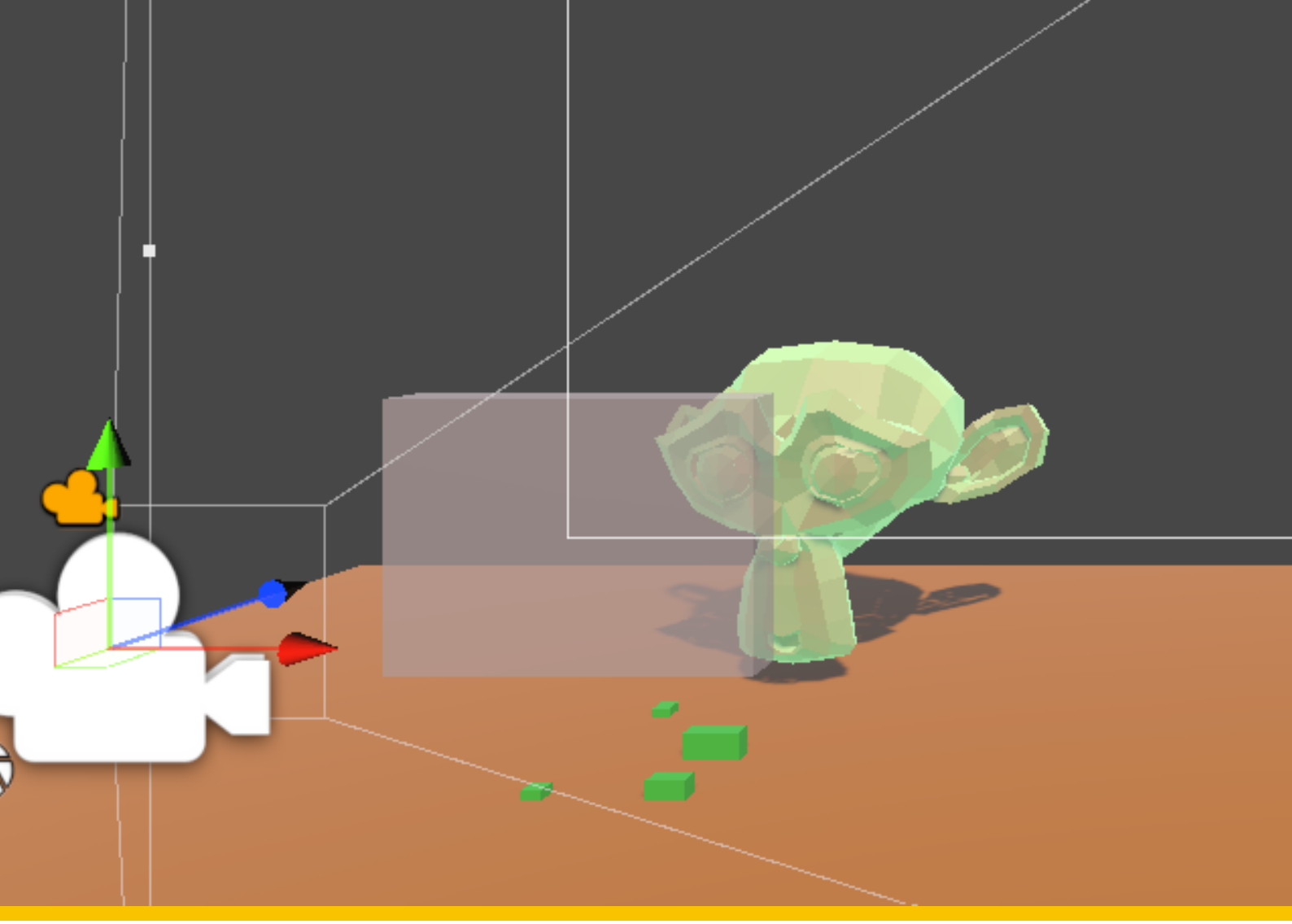


Projekt 2

AR / POSE Estimation

Course of study	Bachelor of Science in Computer Science
Author	Bernhard Messerli
Advisor	Marcus Hudritsch
Expert	Some expert

Version 1.0 of December 16, 2023

- Technik und Informatik
- Informatik

Abstract

In meiner Projektarbeit möchte ich einen tieferen Einblick erhalten wie Augmented Reality an einem eigentlich einfachen Anwendungsfall umgesetzt werden kann.
Der Betrachter welcher vor einem Bildschirm steht, wird von einer Intel Realsense Kamera D435 erfasst und dessen Position relativ zur Kamera wird ermittelt. In einer virutellen Szene, welche mittels der Gameengine Unity dargestellt wird, wird die Position von der getrackten Person auf eine virutelle Kamera gemappt. Der Blick der virtuellen Kamera entspricht dem Blick des Betrachters. In dieser Szene wird nun ein virtuelles Fenster dargestellt. Dieses Fenster wird perspektivisch in 2D dargestellt und erscheint als verzerrtes Rechteck - je nach Betrachtungswinkel. Nun soll dieses Polygon entzerrt und dann auf dem Bildschirm vor dem Betrachter dargestellt werden.
Dem Betrachter erscheint so auf dem Bildschirm ein virtuelles Fenster, deren sichtbare Objekte sich mit der Position des Betrachters perspektivisch verschieben. Der Betrachter soll sich visuell vor einem Fenster in eine virtuelle Welt empfinden.

