摘要

當前的虛擬現實係統通常使用相機來捕捉控制器的輸入或徒手在空中的運動。

在本文中，我們認為這是VR中生產力場景的關鍵障礙，它需要長時間持續交互——這是控制器或空中徒手輸入無法滿足的要求。

為了應對這一挑戰，我們將表面上的觸控引入虛擬現實——用戶已經習慣於在手機和平板電腦上使用的輸入模式。

我們提出了 TapID，這是一種基於手腕的慣性傳感系統，它補足了VR眼鏡的手部姿勢以觸發 VR 中的輸入。TapID 將一對慣性傳感器嵌入柔性錶帶中，腕帶兩側各有一個；

通過註冊信號的組合，TapID 可靠地檢測表面觸摸事件，更重要的是，識別用於觸摸的手指。

我們在一系列關於事件檢測準確性（F1 = 0.997）和與手無關的手指識別準確性（用戶內部：F1 = 0.93；跨用戶：10 次細化輕敲後 F1 = 0.91 和 F1 = 0.87）的用戶研究中評估了 TapID沒有細化）在坐桌場景中。

我們以一系列應用程序作為結束，這些應用程序通過觸摸輸入來補充手部跟踪，並且由 TapID 獨特地啟用，包括 UI 控制、快速鍵盤輸入和鋼琴演奏，以及表面手勢。

Index Terms: Human-centered computing—Human com puter interaction (HCI)—Interaction paradigms—Virtual reality;

以人為中心的計算——人機交互（HCI）——交互技術——手勢輸入

INTRODUCTION

最新的混合現實係統結合了用於輸入檢測的手部姿勢識別，用於增強現實（例如 Hololens 2）和虛擬現實（例如 Quest 2）等。

雖然以前的設備世代在很大程度上依賴於手持控制器進行輸入，但在研究社區（例如，[47]）和商業領域，向無控制器、手勢和手勢操作輸入的轉變是顯而易見的。域（例如，[11,12,33]）。

在增強現實中，手部跟踪非常適合伴隨物理對象的工作，例如在維護和維修 [21] 或製造和組裝 [42] 中。

在物理對像上工作可以讓用戶抓住一些東西，從而防止使用過程中的疲勞 [22]。

相反，對於虛擬現實中與對象的交互，內容在被動觸覺系統之外是無形的（例如，[9,24,30]），一旦超過快速交互，徒手交互就會成為一種負擔[25]。

在本文中，我們建議將 VR 中的所有直接交互轉移到被動表面。

與空中交互相比，表面上的觸摸交互為用戶提供了在交互之間休息手臂的機會，同時在長時間交互期間提供物理支持。

同時，表面提供觸覺反饋以及本體感覺參考框架，提供快速和精確的交互。

VR系統中觸摸交互的核心挑戰是觸摸事件的精確檢測；

因為手是從頭戴式系統本身進行跟踪的，所以這個單一的有利位置會導致深度感應不准確，從而使觸摸/非觸摸辨別和接觸定位 [2] 具有挑戰性。

已經提出了多種方法來使用深度相機檢測表面上的觸摸，例如用於靜止 [46] 和耳機集成系統 [47]。

為了對抗噪聲深入測量，它們集成了過濾器，以實現可靠的觸摸檢測和抑制。

兩者，添加過濾器以及使用的相機的幀率都會限制系統能夠可靠檢測的交互速度。

我們通過用戶佩戴在手腕上的 TapID 手環來應對可靠且快速的觸摸檢測挑戰。

通過內置的慣性傳感器，TapID 不僅可以檢測被動表面上的觸摸事件，還可以識別用於觸摸的手指。

因此，TapID 補充了頭顯對手和手指的光學跟踪，我們將 TapID 檢測到的觸摸與它結合起來觸發 VR 中的輸入事件。因此，我們的方法允許平板電腦和手機已知的應用程序和交互模式無縫且可靠地轉移到 VR 場景。

* 1. VR 中被動表面上的快速觸摸交互

圖 1 顯示了我們方法的概述。

在這裡，用戶通過 VR 中的徒手手勢進行交互，而 TapID 為他們面前的桌子上的手指觸摸提供跟踪信號。

因此，用戶在任一手腕上佩戴一個 TapID 腕帶，將慣性事件流式傳輸到 PC 以進行輕敲檢測和手指識別。

同時，VR 頭戴設備使用內置的供應商算法來跟踪佩戴者的手部，這些算法會處理頭戴設備的攝像頭信息。

然後，我們的系統從頭顯中獲取雙手在視錐體中時的 3D 跟踪手部姿勢。

當 TapID 通過慣性傳感器檢測到點擊事件時，我們的系統會將輸入事件轉發給 VR 應用程序。

在驗證用戶的手是否靠近桌子等被動表面後，我們的 VR 應用會在相應手指的位置觸發輸入事件。

如果用戶的手沒有靠近表面，我們的系統會簡單地拒絕 TapID 檢測到的事件作為無意輸入並忽略它。

圖 1b 顯示了一個特別受益於我們的方法的應用程序：一個使用所有十個手指提供快速可靠的文本輸入的鍵盤。

在鍵盤上打字時，我們的系統會將所有鍵輸入轉發給操作系統以進行文本處理並檢索用戶可能選擇的單詞建議——類似於他們在普通觸摸板上的交互方式。

我們的系統集成了瀏覽器界面並將所有輸入轉發到渲染器，因此任何 Web 應用程序（例如搜索引擎、文字處理器或演示應用程序）都可以使用 TapID 在 VR 中通過觸摸變得可用。如圖 1c 所示，Microsoft PowerPoint 在 VR 中運行，並支持用戶期望從觸摸界面獲得的所有功能，例如點擊以放置光標、雙擊以選擇單詞、工具按鈕等。

