

DepthAnything v1

ViTs Modell (Vision Transformer small = kleinstes Modell)

Auf dem Dell XPS 13 7390 mit 16 GB Arbeitsspeicher

Benötigen wir noch 3D-Kameras wie die Azure Kinect, oder reichen monokulare KI-Modelle aus?

Vergleich: Azure Kinect vs. Depth Anything v1

Installation von DepthAnything v1

Voraussetzung ist es, dass Python und Git auf dem Rechner installiert sind. Ich habe die Version Python 3.13.2 und Git Version 2.49.0.windows.1 auf meinem Rechner.

Anschließend wird DepthAnything v1 über die Eingabeaufforderung (CMD) anhand dieses Befehls heruntergeladen „git clone <https://github.com/isl-org/Depth-Anything.git>“.

Darauffolgend wird über den Befehl „cd Depth-Anything“ in den Zielordner gewechselt, um über den anschließenden Befehl „pip install -r requirements.txt“ die benötigten Python-Pakete zu installieren. Dadurch wird PyTorch, Bildverarbeitungstools und weitere Bibliotheken, die das Modell benötigt, installiert.

Nach der Installation habe ich angefangen, ein einzelnes Bild anhand dieses Befehls „python run.py --encoder vits --img-path meine_bilder/testbild.jpg --outdir ergebnisse/“ zu testen.

Ich habe das ViTs-Modell gewählt, da es das kleinste und effizienteste Modell von Depth Anything v1 ist und sich somit besonders gut für einen schnellen und ressourcenschonenden Vergleich eignet.

Depth Anything v1 gibt es in drei Größen:

ViTs (Small) – ca. 24.8 Mio. Parameter

ViTb (Base) – ca. 97.5 Mio. Parameter

ViTl (Large) – ca. 335.3 Mio. Parameter

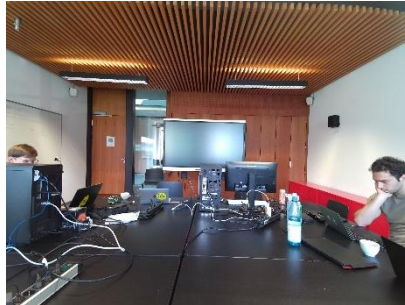
Einrichtung der Azure Kinect

Um die Azure Kinect zu testen, wurde die Azure Kinect SDK heruntergeladen. Anschließend lässt sich über den Azure Kinect Viewer auf die Kamera zugreifen und ein live Bild anzeigen.

Da sich allerdings über den Viewer kein Tiefenbild-Export machen ließ, wurde das Python-Paket pyKinectAzure über den Befehl „pip install pykinect_azure“ installiert und das Skript, das als Beispiel aufgelistet war, auf die Verwendung angepasst. Dadurch ließen sich die RGB- und Tiefenbilder mit einem Skript speichern.

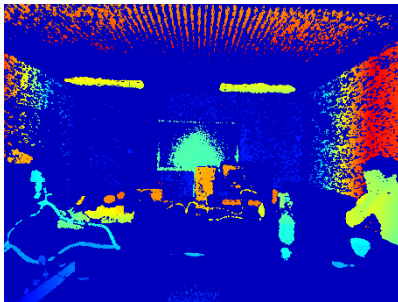
Genauigkeit & Konsistenz

RGB Bild

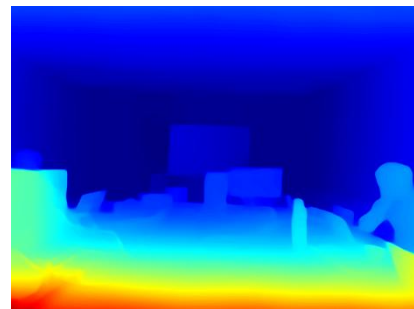


Tiefenfarbbild

Azure Kinect:



Depth Anything v1:



Damit man die Bilder visuell vergleichen kann, wurden die beiden Tiefenbilder über einen Python-Code in die gleiche Farbskala gebracht. Dabei wurden die Tiefenbilder auch auf die gleiche Größenskala gebracht, um unterschiedliche Skalen in den Tiefenwerten von DepthAnythingv1 zu vermeiden.