Contents

Abstract iii

1. First Thesis Chapter 1
   1.1. Introduction . . . 1
   1.2. Komponenten . . . 1
      1.2.1. Physische Komponenten . . . 1
      1.2.2. Software Komponenten . . . 2
      1.2.3. OpenCV . . . 4
      1.2.4. Homographie . . . 4
      1.2.5. Kameramodell . . . 5
   1.3. Results . . . 5

2. Second Thesis Chapter 7
   2.1. Steps . . . 7
      2.1.1. Einbinden von Nuitrack . . . 7
      2.1.2. Virtuelle Szene in Unity . . . 7
      2.1.3. Matrix Shader . . . 8
      2.1.4. Pixel Shader . . . 8
      2.1.5. Entzerrung vom Polygon . . . 8
      2.1.6. Berechnung der Polygon Eckpunkte . . . 9
      2.1.7. Tabular . . . 10
      2.1.8. Math . . . 12
      2.1.9. Include pictures . . . 13
      2.1.10. Code Example . . . 13
      2.1.11. Draw boxes . . . 14
      2.1.12. Some Item-list . . . 16
      2.1.13. Multi column environment . . . 17
      2.1.14. Use Figures . . . 19
   2.2. Example Text With Indices . . . 20
   2.3. Example Text With Glossary . . . 20
   2.4. Example Text With Citations . . . 20
   2.5. Discussion . . . 21
   2.6. Fazit . . . 21
   2.7. Ausblick . . . 21

Bibliography 25

List of Figures 27

List of Tables 29

Listings 31

Glossary 33

A. First Appendix Chapter 35

1. First Thesis Chapter

1.1. Introduction

Bilder faszinieren mich schon sehr lange. Anfangs waren es Gemälde, Zeichnungen, Skizzen, analoge Fotografien, später dann digitale Bilder und bewegte Bilder. Ich bin nun in der Vertiefung Computer Perception and Virtual Reality. Im Gespräch mit dem Dozenten Marcus Hudritsch kamen wir auf die Thematik von Raum- und Farbempfinden im Zusammenhang mit Augmented Reality. Es ging darum ein Projektthema zu finden in dieser Vertiefung der Informatikausbildung. Marcus Hudritsch hat mir dann von dieser Umsetzung erzählt: Ein Bildschirm welcher ein künstliches Fenster in eine virtuelle Welt suggeriert. Ich stelle mir vor der Betrachter steht etwas verwirrt vor diesem Screen und begreift erst allmählich. Zuerst begreift er, dass sich diese Szene dargestellt auf dem Screen bewegt, dann nach einigem hin und her gehen, dass sich die Szene mit ihm im Gleichschritt bewegt und dann nach genauem Beobachten, wie sich die virtuellen Objekte verändern, dass die perspektivische Darstellung stimmt und es ist als schaue er durch ein Fenster in eine künstliche Welt, von Objekten.
Details werde ich in dieser Arbeit später aufzeigen. Wichtig ist mir hier deutlich zu machen, dass es in dieser Umsetzung auch darum geht, den Betrachter von der Realität in die Virtualität hinein zu begeleiten und etwas ins Grübeln zu bringen.
Dies wird nur gelingen wenn die Illusion von einem Fenster unserem gewohnten Empfinden gerecht wird und es sich als realtime Simulation anfühlt.

1.2. Komponenten

1.2.1. Physische Komponenten

Intel RealSense D435

Die Kamera D435 von Intel hat ein breites Sichtfeld, einen globalen Shutter und einen Tiefensensor, der sich für Anwendungen mit schnellen Bewegungen eignet. Gerade durch ihr breites Sichtfeld eignet sich die Kamera fÃŒr Anwendungen in der Robotik, Virtual und Augmented Reality, bei Szenen wo es darum geht möglichst viel zu sehen. Sie hat eine Reichweite bis zu 10 Meter. Das Intel RealSenseSDK bietet plattformunabhängig Unterstüzung bei der Umsetzung.

Figure 1.1.: Intel RealSense D435, Frontside

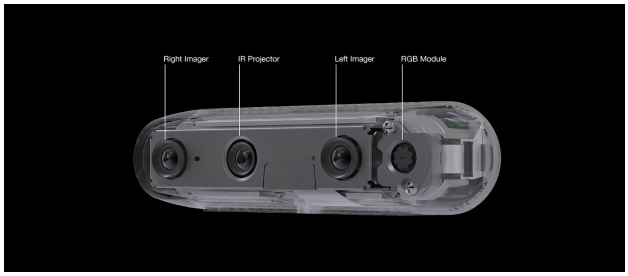


Figure 1.2.: Intel RealSense D435, Backside

Die Global-Shutter-Sensoren haben eine hohe Lichtempfindlichkeit bei schlechten Lichtverhältnissen und ermöglichen die Steuerung von Robotern auch in dunklen Räumen. [1]

1.2.2. Software Komponenten

Intel RealSense Viewer

Der Intel RealSense Viewer hat zwei Frames die er anzeigen kann, den RGB Frame und den Depth Frame. Die Abbildung zeigt einen Screenshot vom Depth Frame.

