

Paralleles Lesen und Schreiben von farbigen 2D-Barcodes mit MPI

**Seminar Wissenschaftliches Rechnen
ZHAW, Zürich**

Florian Lüthi*

14. Juni 2013

*`luethifl@students.zhaw.ch`

Inhaltsverzeichnis

1	Einleitung	3
2	Einführung in das Thema	4
2.1	Existierende Technologien	4
2.2	Vorgeschlagene Methode	5
3	Randbedingungen	8
4	Auswahl der Testdaten für die Arbeit	9
5	Algorithmische Beschreibung des Verfahrens	10
6	Analyse, Design und Implementierung des Verfahrens	11
7	Vergleich der Performancezeiten für ein, zwei oder mehrere CPU-Cores	12
8	Zusammenfassung	13
	Literaturverzeichnis	14

1 Einleitung

Diese Arbeit beschäftigt sich mit der von Md. Mashud Rana, M. E. Kawsar, M. E. Rabbani, S. M. M. Rashid und K. E. U. Ahmed vorgeschlagenen Methode des Lesens und Schreibens von farbigen zweidimensionalen Barcodes mit hoher Datenkapazität [1].

Sie beschreibt diese Methode, zeigt eine mögliche Umsetzung des Verfahrens in `C++` mit Parallelisierung mittels `MPI`, und macht eine quantitative Analyse der Implementation in Bezug auf die Performance. Zu guter Letzt zieht sie ein Fazit und überlegt sich den Sinn der angewandten Methode.

Zur Arbeit gehören dieses Dokument sowie die Beispielimplementation. Beides kann von <https://github.com/foyan/ColorizedBarCodec> bezogen werden.

Der Autor bedankt sich bei Dr. Alexander Herrigel für die Durchführung des Seminars.

2 Einführung in das Thema

2.1 Existierende Technologien

Seit den 1970er-Jahren [3] sind Barcodes aus Logistikbetrieben, Supermärkten etc. nicht mehr wegzudenken. Sie bilden das Rückgrat der automatisierten Warenbewirtschaftung. Das Scanning von eindimensionalen Barcodes wie in Abbildung 2.1 ist milliardenfach erprobt und in der Entwicklung eigentlich abgeschlossen.

Die Informationsdichte von eindimensionalen Barcodes ist aber sehr beschränkt. Beispielsweise erlaubt die eindimensionale Symbologie UPC-A maximal eine Billion eindeutiger Barcodes [6], was gemäss

$$\log_2 10^{12} = 39.863 \dots$$

einer maximalen Kapazität von weniger als 5 Bytes entspricht.

Aus diesem Grund wurden mit dem Aufkommen von billigen und sehr akkuraten Kameras in mobilen Devices zweidimensionale Symbologien entwickelt, welche es zumindest erlauben, ganze URLs von Webseiten oder ähnliches zu codieren. Google beispielsweise vermarktet den in Abbildung 2.2 ersichtlichen, ursprünglich von Toyota entwickelten *Quick Response Code* (QR) [5]. Dieser erlaubt es immerhin, maximal 2953 Bytes zu codieren [5].

Um die Kapazität weiter zu steigern, ist es nötig, eine dritte Dimension einzuführen. Weil aber Hologramme und 3D-Kameras noch nicht marktreif sind, liegt es nahe, die dritte Dimension als Farbe zu codieren. Microsoft geht in seiner Barcode-Technologie *High Capacity Color Barcode (HCCB)* seit 2007 [4] genau diesen Weg [2]. Ein Beispiel ist in Abbildung 2.3 abgebildet. Die maximale Kapazität von HCCB kann nicht genau genannt werden, weil die Barcodes keine definierte Grösse und Skalierung haben. Labortests scheinen aber gezeigt zu haben, dass es möglich ist, circa 2000 Bytes auf eine

Abbildung 2.1: Eindimensionaler Barcode (Quelle: Wikipedia)



Abbildung 2.2: QR Code (Quelle: Wikipedia)



Abbildung 2.3: High Capacity Color Barcode (Quelle: Wikipedia)



Fläche zu codieren, welche nicht viel grösser als eine Ein-Cent-Münze ist. Diese Fläche scheint sich mit einem 600-dpi-Laserdrucker herstellen zu lassen [2].

2.2 Vorgeschlagene Methode

Rana, Kawsar, Rabbani, Rashid und Achmed schlagen nun eine Methode vor, welche sich durch eine hohe Kapazität sowie ein sehr schnelles Leseverfahren auszeichnen soll [1]. Die grundlegenden Eckwerte sind die folgenden:

- Es werden 16 verschiedene Farben verwendet. Jede dieser Farben wird bijektiv einer Sequenz aus 4 Bits (Nibbles) zugeordnet (siehe Tabelle 2.1, [1]).
- Jedes Byte wird auf ein Zeichen abgebildet (siehe [1]).
- Die Fläche des Barcodes wird in ein 102×100 Zellen fassendes Raster aufgeteilt (im folgenden als Pixel bezeichnet). Bei einer angenommenen realen Grösse des Barcodes von 1.5×1.15 Zoll [1] ergibt sich daraus eine horizontale Auflösung von 68 DPI und eine vertikale Auflösung von 87 DPI.
- Jeweils 6 benachbarte Pixel in einem 3×2 -Feld codieren nun ein Byte, wobei ein ein vertikal gespiegeltes L für den ersten Nibble und ein horizontal gespiegeltes L für den zweiten Nibble entstehen (Abbildung 2.4). Über die Reihenfolge der

Tabelle 2.1: Zuordnung der Farben zu den Nibbles

Wert	Nibble (binär)	RGB-Wert
0	0000	rgb(255, 000, 000)
1	0001	rgb(000, 128, 000)
2	0010	rgb(000, 000, 025)
3	0011	rgb(165, 042, 042)
4	0100	rgb(128, 000, 000)
5	0101	rgb(100, 149, 237)
6	0110	rgb(255, 165, 000)
7	0111	rgb(255, 255, 000)
8	1000	rgb(238, 130, 238)
9	1001	rgb(128, 128, 128)
10	1010	rgb(000, 000, 128)
11	1011	rgb(000, 000, 000)
12	1100	rgb(173, 255, 047)
13	1101	rgb(138, 043, 226)
14	1110	rgb(128, 000, 128)
15	1111	rgb(095, 158, 160)

Wertigkeit der Nibbles wird keine Vorschrift gemacht – die abgebildete Rot/Grün-Kombination könnte also sowohl als 0000 0001 = 1 als auch als 0001 0000 = 16 interpretiert werden.

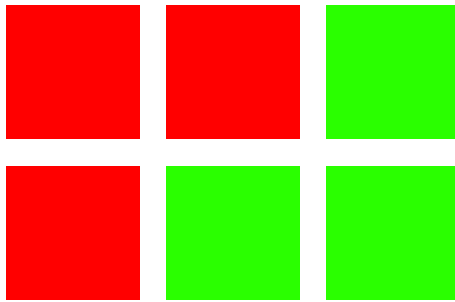
Weil je nach codierter Information die Länge der Codierung variiert, schlägt [1] vor, die ersten Nibbles als Header zu reservieren¹, welche als Information nur die Länge der nachfolgenden Informationen enthalten.

Die maximale Kapazität des Barcodes ist darum

$$C = \frac{102 \text{ px} \cdot 100 \text{ px}}{6 \text{ px}^2 / \text{byte}} - 2 \text{ bytes} = 1698 \text{ bytes}$$

¹Bezüglich der Länge des Headers werden in [1] widersprüchliche Aussagen gemacht. Der Text reserviert die ersten 12 Pixel, was einer Länge von 2 Bytes entsprechen würde, wogegen die dazugehörige Abbildung eine Länge von 9 Pixeln kolportiert, was einer Länge von 1.5 Bytes entsprechen würde. Im Nachfolgenden wird darum eine Headerlänge von 2 Bytes angenommen.

Abbildung 2.4: Ein Byte in der vorgeschlagenen Methodik



3 Randbedingungen

4 Auswahl der Testdaten für die Arbeit

5 Algorithmische Beschreibung des Verfahrens

6 Analyse, Design und Implementierung des Verfahrens

7 Vergleich der Performancezeiten für ein, zwei oder mehrere CPU-Cores

8 Zusammenfassung

Literaturverzeichnis

- [1] Md. Mashud Rana, M. E. Kawsar, M. E. Rabbani, S. M. M. Rashid und K. E. U. Ahmed. An Enhanced Two-Dimensional Color Barcode System. *Journal of Emerging Trends in Engineering and Applied Sciences (JETEAS)*, 2(1):126–131, 2011.
- [2] Microsoft Research. About High Capacity Color Barcode Technology. <http://research.microsoft.com/en-us/projects/hccb/about.aspx>, 2013. [Online; accessed 14-June-2013].
- [3] Wikipedia. Barcode — Wikipedia, The Free Encyclopedia. <http://en.wikipedia.org/w/index.php?title=Barcode&oldid=558925085>, 2013. [Online; accessed 14-June-2013].
- [4] Wikipedia. High Capacity Color Barcode — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=High_Capacity_Color_Barcode&oldid=559150607, 2013. [Online; accessed 14-June-2013].
- [5] Wikipedia. QR code — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=QR_code&oldid=559415969, 2013. [Online; accessed 14-June-2013].
- [6] Wikipedia. Universal Product Code — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Universal_Product_Code&oldid=558763746, 2013. [Online; accessed 14-June-2013].

Abbildungsverzeichnis

2.1	Eindimensionaler Barcode (Quelle: Wikipedia)	4
2.2	QR Code (Quelle: Wikipedia)	5
2.3	High Capacity Color Barcode (Quelle: Wikipedia)	5
2.4	Ein Byte in der vorgeschlagenen Methodik	7

Tabellenverzeichnis

2.1	Zuordnung der Farben zu den Nibbles	6
-----	---	---