来执行操作。请注意如下一些原则：

向用户提供有关可选输入方法的视觉和/或音频提示

针对任意既定任务，尽量提供一种备选输入方法

例如：可采用鼠标或语音来选择某个按钮



同步人机交互操作

同步人机交互操作是指在可供选用的多种输入方法中为每次操作选用多输入方法并按顺序调用。

一些具体示例：

讲话 + 手势

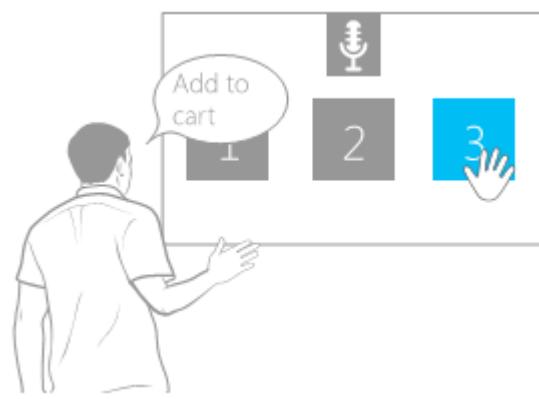
指向某个产品，并说“添加到购物车”

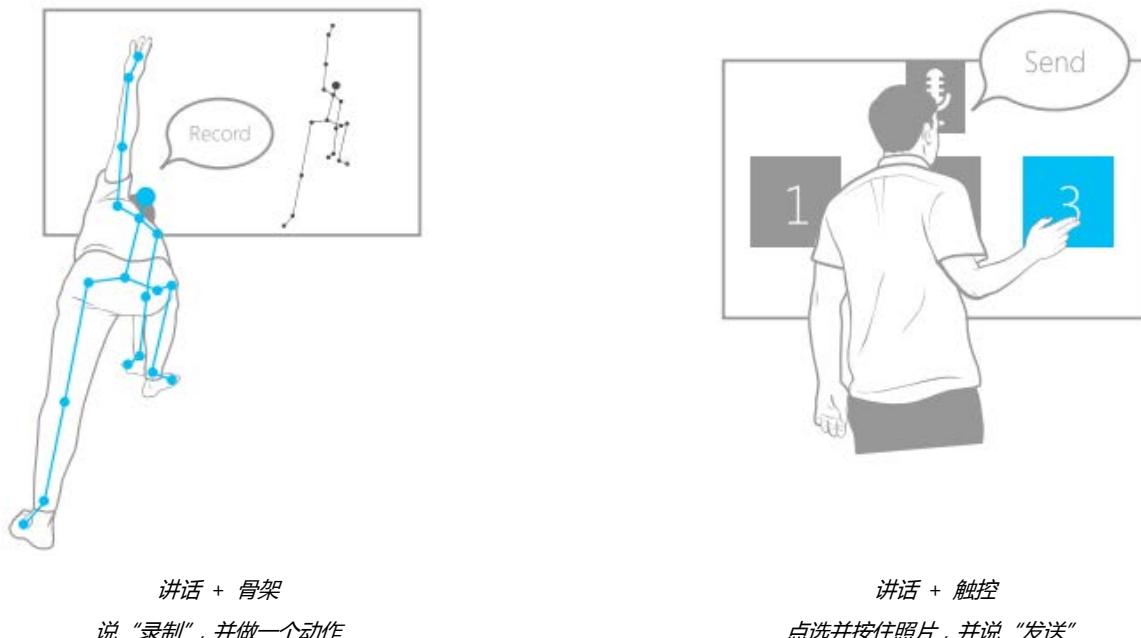
讲话 + 触控

点选并按住某张照片，并说“发送”