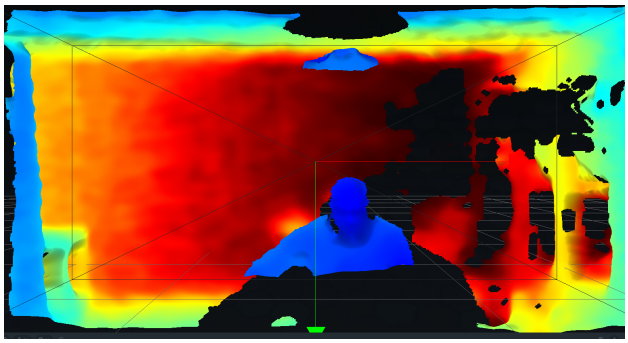


Figure 1.3.: Intel RealSense Viewer, Depth Frame

Die blauen Farbpixel zeigen einen nahen Bildpunkt, die roten Farbpixel einen

entfernten Bildpunkt. Für mein Projekt eignent sich dieser Viewer einzig dazu die verschiedenen Kameraeinstellungen auszuprobieren.

Intel RealSenseSDK 2.0

Intel hatte vor einigen Jahren eine führende Rolle in Anwendungen im Bereich Augmented Reality. Mit dem Intel RealSenseSDK 2016, dem Vorgänger vom RealSenseSDK 2.0 gab es ein ToolKit welches Facetracking implementiert hatte. Die rechte und die linke Hand konnte beispielsweise seperat fürs Tracking angesteuert werden und sogar einzelne Finger einer Hand konnten getrackt werden. Ich habe noch versucht dieses "ältere SDK zu installieren und dann mit einer älteren Unity Version zu verwenden. Leider ohne Erfolg. Intel hatte eine ganze Videoreihe dazu erstellt, welche aber heute vom Netz genommen wurden.
Intel schließt seine Abteilung rund um die RealSense-Kameras, die viele Jahre interessante, aber vom Markt kaum umgesetzte Lösungen hervorgebracht hatte. Sie passt nicht zum neuen Geschäftsmodell, welches rund um die Kernthemen von Intel aufgebaut und von dem neuen Foundry-Geschäft unterstützt wird. [2]-
Das neue SDK 2.0 taugt dagegen nur noch wenig und eignet sich nicht für mein Projekt. Das Toolkit implementiert kein Facedetection für Unity Anwendungen.

Unity

Unity ist eine Game Engine die von einer Open Source Community betreut und weiterentwickelt wird. Unity bietet für dieses Projekt ein anwendungsfreundliches Toll mit welchem die Transformationen der Objekte gemacht werden können. Dies wäre auch mit OpenGL möglich. OpenGL ist die eigentliche Mutter aller Grafik Softwares.
In Unity ist es dann auch möglich komplexe Objekte mit unterschiedlichen Oberflächen und Texturen mit der gewünschten Lichtsituation darzustellen. [3]

Nuitrack

Nuitrack stellt in ihrem SDK viele Anwendungen zur Verfügung, diese sind aber auf 3 Minuten Spieldauer limitiert. Wahrscheinlich haben sie von Intel gelernt und haben ein Pricing welches für volle Anwendungen $99.99 pro Jahr kostet. In der nächsten Abbildung sind alle implementierten Features dargestellt.

Dieses SDK ist enorm mächtig und hat alles zu bieten, was ich für mein Projekt brauche. Wichtig für mein Projekt ist das Facetracking und das Full Body Skeletal Tracking. Die Nutirack SDK hat eine gute Anbindung zu Unity und ist gut domumentiert. [4]

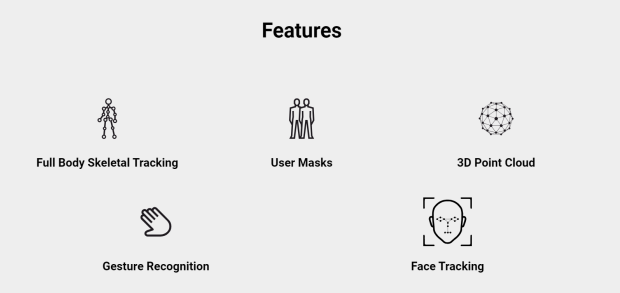


Figure 1.4.: Nutitrack SDK, Features

1.2.3. OpenCV

OpenCV ist eine Open Source Computer Vision Grafik Library. In meinem Projekt benötige ich diese Library um ein bliebiges Polygon mit Seiten zu entzerren und als Rechteck darzustellen. Es gibt einen Wrapper für diese Library, welcher sich in Unity einbinden lässt. Dieser ist kostenpflichtig. Herr Hudritsch konnte mir eine Free Version geben, dies im Rahmen der Ausbildung. Informationen zu OpenCV finden sich unter: https://opencv.org/ und für das Unity Plugin unter: https://enoxsoftware.com/opencvforunity/.