最後，圖 1d 顯示了一個示例應用程序，該應用程序使用用戶自己的身體進行輸入提示，例如當用戶周圍的被動表面不存在時。

在這裡，菜單和其他交互元素在用戶的手臂上註冊，由耳機跟踪，並且在身體上的點擊將輸入傳遞給 UI 元素，因為它被 TapID 拾取和處理。圖 2 說明了我們的傳感原理。

彈性矽膠錶帶嵌入了我們的 SoC 平台，該平台連接到兩個加速度計，一個在手腕的兩側。

加速度計通過帶子機械地耦合到佩戴者的皮膚和腕骨，從而拾取身體耦合的振動信號。

在每個輕敲事件期間，慣性傳感器記錄的振動信號略有不同，這是由於通過用戶手的傳播路徑取決於它們來自的手指。

如圖 3 所示，這些差異是微妙而係統的，它們是我們信號的來源，TapID 使用機器學習管道對其進行分析，以檢測敲擊並識別敲擊手指。

在我們的用戶研究中，我們對 18 名參與者評估了 TapID 在點擊事件檢測和手指識別方面的性能。

TapID 可靠地檢測到敲擊事件 (F1,tap = .996) 並可靠地識別敲擊手指 (cross-session F1,finger = .93, cross-person with refinement F1,finger = .91, cross-person without refinement F1 , 手指 = .87)。

我們將我們的結果與我們在剛性和板載雙加速度計配置上運行的方法的性能進行了比較（例如，在 FITBIT 頻段中發現，例如 Charge [43]），在這種情況下，它在輕敲檢測 F1,tap 中實現了相當的精度= .996 甚至更差，但仍然有用的手指識別（跨會話 F1，手指 = .92，具有細化 F1 的跨人，手指 = .89，沒有細化 F1，手指 = .80）。

我們還在傳統的智能手錶配置中評估了我們的原型，只有一個安裝在板上的加速度計。

這種設置在敲擊檢測 (F1,tap = .996) 和手指識別 (cross session F1,finger = .91, cross-person with refinement F1,finger = .89, 沒有細化 F1,finger) 方面取得了與 FITBIT 相當的結果= .81)。

這些有希望的結果，特別是在現有可穿戴設備的配置中，驗證了我們的方法在 VR 場景中的實際應用。

1.2 貢獻

我們對工作做出了三個主要貢獻。

1) 我們引入了一種基於手腕的傳感原理來檢測表面觸摸，並通過集成到錶帶中的雙加速度計配置來識別導致觸摸的手指。

2) 我們展示了我們的原型實現 TapID，它使用機器學習管道實時處理和分類輸入事件。

然後將檢測到和分類的事件轉發到我們的 VR 系統，該系統將它們與耳機以 3D 方式跟踪的手部姿勢相結合，以觸發 VR 中的輸入。

3）在對 18 位參與者的技術評估中，我們評估了 TapID 在檢測輕敲事件和正確識別手指方面的表現。

因此，我們比較了三種配置中的 TapID：建議在腕帶上實現雙 IMU、剛性安裝在板上的雙 IMU 傳感器 (FITBIT) 和單個板載加速度計 (SMARTWATCH)，研究跨會話和跨人。

值得注意的是，我們證明了我們的方法既不是針對個人也不是針對特定手的。

總的來說，這些貢獻突出了 VR 中觸摸交互的前景，使用平板電腦和手機設備熟悉的輸入模式，但將其擴展並集成到 VR 中常見的空間用戶界面中。我們相信，我們的方法有可能在創造力和生產力場景的背景下為 VR 帶來長時間的交互，其中可靠、精確和不疲勞的輸入優先於已經證明的較短且通常全身參與的交互在遊戲和娛樂方面取得成功

2 相關工作

TapID 與輸入事件檢測相關，特別是使用可穿戴傳感器、手指識別和 VR 交互。

2.1 穿戴式裝置和身體結合之事件

幾個項目已經研究了身體耦合機械事件的交互作用。

方法主要在這些事件的起源上有所不同，在由手持工具、活動期間的重複運動、與被動啟示的接觸以及由敲擊自己的身體引起的機械事件引起的事件之間有所不同。

慣性測量單元 (IMU) 經常用於檢測人體運動和活動，因為它們集成到手機和可穿戴設備中 [44]。

例如，Viband 以高采樣率運行手錶中的 IMU，以檢測手持工具和手動活動 [32]。

隨著基於神經網絡的處理管道的出現，已經為這種識別提出了各種網絡架構 [27, 36, 37, 50]，它已經開始取代以前手工製作的特徵（例如，[13, 26] .

為了感知與被動示能的顯式交互，IMU 和麥克風已被用於檢測觸摸事件（例如，[6, 18]），它們可以使用多個傳感器（例如，[16, 38]）在空間上定位在表面上。

研究人員還檢查了身體位置是否適合安裝傳感器，例如手指（例如 [18,51]）、指關節（例如 [1, 35]）或手腕（例如 [6, 16]） .

