



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

DEKÓDOVÁNÍ ČÁROVÉHO KÓDU V OBRAZE

DECODING BARCODE IN IMAGE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. PETRA BAČÍKOVÁ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ADAM HEROUT, Ph.D.

BRNO 2010

Abstrakt

V práci jsou uvedeny základní pojmy z oblasti zpracování čárových kódů, jejich historie, vývoj a dělení. Vybrané zajímavé typy jsou představeny blíže. Je uvedena specifikace kódů EAN-8, EAN-13 a UPC-A. Klíčová část popisuje algoritmus dekodování čárových kódů v obraze, přičemž byla zvolena symbologie EAN-UCC. Úspěšnost algoritmu byla testována na několika sadách obrázků. Na závěr jsou diskutovány výsledky testování a je vytyčen další možný vývoj projektu.

Abstract

The thesis describes the basic types of barcodes, their development and history. It's mentioned cutting barcodes by dimension, types of barcodes which are the best known and the best used, are described. The key chapter describes details of EAN-8, EAN-13, UPC-A and the additional symbol. It's outlined an algorithm for decoding barcode in image. In conclusion, the results are evaluated and a further development of the project is outlined.

Klíčová slova

Dekodování čárového kódu v obraze, čárový kód, kódové tabulky, doplňkový symbol, lokalizace, segmentace, binarizace, dekodování, EAN-8, EAN-13, ISBN, ISSN, UPC-A, GS1.

Keywords

Decoding barcode in image, barcode, coding tables, add-on symbol, localization, segmentation, binarization, decoding, EAN-8, EAN-13, ISBN, ISSN, UPC-A, GS1.

Citace

Petra Bačíková: Dekodování čárového kódu v obraze, diplomová práce, Brno, FIT VUT v Brně, 2010.

Dekódování čárového kódu v obraze

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně pod vedením pana Adama Herouta. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

.....

Petra Bačíková
24. května 2010

Poděkování

Chtěla bych poděkovat panu Adamu Heroutovi za dobrou motivaci, cenné rady, poskytnuté materiály a jeho čas strávený na pravidelných konzultacích.

© Petra Bačíková, 2010.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	3
2 Čárové kódy	5
2.1 Historie	5
2.2 Terminologie	6
2.3 Přidělování čísel čárových kódů	9
2.4 Typy čárových kódů	10
3 Symbologie EAN-UCC	17
3.1 Kódování	17
3.2 Symbol EAN-8	17
3.3 Symbol EAN-13	20
3.4 Symbol UPC-A	23
3.5 Doplnkové symboly	24
4 Obrazové techniky zpracování čárového kódu	25
4.1 Předzpracování obrazu	25
4.2 Lokalizace	27
4.3 Dekódování	28
4.4 Zhodnocení	30
5 Implementace	31
5.1 Návrh aplikace	31
5.2 Předzpracování obrazu	32
5.3 Výběr řádků	33
5.4 Segmentace	33
5.5 Uložení kódových tabulek	34
5.6 Dekódování čárového kódu	34
5.7 Porovnání kontrolní číslice	36
5.8 Hromadné zpracování	36
5.9 Webové rozhraní	38

6	Vyhodnocení	39
6.1	Testování	39
6.2	Robustnost	41
7	Závěr	45
A	Obsah CD	49
A.1	Obsah datového nosiče	49
B	Jednoduchá nápověda	50
B.1	Spuštění na OS Linux	50
B.2	Spuštění na OS Windows	51
C	Seznam kódů zemí pro EAN-UCC	52

Kapitola 1

Úvod

V práci jsou popsány způsoby kódování čárovým kódem, typy kódů a jejich specifikace, metody a postupy používané k dekódování čárového kódu v obraze. Výstupem této práce je algoritmus pro rozpoznání čárového kódu v obraze. Proces dekódování závisí na kvalitě vstupních dat a dodržování norem, byla proto vytvořena testovací sada vzorků na kterých je účinnost algoritmu ukázána. Tato kapitola (1) má za úkol stručně vystihnout uvedenou práci a dát čtenáři rychlý přehled obsahu jednotlivých kapitol.

Pro uvedení čtenáře do obrazu je zahrnuta teoretická část o čárových kódech, uvedena v kapitole 2, která popisuje historii čárových kódů, jejich postupný vývoj a definuje základní terminologii používanou v této branži. Popisuje rozdělení čárových kódů dle dimenze a ty typy čárových kódů, které přišly autorce zajímavé, detailněji rozebírá a zobrazuje jejich příklady, použití a možnosti.

Autorka pro implementaci algoritmu dekódování vybrala tyto typy čárových kódů: EAN-8, EAN-13 a UPC-A, které jsou přesně definovány v kapitole 3. Pro každý symbol čárového kódu je zmíněn jeho formát, velikost, kódování a výpočet kontrolní číslice. Pro úplnost je uveden i doplňkový symbol, používaný v kombinaci s EAN-13.

Již existující algoritmy dekódování a zpracování obrazu pro čárové kódy jsou vypsány v kapitole 4. Proces zpracování čárového kódu je rozdělen na několik částí: předzpracování obrazu, lokalizace a dekódování. Snahou bylo najít nejen nejrozumnější možnosti zpracování obrazu pro dekódování, ale i samotné algoritmy dekódování čárových kódů. Na závěr kapitoly je provedeno zhodnocení postupu při dekódování kódů z obrazu.

Kapitola 5 se věnuje samotné implementaci algoritmu pro dekódování. Nejdůležitější je návrh aplikace, který udává směr celé práci. Postupně jsou uvedeny algoritmy pro předzpracování obrazu, výběr řádků, segmentaci a dekódování čárového kódu včetně porovnání kontrolní číslice. Je nastíněno hromadné zpracování dat a ukázána možnost testovat program pomocí webového rozhraní.

V kapitole 6 je analyzována úspěšnost na různých sadách obrázků, počínaje jednoduchými a konče těžkými obrázky pro rozpoznání. Nejrozsáhlejší testování proběhlo na typu kódu EAN-13, který je nejvíce dostupný. Testování robustnosti algoritmu proběhlo v zá-

vislosti na velikosti obrázku, rotaci kódu v obraze a rozostření.

Závěrečná kapitola 7 má za úkol zhodnotit provedenou práci. Je diskutován současný stav práce a je udán možný směr vývoje projektu.

V dodatku B je adresářová struktura příloženého nosiče a návod na spuštění programu a testovacích skriptů. Další dodatek C je vložen pouze orientačně, uvádí seznam kódů zemí pro symbologii EAN-UCC.

Kapitola 2

Čárové kódy

Čárové kódy umožňují označování zboží, přičemž tak zjednodušují řízení dodávek zboží po celém světě. Kódy mohou označovat nejen zboží, ale i přepravní jednotky, služby, objekty a místa.

Čárové kódy usnadňují elektronické obchodní procesy tím, že se snímají elektronicky na místě prodeje, při příjmu ve skladech a na dalších místech, kde to podnikové procesy vyžadují. Tyto identifikační čísla se používají i při elektronické výměně dat mezi výrobcem a odběratelem, čímž se zvyšuje rychlost komunikace a zpracování [14].

Kapitola uvádí historii a vznik čárových kódů [3], dále základní pojmy týkající se oblasti zpracování čárových kódů, jako např. orientace čárového kódu, diskrétní a spojitý čárový kód, definice názvosloví popisu čárového kódu. Na konci kapitoly jsou rozděleny čárové kódy dle prostorových vlastností a jsou detailněji popsány nejznámější a nepoužívanější z nich, bylo použito zdrojů několika výrobců etiket čárových kódů, zejména Leonardo technology [1] a IDAutomation [8].

2.1 Historie

Čárové kódy vznikly v polovině minulého století jako efektivní řešení strojového čtení číselných, později i alfanumerických řetězců. Klasický čárový kód tvoří skupina po sobě vytištěných černých pruhů, oddělených od sebe mezerami.

Nejdříve se používalo méně než 20 pruhů, s postupem času a potřeb kódovat více informací se šířka kódu zvýšila na 50 a více pruhů. Šířka dat ale i tak přestávala stačit a bylo nutné rozvinout jinou metodu kódování, s vyšší hustotou informace – 2D čárové kódy. V současnosti nejvyšší hodnotu informace mají kódy nové generace, které jsou většinou vícerozměrné, barevné.

Jako jeden z prvních kódů byl navržen Code 2/5, který byl nadále rozšiřován. K prvnímu 2D kódu dospěl vývoj až v roce 1988 od společnosti Intermec Corporation a jeho označení bylo Code 49. Další kódy následovaly dle potřeby a použití. Některé se používají pouze v daných zemích, jiné jsou specifické svým použitím. V současné době (2010) je definováno

Tabulka 2.1: Chronologie počátku vzniku čárových kódů

rok	typ kódu
1968	Code 2/5
1972	Code 2/5 Interleaved, Codabar
1973	UPC (Universal Product Code)
1974	Code 3/9
1976	EAN (European Article Numbering)
1977	Code 11
1981	Code 128
1982	Code 93
1988	Code 49

přibližně na 200 standardů různých čárových kódů pro různá použití.

2.2 Terminologie

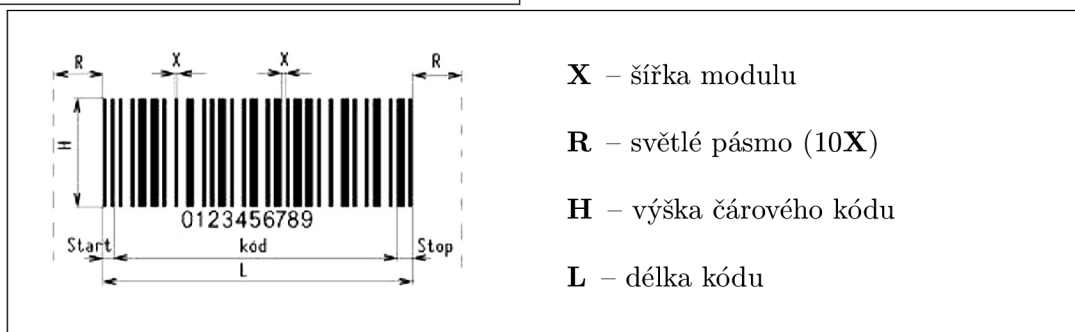
Pro rychlejší orientaci v tomto odvětví je uvedeno několik základních termínů a pojmů. Je zobrazena konstrukce čárového kódu a jeho základní rozměry (obrázek 2.1). V této kapitole bylo využito poznatků z české technické normy ČSN EN 1556 [9], která přesně definuje konstrukci čárových kódů pro výrobce.

Datová oblast – část symbolu čárového kódu nesoucí datovou část (datová kódová slova).

Pomocný znak – část symbolu, nedatový znak, který nenesí informaci.

- Oddělovací znak dat – odděluje konec jedné a začátek další datové oblasti.
- Znak pro začátek – definuje počátek symbolu čárového kódu.
- Znak pro konec – definuje ukončení symbolu čárového kódu.

Obrázek 2.1: Konstrukce čárového kódu



X – šířka modulu

R – světlé pásmo ($10X$)

H – výška čárového kódu

L – délka kódu

Pozadí – světlá plocha mezi tmavými prvky vytištěného symbolu a okolo nich. Pozadí může být bílé nebo může být vhodně zvolena světlá barva.

Čára – jakákoliv z tmavých čar ve vytištěném symbolu čárového kódu.

Šířka modulu – se značí **X**, udává nejužší element kódu (čára či mezera).

Ochranná zóna – světlé pásmo, klidová zóna. Jedná se o plochu prostou rušivých značek, která musí obklopovat symbol čárového kódu. Zejména musí předcházet znaku pro začátek a následovat po znaku pro konec.