1.2.4. Homographie

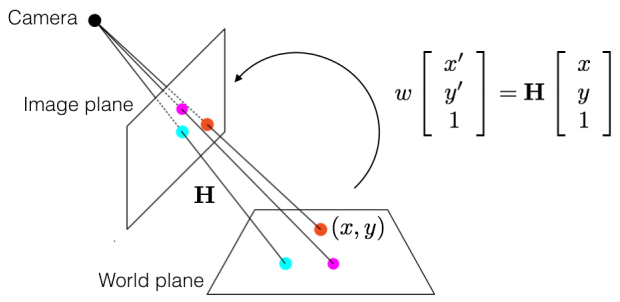


Figure 1.5.: Homographie, Visualisation

Die Homographie Matrix H beschreibt wie die Originalpunkte x,y zu liegen kommen in der Bildebene. Die Homographie Matrix benötigt jeweils 4 Punkte in der Quellebene und die entsprechenden Punkte in der Zielebene.

Diese 8 Punkte werden als Vektoren in einer Matrix A dargestellt. Die Matrixmultiplikation von A mit der Homographie Matrix die wir suchen, wird in einem

$$\begin{bmatrix} \hat{x}_i z_a \\ \hat{y}_i z_a \\ z_a \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

Figure 1.6.: Homographie, Abbildungsgleichung

Gleichungssystem von 8 Gleichungen und 8 Unbekannten gelöst.

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1\hat{x_1} & -y_1\hat{x_1} \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2\hat{x_2} & -y_2\hat{x_2} \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3\hat{x_3} & -y_3\hat{x_3} \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4\hat{x_4} & -y_4\hat{x_4} \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1\hat{y_1} & -y_1\hat{y_1} \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2\hat{y_2} & -y_2\hat{y_2} \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3\hat{y_3} & -y_3\hat{y_3} \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4\hat{y_4} & -y_4\hat{y_4} \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = h_{33} \begin{bmatrix} \hat{x_1} \\ \hat{x_2} \\ \hat{x_3} \\ \hat{x_4} \\ \hat{y_1} \\ \hat{y_2} \\ \hat{y_3} \\ \hat{y_4} \end{bmatrix}$$

Figure 1.7.: Homographie, Gleichungssystem

Der Streckungsfaktor h33 zeigt, dass die Homographie Matrix noch skaliert werden kann. Im Default Fall wird h33 eins gesetzt. [5] [6]

1.2.5. Kameramodell

Um auf die Eckpunkte vom Screen zu kommen, sollte es reichen, wenn wir die Position der Kamera, deren Rotationswinkel, sowie den Winkel vom Field of View und den Viewport der Kamera kennen. Weiter kennen wir die World Positions von den Eckpunkten vom Screen. Da der Screen statisch ist und sich nicht bewegt, können wir dieses Model ausser acht lassen. Was bleibt ist: der Viewport (1), die Projektion (2) (Kamera intrinsisch) und das Kamera-Modell (3) (Kamera extrinsisch oder View). Unsere Abbildungsmatrix ist also: Matrix m = (1) * (2) * (3) Und mittels Anwendung auf die Eckpunkte vom Screen: P1' = P1 * m

1.3. Results

What did you find? – a section which describes the data that was collected and the results of any statistical tests that were performed. It may also be prefaced by a description of the analysis procedure that was used. If there were multiple experiments, then each experiment may require a separate Results section.

2. Second Thesis Chapter

2.1. Steps

2.1.1. Einbinden von Nuitrack

Nuitrack kann unter folgendem Link für alle Platformen runtergeladen werden: .

Import Nuitrack Wrapper in Unity

Unter folgendem git repo kann das Nuitrack Plugin runtergeladen werden: . Da dieses Plugin für verschiede Sensoren, also Kameras konfiguriert wurde, muss dann noch der Sensor spezifiziert werden. In meinem Projekt ist dies die Intel-Realsense D435.

Skeleton Tracking mit NuitrackSDK

Das Tutorial von Nuitrack ist meine Ausgangslage um eine Person zu tracken. Der Link führt zu diesem Tutorial: .

2.1.2. Virtuelle Szene in Unity

In Unity baue ich die Szene so auf, dass die Render-Kamera in positiver z-Achse ausgerichtet ist. Anfangsposition der Render-Kamera ist (0, 0.2, -1.5). Den Screen welcher dann entzerrt werden soll, setze ich auf den Nullpunkt. Also genauer die Mitte vom Screen hat die Position (0, 0, 0). Die Eckpunkte vom Screen haben die folgenden Positionen: P0(-0.8, -0.45) P1(0.8, -0.45) P2(0.8, 0.45) P3(-0.8, 0.45) Der Screen hat die Grösse 1600 x 900 mm. Dieser wird als schmaler Kubus dargestellt und transparent leicht grau eingefärbt, damit ich dann erkannen kann ob mein Rendern und Entzerren richtig funktioniert. Die Main-Kamera setze ich als Referenz auf die Position der Render-Kamera. Dies damit ich eine Ansicht in der Szene bekomme. Die Main-Kamera zeigt dann die Szene und für den Output vom entzerrten Screen benötige ich das gerenderte Bildmaterial der Renderkamera. Gut möglich, dass man dies auch mit weniger Kameras darstellen kann.