其他項目則研究了對用戶自己身體的輕敲檢測。

此類項目通常涉及傳感器陣列以檢測生物聲波並安裝在前臂（例如，用於前臂和手部輕拍 [20]）或手腕（例如，用於輕拍手指關節和其他身體部位 [8、49]） .

除了輕敲事件之外，AudioTouch 還使用手背上的兩個壓電元件來解析觸摸力，並在觸摸過程中對 12 種手勢進行分類 [31]。

與 TapID 類似，Actitouch 補充了 VR 交互以檢測身體上的觸摸輸入 [52]，它使用具有皮膚接觸的可穿戴設備將射頻信號調製到身體上，接收器拾取以檢測身體觸摸，而耳機提供位置通過手部追踪。

2.2 觸摸時的手指識別

一些努力已經表明了在交互過程中手指識別的相關性。

在光學桌面上，這些努力使用基準標記 [34] 進行手指特定的 UI 控製或指紋掃描 [23] 以在交互過程中進行身份驗證。

Gupta 和 Balakrishnan 將光學傳感器安裝到指尖以區分兩個手指，使用新功能將觸摸屏鍵盤縮小一半，並為每個鍵重載兩個字母以進行手指特定輸入。

當安裝在靠近源（即單個手指）附近時，身體佩戴的 IMU 能夠識別手指。

哪個手指檢測到來自壓電換能器的敲擊手指，在一個實施例中放置在指尖上，在另一個實施例中安裝在指關節上[35]。

TapStrap 在設計上類似，是一種消費產品，具有 5 個嵌入 IMU [1] 的鬆散耦合環。

使用它們，TapStrap 通過被動表面上的敲擊檢測來實現單手和弦鍵盤。

Booth 和 Goldsmith 的手腕佩戴式 8 傳感器陣列沿手腕底部檢測到手勢輸入 [7]，儘管用戶的手指檢測精度低於 0.55。

許多場景受益於不受阻礙的手和手指，這帶來了替代的傳感方法。

例如，Becker 等人。使用肘部佩戴的 Myo 帶並從肌電圖信號中檢測到單個手指 [4]，達到了高達 0.75（用戶內）的精度，而用戶之間的精度則低於 0.50。

也使用 EMG，Benko 等人。以 0.91 的準確度對兩個手指進行分類，並在觸摸桌上展示了用例 [5]。

在研究了這些領域的相關工作後，我們得出結論，以前的方法顯示了手指特定交互的前景。

然而，它們通常在方便性（即，將傳感器定位在手指附近或要求用戶戴手套）與適合可靠地區分來自五個手指的輸入的準確度之間進行權衡。

我們認為 TapID 在這方面是一種令人鼓舞的方法，它通過用戶手腕將傳感器安裝在方便且社會可接受的外形尺寸中。

TapID 在經過最少的改進後可靠地達到了高跨用戶檢測精度，這充分建立了我們的交互式使用方法。

2.3 VR 中的觸摸交互

Surface 觸摸交互在混合現實社區中引起了人們的興趣。

出於在 VR 中輸入文本的目的，商用的高 fps Optitrack 系統被證明是一個有用的輸入平台，每隻手上都貼有 23 個動作捕捉標記[29]。

如果在 MR 場景中沒有固定的多攝像頭設置，人們很容易使用頭戴式耳機中內置的攝像頭來同時估計手部姿勢並檢測觸摸事件。

例如，MRTouch 結合了 Microsoft Hololens 中的深度和紅外攝像頭，用於實時表面觸摸檢測 [47]，並討論了其評估中的主要挑戰：高遺漏觸摸率 (3.5%) 和虛假額外觸摸 (19%) ）。

使用相機檢測觸摸一直是一個長期的挑戰。

在他的數字辦公桌上，Wellner 在 90 年代初期探索了這一雄心壯志 [45]，但得出的結論是可靠的接觸傳感需要表面儀器。

阿加瓦爾等人。檢測到觸摸位置和與頭頂立體相機的接觸時刻 [2]，討論立體深度的噪聲如何導致錯誤的接近度讀數。

Wilson 後來介紹了一種建立在單個深度相機上的方法，該方法提供了更可靠的檢測，並且還可以擴展到移動場景和身體輸入（例如，Imaginary Phone [19]）。

這些基於攝像頭的方法共享識別準確觸摸位置的好處，但所有與噪聲深度估計的鬥爭都是為了可靠的事件檢測以及它們對輸入速度的限制，這些限制受攝像頭幀速率和應用的過濾器的限制。

我們試圖通過 TapID 來補充這一挑戰，使用腕戴式慣性傳感器提供有關快速觸摸交互的缺失信息：精確事件和手指識別。

TapID 還從替代現實中獲得靈感，即在 VR 應用程序的環境中採用被動可供性 [40]。

此類可供性已用於提供其他缺失的觸覺反饋以響應輸入事件（例如，稀疏觸覺代理 [9]）。

我們設想使用 TapID 的設置是在我們之前將虛擬現實交互置於不同物理現實中的工作之後使用的（例如，VRoamer [10] 和 Dreamwalker [48]）