Kontrolní znak/číslice symbolu – číslice nebo znak vypočtený dle definovaného algoritmu z ostatních znaků kódu. Používá se ke kontrole, zda je kód správně sestaven. Kontrolní znak symbolu netvoří součást dat zakódovaných v symbolu.

Datový kontrolní znak/číslice – číslice nebo znak vypočtený z dat a napojený jako součást datového řetězce k potvrzení, že data byla správně sestavena a přenesena.

Kódová stránka – tabulka, ve které je každému znaku přiřazena posloupnost nul a jedniček, udávající kód znaku. Kódová stránka je vždy omezena na soubor znaků, který je typem kódu podporován.

Velikost čárových kódů

Jeden a týž kód může být vyroben v různých velikostech. Záleží na zvolení základní šířky modulu **X**. Typicky [2] se používají tyto velikosti: malá, standardní, velká, kruhová.

Na obrázku 2.2 jsou zobrazeny velikosti čárových kódů. Malá velikost (vlevo nahoře) je 0,5 palce vysoká, 1,75 palce široká. Standardní velikost je 1 palec na výšku a 1,5 palce na šířku (vpravo nahoře). Veliký čárový kód (vlevo dole) má prodlouženou ochrannou zónu na 2,5 palce a je 1 palec vysoký. Kruhový čárový kód (vpravo dole) má průměr zóny 1,36 palce.

Technika navazování znaků

Jedná se o techniku navazování znaků v symbolu kódu mezi sebou. Symbolika kódu může být buď spojitá nebo diskrétní. Obrázek 2.3 vlevo ukazuje výřez diskrétního a vpravo spojitého kódu. Jsou naznačeny délky jednotlivých znaků symbolu.

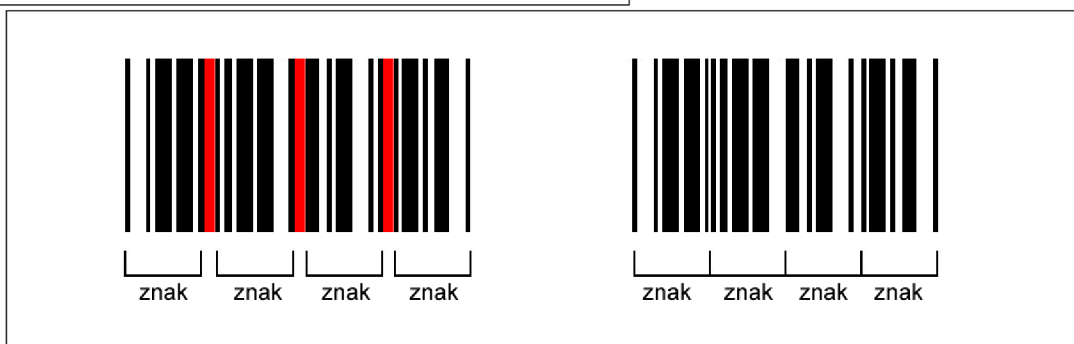
Diskrétní symbolika je taková, kde je každý znak interpretován zvlášť, bez ohledu na zbytek kódu. To znamená, že meziznakové mezery neobsahují informace (neexistují), protože každý znak začíná a končí čarou. Meziznakové mezery jsou v obrázku označeny červenou barvou.

Spojitá (souvislá) symbolika nedefinuje žádné mezery mezi znaky, tj. koncový prvek jednoho znaku symbolu navazuje na první prvek dalšího znaku symbolu. Všechny tyto prvky přenášejí data souvisle. Musí být dbáno na dodržení střídání čar a mezer, aby čára nebo mezera nebyla zdvojená.

Obrázek 2.2: Velikosti čárových kódů



Obrázek 2.3: Diskrétní a spojitá symbolika kódu



Orientace čárového kódu

Orientace čárového kódu udává volbu polohy ve vztahu k určitému směru nebo rovině. Užívá se žebříková orientace (vpravo) a plotová orientace (vlevo) zobrazena na obrázku 2.4 převzatého z [14].

Plotová orientace je taková poloha symbolu čárového kódu, kde je osa čar svislá, aby horizontální snímání mohlo projít přes celý symbol. Naopak *žebříková orientace* je taková, kde osa čar je vodorovná, aby mohlo přes symbol projít snímání svislé.

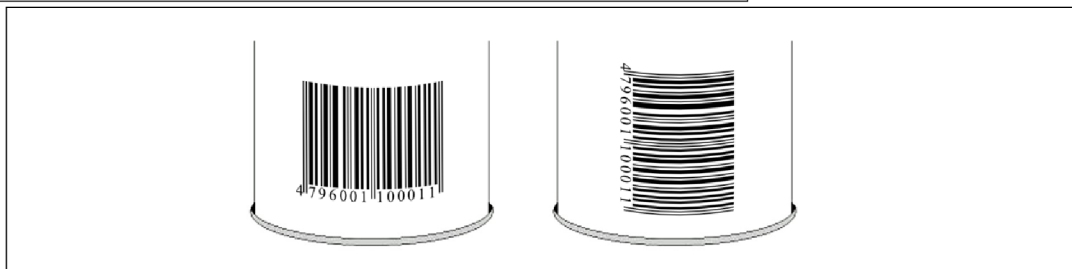
Parita

Symbol čárového kódu má sudou nebo lichou paritu, jestliže všechny znaky symbolu mají stejnou paritu (buď sudou nebo lichou).

Pokud je počet tmavých modulů ve znaku sudý, jedná se o *sudou paritu*. *Lichá parita* říká, že počet tmavých modulů obsažených ve znaku je lichý.

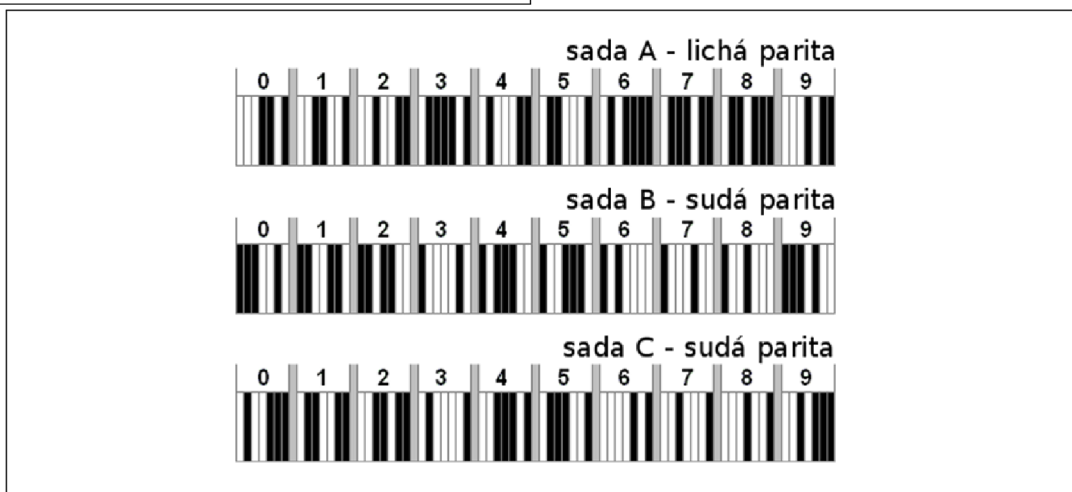
Parita se využívá pro kontrolu správnosti symbolu při čtení čárového kódu. Lichá i

Obrázek 2.4: Plotová a žebříková orientace čárového kódu



sudá parita je charakteristická pro kódování znaků v symbologii EAN-UCC, jak je vidět na obrázku 2.5, který znázorňuje paritu pro všechny tři sady kódové tabulky.

Obrázek 2.5: Parita kódování EAN-UCC



Kontrast

Úspěšné čtení čárového kódu závisí na kontrastu C , který je dán poměrem rozdílu odrazu pozadí R_{bg} a odrazu čárky R_{line} k odrazu pozadí.

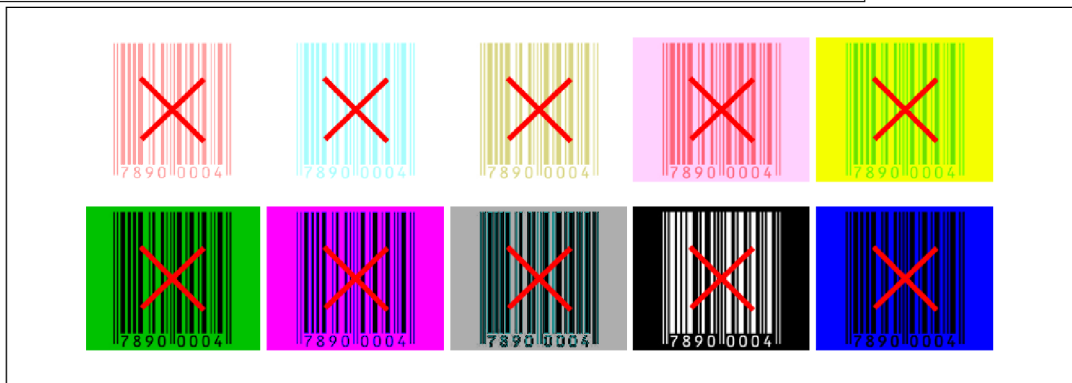
$$C = \frac{R_{bg} - R_{line}}{R_{bg}} \geq 0,7 \quad (2.1)$$

Pokud je poměr větší než 0,7, symbol bude pravděpodobně bezproblémově snímatelný. Riziko nečitelnosti zvyšuje lesklý povrch potištěného materiálu, průhlednost materiálů, přímý tisk na materiály barevné, tmavé, barevně nejednotné (obrázek 2.6).

2.3 Přidělování čísel čárových kódů

První patent na čárové kódy byl udělen v roce 1949. Organizace GS1 Czech Republic [7], zabývající se nejen tvorbou globálních standardů pro automatickou identifikaci, poskytuje jako jediná v ČR autorizované čárové kódy.

Obrázek 2.6: Nevhodné barevné kombinace se špatným kontrastem



Nadnárodní organizace GS1 se sídlem v Bruselu přiděluje jednotlivým státům čísla, zpravidla třímístná, určující stát původu zboží. Seznam kódů přiřazený k jednotlivým zemím pro EAN-UPC je uveden v dodatku C. Čísla výrobce (většinou 4–6 místné) určuje národní organizace, v případě ČR již zmíněná GS1 Czech Republic.

2.4 Typy čárových kódů

Kódy je možné dělit dle způsobu kódování černých a bílých pruhů. Existuje mnoho čárových kódů, které mají své specifické využití. Některé druhy kromě číslic mohou kódovat i písmena a znaky.

Konstrukci čárového kódu můžeme pojmovit v závislosti na prostoru. Rozeznáváme čárové kódy jednodimenzionální (1D) a dvoudimenzionální (2D). 1D kódy obvykle kódují číslice nebo alfanumerické znaky, které určují ve většině případů kódy předmětů, které jsou s dalšími informacemi uloženy v externí databázi. 2D kódy mají větší kapacitu a obsahují většinou více informací o předmětu.

Kódy nové generace jsou především určeny pro čtení z mobilních telefonů, obvykle obsahují zakódovanou internetovou adresu, např. článek v časopise. Umožňují rychlé získání informace – není třeba opisovat internetovou adresu do mobilu, či ukládat kontaktní informace do adresáře, stačí pouze čárový kód vyfotit. Poznatky o kódech nové generace byly převzaty z článku Budoucnost a současnost čárových kódů pro mobily [5].

Dále jsou popsány typy čárových kódů v závislosti na prostoru, uvedeny příklady kódů a jejich další dělení.

Jednodimenzionální kódy

Lineární symbolika, ve které symbol tvoří jediná řada znaků symbolu.

