

Protokoll der Sitzung vom: 10.11.2016

Peter Enenkel
Hochschule Darmstadt
Haardtring 100, 64295 Darmstadt Deutschland
Email: peter.enenkel@stud.h-da.de

I. ARBEITSPAKETE

Datentransfer via Sockets: Ist grundsätzlich möglich, aufgrund der notwendigen asynchronen Entkopplung aber nicht trivial umzusetzen. Es existiert eine Referenzimplementierung im HoloToolkit.

Zugriff auf Webcam via App: Die Hololens stellt eine highlevel API für den Zugriff auf die Webcam zur Verfügung. Das Abspeichern eines Bildes mittlerer Qualität dauert etwa 2 Sekunden. Darüber hinaus besteht die Möglichkeit Videos in wählbarer Qualität und Framerate zu erstellen. Es gibt die Option sowohl Bilder als auch Videos als Stream abzugreifen. Es besteht die Hoffnung, dass sich hierdurch eine Beschleunigung des Vorgangs erreichen lässt. Ein lowlevel Zugriff auf die Webcam ist nach derzeitigem Erkenntnisstand nicht möglich.

Zugriff auf Webcam via REST API: Es hat sich herausgestellt, dass die über die REST API verfügbaren Bilddaten den gleichen erheblichen Verzögerungen unterliegen, wie die „Live-View“ des Device Portals.

Fähigkeiten des Spatial-Mappings: Es wurden erfolgreich Parameter identifiziert, die die Erfassungsfähigkeiten verbessern können.

Erstellen einer Referenz-App: Um die oben genannten Vorgänge testen zu können wurde ein minimales funktionsfähiges Hololens-Unity-Projekt benötigt. Die Erstellung eines solchen hat sich als schwieriger als erwartet erwiesen, wurde aber schlussendlich gemeistert.

Offene Punkte:

- Erkennung eines Tennisballs durch CV

- Auslesen von Metainformationen (wie Blickrichtung, etc.) aus Bilddaten der Hololens. Laut Dokumentation sind diese Informationen hinterlegt, bislang konnten wir sie allerdings noch nicht auslesen.

II. DERZEITIGER STAND

Es besteht ein erstes, eigenes, funktionsfähiges Hololens-Unity-Projekt. Es wurden alle vom HoloToolkit-Repository bereitgestellten Assets eingebunden, und darüber hinaus einige ausgewählte davon aktiviert/verwendet um eine minimale lauffähige App zu erhalten, die grundlegende Benutzer-Interaktion erlaubt. Von, für unsere Zwecke, größter Bedeutung sind hier das Spatial-Mapping, der Gaze gesteuerte Cursor, sowie das Aufheben und gezieltes Platzieren von geometrischen Primitiven (auf dem erstellten Spatial-Mesh) per Tap.

Es wurde erfolgreich mit der Auflösung des Spatial-Meshings experimentiert, so dass nun die Erfassung von tennisballgroßen Objekten zumindest rudimentär funktioniert. Gleichzeitig wurde allerdings auch festgestellt, dass die Erfassung von bewegten Objekten via der ToF-Kamera definitiv nicht möglich ist. Zwar lässt sich die Auflösung und die Refreshrate des Spatial-Mappings theoretisch beliebig anpassen, allerdings ist die Leistung der verbauten Hardware hierzu zu limitiert. Dass als Standard eine Refreshrate von mehr als drei Sekunden eingestellt ist verdeutlicht dies. Auch sollte man sich darüber im Klaren sein, dass Erfassung nicht gleich Erkennung ist. tennisballgroße Objekte werden auf kurze Distanz (d.h. 1-2m) als ein bis zwei Tetraeder erfasst, dies reicht für eine Erkennung natürlich nicht aus.

Somit wurde die Notwendigkeit der zusätzlichen Bildverarbeitung wie erwartet bestätigt. Nach der Recherchearbeit der letzten Wochen und den bisher gewonnenen Erfahrungen, wird es immer unwahrschein-

licher, dass eine solche Bildverarbeitung direkt auf der Hololens ausgeführt werden kann. Neben den Performance-Beschränkungen fallen hier vor allem die Inkompatibilitäten der Hololens mit den gängigen CV-Bibliotheken ins Gewicht. Ein Einsatz auf der Hololens würde den Einsatz von kaum getesteten und schlecht dokumentierten Nischenlösungen mit stark eingeschränktem Funktionsumfang erfordern, was unserer Einschätzung nach nicht vertretbar ist.