Render Kamera zeigt Position vom Betrachter

Die Render-Kamera wird als Referenz auf den obersten Knochen vom Skelett gemappt. Dieser oberste Knochen entspricht dem Kopf, genauer der Stirn. Damit wird dann auch die Main-Kamera, welche die Render-Kamera referenziert auf den Kopf gemappt.

Render Kamera zeigt RenderTexture

Die Render-Kamera wird nun stets auf den Mittelpunkt vom Screen gerichtet. Der Output der Render-Kamera erzeugt dann eine RenderTexture der Grösse 750 x 750 Pixel. Je grösser die Anzahl Pixel, desto besser wird dann die Qualität vom Endbild.

2.1.3. Matrix Shader

Im Netz habe ich einen Rotations Shader gefunden und versucht diesen so anzupassen, dass die Entzerrung gelingt. Leider ohne Erfolg. Das Probelm bei diesem Shader funktioniert das Scaling und das Shearing, nicht aber die Translation. Zudem wüsste ich nicht, wie ich mit dieser Rotationsmatrix das Polygon selektieren sollte. Als Einstieg in die Unity Shader hat es trotzdem etwas geholfen, mehr aber auch nicht.

2.1.4. Pixel Shader

Damit ich diese Entzerrung vom Polygon machen kann, benötige ich einen Pixel-Shader. Oder einen Shader welcher die Maintexture ändert. So habe ich nun einen Shader welche als Parameter die Maintextur als Eingabewert hat. Im Script wird dann in der Update Funktion für jedes Pixel der Farbwert gesetzt und am Ende die Textur dem Shader übergeben.

2.1.5. Entzerrung vom Polygon

Homographie

Zuerst habe ich versucht die Entzerrung mit einer Homographie Matrix zu erreichen. Leider hat dies nur geklappt für die Ansicht ohne seitliche Verschiebung, also mit einem x-Wert von Null. Sobald ich etwas abweiche von dieser zentralen Position, zeigt sich ein Shearing im Output Bild, welches sich auch noch anders verhält, je nachdem ich in positiver x-Achse gehe oder in negativer x-Achse. Ich habe da länger darüber nachgedacht, meine Homographie Matrix mehrmals geprüft, fand aber nicht heraus was das Problem war. Komisch ja auch, dass es in zentraler Position den Richtigen Output gab, bei seitlicher Verschiebung sich dann

aber ein Shearing zeigt, welches mit grösserer Abweichung nach links oder rechts dann auch noch grösser wurde. Somit brauche ich einen anderen Ansatz um die Entzerrung zu lösen.

OpenCV

OpenCV hat bereits Methoden, welche diese Entzerrung vornehmen. Das Plugin von OpenCV findet sich unter: . Die grösste Schwierigkeit ist bei OpenCV, dass Unity einen anderen Aufbau der Matrizen hat und dass OpenCV keine Textures kennt. Die grösste Schwierigkeit war dann zu merken, dass die Pixelwerte der Textures in OpenCV nicht einfach so auf der CPU verfügbar sind. Diese werden auf der GPU gespeichert und mössen dann exklusiv abgefragt werden. Die Funktionen von OpenCV funktionieren tadellos. Damit die Entzerrung gelingt, gehe ich vom Zielbild aus und berechne die Perspektivische Matrix rückwärts vom Zielbild zum Inputbild, was dann dem Inversen der Perspektivischen Matrix entspricht. Nun wird diese Matrix auf jedes Pixel des Zielbildes angewendet und so den Farbwert an der entsprechenden Stelle im Inputbild geholt. Da diese Operation mit der Anzahl Pixel steigt, habe ich hier ein Format vom Zielbild 800 x 450 Pixel gewählt.

2.1.6. Berechnung der Polygon Eckpunkte

Kameramodell

Ich habe versucht mittels Kameramodell die Transformationsmatrix für die Eckpunkte vom Screen zu berechnen. Dazu braucht man den Viewport, die intrinsiche (Projektionsmatrix) und die extrinsische Kameramatrix (Viewmatrix). Da sich der Screen nicht verschiebt, braucht es die Modellmatrix nicht.

Projektion der Eckpunkte auf der Zielebene

Mein Ansatz den ich umgesetzt habe funktioniert wie folgt: Unity hat eine Funktion welche mir die Eckpunkte von der Clipping Ebene entsprechend dem z Wert herausscheibt. Ich wähle diese Ebene durch den Nullpunkt, auf welchen sich die Render-Kamera immer richtet. Nun projeziere ich die Eckpunkte vom Screen in der Verlängerung von der Render-Kamera durch die Eckpunkte. Die Schnittpunkte mit der Ebene müssen nun nur noch relativ zu den Ecken der Clipping Ebene berechnet werden.