- **lineární** – Code 2/5, EAN, Code 128, Code 3/9, Code 11, Code 93, UPC, ISBN, ISSN, Codabar, MSI/Plessey

Dvoudimenzionální kódy

Víceřádková symbolika (též vrstvená symbolika), kde se symbol skládá ze dvou nebo více vertikálně přilehlých řádek modulů symbolu.

- **dvourozměrné (matrix)** – data jsou definována dvourozměrnými souřadnicemi tmavých bodů v matici, např. Data Matrix, QR code, Aztec Code, UPS MaxiCode, Semacode
- **kvazidvouzměrné (stacked)** – skládané kódy vznikající složením jednodimenzionálních kódů, např. Code 49, PDF 417
- **kruhové** – jedná se o jiné zobrazení jednorozměrného kódu, transformace sledu čar do soustředných kružnic, typicky Kruhový kód

Kódy nové generace

Speciální tvary kódů, většinou vícerozměrné, některé barevné.

- BeeTagg, Multi Color Barcode

Nyní jsou vybrány a popsány nejznámější typy čárových kódů. Vždy je uvedeno jakou má kód délku a kde se nejčastěji využívá. Pro názornost je uveden jeho příklad na obrázku. O kategorizaci bylo pojednáváno dříve, proto zde není uvedena.

2.4.1 Code 2/5

Jedná se o univerzální numerický kód s proměnnou délkou, sudé parity, který je samoopravný. Kód dostal název podle své konstrukce: dvě tloušťky čar z celkových pěti čar pro každý zakódovaný znak, čili 2 z 5.

Code 2/5 je užívaný nejčastěji pro interní aplikace a označování obchodních jednotek. Zpravidla pro pořadač zásobníku v identifikaci obálek, leteckých lístků a zavazadel. Je znám v těchto modifikacích: Datalogic, IATA, Industrial, Interleaved, Inverted a Matrix.

V obrázku 2.7 je zakódován řetězec 16021986. Čárový kód 2/5 lze vygenerovat například na www.barcode-soft.com.

Obrázek 2.7: Čárový kód Code 2/5 interleaved

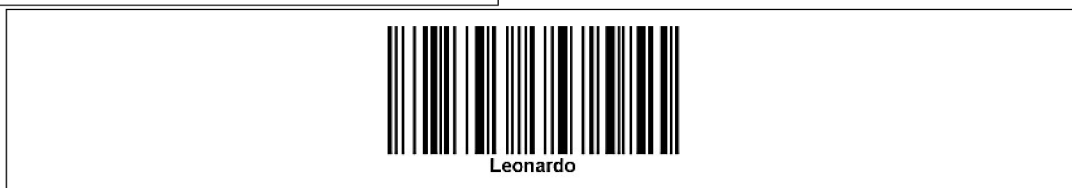


2.4.2 Code 128

Univerzální, volně použitelný čárový kód ke kódování alfanumerických znaků s vysokou hustotou informace (obrázek 2.8), je schopný zakódovat 128 znaků (spodní polovinu ASCII). Je souvislý, proměnné délky a umí rozlišovat velikost písmen v kódu. Znaky se sestávají ze 3 čar a 3 mezer tak, že celková šířka znaku je 11 modulů. Čáry a mezery mohou mít šířku 1–4 moduly.

Umožňuje zakódovat hodně informací o výrobku, jak jsou např. datum výroby, datum balení, minimální trvanlivost, hmotnost, sériové číslo, délka, šířka, plocha, objem a další.

Obrázek 2.8: Čárový kód Code 128 [1]



2.4.3 UPC – Universal Product Code

Univerzální numerické kódy navržené pro jednoznačnou identifikaci výrobku a jeho výrobce mají souvislou symboliku a pevnou délku, každý znak se skládá ze 4 modulů. Užívá se pro jednoznačné označování výrobků v maloobchodě od roku 1973 v USA a Kanadě.

Nepoužívanějšími variantami jsou UPC–A, a jeho kratší varianta UPC–E mající také název *kód s potlačenými nulami*. UPC verze A se používá k zakódování dvanáctimístného čísla. První číslice udává systém číslování, dalších 5 čísel je pro identifikaci výrobce, 5 čísel pro identifikaci výrobku a jako poslední kontrolní znak. UPC verze E komprimuje 12 čísel do kódu obsahujícího pouze 6 znaků (obrázek 2.9).

Obrázek 2.9: Čárový kód UPC–E a UPC–A



2.4.4 EAN – European Article Numbering

EAN je nadstavbou UPC, je to nejnámější kód používaný pro zboží prodávané v obchodní síti. Nepoužívanější varianty jsou EAN–8 a EAN–13, mají pevnou délku, mohou kódovat

pouze číslice, přičemž každá číslice je kódována dvěma čarami a dvěma mezerami. Obě varianty kódu jsou zobrazeny na obrázku 2.10.

Varianta EAN-13 je užívána v maloobchodě a velkoobchodě k označení výrobků, obsahuje 13 číslic, přičemž přímo zakódováno je pouze 12. Speciální varianta EAN-8 je používána v maloobchodě pro označení malých výrobků, na kterých se nedá uplatnit kód EAN-13. Kóduje pouze 8 číslic.

Obrázek 2.10: Čárový kód EAN-8 a EAN-13

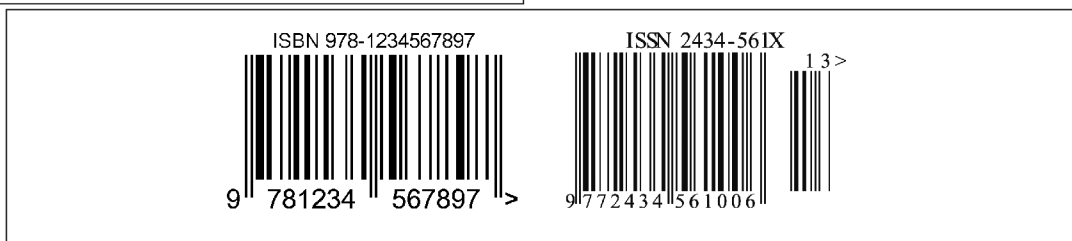


Pro implementaci byl vybrán tento typ čárového kódu pro jeho masivní rozšíření. Detailní popis variant (EAN-8 a EAN-13) je uveden v následující kapitole.

2.4.5 ISBN a ISSN

Kódy ISBN (*International Standard Book Number*) a ISSN (*International Standard Serial Number*) mají v dnešní době identickou strukturu jako EAN-13 (obrázek 2.11), nejedná se tedy o specificky další kódový typ. Starší verze těchto kódů měly pouze 10 míst, a proto je bylo třeba upravit.

Obrázek 2.11: Čárový kód ISBN a ISSN



ISBN (obrázek 2.11 vlevo) je mezinárodní kód používaný k označování neperiodických publikací (knih a hudebnin). V Česku ho definuje norma ČSN 977115 z ledna 2002. Dále existuje rozšířená varianta kódu ISBN (tzv. add on kód), používaná především k označování jednotlivých čísel periodické publikace.

Mezinárodní kód ISSN je používaný k označování periodických tiskovin, definovaný normou ČSN 977116. Publikace se označují pomocí kódu EAN-UCC 13 využívající čísla ISSN. Ke klasickému symbolu EAN-13 se přidávají doplňkové kódy, které mají délku 5 pro deníky a délku 2 pro časopisy. Na obrázku 2.11 vpravo je ISSN s doplňkovým kódem o délce 2, který udává, že číslo časopisu je 13.

2.4.6 PDF 417

PDF 417 nebo-li Portable Data File 417 (obrázek 2.12) je kvazidvourozměrný kód s velmi vysokou informační kapacitou. Skládá se ze 3 až 90 řádků, přičemž v každém řádku může být zakódováno 1 až 30 znaků. Každý znak tvoří 4 čáry a 4 mezery o šířce minimálně jednoho a maximálně 6 modulů. Celkem je však modulů ve znaku přesně 17. Při porušení kódu je možné detekovat a opravit chybu algoritmem Reed-Solomon. Tento algoritmus je více popsán v knize E. Blahuta *Theory and practise of error control codes* [4].

Na rozdíl od tradičních čárových kódů, které obvykle slouží jako klíč k vyhledání údajů v externí databázi, PDF 417 nese všechny údaje s sebou a stává se tak nezávislý na vnějším systému. Příkladem použití jsou nejrůznější identifikační karty, řidičské průkazy v některých státech USA, dokumenty, ale například i zakódování diagnózy pacientů.

Obrázek 2.12: Čárový kód PDF-417



2.4.7 QR-Code

QR-Code (Quick Response Code) pochází od společnosti Denso-Wave a existuje od roku 1994. U nás je standardizován od roku 2000 normou ISO/IEC 18004.

Dekóduje informace vysokou rychlostí. Umožňuje uložit 7089 numerických znaků, používá Reed-Solomon algoritmus pro opravu chyb. Rohové záchytné body přispívají k lepšímu čtení na nerovných plochách.

Kód (obrázek 2.13) byl vygenerován pomocí `qrcode.kaywa.com`, je vhodný pro čtení z mobilních telefonů. Existuje pro něj software na čtení přímo z integrovaného fotoaparátu zařízení.

Obrázek 2.13: QR-Code – zakódována URL <http://www.fit.vutbr.cz>

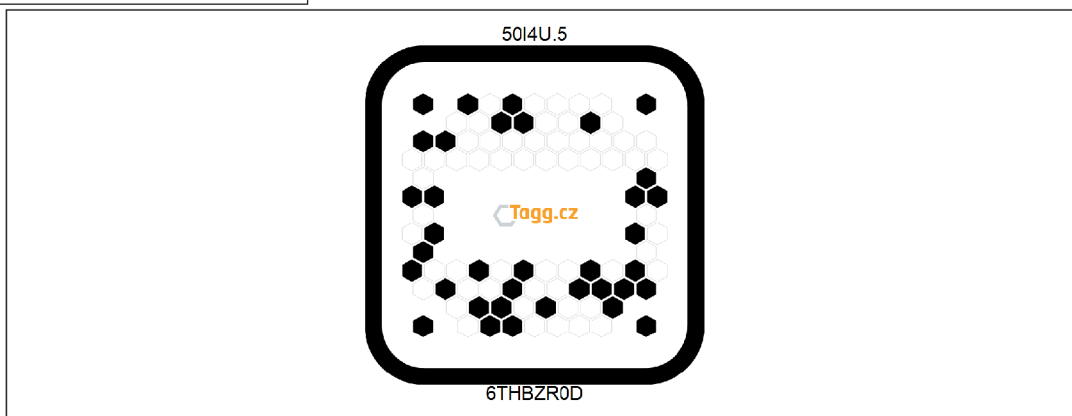


2.4.8 BeeTagg

BeeTagg, někdy označován jako *bee code* – volně přeloženo „včelí kód“, je další z nové generace kódů. Jeho označení je odvozeno zejména proto, že kód vypadá jako včelí plástev.

V české verzi <http://tagg.cz> je možné vygenerovat tyto kódy a dokonce i stáhnout jejich čtečku. Na obrázku 2.14 je zakódována adresa <http://www.fit.vutbr.cz>. Výhoda je v možnosti umístění loga a relativně vizuálně pěkném vzhledu, který může být použit např. na vizitkách. Může obsahovat informace z vizitky, jako jméno, příjmení, telefonní číslo, adresu, a další.