III. PROBLEME

Die Ursache der sporadisch auftretenden Schwierigkeiten beim Deployment via WiFi sind nach wie vor unklar. Glücklicherweise funktioniert das Deployment via USB mittlerweile weitestgehend problemlos. Durch die geringe Anzahl an Teilnehmern, die momentan vorherrschende gemeinschaftlich explorative Arbeitsweise, sowie natürlich die Tatsache, dass es sich um ein einzelnes Endgerät handelt, ist allerdings die Datenbasis für derlei Aussagen limitiert: Es wird weitestgehend von einem einzelnen privatem Laptop aus deployed. Der Einsatz des Projektrechners hat sich bislang noch nicht durchgesetzt.

Im Hinblick auf die, aus dem letzten Protokoll entstandenen, Fragen sei an dieser Stelle noch einmal auf die korrekte Nomenklatur hingewiesen. Es sind dabei die Begriffe des Pairing, Building, Deployment und des Debugging von einander zu unterscheiden. Hinzu kommt das über das Device Portal ausgelöste Pairing, welches wiederum anders funktioniert.

Pairing beschreibt den Vorgang der Identifikation des Entwicklungsrechners gegenüber der Hololens. Theoretisch ist dies für jede Kombination von Entwicklungsrechner und Device (d.h. Hololens) ein einmaliger Vorgang. Zumindest solange das Listing der bekannten Rechner nicht explizit auf der Hololens gelöscht wird. Das Pairing eines Entwicklungsrechners lässt sich nicht explizit starten, es erfolgt implizit beim ersten Deploymentvorgang, zumindest theoretisch. Eine Lösung für den Fall, dass, wie vor zwei Wochen geschehen, das Pairing nicht automatisch initiiert wird ist bislang nicht bekannt.

Building bezeichnet zum einen den Vorgang aus dem Unity-Projekt ein Hololens-geeignetes UWP-App Projekt für Visual Studio zu generieren und zum anderen dieses dann für den Einsatz auf der Ho-

lolens zu kompilieren. Größtes Hindernis hierbei ist die korrekte Installation der Entwicklungs-Toolchain, wobei jeweils auf die richtigen Versionen zu achten ist.

Deployment bezieht sich auf die Übertragung der kompilierten App auf die Hololens (oder den Emulator). Dies ist, ein erfolgreiches Pairing vorausgesetzt, grundsätzlich per WiFi, USB oder dem Device Portal möglich. Das Deployment via Device Portal ist relativ umständlich und wurde bislang von uns noch nicht eingesetzt. Das Deployment via WiFi setzt eine geeignete WLAN Konstellation voraus und hat sich nach unserer Erfahrung als anfällig und vergleichsweise langsam erwiesen. Das Deployment via USB funktionierte bislang am zuverlässigsten.

Debugging ermöglicht das Stoppen der deployten App durch Haltepunkte im Quellcode. Hierfür ist eine dauerhafte Verbindung mit dem Entwicklungsrechner notwendig, was bei USB-Deployment etwas unpraktisch ist. Alternativ kann mittels [STRG-F5] auch ohne Debugger deployt werden. In diesem Fall kann die Verbindung nach erfolgreichem Deployment getrennt werden. (Es gibt Anzeichen, dass nicht alle Kombinationen von WiFi/USB und Debug/No-Debug zuverlässig funktionieren.)

IV. WEITERE VORGEHENSWEISE

Nachdem nun ein grundlegendes Basis-Projekt besteht mit dem die Fähigkeiten und Schwächen der ToF-Kamera demonstriert werden können, ist das nächste Etappenziel eine Proof of Concept Implementierung der Remote-Bildverarbeitung.

Darüber hinaus ist es notwendig, dass das ultimative Projektziel konkretisiert bzw. bestätigt wird. Momentaner Stand ist hier das Erfassen, Erkennen und Tracken von (bewegten) Tennisbällen; unabhängig davon ob dies im Umfeld der Bühnentechnik einen sinnvollen Anwendungsfall darstellt oder nicht.

Unabhängig von der eigentlichen Entwicklungsarbeit ist angedacht, das Deployment von weiteren Rechnern aus zu testen, in der Hoffnung weitere Erkenntnisse über die bislang aufgetretenen Ausfälle und deren Ursachen zu erhalten.