2.1.7. Tabular

Measure	Data	Unit
1	2	3
4	5	6

Stadtteil	Anzahl Personen	Ausländische Bevölkerung
Innere Stadt	3748	17.9 %
Länggasse-Felsenau	17 976	17.1 %
Mattenhof-Weissenbühl	26 895	22.4 %
Kirchenfeld-Schlosshalde	23 384	13.4 %
Breitenrain-Lorraine	24 082	19.4 %

Table 2.1.: Anzahl Personen, ausländischer Bevölkerungsanteil und Arbeitslosenquote pro Stadtteil Ende 2005 (Statistikdienste der Stadt Bern, 2006)

Stadtteil	Anzahl Personen	Ausländische Bevölkerung
Innere Stadt	3748	17.9 %
Länggasse-Felsenau	17 976	17.1 %
Mattenhof-Weissenbühl	26 895	22.4 %
Kirchenfeld-Schlosshalde	23 384	13.4 %
Breitenrain-Lorraine	24 082	19.4 %

Table 2.2.: Anzahl Personen, ausländischer Bevölkerungsanteil und Arbeitslosenquote pro Stadtteil Ende 2005 (Statistikdienste der Stadt Bern, 2006)

More about Tables Further information about tables : `https://www.latex-tutorial.com/tutorials/tables/`

Long Tables Further information about long tables : `https://www.overleaf.com/learn/latex/tables`

Table 2.4.: A sample long table

First column	Second column	Third column
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778

Continued on next page...

Table 2.4: ... continued from previous page

First column	**Second column**	**Third column**
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778

Continued on next page...

Table 2.4: ... continued from previous page

First column	**Second column**	**Third column**
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778
One	abcdef ghjijklmn	123.456778

2.1.8. Math

$$\begin{pmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \end{pmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{b_f}{J} & \frac{K_m}{J} \\ 0 & -\frac{K_g}{L} & \frac{R}{L} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{bmatrix} 0 & 0 \\ -\frac{1}{J} & 0 \\ 0 & \frac{1}{L} \end{bmatrix} \begin{pmatrix} t_L \\ v_a \end{pmatrix} \tag{2.1}$$

Stadtteil	Anzahl Personen	Ausländische Bevölkerung
Innere Stadt	3748	17.9 %
Länggasse-Felsenau	17 976	17.1 %
Mattenhof-Weissenbühl	26 895	22.4 %
Kirchenfeld-Schlosshalde	23 384	13.4 %
Breitenrain-Lorraine	24 082	19.4 %

Table 2.3.: Anzahl Personen, ausländischer Bevölkerungsanteil und Arbeitslosenquote pro Stadtteil Ende 2005 (Statistikdienste der Stadt Bern, 2006)

2.1.9. Include pictures

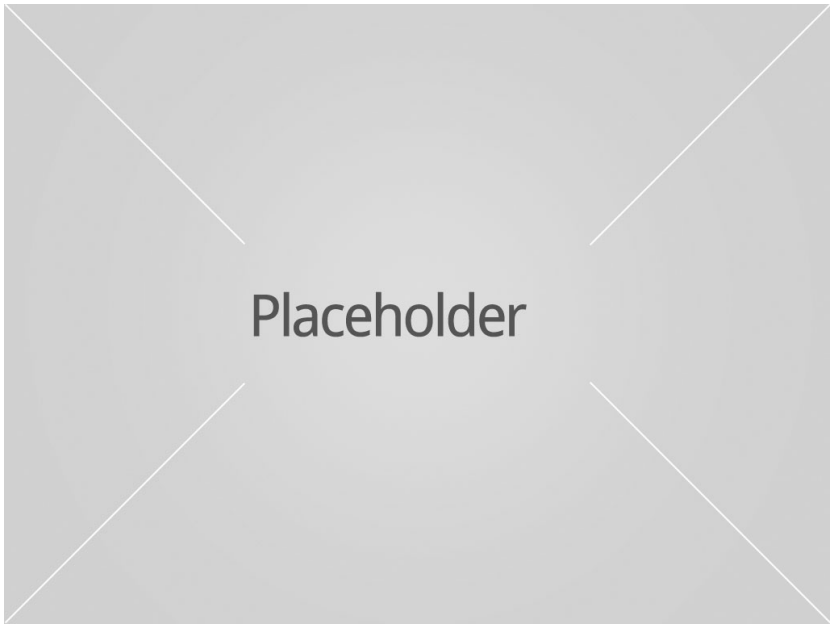


Figure 2.1.: Some meaningful caption

2.1.10. Code Example

```c
#include <stdio.h>
#include <stdlib.h>

int main( /* int argc, char **argv */ )
{
  printf("Hello World!\n");
  return EXIT_SUCCESS;
}
```

Listing 2.1: My very first C program.

