

9 Appendix B

9.1 Das BND- Bildcontainerformat

Wie auf Seite 20 beschrieben, liefert die Kamera des Naos ein komprimiertes YUV Bild. Dieses kann nicht direkt gespeichert und mit jedem Bildbetrachter geöffnet werden. Aus diesem Grund wurde das BND- Bildcontainerformat entwickelt, da neben den eigentlichen Bilddaten noch zusätzliche Informationen und das klassifizierte Bild abgespeichert werden müssen.

Zum Analysieren des Verhaltens des Roboters ist es notwendig, die aufgenommenen Bilder und den Zustand zu diesem Zeitpunkt zu betrachten. Daraus können Schlüsse gezogen werden, was der Roboter gesehen, welche Ergebnisse der in dieser Diplomarbeit beschriebene Klassifizierer geliefert und die anschließende Bildverarbeitung geleistet hat. Fehler in der Abfolge können so rekonstruiert und beseitigt werden.

Abbildung 9.1 zeigt anschaulich das Containerformat, Tabelle 9.2 den Aufbau der Datei an sich und Tabelle 9.2 die darin zusätzlich gespeicherten Informationen.

9.2 Aufbau

Das BND- Bildcontainerformat kann bis zu zwei Bilder und zusätzliche Informationen enthalten. Neben den Daten des YUV422 Bildes kann das Ergebnis des Klassifizierers abgespeichert werden. Es können jedoch zum Beispiel nur Originaldaten auf dem Roboter abgespeichert und anschließend offline mit einer speziell angepassten LUT (vgl. Seite 39) erzeugt und eingefügt werden. Ein Originalbild muss ebenfalls nicht abgespeichert werden.

Der Block mit zusätzlichen Informationen ist von variabler Größe und befindet sich am Ende der Datei. Diese Informationen beziehen sich jeweils auf den Zeitpunkt, zu dem das Originalbild aufgenommen wurde. Die verschiedenen Informationen sind mit $\backslash xFF$ getrennt, werden eindeutig mit einer ID versehen (siehe Tabelle 9.2). Sie haben die Form $ID = \text{wert}_1; \text{wert}_2; \dots; \text{wert}_n$. Zusätzliche Felder werden mit einem $:$ getrennt (wie zum Beispiel die verschiedenen Kameraparameter).

9.3 Benutzung

stay tuned, gibts bald für C++, Python und Matlab! yeah! und extrahieren von ganzen Verzeichnissen!

BLD- Bildcontainerformat



Abbildung 9.1: Das BLD- Bildcontainerformat. Es beinhaltet neben dem Originalbild das klassifizierte Bild und zusätzliche Informationen.

Byte	Typ	Beschreibung
0	Bildgröße	2 = VGA (640x480), 1 = qVGA (320x240), 0 = qqVGA (160x120)
1	Originalbild	0 = kein Originalbild, 1 = Originalbild eingebettet
2	klassifiziertes Bild	0 = kein klassifiziertes Bild, 1 = klassifiziertes Bild eingebettet
3 ⋮ $n + 3$	Bilddaten	Daten des Originalbildes, wobei $Byte[3] = 0$, wenn kein Originalbild eingebettet ist. Ansonsten $n = Byte[1] \cdot 4^{Byte[0]} \cdot 160 \cdot 120 \cdot 2$
$n + 4$ ⋮ $m + 4$	Bilddaten	Daten des klassifizierten Bildes, wobei $Byte[4] = 0$, wenn kein klassifiziertes Bild eingebettet ist. Ansonsten $m = Byte[2] \cdot 4^{Byte[0]} \cdot 160 \cdot 120$
$n + m + 5$	Bildinformationen	Zusätzliche Informationen, die während der Aufnahme des Bildes gespeichert wurden. Eine detaillierte Auflistung der Informationen findet sich in Tabelle 9.2.

Tabelle 9.1: detaillierte Bytereihenfolge des BLD- Bildcontainers.

ID	Name	Anz.	Beschreibung
0	Timestamp	1	Zeitpunkt der Aufnahme.
1	Name	1	Name des Roboters, bzw. Simulator.
2	Quelle	1	wurde die Daten während der Laufzeit aufgenommen oder Offline.
3	Kameraparameter	1	Kameraeinstellungen während der Aufnahme.
4	Ground Truth	1	Ground Truth Position auf dem Spielfeld des Roboters (x, y, Ausrichtung).
5	Position	1	errechnete Position des Roboters auf dem Spielfeld (x, y, Ausrichtung, Wahrscheinlichkeit).
6	Gyrometer	1	Gyrometerwerte (x, y), vgl. Seite 18.
7	Accelerometer	1	Accelerometerwerte (x, y, z), vgl. Seite 18.
8	Winkel	1	Eulerrepräsentation der Winkel der oberen Kamera des Roboters.
9	Ergebnisse	n	Ergebnisse der Bildverarbeitung. Aus diesen Informationen lässt sich ablesen, welche Objekte der Roboter vermeintlich gesehen hat.
10	Version	1	benutze Softwareversion.

Tabelle 9.2: zusätzliche Informationen im BLD- Format. Diese werden während der Aufnahme des Bildes aufgezeichnet und teilweise berechnet.