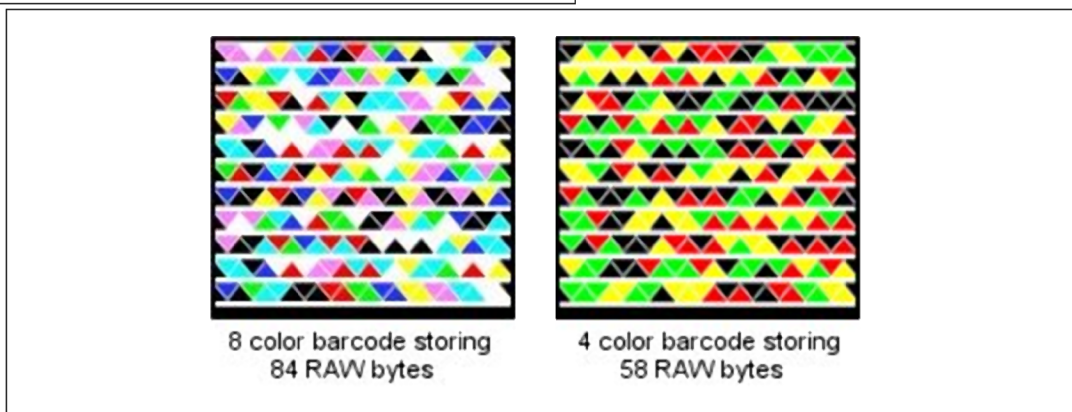
Obrázek 2.14: BeeTagg



2.4.9 High Capacity Color Barcode

Tento relativně nový kód, zveřejněn v roce 2007 firmou Microsoft, je určen pro identifikaci multimediálních děl [10]. Z názvu je zřejmé, že informace jsou v kódu opravdu zakódovány dle barev.

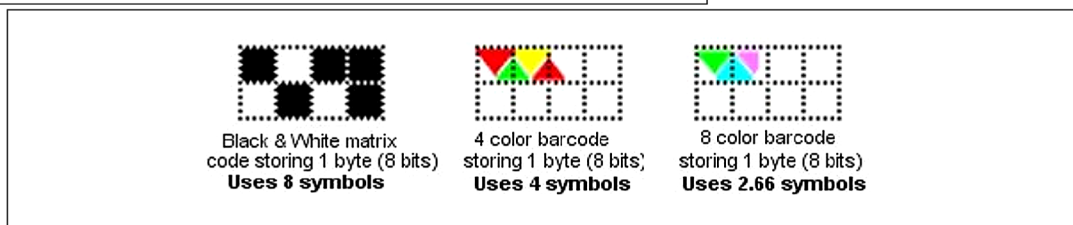
Obrázek 2.15: High Capacity Color Barcode



Existují dvě obdoby tohoto kódu zobrazeny na obrázku 2.15 – čtyřbarevná (vlevo) a osmibarevná (vpravo). Pokud si vezmeme 2D maticový kód, víme, že dvěma barvami

lze zakódovat 1 byte pomocí 8 symbolů. Čtyřbarevný kód je schopen uložit 1 byte pomocí 4 symbolů, ještě lépe je na tom osmibarevný kód, kde je možné uložit 1 byte na 2,66 symbolu. Toto rozložení je demonstrováno na obrázku 2.16.

Obrázek 2.16: Kódování High Capacity Color Barcode



Kapitola 3

Symbologie EAN-UCC

EAN je zkratka pro *European Article Numbering* – druh čárového kódu, který může užívat každý stát zapojený do světového sdružení *EAN International* (od roku 2005 přejmenován na GS1). Celosvětová podpora standardu byla označována jako EAN-UCC (EAN International-Uniform Code Council). Od roku 2005 byla organizace UCC pro správu a řízení v USA přejmenována na GS1 US. Formát symbolu čárového kódu definuje norma ISO/IEC 15420.

Budou popsány vybrané typy těchto kódů, kódování a kódové stránky, společné pro všechny varianty. Dále následuje detailní popis variant EAN-8, EAN-13 a UPC-A, včetně výpočtů kontrolních číslic, které jsou vybrány pro implementaci. Na konci kapitoly naleznete popis doplňkových symbolů používaných pro kódování ISBN a ISSN pomocí EAN-13.

3.1 Kódování

Symbole EAN a UPC jsou lineární, spojitě a numerické (zakódovatelné znaky jsou 0–9). Každá z číslic 0 až 9 je kódována dvěma mezerami a dvěma čarami. Kód může obsahovat 8 číslic, pak se jedná o kód EAN-8, 13 číslic – jedná se o kód EAN-13 (obrázek 2.10) nebo 12 číslic pro kód UPC-A. Upravená hodnota kódu EAN může uchovávat informace o ISBN či ISSN kódech, zmíněných výše, pomocí přidání dodatkového symbolu.

Pro kódování číslic jsou definovány 3 sady kódových tabulek (tabulka 3.1), kde pro každé číslo od 0 do 9 je přiřazena posloupnost čar a mezer, přičemž 0 značí mezeru a 1 čáru. Každé číslo je složeno ze 7 modulů, kde modul je definován jako minimální šířka čáry. Podle počtu modulů se určuje šířka čáry nebo mezery.

3.2 Symbol EAN-8

Kód EAN-8 je určen pro identifikaci malých spotřebitelských jednotek. Je zásadně používán jen na zboží malých rozměrů, kde nelze plochu pro identifikaci jakkoliv zvětšit, příkladem jsou cukrovinky, žvýkačky, a další.

Tabulka 3.1: Kódová stránka pro čárové kódy EAN

číslice	sada A	sada B	sada C
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

3.2.1 Struktura a formát

Kód má pevnou délku datového pole – 8 znaků, nesmí následovat pole dodatkového symbolu. Formát a jeho struktura jsou na obrázku 3.1. Technické údaje a rozměry jsou uvedeny v tabulce 3.2, tyto údaje specifikuje [7].

Obrázek 3.1: Formát symbolu EAN-8



Struktura kódu: 2–3 pozice mezinárodního prefixu (pro Českou republiku 859 viz dodatky C), 3–4 pozice (dle prefixu) identifikace položky (přidělená smluvně od GS1 Czech Republic), jako poslední je uvedena povinná kontrolní číslice *K*.

Závazný formát symbolu: levá ochranná zóna + levý ochranný znak + 4 datové znaky + středový ochranný znak + 4 datové znaky + pravý ochranný znak + pravá ochranná zóna.

Výška čar: Všechny ochranné znaky musí přesahovat spodní linii datových čar symbolu o 5X.

Tabulka 3.2: Technické údaje symbolu EAN–8

symbol EAN–8	technické údaje
Jmenovitá šíře X (100%)	0,33 mm
Jmenovitý rozměr (šířka × výška)	26,73 × 21,64 mm
Rozměrové možnosti	80 – 200 %
Ochranná zóna	Levá: 7X , Pravá: 7X
Ochranná zóna horní	Min. 1X
Minimální kvalitativní stupeň	1,5/06/670

Číslicové vyjádření: Umístěno pod symbolem ve vzdálenosti **1X** od spodní linie čar. Celková výška symbolu zahrnuje i výšku číslic. Při jmenovitém rozměru kódu činí výška číslic 2,75 mm.

3.2.2 Kódování

Symbol EAN–8 se kóduje dle tabulky 3.1, přičemž první 4 datové znaky se kódují podle sady A a druhé 4 datové znaky se kódují podle sady C.

Příklad vytvoření čárového kódu

Mějme řetězec „8652164“, který chceme zakódovat do čárového kódu EAN–8. Počet datových číslic je 8, ale vstupem je pouze 7 číslic, poslední číslice (kontrolní) se musí dopočítat, aby odpovídala normě.

V sadě A nalezneme příslušné zakódování pro první čtyři znaky 8, 6, 5, 2, zapíšeme za sebe: 0110111 0101111 0110001 0010011, stejně provedeme pro druhé tři znaky 1, 6, 4 ale ze sady C: 1100110 1010000 1011100.

Nyní je potřeba dopočítat kontrolní číslici, dle algoritmu 3.1: $\beta = 3 \cdot 8 + 1 \cdot 6 + 3 \cdot 5 + 1 \cdot 2 + 3 \cdot 1 + 1 \cdot 6 + 3 \cdot 4 = 68$. Nejbližší vyšší násobek deseti k číslu 68 je 70. Potom kontrolní číslice $K = 70 - 68 = 2$. Kontrolní číslice patří do druhé poloviny datových znaků, kóduje se tedy podle sady C: 1101100.

Výsledný řetězec čárového kódu tedy bude „86521642“ a jemu odpovídající zakódování 0110111 0101111 0110001 0010011 1100110 1010000 1011100 1101100 je znázorněno na obrázku 3.2.

Obrázek 3.2: Příklad zakódování 86521642 v EAN–8

3.2.3 Výpočet datové kontrolní číslice

Pro výpočet kontrolní číslice je třeba znát vážený výpočet β , který je dán součtem všech číslic kódu, přičemž liché pozice v čárovém kódu se násobí třikrát a sudé pozice pouze jednou.

Tabulka 3.3: Výpočet váženého výpočtu β pro EAN-8

čárový kód	a	b	c	d	e	f	g
pozice	lichá	sudá	lichá	sudá	lichá	sudá	lichá
váha α	3	1	3	1	3	1	3
výpočet	$3 \cdot a$	$1 \cdot b$	$3 \cdot c$	$1 \cdot d$	$3 \cdot e$	$1 \cdot f$	$3 \cdot g$
vážený výpočet β	$\beta = 3 \cdot a + 1 \cdot b + 3 \cdot c + 1 \cdot d + 3 \cdot e + 1 \cdot f + 3 \cdot g$						

Kontrolní číslici lze vypočítat dle algoritmu 3.1, který popisuje norma [7]. Kódování kontrolní číslice je stejné jako pro druhou polovinu datových znaků, tedy dle sady C.

Algoritmus 3.1: Výpočet kontrolní číslice pro EAN-UCC

1. Vypočítej vážený výpočet β dle tabulky pro daný symbol.
2. K váženému výpočtu β najdi C , nejbližší vyšší násobek 10.
3. Vypočítej kontrolní číslici K , dle výrazu $K = C - \beta$.

Pokud se přečtená a vypočítaná kontrolní číslice shoduje, dekódování bylo s nejvyšší pravděpodobností správné.

3.3 Symbol EAN-13

Kód EAN-13 je určený pro identifikaci spotřebitelských či obchodních jednotek po celém světě. Jedná se o pravděpodobně nejčastěji používaný kód vůbec, je možné ho vidět na potravinách, oblečení a tištěných etiketách v obchodech.

Jeho upravená podoba může nést informace o knížkách – ISBN nebo o časopisech a jiných periodikách – ISSN, přičemž je k symbolu EAN-13 přidán dodatkový symbol, popsán v poslední sekci této kapitoly.

3.3.1 Struktura a formát

EAN-13 je spojitý, lineární, délka datového pole je pevná – 13 znaků, oproti kódu EAN-8 může následovat pole dodatkového symbolu pro rozšířené kódování [7]. Technické údaje jsou uvedeny v tabulce 3.4.

Obrázek 3.3: Formát symbolu EAN-13



Struktura kódu: zleva 2–3 pozice mezinárodního prefixu, 4–6 pozic (dle prefixu) neměnné identifikace firmy (přidělená smluvně od GS1 Czech Republic), 3–5 pozic identifikace položky (definuje firma), povinná kontrolní číslice *K*.

Závazný formát symbolu: levá ochranná zóna + levý ochranný znak + 6 datových znaků + středový ochranný znak + 6 datových znaků + pravý ochranný znak + pravá ochranná zóna. První datová číslice je předtištěna vlevo od symbolu, je zakódována nepřímo a formát symbolu neprodlužuje.