Anhand der beiden Tiefenbilder kann erkannt werden, dass die Kinect eine hohe Detailgenauigkeit besitzt. Es können feine Strukturen an den Personen links und rechts sowie an den Objekten auf dem Tisch erkannt werden. Zudem werden die unterschiedlichen Tiefen klar und kontrastreich dargestellt. Allerdings lassen sich leichte Rauscher in der Tiefe erkennen, besonders an Kanten und in dunkleren Bildbereichen.

Das Tiefenfarbbild des DepthAnything v1 wiederum wirkt sehr gleichmäßig. Es fehlen Details wie kleine Objekte auf dem Tisch oder feine Strukturen. Die Tiefenkarte ist auf größere Strukturen bezogen plausibel, zeigt jedoch kaum Details kleiner Objekte. DepthAnythingv1 generiert relative Tiefenkarten und liefert daher keine echten Entfernungen. Für genaue Messungen ist es dadurch weniger geeignet. Die Azure Kinect wiederum liefert echte Tiefenwerte und ist daher für präzise Messungen besser geeignet.

RGB Bild

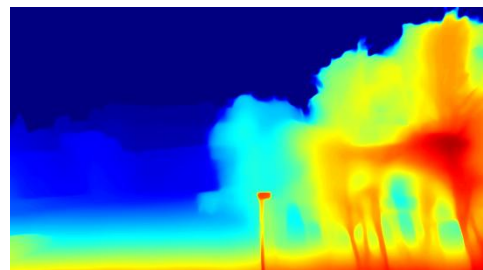


Tiefenfarbbild

Azure Kinect:



DepthAnythingv1:



Hier lässt sich erkennen, dass das Bild von Azure Kinect fast keine Tiefe zeigt. Sehr dunkelblau und kaum erkennbar, nur ein paar kleine Punkte/Tiefenwerte sind sichtbar. Die Azure Kinect ist somit bei Außenszenen und großen Entfernungen überfordert.

Bei DepthAnything v1 wird die Szene gut erkannt, trotz der vielen Details. Die Tiefe der Bäume, der Hintergrund, die Wiese haben eine deutliche Tiefenabstufung. Die Darstellung ist Glatt, aber relativ konsistent. Somit ist DepthAnythingv1 für Außenszenen deutlich besser geeignet als die Azure Kinect.

Weitere Beispiele:

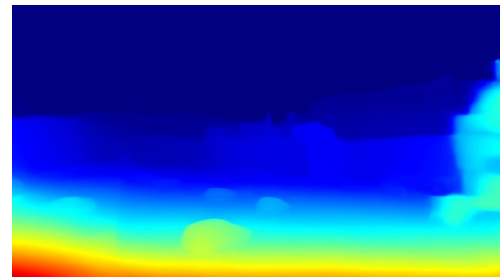
RGB Bild



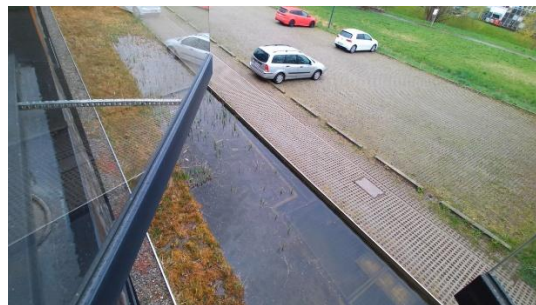
Azure Kinect:



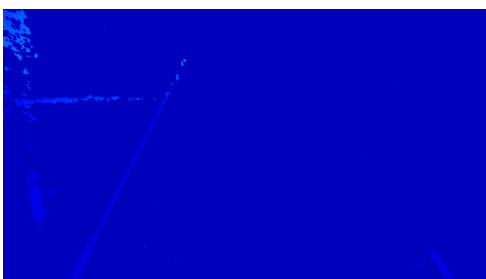
DepthAnythingv1:



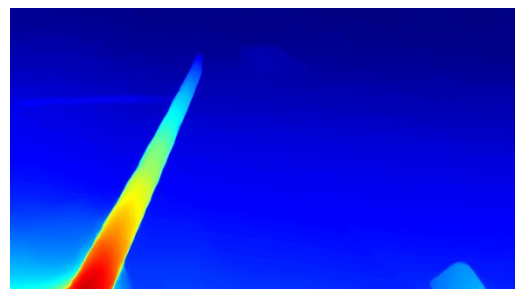
RGB Bild



Azure Kinect:



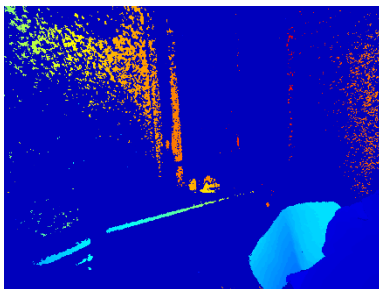
DepthAnythingv1:



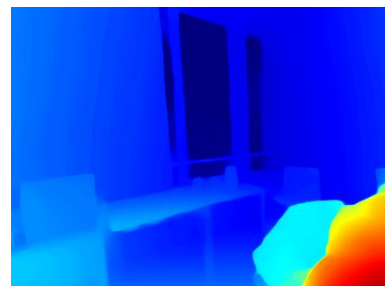
RGB Bild



Azure Kinect:



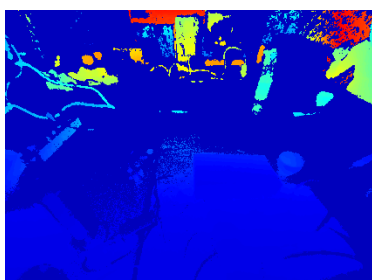
DepthAnythingv1:



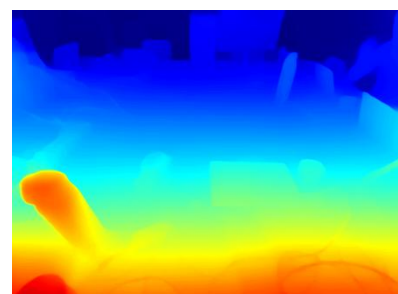
RGB Bild



Azure Kinect



DepthAnythingv1:



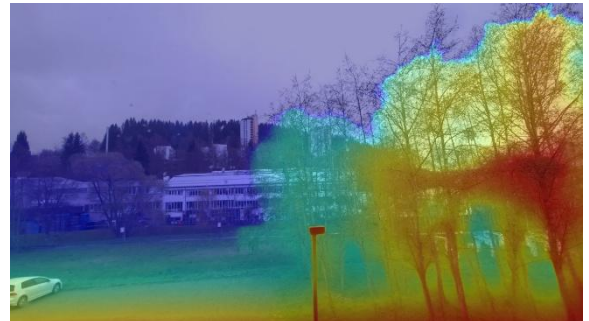
Tiefenbilder überlagern (Overlay)

Durch das Überlagern können Tiefenbilder anschaulich mit dem realen Bild kombiniert werden.