讲话 + 骨架

说“录制”，并做一个动作





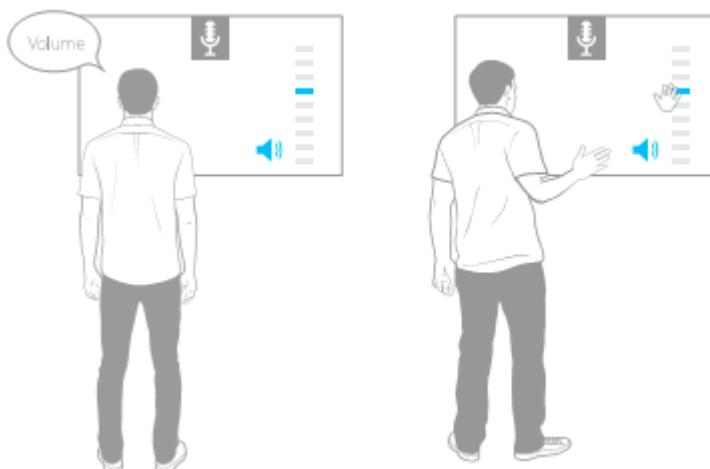
采用同步人机交互操作的一些理由

减少某个复杂操作中的步骤数量

例如：如果要使用同一种输入方法来发送一张照片，通常需要完成两个步骤：单击照片和发送照片。如果使用多模态技术，就可以将两种输入方法合并到一个操作之中。

无需移动您的身体，即可触发控件

例如：如果要使用同一种输入方法来发送一张照片，通常需要完成两个步骤：单击照片和发送照片。如果使用多模态技术，就可以将两种输入方法合并到一个操作之中。



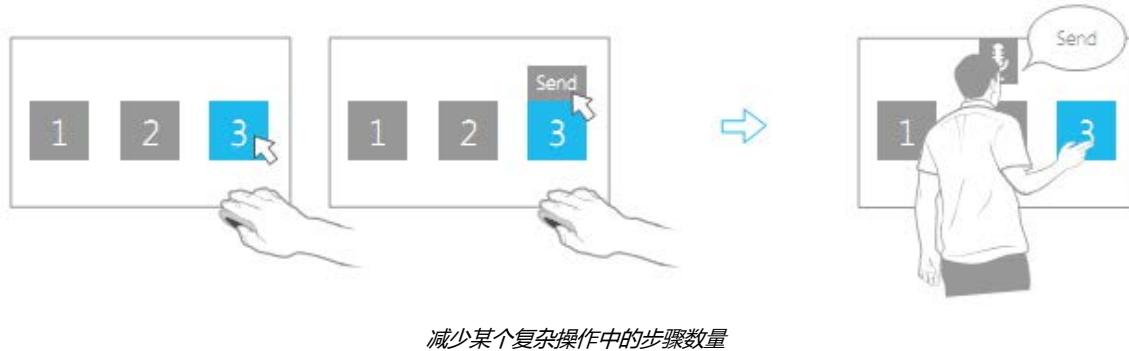
减少某个复杂操作中的步骤数量

增强操作信心

例如：如果某位用户为了执行一个命令而必须做两件不同的事情，这种命令执行方式就显得过于刻板和僵硬。

以最合理的方式利用每种输入方法

例如：使用语音来将系统切换到音量设置模式，然后，使用手势来调节音量大小。



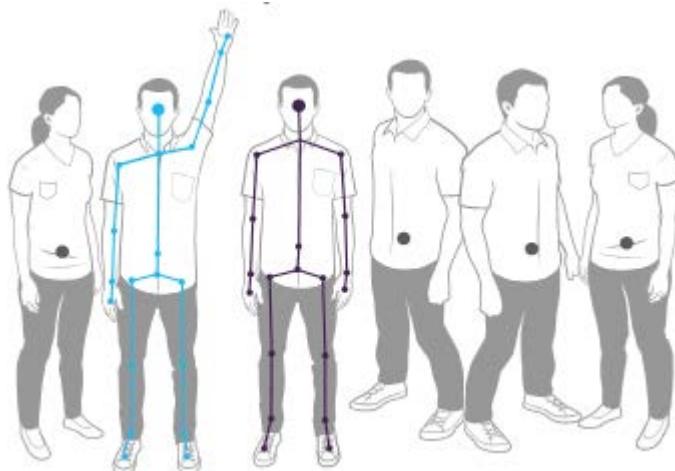
减少某个复杂操作中的步骤数量

多用户

Kinect 中的另一项神奇功能就是跟踪多位用户。该功能将让更加生动逼真的多人协作式人机交互操作成为可能，但是，这也会为操控和输入带来一些新的挑战。本节内容将介绍我们目前已知的多用户处理技术。

多用户跟踪能力

Kinect 最多可以识别 6 个人，并可以对其中的 2 个人进行细节跟踪。对于这 2 个被跟踪的人而言，Kinect 将返回完整的关节信息。对于其他骨架而言，Kinect 仍可以返回一些有用的信息：一个用于指定当前玩家身体外型的“玩家轮廓”，以及一个中心位置。当前应用程序可以选择应该跟踪这 6 位玩家中哪两位玩家的完整骨架。



多用户及其距离

在您设计用户体验和实现预期应用目标的同时，请随时留意距离因素对可识别人数的影响。

在每种距离范围内，如下给出的人数能够被您的应用程序识别，或者说，他们可以自如操控：

	触控	近距	远距	超出范围
人数	0	1-2	1-6	0

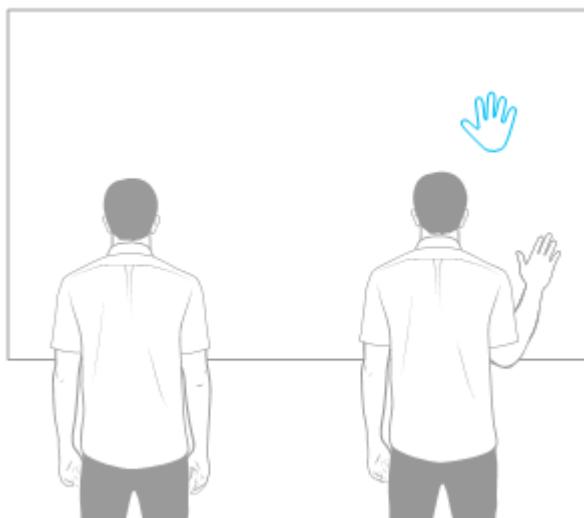
多用户、多模态以及协作

多模态和多用户的实现意味着，您的用户可以位于不同距离、采用不同模态。虽然您可以采用多种不同的手段来实现既定的协作模型，但是，在这一过程中，需要考虑到一些重要的注意事项。