3 TAPID 可穿戴設備和電子設備

TapID 可精確檢測表面上的敲擊並識別敲擊手指，這得益於我們定制的腕帶設備的集成。

用戶佩戴兩個這樣的設備（圖 1a），然後每隻手獨立操作。

圖 4 顯示了我們的電子平台，它以片上系統（DA14695，Dialog Semi）為中心，以 1344 Hz 的頻率對所有慣性傳感器進行採樣，並通過串行連接將 IMU 數據流式傳輸到 PC。

TapID 具有兩個通過柔性 PCB 嵌入腕帶的低功耗加速度計（LIS2DH、意法半導體）以及兩個用於測試和比較不同傳感器配置的板載加速度計。

我們使用 Shore-32 矽膠澆鑄 TapID 的錶帶，並在固化過程中嵌入所有電子設備。

4 點擊事件處理管道。

我們的系統通過時間戳、導致點擊的手指的身份以及世界坐標中表面上的 2D 位置來檢測並表示所有點擊事件。

每當觸摸事件發生時，系統都會將此記錄傳遞給 VR 應用程序。

在本節中，我們描述了確定每個條目的三個相互依賴的步驟，如圖5所示

這包括我們用於輕敲檢測的信號處理、我們基於機器學習的識別管道，以及我們與 VR 耳機的手部跟踪的結合。

4.1 點擊事件檢測和時間戳計算

為了將點擊事件與其他手臂和手部運動區分開來，我們通過它們在加速度計信號中產生的尖銳尖峰來選擇事件。

TapID 不是執行簡單的幅度閾值處理，而是累積所有加速度計軸的原始信號 (RS) 的變化率來計算變化率分數 (RCS)。

為此，我們添加了指數下降，以僅關注信號的快速變化。

TapID 使用依賴於最後一個局部最小值的 RCS 峰值突出閾值來檢測敲擊事件。

我們將敲擊事件的確切時間戳作為超過閾值後 20 個樣本（15 毫秒）的窗口中的最高 RCS 尖峰推導出來。

在每個事件之後，我們的處理會在檢測到新事件之前產生 200 毫秒的回退期。

這可以防止單擊觸發多個事件，同時在使用雙手時仍允許每分鐘最多 600 個輸入事件。

4.2 點擊事件手指識別

在檢測到點擊事件的發生（包括其精確時間戳）後，TapID 將圍繞事件時間戳的原始加速度計樣本的對稱窗口轉發到我們的神經網絡分類器以識別手指。

我們的分類器估計每個手指引起記錄信號的可能性。

使用監督學習，我們通過反向傳播調整其權重來訓練網絡，以最小化訓練數據集上的交叉熵損失，該訓練數據集具有分配給每個記錄的點擊事件的單個手指標籤。

請注意，我們只訓練五個這樣的類，每個手指一個，創建一個獨立於單個手的分類器。

我們觀察到在給定任務上的最佳性能，我們的分類器實現為遵循 VGG 風格架構的多層前饋一維卷積神經網絡 [41]。

圖 5 顯示了我們的捲積架構，考慮到輸入窗口的固定長度和卷積層的強度以捕獲信號中的局部依賴性，這是一個合適的選擇。

為了增加感受野，我們使用更深的網絡和更小的內核。

直觀地說，與具有較大內核的淺層網絡相比，由於層之間存在額外的非線性，這使得網絡能夠提取更強大的特徵。

具體來說，該網絡由五個塊組成，每個塊由兩個卷積層組成，內核大小為 3，它們連接到一個最大池化層，以將輸入沿時間維度減少 1 倍

2. 卷積塊後面是一個自適應平均池化層和兩個線性層。

最終線性層的輸出饋入一個 softmax 激活函數，該函數估計手的五個手指上的概率分佈。

我們將時域中的加速度計數據表示為維度的輸入張量 [傳感器的堆疊軸 × 窗口大小]，其中在時間軸上應用卷積，並將每幀堆疊的傳感器數據視為輸入通道。

我們應用了幾個數據預處理步驟來提高 TapID 的性能。

利用左右手的對稱性，我們鏡像左手的數據，以接收來自雙手的可比輸入信號。

我們交換對稱排列的加速度計的數據，並根據傳感器的方向反轉 x 或 y 軸。

這減少了組合數據集的可變性並有助於我們分類器的學習過程，現在可以將其應用於來自任一腕帶的信號。

接下來，我們通過沿每個時間軸應用前向差分算子來估計每個傳感器經歷的抖動。

這個操作可以被認為是尋找信號的離散導數，它放大了傅里葉域中的高頻分量，我們認為這些分量包含我們尋求的判別手指信息。

此步驟還減少了由手臂和手部運動引起的低頻偽影。

最後，我們沿著數據集中每個通道的時間維度對生成的片段進行標準化。

完整的分類網絡包含超過 210 萬個可訓練參數。

我們在 Pytorch 中實現了該模型，並使用 Adam 優化器 [28] 以 10−4 的學習率、ε = 10−8 和 β = (0.9,0.999) 對它進行了 30 個 epoch 的訓練，批量大小為 16。