Tabulka 3.4: Technické údaje symbolu EAN-13

symbol EAN-13	technické údaje
Jmenovitá šíře X (100%)	0,33 mm
Jmenovitý rozměr (šířka × výška)	37,29 × 26,26 mm
Rozměrové možnosti	80–200 %
Ochranná zóna	Levá: 11 X , Pravá: 7 X
Ochranná zóna horní	Min. 1 X
Minimální kvalitativní stupeň	1,5/06/670

Výška čar: Všechny ochranné znaky musí přesahovat spodní linii datových čar symbolu o 5**X**.

Číslicové vyjádření: Umístěno pod symbolem ve vzdálenosti 1**X** od spodní linie čar. Celková výška symbolu zahrnuje i výšku číslic. Při jmenovitém rozměru kódu činí výška číslic 2,75 mm. První levá číslice je předsazena před levý ochranný znak do oblasti levé ochranné zóny, 5**X** od krajní čáry ochranného znaku.

3.3.2 Kódování

Oproti EAN–8, kde je kódování první poloviny datových znaků dáno sadou A, je kódování první poloviny EAN–13 složitější. Prvních 6 datových znaků se kóduje sadou A nebo B podle toho, jaká je uvedena první číslice před kódem. Druhých 6 datových znaků se kóduje dle sady C. Sady A, B a C jsou definované v tabulce 3.1.

Tabulka 3.5: Kódová stránka symbolu EAN–13

1. číslice	2.–7. číslice	8.–13. číslice
0	AAAAAA	CCCCCC
1	AABABB	CCCCCC
2	AABBAB	CCCCCC
3	AABBBA	CCCCCC
4	ABAABB	CCCCCC
5	ABBAAB	CCCCCC
6	ABBBAA	CCCCCC
7	ABABAB	CCCCCC
8	ABABBA	CCCCCC
9	ABBABA	CCCCCC

Tabulka 3.5 definuje, dle jaké sady má být kódována první polovina datových znaků v závislosti na první číslici umístěné před samotným kódem. Tato číslice není v datech zakódována.

3.3.3 Výpočet datové kontrolní číslice

Datovou kontrolní číslici pro EAN–13 lze vypočítat již uvedeným způsobem (algoritmus 3.1). Rozdíl oproti EAN–8 je ve váženém výpočtu β , kde se počítá v sumě přes dvanáct číslic, nikoliv přes sedm a je vyměněna váha lichých a sudých pozic kódu (tabulka 3.6).

Tabulka 3.6: Výpočet váženého výpočtu β pro EAN–13

čárový kód	a	b	c	d	e	f	g	h	i	j	k	l
pozice	L	S	L	S	L	S	L	S	L	S	L	S
váha α	1	3	1	3	1	3	1	3	1	3	1	3
výpočet	$1 \cdot a$	$3 \cdot b$	$1 \cdot c$	$3 \cdot d$	$1 \cdot e$	$3 \cdot f$	$1 \cdot g$	$3 \cdot h$	$1 \cdot i$	$3 \cdot j$	$1 \cdot k$	$3 \cdot l$
vážený výpočet β	$\beta = 3 \cdot a + 1 \cdot b + 3 \cdot c + 1 \cdot d + 3 \cdot e + 1 \cdot f +$ $+ 3 \cdot g + 3 \cdot h + 1 \cdot i + 3 \cdot j + 1 \cdot k + 3 \cdot l$											

3.4 Symbol UPC–A

UPC–A byl původně (1973) vyvinut pro identifikaci spotřebitelských jednotek na americkém trhu. Je navržen z hlediska jednoznačné identifikace výrobku a jeho výrobce. Jeho struktura je založena na principech shodných s EAN–13. Rozdíly jsou uvedeny níže, pokud není uvedeno jinak, platí podmínky jako u EAN–13 a nejsou znovu vypisovány.

Obrázek 3.4: Formát symbolu UPC–A



3.4.1 Struktura a formát

Struktura kódu: zleva 2 pozice prefixu, 5 pozic neměnné identifikace firmy (přidělená smluvně od GS1 US), 5 pozic identifikace položky (definovaná firmou), povinná kontrolní číslice *K*.

Závazný formát symbolu: levá ochranná zóna + levý ochranný znak + 6 datových znaků + středový ochranný znak + 6 datových znaků + pravý ochranný znak + pravá ochranná zóna. Kontrolní číslice je vytištěna vpravo vedle symbolu. První datová číslice je předtištěna vlevo od symbolu, je zakódována nepřímo a formát symbolu neprodlužuje.

Tabulka 3.7: Technické údaje symbolu UPC–A

symbol UPC–A	technické údaje
Jmenovitá šíře X (100%)	0,33 mm
Jmenovitý rozměr (šířka × výška)	37,29 × 26,26 mm
Rozměrové možnosti	80–200 %
Ochranná zóna	Levá: 9X , Pravá: 9X
Ochranná zóna horní	Min. 1X
Minimální kvalitativní stupeň	1,5/06/670

Výška čar: Všechny ochranné znaky symbolu doplněné o čáry prvního a posledního datového znaku musí přesahovat spodní linii datových čar symbolu o $5X$.

Číslicové vyjádření: Umístěno pod symbolem ve vzdálenosti $1X$ od spodní linie čar. Celková výška symbolu zahrnuje i výšku číslic. Při jmenovitém rozměru kódu činí výška číslic 2,75 mm. První levá číslice je předsazena před levý ochranný znak do oblasti levé ochranné zóny, $5X$ od krajní čáry ochranného znaku. Technické údaje jsou uvedeny v tabulce 3.7.

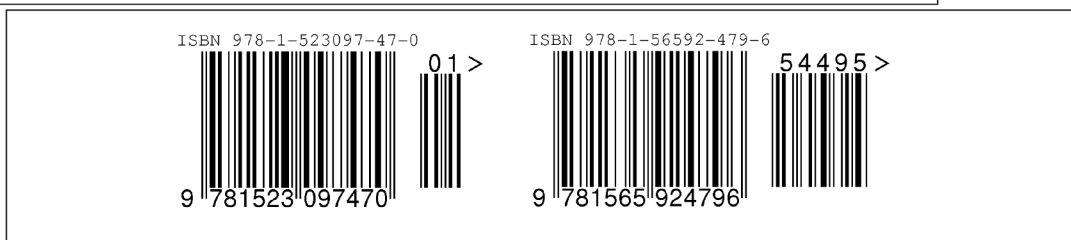
3.4.2 Kódování

Kódování je shodné s EAN-13, přičemž EAN-13 s první číslicí 0 odpovídá UPC-A. Uvedme na příkladě. Mějme UPC-A se zakódovaným řetězcem „075678164125“, pokud před řetězec vložíme 0, dostaneme symbol EAN-13 s řetězcem „0075678164125“. Oba tyto kódy jsou totožné a mají stejné zakódování, ale jiný formát symbolu a výšku čar.

3.5 Doplnkové symboly

Doplňkový symbol je užíván pro kódování ISBN a ISSN, může být použit ve spojení s EAN-13, UPC-A a UPC-E. Existují dva druhy doplňkových symbolů o dvou či pěti znacích, ve kterých je možné zakódovat doplňkové informace (obrázek 3.5). Doplnkové kódy nesmí být použity pro kódování informací, které běžně obsahuje symbol základní.

Obrázek 3.5: Čárový kód EAN-13 s doplňkovým symbolem o délce 2 a 5



Závazný formát symbolu: Levá ochranná zóna, částečně sdílená s pravou ochrannou zónou základního symbolu + levý ochranný znak + první datový znak + oddělovací znak + druhý datový znak + pravá ochranná zóna, přičemž pravý ochranný znak není použit. Doplnkový symbol o 5 znacích je rozšířen o příslušný počet datových a oddělovacích znaků.

Výška čar: činí při jmenovité velikosti symbolu 21,9 mm, přičemž spodní linie čar je zarovnána s linií hlavního symbolu.

Číslicové vyjádření: Poloha čísel je zarovnána s horní linií čar hlavního symbolu, velikost čísel je stejná jako u hlavního symbolu.

Kapitola 4

Obrazové techniky zpracování čárového kódu

Obrazové techniky rozpoznání čárového kódu založené na zpracování obrazu jsou alternativou ke klasickým technikám snímání čárových kódů, např. snímání pomocí laseru nebo LED diod. Klasické systémy se typicky rozdělují na dvě části: lokalizaci čárového kódu a dekodování informace z kódu.

Techniky zpracování obrazu zvyšují efektivnost obou částí klasického zpracování, jak lokalizaci, tak dekodování. V této kapitole jsou popsány různé techniky zpracování obrazu používaných při dekodování čárových kódů. Úsilím bylo nalézt nejrůznější způsoby zpracování obrazu.

Postup při komplexním dekodování čárového kódu z obrazu se ve většině článků shoduje. Nejprve by měl být obraz předzpracován, dále by měla proběhnout lokalizace čárového kódu a jeho zpracování (dekodování). Pro správnost dekodování je u většiny kódů vhodné vypočítat kontrolní číslici a porovnat jí s přečtenou.

4.1 Předzpracování obrazu

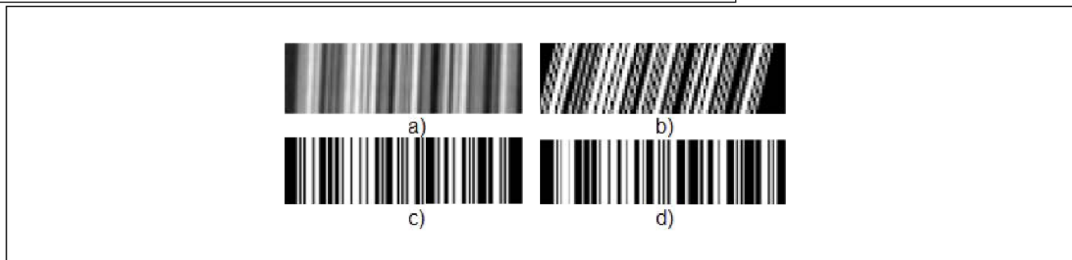
Předzpracování obrazu je první klíč k úspěchu dekodování. V obraze se mohou vyskytnout artefakty, kód může být otočen o 360° kolem své osy, nemusí být pořízen z přímého pohledu na kód, ale s perspektivou.

Geometrické zkreslení – zkosení

R. a L. Oktem ve svém článku [12] uvádějí problematiku rekonstrukce zkoseného čárového kódu. Metoda je založena na hledání vektoru zkosení kódu, ze kterého je možné vypočítat úhel zkosení. Z těchto dvou parametrů je již možné rekonstruovat čárový kód tak, aby byl pohled na něj z vrchu. Příklady rekonstrukce jsou uvedeny na obrázku 4.1, kde a) a b) jsou původní kódy a c) a d) jsou kódy rekonstruované.

Algoritmus má problém s kódy, které mají perspektivu, např. kódy vyfocené na válcových předmětech, nicméně je možné je aplikovat nejen na 1D kódy (UPC), ale i na 2D kódy jako je PDF-417.

Obrázek 4.1: Rekonstrukce zkoseného čárového kódu [12]

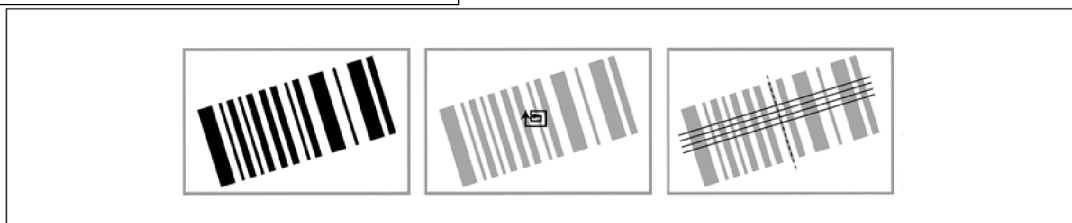


Geometrické zkreslení – otočení

V článku od japonských vědců [11] je uveden algoritmus pro otočení obrazu čárového kódu do svislé (plotové) polohy. Algoritmus pracuje spolehlivě na většině kódů.