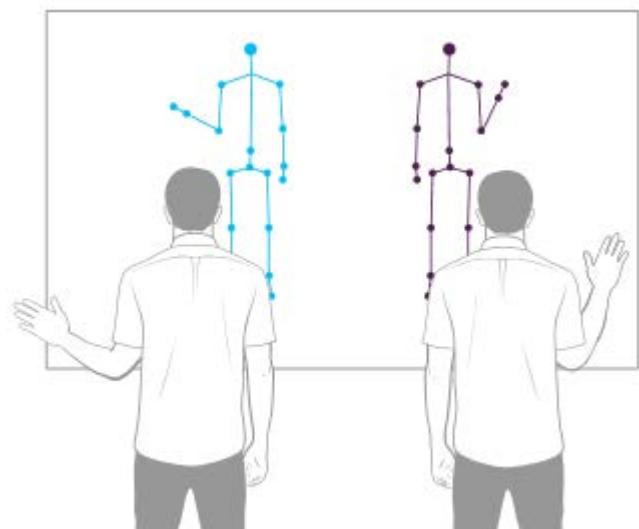
协作式人机交互操作

在协作式人机交互操作过程中，两位用户都在操控同一个屏幕空间和同一组输入设备。这就意味着，其中一位用户所执行的操作将影响到当前应用程序的整体状态，进而同时影响到这两位用户。在这种情况下，可以使用两种选项来平衡操控权：

单方控制模式：在该模式下，只有一位用户始终作为“控制者”，而且只有该用户所执行的操作会被跟踪。控制者角色的选择或切换可以采用多种方式来完成，例如，将首位进入监控范围的用户或者距离传感器最近的用户选为控制者。这是避免输入冲突的一种手段。该模式通常采用当前所跟踪用户的画面来指定，而且屏幕上始终只存在一个游标。



单方控制模式



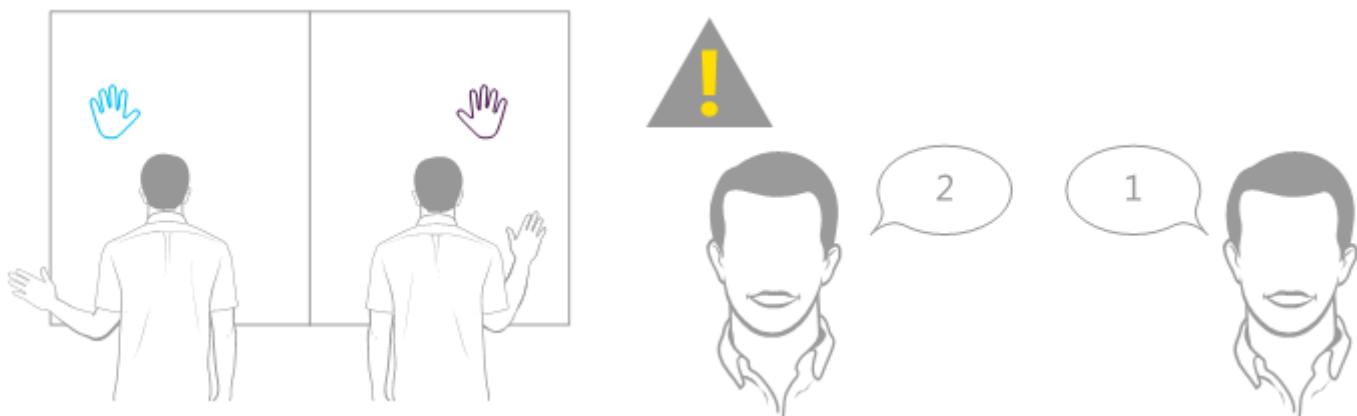
平等参与模式

非协作式人机交互操作

非协作式人机交互操作让同一个屏幕中的各位用户都能拥有自己的操作体验。其实现方式是屏幕分区或屏幕分割。通常，这种用户体验的操控方式应该更类似于单位用户的操作体验；但是，现存的另一个挑战是如何将当前用户的语音和动作正确映射到相应的操作界面。语音命令可能更具挑战。手势和动作的视觉反馈应该以单位用户为单位进行分别显示。

平等参与模式

该模式可以从两位用户采集输入信息，通常按照他们所定义的顺序进行。对于导航或基本的人机交互操作而言，这是一种非常复杂的模式，但是，极其适用于每位用户独立操控各自的用户体验，例如操控虚拟人物或其它一些游戏。



总结

采用 Kinect 所打造的人机交互及其操作界面可以实现出神入化的用户体验。当用户掌握了如何使用这套操控界面之后，就可以自然而然地对其灵活运用，Kinect 的神奇和魅力也就能够得以淋漓尽致地展现。