此外，我們在峰值事件周圍嘗試了不同的段大小，並在 128 幀（~95 ms）的窗口長度下實現了最佳性能。

我們在跨越多個用戶和會話的數據集上優化了這些超參數。

4.3 將手部跟踪與TapID 事件相結合

在基於小部件的VR 用戶界面（例如，使用按鈕、滑塊、複選框等的用戶界面）中使用TapID 時，我們需要在TapID 檢測到它的發生後檢索點擊的位置。

為了實現這一點，我們通過集成到現代耳機中的商品手部跟踪來補充 TapID 的分類輸出。

在確定了哪個手指導致了點擊後，我們從一組被跟踪的手中檢索指尖位置，並將其分配到 TapID 的點擊事件中。

如果檢測到的手指尖不在 VR 中的表面附近，我們的系統會簡單地將檢測到的事件視為無意輸入。

在我們的實驗中，我們發現 30 毫米是可靠檢測和拒絕的可行最大表面距離。

在 VR 中觸發輸入事件的有效表麵包括用戶自己的身體，因此我們的系統可以在 UI 元素錨定到用戶手臂時檢測到身體上的觸摸輸入（圖 1d）。

在這種情況下，我們的系統將指尖位置與手臂表面的位置進行比較，我們從具有恆定半徑（40 毫米）加上上述增量區域的 3D 跟踪關節近似得出。

我們的 VR 系統使用 Unity 的 3D 遊戲引擎進行渲染，並使用 Velt [15] 實現整體數據流。

Oculus Quest 頭戴式設備可渲染我們的 VR 應用程序，其 SDK 內置了對頭戴式設備位置和手部姿勢的內向外跟踪。

雖然我們發現來自 SDK 的跟踪數據在我們的實驗中具有足夠的精度來確定用戶的指尖位置，但它無法檢測具有任何實際可靠性的被動表面上的觸摸或具有任何使用由內向外的可靠性的身體觸摸輸入單獨跟踪。

更嚴重的是，快速的手指敲擊表現為被跟踪手部的抖動運動，儘管在允許觸摸檢測的位置更新中沒有。

當手指從相機的角度被部分遮擋時，例如當手指在手的向下運動期間被其他手指遮擋時，跟踪此類事件變得更加不合適。

我們還發現，指尖位置略微落後於真實手指位置，這對於檢索表面上的位置是可以容忍的，但對於檢測觸摸輸入或檢索交互過程中觸摸事件的確切時間戳而言，這是可以容忍的。

出於這些原因，我們僅從手部跟踪中提取指尖位置，但依靠 TapID 來檢測點擊事件並顯示點擊手指。

然後，我們的系統在通過將指尖位置投影到 UI 所在的相應表面上獲得的 2D 位置觸發事件。

4.4 管道延遲

我們將傳感和支持 VR 的手部跟踪相結合，引入了多種延遲來源。

連續手部跟踪本身需要一些延遲，這取決於 VR 設備和實施的手部跟踪方法。

關於我們自己處理的延遲，從物理點擊到在 Unity 中顯示其效果的總延遲大約為 130 毫秒。

我們通過手動計算高頻視頻記錄中的幀來估計這個值。

此持續時間包括由所有硬件操作、通信、處理和輸出引起的延遲。

我們憑經驗確定了 TapID 的傳感、板載處理和通信管道的延遲，平均約為 10 毫秒。

基於機器學習的分類，包括我們對每個點擊事件的緩衝數據採集，增加了 50–

60 毫秒。

因此，檢測敲擊和識別手指的延遲包括：

1. 傳感器 FIFO 中的數據：3 ms（最大值：6 ms）

2. SoC處理和串行傳輸到主機：3 ms

3.點擊事件細化（等待最高幅度）：15 ms

4. 緩衝直到數據窗口完成：47 ms（64 個樣本）

5. 神經網絡推理時間 2 ms (GeForce GTX 1050 Ti) 請注意，第 3 步和第 4 步平均重疊約 10 ms，因為細化的峰值通常位於細化窗口的開頭。

此外，通過串口傳輸數據的延遲取決於主機上的緩衝區設置。

總而言之，我們將低延遲點擊輸入與由內而外的指尖跟踪相結合，獨特地使用戶能夠使用他們從常規觸摸設備中知道的快速輸入，通過觸摸可靠地與 VR 應用程序交互。

我們在第 6 節展示了幾個利用我們的低延遲方法的用例。

5 技術評估

本次評估的目的是驗證我們檢測手指敲擊事件並識別導致它的手指的傳感原理和信號處理方法。

我們還旨在確定不同位置和組合的腕式加速度計在多大程度上有助於我們方法的準確性。

為此，我們比較了圖 6 所示三種配置中慣性數據分類的 F1 分數：TAPID（嵌入彈性矽膠帶中的兩個加速度計）、FITBIT（剛性安裝在主 PCB 上的兩個加速度計）、SMARTWATCH（單個低噪聲板載加速度計）。