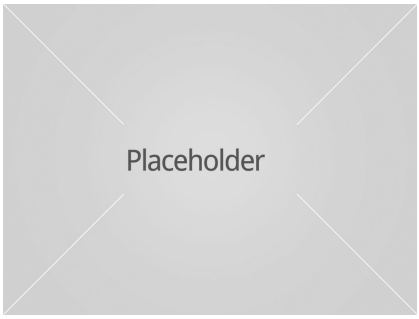


Figure 2.2.: PLACEHOLDER

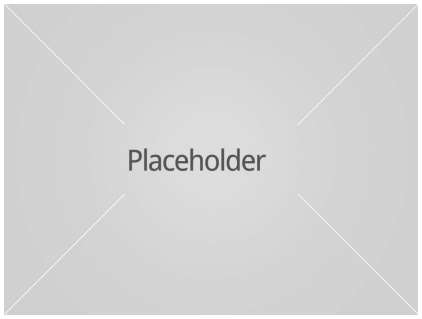


Figure 2.3.: PLACEHOLDER

I developed this very nice application writing "Hello World" to my terminal. The implementation is shown in listing 2.1.

2.1.11. Draw boxes

Purple

Note

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Blue

Note

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Red

Note

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

An alert box.

A warning box.

A note box.

A recycle box.

No color set

Some BFH box without color option set. Using default.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a

meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

2.1.12. Some Item-list

Sometimes you explain this and that using a bullet points. This can be done in LaTeX using an item list in a item environment.

- My first item
- The second
- ...
- ...

It is also possible to nest such environment and/or enumerate.

- My first item
  1. My first enumerated item
  2. The second
  3. ...
- The second
  1. An other enumerated item
  2. ...
  3. ...

2.1.13. Multi column environment

Split a part of a document in multiple columns is not so easy with WYSIWYG tools. Whit multicols LaTeX package· · · well you may know.

- My first item
- The second
- ...
- ...
- My first item
- The second
- ...
- ...
- My first item
- The second
- ...
- ...

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are

written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

- My first item
- The second
- ...
- ...

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

- My first item
- The second
- ...
- ...

2.1.14. Use Figures

Figure 2.4.: An example of including a PDF figure.

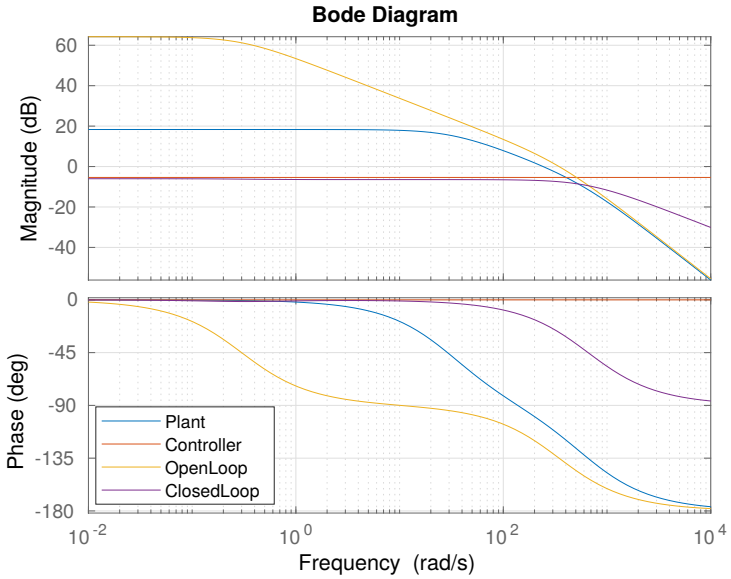
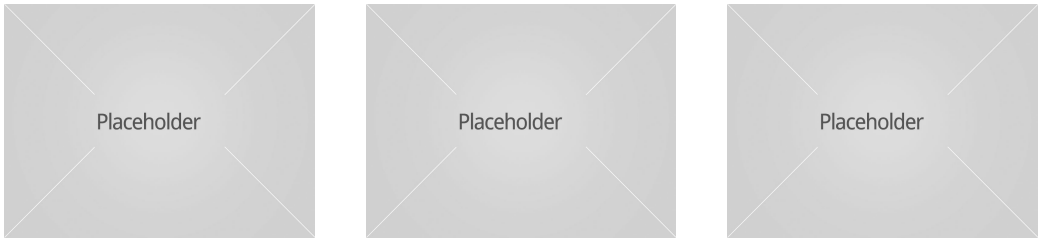


Figure 2.5.: An example of including a PDF figure.

Use Subfigures

These subfigures requires the package `subcaption`.



(a) $y = x$ (b) $y = 3sinx$ (c) $y = 5/x$

Figure 2.6.: Three simple graphs

2.2. Example Text With Indices

In this example, several keywords will be used which are important and deserve to appear in the Index.

Terms like generate and some will also show up.