Jedná se o tzv. „spirálovou metodu“ (obrázek 4.2). Do středu obrazu se umístí bod (uprostřed), ze kterého se odvíjí spirála, která má za úkol vyhledat černé čáry kódu. Na základě nalezení čar se vypočítá směr natočení (vpravo) a obraz se transformuje do normalizované polohy.

Obrázek 4.2: Spirálová metoda [11]

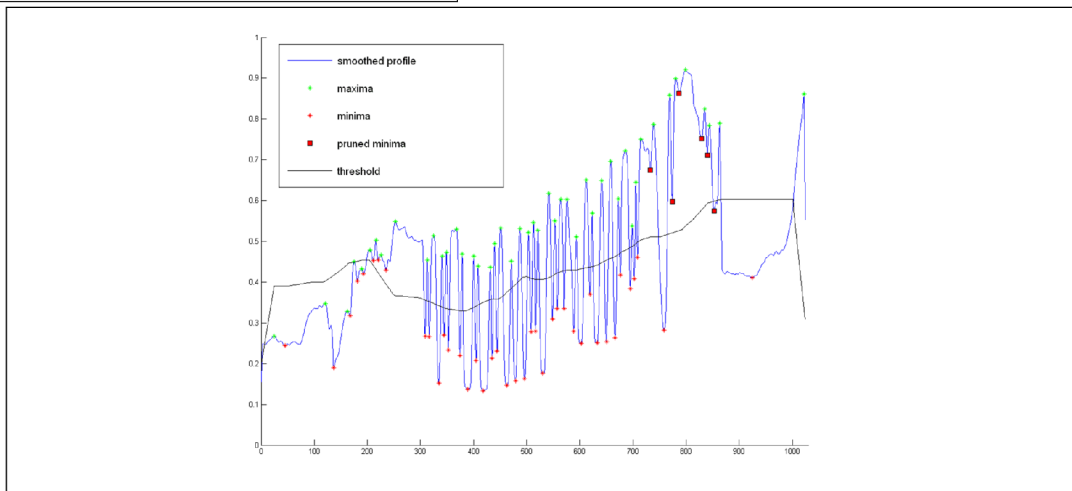


Šum

Dekódování obrazu s velkým šumem zaměřený na kód UPC je dobře popsán v *A Bayesian Algorithm for Reading 1D Barcodes* [15]. Využívá statistických metod pro detekci kódu, a tím je velice odolný vůči šumu. Dekódování pomocí tohoto algoritmu je popsáno v sekci Bayesovský algoritmus.

Computer Vision and Pattern recognition Group [16], která bude ještě dále zmíněna používá několikakrokové zpracování obrázku před samotným dekodováním. Prvním krokem je rozostření načteného řádku a převedení pomocí standardního algoritmu do stupňů šedi. V tomto obraze jsou vyhledány lokální maxima a minima, přičemž tmavá maxima a světlá minima se z obrázku odstraní, protože se může jednat hlavně o odlesky materiálu, případně

Obrázek 4.3: Dynamické prahování



o tmavý stín. Pro každou skupinu okolo lokálních minim/maxim je vypočítán dynamicky práh dle okolí. Tím je zaručeno, že se tenké čáry nepřevodou na opačný extrém. Na obrázku 4.3 jsou vidět minima, maxima a křivka dynamického prahu.

4.2 Lokalizace

Lokalizaci je možné provádět několika různými způsoby, nepoužívanější metodou je ROI, dalšími možnostmi jsou lokalizace pomocí automatického rozpoznání textu – OCR, vhodné zejména pro skenované kódy, spirálová metoda nebo vlnková transformace.

ROI – *Regions of Interest*

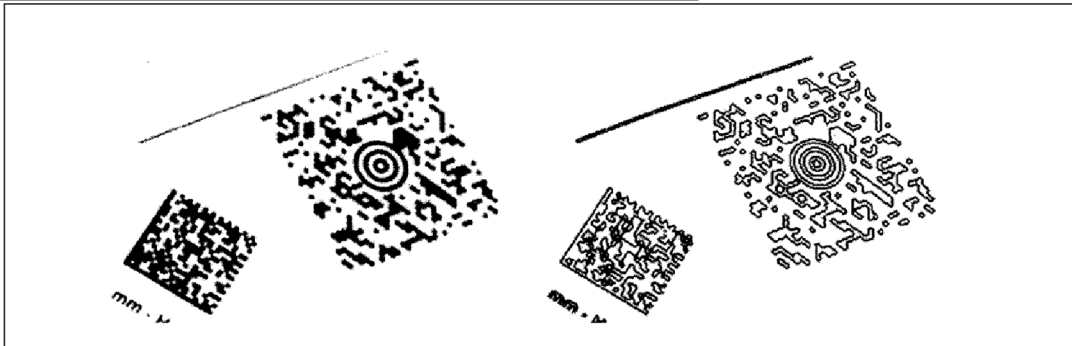
Pro zpracování 2D čárových kódů používají v [13] hledání „oblastí zájmu“ – ROI. Před samotným vyhledáním ROI je obraz zpracován pomocí Sobelova filtru o rozměru 3×3 , čímž se zvýrazní hrany v obraze (obrázek 4.4, vlevo originál, vpravo po úpravě filtrem). Na vyhledání oblastí zájmu pro obrázky o velikostech od 100 do 400 px se používá velikost vyhledávacího bloku 32. Na vyhledaných oblastech se provádí lokalizace kódu pomocí standardního hranového detektoru.

Asterisk Search

V článku Automatic recognition of Noisy Code-39 Barcode [6] je popsán další algoritmus pro lokalizaci kódu. Článek je zaměřen na zašuměné skenované formuláře. Lokalizace probíhá ve dvou bodech: lokalizace oblasti *asterisk* a vybrání oblasti s čárovým kódem. Tento algoritmus je principiálně velice podobný ROI.

Skenovaný obraz je prohledáván horizontálně, vertikálně a diagonálně, z důvodu různého natočení a pozice čárového kódu. Vyhledává se začátek a konec čárového kódu – tedy daná

Obrázek 4.4: Sobelův filtr aplikovaný na QR-code [13]



posloupnost černých čar. Pokud je posloupnost nalezena, označí se jako oblast asterisk. Pro vybrání oblasti s čárovým kódem je třeba porovnávat asterisky mezi sebou, zda patří ke stejnému kódu, což je v kódu dáno směrem čar, který musí být stejný. Všechny asterisky patřící k jednomu regionu se sloučí dohromady. Nyní může následovat proces dekódování.

Vlnková transformace – *Wavelet transform*

Vlnková transformace slouží k reprezentaci obrázků s několika rozlišeními uloženými v sobě (obrázek 4.5), postup najdeme ve velice dobře vypracovaném článku *1D Bar Code Reading on Camera Phones* [17]. Kód je lokalizován pomocí detekce vysokých frekvencí v obrázku, autoři článku uvádí, že se jedná o dokonalejší lokalizaci, než je *zero-crossing*.

Obrázek 4.5: Příklad aplikace vlnkové transformace



4.3 Dekódování

Proces dekódování je vždy aplikován na sekvenci znaků, která reprezentuje bílé a černé pruhy čárového kódu. Nejčastěji používaná je klasifikace dle tabulky.

Semínkové vyhledávání

Computer Vision and Pattern recognition Group z Rakouska publikovala robustní a rychlý algoritmus pro rozpoznání 1D čárových kódů UPC-A, EAN-8, ISBN-13 pomocí mobilního

telefonu [16]. Kód může být natočen či deformován perspektivou $\pm 15^\circ$, což je při focení integrovaným fotoaparátem v mobilním telefonu běžné. Lokalizace v tomto algoritmu vůbec neprobíhá. Předpokladem je, že střed obrázku pokrývá plochu čárového kódu, pokud se na tomto vzorku nepodaří kód rozpoznat, budou dále zkoušeny řádky nad a pod touto horizontální linií. Tím bude zabezpečen několikanásobný průchod obrázkem.

Prohledávání binarizovaného řádku probíhá z prostřed obrázku (semínko). Jako první se pokusí najít nejbližší čáru, vlevo nebo vpravo. Od takto nalezené startovací čáry pak vyhledává čáry a mezery na obě strany, které uloží jako sekvenci. Hranice kódu v takovéto sekvenci je pak nalezena pomocí sekvence čar pro start a stop znak čárového kódu. Dekódovací tabulka je předzpracována pomocí obrázků. Ze sekvence se určí minimální šířka čáry, která odpovídá jednomu modulu. Pomocí tohoto rozměru je možné již klasifikovat sekvence tak, aby je bylo možné vyhledat v tabulce, viz další sekce.

Klasifikace dle tabulky

Tento algoritmus navazuje na předzpracování [16]. Jak již bylo zmíněno zpracování obrazu spočívá ve skenování prostředního řádku obrazu a převedení do černé a bílé. Následně se hledá změna bílých a černých pruhů pro detekování okrajů čárového kódu. Načtené pruhy se klasifikují jako číslice, které se podle standardní kódové tabulky dekodují. Pokud je hypotéza přidělení číslic správná a vyjde i výpočet kontrolní číslice, tak výsledek je pravděpodobně správný.

Algoritmus implementovaný v Matlabu byl testován na 1000 obrázcích, přičemž se pohybovala úspěšnost od 90,5 % do 99,2 % podle rozlišení obrázku (640×480 do 2592×1944 pixelů).

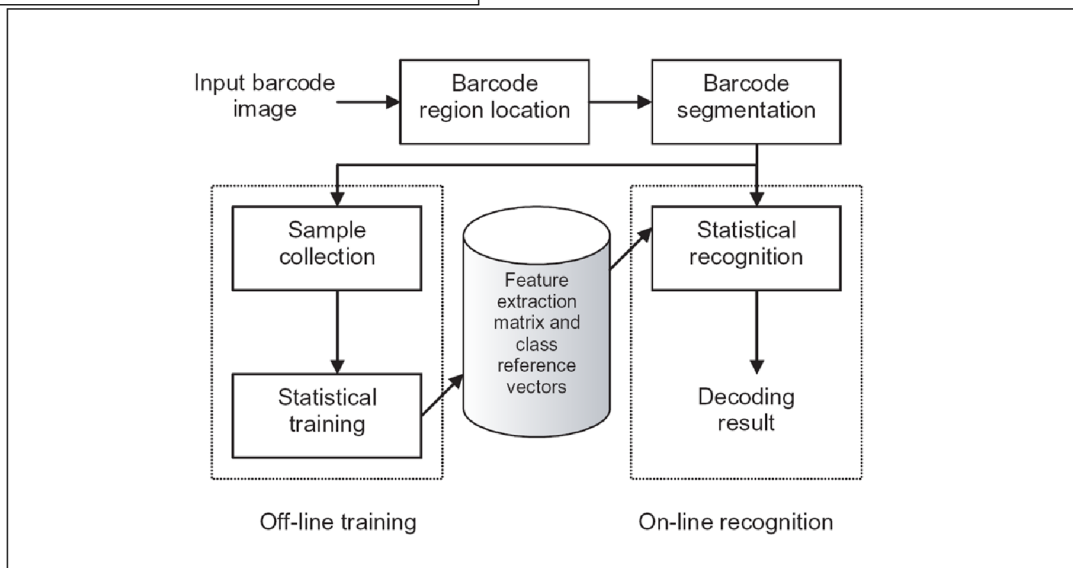
Učení s učitelem

Již ve zmíněném článku [17] je použito dekodování na základě učení s učitelem. Celý proces dekodování je načrtnut na obrázku 4.6, jsou zřetelné tři části zpracování, a to: vlnková transformace pro lokalizaci, učení se typu kódu ze známé trénovací sady, statistické rozpoznání čárového kódu. Trénovací sada musí obsahovat správně označené kódy, aby byl algoritmus schopen natrénovat co největší množinu různých obrázků. Pro statistické rozpoznání je použita modifikovaná GLVQ metoda. Algoritmus je opravdu robustní a velice účinný.