Azure Kinect:



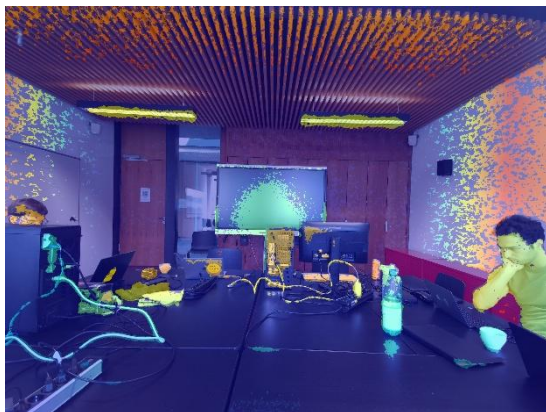
DepthAnythingv1:



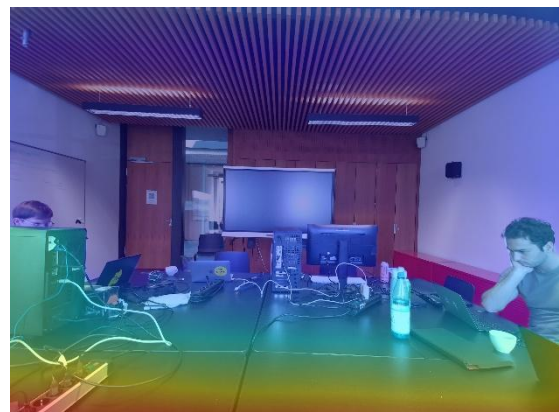
An dem linken Overlay sehen wir, dass die Tiefeninformationen bei der Kinect fast nicht sichtbar sind. Die Kinect hat somit draußen kaum Tiefe erkannt.

Bei dem Overlay von Depth Anythingv1 sind die Tiefeninformationen der KI klar über das RGB-Bild gelegt. So wird gezeigt, dass Depth Anything die Szene sinnvoll interpretiert und eine brauchbare Tiefenschätzung liefert.

Azure Kinect:



DepthAnythingv1:



Hier sehen wir, dass die Kinect sehr genaue Tiefeninformationen für kleine Objekte liefert. Allerdings ist ein Tiefenrauschen erkennbar. Besonders an Wänden oder Decken gibt es Störungen. Dies könnte durch die schwierigen Lichtbedingungen verursacht worden sein.

Bei DepthAnythingv1 ist die Tiefenabstufung konsistent, aber es fehlen feine Details. Die Personen, Kabel, Flaschen, sind eher glatt.

Weitere Beispiele:

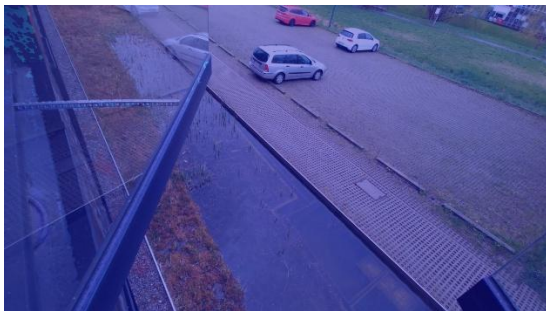
Azure Kinect:



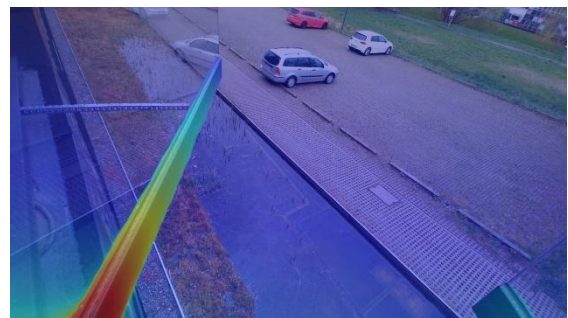
DepthAnythingv1:



Azure Kinect:



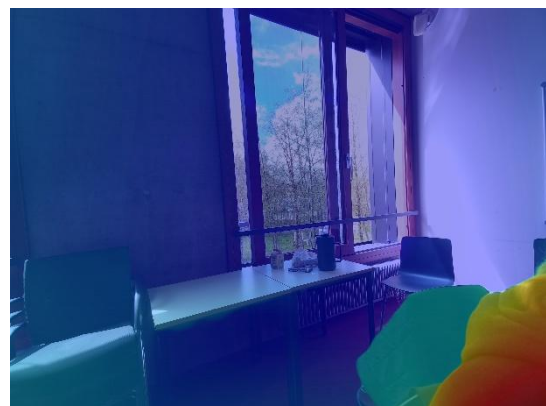
DepthAnythingv1:



Azure Kinect:



DepthAnythingv1:



Azure Kinect:



DepthAnythingv1:



Azure Kinect:



DepthAnythingv1:



Tiefe vergleichen – Pixel für Pixel Fehler messen

Um die genaue Abweichung der KI-Tiefe von der echten Tiefe zu berechnen, werden Tiefenbilder Pixel für Pixel verglichen. Der berechnete Wert ist der durchschnittliche Pixel-Fehler zwischen dem echten Tiefenbild der Azure Kinect und dem KI-generierten Tiefenbild. Die Tiefenkarten wurden vor dem Vergleich in dieselbe Skala (0–255) normalisiert, um vergleichbare Werte zu erhalten. Die Fehlerwerte (z.B. 95.94) beziehen sich auf normierte 8-bit Tiefenkarten (Skalierung 0–255).

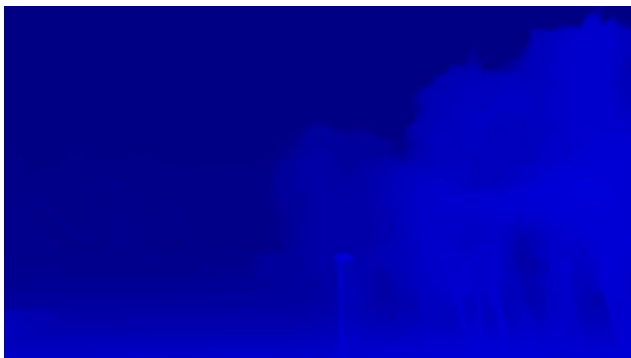
Azure Kinect Tiefenbild:



DepthAnythingv1 Tiefenbild:



Fehlerbild:



Durchschnittlicher Pixelfehler lag bei 95.94, was eine deutliche Abweichung von der realen Tiefenverteilung zeigt.

Azure Kinect Tiefenbild:



DepthAnythingv1 tiefenbild



Fehlerbild:



Durchschnittlicher Pixelfehler lag bei 80.94, was auf eine spürbare Abweichung der KI-Tiefeinschätzung im Vergleich zur realen Tiefe der Azure Kinect hinweist.

Weitere Beispiele:

Azure Kinect Tiefenbild:



DepthAnythingv1 tiefenbild



Fehlerbild:



Durchschnittlicher Pixel Fehler lag bei 59.29.

Azure Kinect Tiefenbild:



DepthAnythingv1 tiefenbild



Fehlerbild:



Durchschnittlicher Pixel-Fehler lag bei 115.18.

Rechenaufwand & Geschwindigkeit:

Azure Kinect arbeitet in Echtzeit. Es ist somit kein zusätzlicher Rechenaufwand nötig.

DepthAnythingv1 benötigt wiederum eine GPU-Beschleunigung für akzeptable Geschwindigkeiten. Auf dem Dell XPS 13 unter dem kleinsten Modell (vits) beträgt die Laufzeit ca. 10 Sekunden pro Bild. DepthAnythingv1 ist somit ohne High-End-Hardware nicht für Echtzeitanwendungen geeignet.

Flexibilität & Kosten:

Die Hardwarekosten der Azure Kinect betragen sich auf ca. 400–500 €. Da die Kinect im Innenraum detaillierter ist, wie DepthAnythingv1 eignet sie sich besser für industrielle und wissenschaftliche Anwendungen. Für Außenanwendungen ist sie allerdings nicht geeignet.

Die Nutzung von DepthAnythingv1 ist kostenlos, und es ist keine zusätzliche Hardware nötig. Es eignet sich somit gut für günstige, schnelle Prototypen oder Visualisierungen.