2.3. Example Text With Glossary

This Zynq introduction summary has been written for bachelor students due to the introduction workshop in the "Embedded Systems" course at Bern University of Applied Sciences. The topic SoC is introduced by using Xilinx' AP SoC platform Zynq. The subsequent summery is a brief introduction only. It is based on several tutorials in the field of SoC such as the Zynq Book or Xilinx' AP SoC manual. We think the script provides a good introduction and helps getting the overall picture of the SoC basics. In addition we reference to our wiki tutorials that provide lots of information on how to get started with the ZedBoard.

Hey folks let's do an ASIC design and develop some awesome RTOS! Yea ARM is nice but we can do better, can we?

2.4. Example Text With Citations

This document is an example of BibTeX using in bibliography management. Three items are cited: The LaTeX Companion book [7], the Einstein journal paper [8], and the Donald Knuth's website [9]. The LaTeX related items are [7, 9].

2.5. Discussion

What is the significance of your results? – the final major section of text in the paper. The Discussion commonly features a summary of the results that were obtained in the study, describes how those results address the topic under investigation and/or the issues that the research was designed to address, and may expand upon the implications of those findings. Limitations and directions for future research are also commonly addressed.

2.6. Fazit

2.7. Ausblick

Declaration of Authorship

I hereby declare that I have written this thesis independently and have not used any sources or aids other than those acknowledged.

All statements taken from other writings, either literally or in essence, have been marked as such.

I hereby agree that the present work may be reviewed in electronic form using appropriate software.

December 16, 2023

A. Muster

C. Example

Bibliography

[1] Intel. Intel.

[2] Computer Base. Tiefenkameras & co: Intel gibt realsense zugunsten des kerngeschäfts auf.

[3] Unity. Unity.

[4] Nuitrack. Nuitrack sdk.

[5] Faisal Qureshi. Homography.

[6] Yalda Shankar. Homography estimation.

[7] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley, Reading, Massachusetts, 1993.

[8] Albert Einstein. Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]. *Annalen der Physik*, 322(10):891–921, 1905.

[9] Donald Knuth. Knuth: Computers and typesetting.

List of Figures

1.1. Intel RealSense D435, Frontside 2
1.2. Intel RealSense D435, Backside 2
1.3. Intel RealSense Viewer, Depth Frame 2
1.4. Nutitrack SDK, Features 4
1.5. Homographie, Visualisation 4
1.6. Homographie, Abbildungsgleichung 5
1.7. Homographie, Gleichungssystem 5

2.1. Some meaningful caption 13
2.2. PLACEHOLDER 14
2.3. PLACEHOLDER 14
2.4. An example of including a PDF figure. 19
2.5. An example of including a PDF figure. 19
2.6. Three simple graphs 20

List of Tables

2.1. Anzahl Personen, ausländischer Bevölkerungsanteil und Arbeitslosenquote pro Stadtteil Ende 2005 (Statistikdienste der Stadt Bern, 2006) 10
2.2. Anzahl Personen, ausländischer Bevölkerungsanteil und Arbeitslosenquote pro Stadtteil Ende 2005 (Statistikdienste der Stadt Bern, 2006) 10
2.4. A sample long table 10
2.3. Anzahl Personen, ausländischer Bevölkerungsanteil und Arbeitslosenquote pro Stadtteil Ende 2005 (Statistikdienste der Stadt Bern, 2006) 13

Listings

2.1. My very first C program. 13

Glossary

AP SoC **All Programmable System-on-Chip (AP SoC)** was introduced by Xilinx. It represents a IC which comprise a hard-core processor core surrounded by an FPGA fabric. This type of ICs are highly configurable and provide algorithm partitioning capabilities. This provides high benefit for highly scale-able applications as well as fast time-to-market

ARM **ARM** A family of processor architectures. The hard processor type which forms the basis of the Zynq processing system is an ARM Cortex-A9 version. The term 'ARM' may also be used to refer to the developer of the processor, i.e. a company of the same name

ASIC **Application-Specific Integrated Circuit (ASIC)** An integrated circuit which is designed for a specific use, rather than general-purpose use

BibTeX Program for the creation of bibliographical references and directories in TEXor LATEX documents

RTOS **Real-Time Operating System (RTOS)** A category of operating systems defined by their ability to respond quickly and predictably for a given task

SoC **System-on-Chip (SoC)** A single chip that holds all of the necessary hardware and electronic circuitry for a complete system. SoC includes on-chip memory (RAM and ROM), the microprocessor, peripheral interfaces, I/O logic control, data converters, and other components that comprise a complete computer system

ZedBoard **ZedBoard** A low cost development board featuring a Zynq-700 0 SoC, and a number of peripherals

Zynq **Zynq** Xilinx' AP SoC. The characteristic feature of Zynq is that it combines a dual-core ARM Cortex-A9 processor with traditional Series-7 FPGA logic fabric

Zynq Book **Zynq Book** A book that summarizes all the important aspects when working with Zynq and provides a strong and easy understandable introduction to the topic. The book has been written by a team of University of Strathclyde Glasgow in cooperation with Xilinx

A. First Appendix Chapter