我們還報告了同時使用所有 4 個傳感器的基線的 F1 分數。

數據捕獲的第二個目的是建立一個語料庫，用於訓練我們在演示應用程序中用於在線操作的跨人點擊分類器。

我們分兩部分進行評估；

在第 1 部分中，我們研究了點擊事件檢測的準確性。

在第 2 部分中，我們評估了我們的手指識別網絡實現的準確度。

評估的兩個部分共享相同的任務和程序，但由不同的參與者進行。

5.1 任務

被試舒適地坐在桌椅上，將下臂放在桌子邊緣，用實驗者指定的手指反复敲擊桌面。

沒有提供其他說明，參與者控制自己的敲擊頻率、強度和敲擊位置、手指和手部動作。

實驗者小心翼翼地簡單地使用“點擊”這個詞來指代實驗期間的所有觸摸事件。

5.2 程序

本研究從簡單介紹TapID開始，實驗者將其與普通智能手錶相關聯。

然後，實驗者記下參與者的年齡和性別，並測量他們的腕圍。

在整個研究過程中，實驗者指示參與者在每次訓練期間將帶子戴在哪隻手臂上，以及在每一輪敲擊時使用哪根手指。

對於每一輪，參與者將 TapID 設備放在指定的手腕上，並按照指示用每根手指重複敲擊 30 次，然後再次取下腕帶。

每個會話中的一個塊由每隻手的每個手指敲擊 30 次組成，總計 30 × 5 個手指 × 2 個手臂 = 每塊敲擊 300 次。

參與者執行了三個這樣的塊，中間有短暫的休息，總共產生了 300 個抽頭×3 個塊 = 900 個抽頭。

每個參與者在 20 分鐘內完成了研究。

5.3 參與者 我們招募了 18 名參與者（14 名男性，4 名女性，年齡 19-

57，平均 = 28.4 歲）。

參與者的平均腕圍為 168 毫米（SD = 10 毫米，最小 = 146 毫米，最大 = 185 毫米）。

每個參與者都收到了一小筆酬金。

在 18 名參與者中，4 名（1 名女性，年齡 23-

29，平均 = 25.3 年）還參與了評估我們的敲擊檢測的性能。

他們的手腕周長從 158 毫米到 176 毫米不等。

5.4 評估敲擊事件檢測

為了記錄表面敲擊的地面實況註釋，我們使用聽診器記錄表面聲學信號，聽診器被覆蓋並貼在桌面上（圖 9）。

在評估過程中，參與者保持安靜以避免將偽影引入記錄的信號中。

該子評估共產生了 3600 次點擊事件。

使用我們的 RC 分數作為衡量標準（方程式 1），我們的抽頭檢測在 TAPID 配置中實現了 F1,tap = .996（精度 99.7%，召回率 99.5%）的準確度。

對於 FITBIT 和 SMARTWATCH，檢測也達到了 F1,tap = .996。

5.5 評估敲擊事件識別

在這個子評估中，基本事實來自於實驗者的指令，該指令用於敲擊，實驗者登錄我們的數據收集工具進行事後標籤分配。

5.5.1 TAPID 配置中的結果 我們計算所有參與者褲子每個手指的平均 F1 分數。

此外，我們報告了所有參與者的宏觀 F1 分數的平均值和標準差，通過在類 F1 分數中取未加權平均值，將所有手指視為同等重要。

跨會話識別準確性：我們按參與者和塊分割記錄的數據。

對於每個參與者，我們進行了 3 折交叉驗證，使用兩個塊進行訓練和一個塊進行測試，同時忽略其他參與者的所有數據。

在所有參與者中，準確度平均為 F1,finger = .93 (SD=0.06)。

跨人員識別準確性：我們進行了 18 倍交叉驗證，對每個參與者的事件進行測試，並對所有其他參與者進行培訓。

準確度為 F1,finger = .87 (SD=0.09)。

帶細化的跨人識別準確度：除了純粹的跨人測試外，我們還評估了來自參與者集合中的單個輕敲事件的細化對跨人識別準確度 F1,finger 的影響。

同時，我們評估了訓練數據集大小對平均跨人識別準確率的影響。

圖 10 顯示了參與者之間的平均 F1 分數，取決於用於細化的隨機抽頭數（行）和取決於用於訓練的參與者數據的數量（列）。

對於這個評估，我們首先在記錄的（i）0、（ii）4、（iii）8、（iv）12和（v）17個隨機抽取的參與者（不包括在測試集。

然後，我們僅使用測試參與者的每個手指的前 (a) 1、(b) 10、(c) 30 和 (d) 60 個輕敲事件對網絡進行了另外 30 個 epoch 的微調，並從測試中刪除了這些樣本放。

我們在該參與者的剩餘塊上測試了精煉分類器。

微調後，我們訓練的網絡產生的 F1 分數隨著訓練集中參與者的數量以及細化抽頭的數量而上升。

通過 18 倍交叉驗證和 (iv) 所有 17 名參與者的訓練數據進行訓練，我們的分類器達到 (a) F1,finger = .86 (SD=0.07), (b) F1,finger 的平均 F1 分數= .91 (SD=0.07), (c) F1,finger = .94 (SD=0.05), 和 (d) F1,finger = .96 (SD=0.05) 經過相應的細化。

5.5.2 其他配置的結果 對於我們的基線比較，圖 7 列出了使用所有 4 個 IMU 的基線以及相應傳感器配置獲得的所有分數。

與 FITBIT 和 SMARTWATCH 配置相比，TAPID 始終產生更高的 F1 分數。

相比之下，使用所有 4 個 IMU 進行訓練和測試比 TAPID 增加了邊際改進，並且僅在某些情況下。

5.6 討論

評估證實了我們的傳感原理的有效性以及我們在腕戴設備上捕獲身體耦合事件的實現。

它還證實了我們原型的 TAPID 配置，在矽膠錶帶中嵌入了兩個低成本、低功耗的 IMU。

關於敲擊檢測，TapID 達到的準確度對於實際用途來說已經足夠高了，但它也在 FITBIT 和 SMART WATCH 配置中。

這證實了最近相關工作的結果（例如，[18]），儘管 IMU 傳感器放置在手腕上，用於不受阻礙的手，而不是將它們放在手指上。

在我們的點擊檢測中仍然存在一個折衷：減少回退時間允許更快的交互，但會增加一個（強）點擊產生兩個事件（即一個誤報）的可能性。

至於我們的分類器識別單個手指的性能，不出所料，我們在盡可能多的訓練中看到了最好的跨人準確率，並使用特定於個人的訓練數據進行了細化，通過這種方式達到了高達 0.96 的 F1 分數（圖 10 ）。

然而，值得注意的是，單個人特定的校準程序在記錄會話中保持不變，通過一次性的入職步驟設置新用戶以達到 .