Bayesovský algoritmus

Dekodování založené na Bayesovském modelu provádí Tekin a Coughlan [15] na Smith-Kettlewell Institutu v Kalifornii. Jejich řešení je robustní, odolné vůči šumu, nerovnoměrnému osvětlení, geometrické a perspektivní distorzi, umí dekodovat i obrazy s velmi malým rozlišením.

Obrázek 4.6: Postup dekódování [17]



V článku jsou popsány dva různé modely, přičemž je pro výslednou implementaci zvolen jednodušší, který prochází obrazem pouze jednou. Dekódovací program je možné používat v mobilním telefonu s vestavěným fotoaparátem, ale fotografie musí obsahovat pouze čárový kód s jeho ochrannou zónou.

4.4 Zhodnocení

Z uvedených výtahů z článků vyplývá, že obraz je nejprve třeba předzpracovat tak, aby mohly být použity jednoduché metody pro dekódování. Samotný převod jednoho řádku obrazu může probíhat několika způsoby. Algoritmy zaměřené na použití v mobilních zařízeních musí být hodně robustní a odolné na šum, protože vestavěné fotoaparáty nedosahují takových kvalit jako průmyslové CCD kamery nad přepravními pásy.

Je třeba také brát v úvahu natočení kódu, osvětlení, jeho perspektivu a velikost, pokud se nejedná např. o rozpoznávání čárových kódů na skenovacích formulářích, kde je třeba se zaměřit pouze na natočení papíru ve skeneru. Vybrané metody pro zpracování obrazu budou popsány v následující kapitole, stejně tak i vlastní návrh aplikace.

Kapitola 5

Implementace

Úkolem implementace bylo vytvoření funkčního prototypu rozpoznávání čárového kódu. Tato kapitola shrnuje návrh samotné aplikace a postupy zpracování dílčích úkolů při běhu programu – předzpracování obrazu, segmentace, dekódování a kontrola čtení.

Pro potřeby testování zpracování obrazu byla zvolena knihovna OpenCV, se kterou se dobře pracuje v programovacím jazyce C++, který byl zvolen pro implementaci. Program byl vytvářen v běhovém prostředí Microsoft Visual Studio 2008, které velmi usnadňuje ladění programu. Testovací skript byl vytvořen v jazyce Python, pro svojí jednoduchost při práci s adresáři a funkcemi operačního systému. Popis tříd nepovažuje autorka za důležitý, je možné ho najít v programové dokumentaci na přiloženém datovém nosiči.

5.1 Návrh aplikace

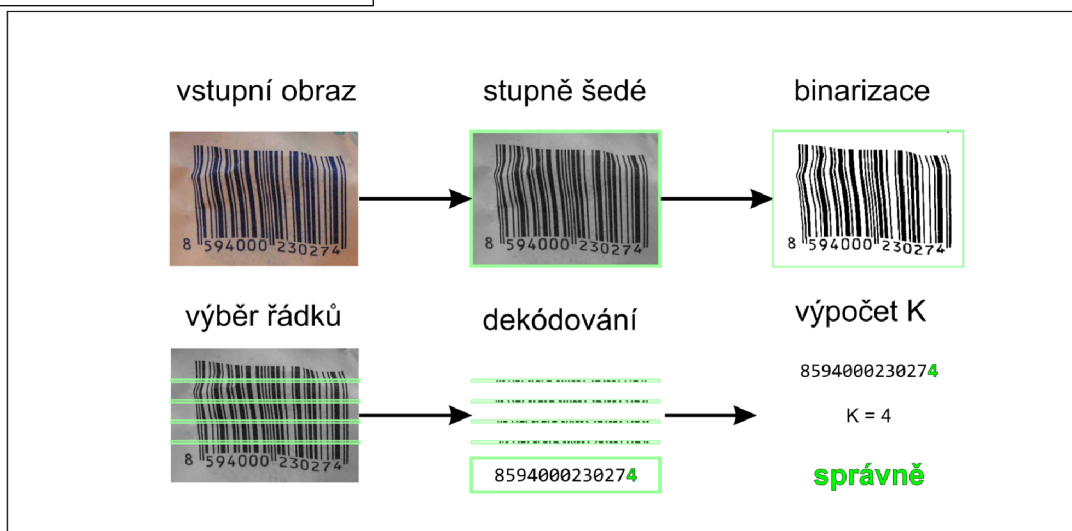
Nejpoužívanějšími kódy pro označení zboží v ČR jsou kódy EAN-UCC, proto autorka vybrala právě tyto typy kódů pro implementaci. Z variant EAN-UCC jsou masivně využívány kódy EAN-8 a EAN-13. Pro porovnání s kódy používanými v USA byl vybrán kód UPC-A, který je téměř ekvivalentní kódu EAN-13. Další typy kódů nemělo smysl implementovat, vzhledem k podobnosti postupu dekódování s již implementovanými kódy.

Dekódování je zaměřeno především na správné zpracování načteného kódu, nikoliv na jeho lokalizaci. Aplikace je vhodná na pro obrazy pořízené z fotoaparátu či skeneru, které nejsou nerovnoměrně osvětlené. Příkladem budiž skenovací formuláře, kde se kód očekává vždy ve stejném místě, natočený maximálně o $\pm 15\%$.

Na obrázku 5.1 je zobrazen postup při dekódování čárového kódu z obrazu. Nejprve je načten obrázek ze vstupního souboru. Obraz se převede do stupňů šedi a následně se vypočítá práh pro převedení do černobílé reprezentace. V obrázku se vybere standardně 10 řádků, na které bude aplikován segmentační algoritmus. Segmentace je řízena konečnými automaty, kde se v sekvenci černých a bílých pixelů vyhledává definovaná sekvence pro start a stop symbol čárového kódu. Pokud je nalezený počáteční i koncový znak, pokusí se program načíst správný počet datových znaků pro dekódovací algoritmus. Načtená data se

ze sekvencí 0 a 1 převedou na poměry tak, aby mohly být porovnány s tabulkou, jak je ukázáno v příkladě 5.1. Dekódovací algoritmus vrací řetězec symbolu čárového kódu. Nyní je třeba z přečtených čísel vypočítat kontrolní číslici, aby mohla být porovnána s přečtenou kontrolní číslicí. Pokud je porovnání shodné, pravděpodobně se dekodování zdařilo a program vrátí řetězec čárového kódu.

Obrázek 5.1: Návrh aplikace



Autorka vytvořila velikou testovací sadu (přes 1000 obrázků) pro vybrané 3 typy kódů, které jsou detailně popsány v kapitole 6. Byla snaha najít již vytvořenou a testovanou sadu obrázků jinými algoritmy, ale žádný z nalezených odkazů na databázi obrázků bohužel nefungoval.

5.2 Předzpracování obrazu

Program načítá obrázek ze souboru daného parametrem při spuštění programu. Podmínkou pro další úspěšné zpracování je bílé, či hodně světlé pozadí. Vlevo a vpravo od symbolu čárového kódu v obrázku se nesmí vykytovat nic jiného než ochranná zóna kódu.

Dle možností knihovny OpenCV je možné načítat obrázky typu `jpg`, `jpeg`, `png`, `bmp` a `tif`. Obrázek je uložen do paměti a převeden do stupňů šedi. Převod probíhá dle standardního algoritmu (algoritmus 5.1).

Pro převod na binarizovaný obraz bylo vyzkoušeno několik možných algoritmů. Nejlepší výsledky byly očekávány od adaptivního prahování implementovaného v knihovně OpenCV. Zde nastal problém při převodu „dokonalých“ čárových kódů, které obsahovaly pouze 2 hodnoty barev, kde algoritmus nepracoval správně. Je proto implementován algoritmus pro prahování s adaptivním prahem předpočítaným pro celý obraz (algoritmus 5.2). Převode šedotónový obraz na reprezentaci 0 a 1, přičemž čáry (černé) jsou nuly a mezery (bílé) jsou jedničky.

Algoritmus 5.1: Převod barevného obrazu na šedotónový

1. $C(x, y)$ je vstupní obraz, $G(x, y)$ je výstupní šedotónový obraz.
2. Pro každý pixel barevného obrazu $C(x, y)$:

$$G(x, y) = 0,229 \cdot C(x, y).red + 0,587 \cdot C(x, y).green + 0,114 \cdot C(x, y).blue$$

Algoritmus 5.2: Převod šedotónového obrazu na černobílý

1. $G(x, y)$ je vstupní obraz, $B(x, y)$ je výstupní binární obraz, w je šířka obrazu, h je výška obrazu v pixelech.
2. Vypočítej průměrnou intenzitu pixelu:

$$threshold = \frac{\sum_{i=0, j=0}^{i=w, j=h} G(i, j)}{w \cdot h}$$

3. Pro každý pixel obrazu vstupního obrazu $G = (x, y)$:

$$B(x, y) = \begin{cases} 0(white), & G(x, y) < threshold \\ 1(black), & G(x, y) \geq threshold \end{cases}$$

5.3 Výběr řádků

Nyní je již obraz připraven pro další zpracování. Program vybere dle parametru daný počet řádků, které budou segmentovány a dekodovány. Tento výběr je proveden dle algoritmu 5.3.

5.4 Segmentace

Zde se program větví, dle vstupních parametrů se zavolá buď vyhledání EAN–8, EAN–13 nebo UPC–A. Pro všechny typy kódů je vyhledání postaveno na stejném principu. Pomocí konečných automatů se vyhledávají sekvence 0 a 1. Nejprve se vyhledá startovací znak, poté se odečte příslušný počet čar, definující první polovinu datových znaků, zkontroluje se, zda následuje oddělovací znak dat, přečte se opět příslušný počet čar pro druhou polovinu datových znaků a zkontroluje, zda je na konci znak pro konec.

U kódů EAN-UCC je vždy jedna číslice kódována čtyřmi čarami (2 bílé a 2 černé). Dle délky kódu se čtou 4, resp. 5, resp. 6, číslice pro první polovinu kódu EAN–8, resp. UPC–A, resp. EAN–13. Stejně tak se provede pro druhou polovinu dat mezi středovým oddělovacím

Algoritmus 5.3: Výběr řádků

1. $G(x, y)$ je vstupní obraz, w je šířka obrazu, h je výška obrazu v pixelech, c je počet řádků.
2. Vypočítej ochranou zónu o jako 10 % z výšky obrazu h .
3. Vyber počet řádků c tak, že číslo vybraného řádku k bude:

$$k = o + 6i \cdot \frac{o}{c}$$

přičemž $i \in \langle 1, c \rangle$.

znakem a koncovým znakem.

Tímto postupem jsme získaly sekvence 0 a 1 pro levou a pravou část čárového kódu. Aby bylo možné jednoduše tyto čísla porovnávat, je třeba stanovit, jak se budou kódovat a ukládat již samotné tabulky pro dekódování.

5.5 Uložení kódových tabulek

Číslice pro EAN-UCC mají vždy 7 modulů na jedno číslo, přičemž jedno číslo je kódováno 4 čarami. Kódová tabulka 3.1 pro EAN-UCC je uložena v paměti jako sekvence čtyř desetinných čísel. Tato čísla udávají poměr počtu modulů jedné čáry k celkovému počtu modulů pro jedno číslo.