96. 即使只用 10 個樣本進行微調（可以在一分鐘內輕鬆記錄），TapID 也能以 F1,finger = .91 的準確度識別正確的手指，尤其是對於任何給定的觸摸事件。

然而，如果沒有細化，跨人分類達到了 F1,finger = .87 的準確率，我們認為這仍有改進的空間，以提高到實用水平。

也就是說，這個 F1 分數與在相關工作中取得的結果相比，在跨人分類中使用不受阻礙的手指和手（例如，[4,7]）的可比設置。

有趣的是，與 FITBIT 和 SMARTWATCH 配置相比，我們的方法在 TAPID 配置中實現了最大的性能提升（圖 7），而針對個人的改進縮小了配置之間的差距。

看看單個手指的表現，我們的方法最可靠地識別拇指和小指敲擊，而區分中指和無名指被證明更具挑戰性。

雖然跨會話的所有手指 F1 分數都 > .89，但對於跨人 F1 分數，無名指下降到 0.82。

圖 1 正確的混淆矩陣

圖 8 說明該模型特別難以區分食指、中指和無名指。

然而，僅輕敲 10 次即可改進手的三個內指之間的區分，並將所有手指的 F1 分數恢復到 > 0.87 的值（參見圖 10 和圖 7）。

詳細檢查結果，人的腕圍等物理測量可能會影響分類準確性。

對手腕周長最接近我們人群平均值的 9 名參與者進行交叉驗證，沒有細化的交叉人分類上升到 TAPID 的 F1,finger = .92（與使用 F1,finger = .82 相比我們人口的另一半進行驗證）。

所有傳感器配置都出現了類似的趨勢，原因之一可能是 IMU 之間的距離恆定。

基於這些結果，我們假設來自越來越多手腕越來越大的參與者的訓練數據會降低這種影響，但會在不改進的情況下增強我們的跨人準確性。

6 用例

TapID 可用於許多 VR 場景中的互補高保真輸入。

在本節中，我們首先描述 TapID 的更廣泛場景和設想用例。然後，我們展示了我們實施的幾個應用程序，以展示我們方法的有用性。

6.1 場景

在典型的 VR 場景中，用戶與無形物體進行交互，這些物體缺乏所有物理可供性，因此需要通過手持控制器或僅手進行交互。

然而，空中輸入——尤其是在長時間互動的情況下——通常會引發疲勞，因為缺乏身體可供選擇的休息方式 [40]。

因此，我們認為，迄今為止，VR 應用程序的主要成功在於涉及全身的遊戲（例如，Beat Saber [3]）。

相比之下，即使是最近嘗試複製辦公環境的生產力應用程序（例如，Facebook Infinite Spaces [14]）也關注內容消費而不是內容創建。

我們認為，這是空中輸入的交互模式不充分的直接結果，通常會改裝為專為鍵盤、鼠標和触摸而設計的界面。

在 VR 中，除了手持控制器之外，目前沒有任何設備可以促進這種交互，這會帶來短期的新奇感，但迄今為止在實現長期生產力方面還沒有說服力，尤其是與傳統的桌面輸入設備相比。

我們認為，如果輸入方式不有效、精確和不疲勞，沉浸式 VR 生產力應用程序的潛力仍未開發。

許多生產力工作的特點是持續交互，通常以流暢和快速輸入的形式，通常是雙手，並且通常涉及完成任務的工具。

我們的原型 TapID 現在提供了另一種交互概念，其靈感來自實體辦公室中的交互，但適用於支持 VR 中潛在的無限虛擬內容空間。除了基於生產力的場景之外，TapID 還可以用於與以前（主要是基於半空中）VR 體驗不同的更有趣的用例。

例如，TapID 支持需要精確計時的表面輸入的 VR 遊戲（例如，節奏遊戲）。

6.2 應用程序原型

我們實現了幾個應用程序，展示了我們通過 TapID 在 VR 中快速點擊交互與 VR 耳機介導的空間接口的獨特組合。

在這些演示中，用戶位於 VR 中，並被桌面和触摸用戶界面中典型的內容所包圍，例如照片集和 Web 瀏覽器。

為了促進與用戶界面的交互，我們開發了一系列基於點擊的虛擬輸入設備，用戶可以通過用特定手指點擊表面上所需的位置來打開和定位。

例如，左小指可以召喚箭頭鍵，而右小指觸發數字鍵盤出現，拇指可以在雙手之間召喚一個簡單的鍵盤。

雖然我們的應用程序在交互過程中顯示手，但在以自我為中心的使用過程中，我們使用彩色球體渲染指尖和触摸，顯示一條向下投射到表面的線並遵循之前的 UI 建議 [17]。

我們的應用程序的屏幕截圖如圖 1 和

11. 照片排序：熟悉多點觸控設備，我們實現了一個照片排序應用程序，如圖 11a) 所示。

該應用程序使用 TapID 可靠的點擊檢測來觸發拖動事件以移動和調整圖片大小。

命令輸入菜單支持逐步調整大小和旋轉圖像。

通過我們的方法，我們可以輕鬆地將這些命令分配給每個手指。

該原型在沒有桌子的情況下也可以在身體配置中工作，一條腕帶戴在一隻手臂上，用戶的另一隻手臂作為敲擊表面（見圖 1d）。

文檔編輯：我們設計了一個簡單的鍵盤輸入小部件來向文檔添加文本（圖 11b），補充了現有文檔創作工具（例如，Microsoft Word、Power Point）的使用。

可以通過雙擊來選擇單詞，例如更改其格式。

此外，我們的箭頭鍵虛擬輸入（用左手小指召喚）有助於快速調整文本光標的所需位置。

這證明了我們對傳統觸摸功能的支持，將我們基於手指特定敲擊的虛擬輸入設備與操作系統提供的控件和為觸摸設計的現有應用程序無縫集成。

數字輸入：用戶可以打開一個虛擬數字鍵盤小部件來輸入數字和操作，如圖 11c 所示。

這個小部件方便地放在手的下方，以便用戶可以用一隻手的所有五個手指快速輸入數字。

例如，當召喚和使用計算器作為用戶界面中交互流程的一部分時，這種可供性很有用。

鋼琴：我們的最終演示是一個虛擬鋼琴應用程序，如圖 11d 所示。

這款 VR 鋼琴充分利用 TapID 進行時間精確輸入，讓用戶可以用任何手指敲擊鋼琴鍵。

VR 中的視覺反饋、表面的被動觸覺反饋以及我們方法的低延遲相結合，使我們能夠在演奏這種虛擬樂器的同時保留用戶敲擊輸入的準確節拍。

7 限制

在達到生產級性能之前，我們當前的實施中仍然存在一些限制。

除了當前必要的用戶特定改進以實現可靠的準確性外，我們還沒有調整或評估我們的方法識別多點觸摸事件的能力——此時，雖然同時進行雙手輸入，但僅檢測到單個手指。

我們當前實施背後的第二個假設是使用桌子或用戶自己的身體作為被動輸入表面我們沒有通過實驗和定量評估我們當前實施在由塗層木材製成的辦公桌以外的表面上的檢測準確性。

我們也沒有明確改變或控制敲擊強度。

我們已經在靜止和坐著的環境中測試和評估了我們的系統，沒有考慮使用期間的身體運動（例如，在移動場景中）。

對於支持的輸入姿勢，TapID 檢測並識別彎曲手指的敲擊。

但是，在這種情況下，由於我們當前依賴耳機報告的輸入位置，因此導出的輸入位置可能不准確甚至不可用，因此未知。

為了使我們當前的系統正常工作，產生輸入事件的手必須位於 VR 耳機的跟踪區域，否則腕戴式傳感器上的輸入事件將被視為疏忽而忽略。

這意味著，目前，用戶需要查看他們在 VR 中與之交互的內容，包括在所示鍵盤上輸入的文本，因此不支持裸眼操作。

所有交互也必須是直接的，以前的工作發現這會導致 3D 場景中的更高錯誤 [39]，但 Knierim 等人的工作。VR 中的文本輸入顯示是有益的，因為經驗豐富的打字員通過在 VR 中看到他們的手和鍵盤來提高他們的打字性能 [29]。

關於我們的方法在商品智能手錶上實施的適用性，我們承認用戶佩戴兩塊智能手錶可能顯得不合理，而我們目前的方法需要雙手操作。

然而，就像目前使用的輸入控制器一樣，未來用戶戴上兩個小型腕帶進行 VR 中的所有交互似乎是可以理解的。最後，我們當前的推理管道依賴於中等 GPU，目前不適合嵌入式處理。

8 結

因此，我們的方法為持續使用期間的可靠和長時間交互奠定了基礎，用戶可以將手臂和手放在表面上。

對於被動表面（例如桌面）上的每次點擊，TapID 可靠地檢測事件並識別用於觸摸的手指。

這允許完全跟踪用戶手部的 VR 系統添加缺少的關鍵元素：快速的表面接觸檢測，另外還可以區分是哪個手指導致了動作，這樣 VR 應用程序就可以觸發應用程序中的輸入事件。

在我們對 18 名參與者的評估中，TapID 在跨會話設置中為輕敲檢測和手指識別提供了令人信服的準確度。

使用僅在跨人員數據上訓練的分類器進行的交叉驗證測試顯示準確度下降，但是 TapID 可以從中恢復，但是，使用少量用戶的點擊事件進行細化訓練。

總體而言，我們認為我們在本文中概述的方法是對現有 VR 系統的可行補充，它允許熟悉的輸入模式（即觸摸輸入）轉移到空間 VR 系統，甚至可能在日常場景中。

在這方面，一個有希望的見解是評估我們在 FITBIT 配置中的方法，僅使用現有商品可穿戴設備中發現的 TapID 的板載 IMU 傳感器 [43]，在實施我們的方法時產生了相當的精度。