Příklad 5.1: Výpočet poměru pro jedno číslo

Kód čísla 1 v sadě C	1	0	111	00
Počty modulů 0 a 1	1	1	3	2
Uložené poměry v tabulce	0.1428	0.1428	0.4285	0.2857

Hodnoty poměrů se vzhledem k délce kódu mohou vyskytnout maximálně 4, jsou předpočítány a namapovány do paměti pro rychlý přístup při vyhledávání.

5.6 Dekódování čárového kódu

Předzpracování sekvence 0 a 1 jsou pro všechny typy čárových kódů stejné. Kód může mít různé měřítko, proto je poměr vypočítán z šířky čar. Algoritmus 5.4 ukazuje, jak poměr vypočítat. Vstupem algoritmu je vektor čtyř sekvencí 0 a 1 pro jedno číslo.

Nyní následují sekce popisu porovnání poměrů s tabulkou pro každý typ čárového kódu zvlášť. To, dle jakých sad je typ kódován, je uvedeno v jedné z předchozích kapitol (3).

Algoritmus 5.4: Výpočet poměru pro jedno číslo

1. Nechť $u(a_1, a_2, a_3, a_4)$ je vstupní vektor, kde a_i je i -tá sekvence, $a_i \in \{0\}^+ \cup \{1\}^+$, $v(x_1, x_2, x_3, x_4)$ je výstupní vektor poměrů, $x_i \in \mathbb{R}$.
2. Pro všechny a_i , kde $i \in \langle 1; 4 \rangle$ vypočítej poměr x_i k celkové délce sekvence:

$$x_i = \frac{a_i}{\sum_{i=1}^4 |a_i|}$$

Porovnání je realizováno pomocí vzdálenosti dvou vektorů (algoritmus 5.5), kde pro každé porovnání se vypočítá vzájemná vzdálenost, vybere se to číslo, které má s hledanými poměry nejmenší vzdálenost, dále bude označována jako nejbližší.

Algoritmus 5.5: Vzdálenost dvou čísel

1. Nechť $d(p_1, p_2, p_3, p_4)$ je vstupní vektor poměrů, $s(r_1, r_2, r_3, r_4)$ je referenční vektor poměrů v tabulce, e je vzdálenost čísel.
2. Vypočítej e jako:

$$e = |ds| = \sqrt{(p_1 - r_1)^2 + (p_2 - r_2)^2 + (p_3 - r_3)^2 + (p_4 - r_4)^2}$$

5.6.1 EAN–8

Dekódování symbolu EAN–8 je ze zvolených nejjednodušší (algoritmus 5.6). První polovina dat je kódována dle sady A a druhá polovina dat (včetně kontrolní číslice) dle sady C (tabulka 3.1).

Algoritmus 5.6: Dekódování EAN–8

1. Nechť D_1 jsou první 4 znaky (levá polovina), D_2 jsou druhé 4 znaky (pravá polovina) čárového kódu.
2. Pro každé číslo d_1 v D_1 :
 - vyhledej nejbližší číslo s_1 k číslu d_1 v sadě A.
3. Pro každé číslo d_2 v D_2 :
 - vyhledej nejbližší číslo s_2 k číslu d_2 v sadě C.
4. Sestav výsledný řetězec $S = s_1 + s_2$.

5.6.2 UPC–A

Kód UPC–A je kódován shodně jako kód EAN–13, jak je zřejmé z kapitoly 3. První číslo je 0, tedy je první polovina kódu kódována podle sekvence sad AAAAAA (tabulka 3.5). Dekódování je tedy shodné s EAN–8, přičemž je dekodováno dvanáct číslic místo osmi (algoritmus 5.7).

Algoritmus 5.7: Dekódování UPC–A

1. Nechť D_1 je prvních 6 znaků (levá polovina), D_2 je druhých 6 znaků (pravá polovina) čárového kódu.
2. Pro každé číslo d_1 v D_1 :
 - vyhledej nejbližší číslo s_1 k číslu d_1 v sadě A.
3. Pro každé číslo d_2 v D_2 :
 - vyhledej nejbližší číslo s_2 k číslu d_2 v sadě C.
4. Sestav výsledný řetězec $S = s_1 + s_2$.

5.6.3 EAN–13

Nejsložitější je dekodování symbolu EAN–13, kde je potřeba zjistit první číslici s_0 , která je předřazena před kódem a není v datových znacích přímo zakódována. Dle prvního znaku se určuje, v jakých sadách jsou zakódovány číslice vyskytující se v první polovině kódu. Je tedy třeba vyzkoušet všechny možnosti dekodování první poloviny dat dle tabulky 3.5 a vybrat takovou první číslici, kde je součet vzdáleností, pro všech šest čísel, od referenčních čísel v tabulce minimální.

5.7 Porovnání kontrolní číslice

Správnost dekodování je možné ověřit pomocí datové kontrolní číslice (algoritmus 5.9). Z dekodovaných znaků čárového kódu se dle typu symbolu vypočte kontrolní číslice. Výpočty jsou uvedeny v kapitole 3. Provede se porovnání vypočtené a dekodované kontrolní číslice. Pokud se číslice shodují, je sekvence znaků validní a program vrátí výsledek dekodování.

5.8 Hromadné zpracování

Hromadné zpracování dat je umožněno pomocí skriptu napsaném v jazyce Python. Tento skript spouští kompilovaný program C++. Skript je možné použít pouze na platformě unixového typu, vzhledem ke špatné podpoře knihovny OS ve Windows.

Algoritmus 5.8: Dekódování EAN-13

1. Nechť D_1 je prvních 6 znaků čárového kódu (levá polovina), D_2 je druhých 6 znaků (pravá polovina) čárového kódu, e_i je součet vzdáleností od referenčních čísel pro D_1 .
2. Pro všechny první číslice i , $i \in \langle 0; 9 \rangle$:
 - Pro každé číslo d_1 v D_1 :
 - Vyhledej nejbližší číslo s_1 podle dekodovacího řetězce pro i .
 - Připočti k celkové vzdálenosti e_i vzdálenost d_1 od s_1 .
3. Vyber s_1 takové, kde e_i je minimální.
4. Dle kódování s_1 najdi hodnotu první číslice s_0 .
5. Pro každé číslo d_2 v D_2 :
 - vyhledej nejbližší číslo s_2 k číslu d_2 v sadě C.
6. Sestav výsledný řetězec $S = s_0 + s_1 + s_2$.

Algoritmus 5.9: Porovnání kontrolních číslic

1. Vypočti kontrolní číslici K z dekodovaného řetězce S .
2. Porovnej vypočtenou číslici K a přečtenou číslici $S[8]$ (osmá číslice dekodovaného řetězce).
3. V případě shody vrať dekodovaný řetězec S .

Na vstupu skriptu jsou dva parametry, a to typ čárového kódu a cesta k obrazovým datům. Podrobný synopsis je možné nalézt v dodatku B. Aby bylo možné vypisovat statistiky úspěšnosti pro daný test, je třeba pojmenovat soubor s obrázkem tak, aby reprezentoval pouze hodnotu čárového kódu, tedy řetězec čísel.

Výstupem skriptu je výpis testu pro každý zadaný obrázek. Skript umožňuje zadat jméno adresáře, sám si načte všechny obrázky i z podadresářů. Správný čárový kód (uložený jako název souboru), který je v obrázku a dekodovaný čárový kód z obrazu jsou vyhodnoceny. Pokud je dekodování správné, skript vypíše [OK], pokud ne, je vypsána hláška, na jakém prvním problému program selhal.

V programu je možné nastavit parametr show, který říká, že bude ke každému souboru zobrazeno okno s načteným obrázkem, kde v horní části obrázku bude vidět šedotónový a binarizovaný řádek dat. Jako poslední se vypisují statistiky úspěšnosti rozpoznání čárového kódu na dané testovací sadě (adresáři).

5.9 Webové rozhraní

Prototyp aplikace je možné vyzkoušet přes webové rozhraní, které je umístěné na adrese <http://barcode.lwg.cz>. Webovou aplikaci se nepodařilo umístit na školní server z důvodu zákazu spouštění skriptů z formuláře. Je použit jazyk Python a CGI skript, který spouští na serveru program vytvořený v C++.

Aplikace očekává na vstupu typ čárového kódu a obrázek s čárovým kódem. Po spuštění webová stránka zobrazí obrázek s kódem a jeho přečtenou hodnotu. Pokud se obrázek nepodaří přečíst, vrátí chybu s popisem, kde dekodování selhalo.

Kapitola 6

Vyhodnocení

V této kapitole jsou diskutovány výsledky aplikace na různých testovacích sadách obrázků. Pro každou sadu je zobrazena úspěšnost dekodování a jsou vysvětleny důvody špatného rozpoznání kódu. Testovací sady jsou seřazeny podle stupně rozpoznání od nejlehčích po nejtěžší, obsahují vždy kódy EAN-8, EAN-13 a UPC-A.

Nejdůležitějším prvkem kapitoly jsou analýzy robustnosti programu, které byly provedeny na třech parametrech, a to na velikosti čárového kódu, rotovaných obrázcích a rozostření.

6.1 Testování

6.1.1 Generované kódy

První testovací sada obsahuje vždy pro každý typ kódu 30 obrázků, které jsou většinou černobílé, vygenerované. Mají správnou ochrannou zónu. Rozlišení obrázků je v rozmezí 85×45 pixelů až 688×1028 pixelů. Ukázka kódů je na obrázku 6.1.

Obrázek 6.1: Ukázky syntetických obrazů EAN-8, EAN-13, UPC-A



Program na těchto testovacích datech pracuje velice dobře, při rozlišení pod 300 pixelů nastává problém s neostrým obrázkem. Pokud je rozlišení větší jak 400 pixelů na šířku, je kód rozpoznáván správně. Úspěšnost na této datové sadě se pohybuje kolem 85 %. Detailní statistika je uvedena v tabulce 6.1.

Tabulka 6.1: Úspěšnost dekodování pro generované kódy

typ kódu	EAN-8	EAN-13	UPC-A
počet vzorků	30	30	30
úspěšnost	90 %	86 %	83 %

6.1.2 Fotografované kódy

Další testovací sadou jsou vyfocené čárové kódy (obrázek 6.2), které mají minimální natočení či zkreslení, rovnoměrné osvětlení, dodrženu ochrannou zónu. Rozlišení obrázků se pohybuje od 370×110 do 1000×768 pixelů.

Obrázek 6.2: Ukázky fotografovaných kódů EAN-8, EAN-13, UPC-A

Dekodování je stále účinné, pohybuje se v rozmezí 73–96 % (tabulka 6.2). Problémy s dekodováním nastávají u neostrých obrázků, případně u obrázků se špatným barevným vyvážením, osvětlením nebo s porušenou ochrannou zónou.

Tabulka 6.2: Úspěšnost dekodování pro fotografované kódy

typ kódu	EAN-8	EAN-13	UPC-A
počet vzorků	30	50	30
úspěšnost	93 %	96 %	73 %

6.1.3 Natočené a barevné kódy

Pro EAN-8 bylo nasbíráno 23 obrázků, které nejsou ořezané na ochrannou zónu nebo jsou natočené, různě barevné. EAN-13 má opět 50 obrázků, většina je natočená o $\pm 20^\circ$. Kódové čáry jsou různě barevné, ale splňují podmínky kontrastu čar na pozadí. Kódy UPC-A se většinou vyskytují na elektronice, proto je jejich pozadí bílé, a dále na CD, kde je okolí kódu barevné, těchto typů čárových kódů je nedostatek.

Program selhává na obrázcích, kde je tmavé okolí za ochranou zónou. Tento problém by vyřešila lokalizace kódu v obraze. Úspěšnost čtení natočených kódů je kolem 80 % (tabulka 6.3), přičemž natočení nesmí přesáhnout úhel α který je dán tangensem výšky kódu k jeho šířce. To zaručí, že alespoň jeden řádek obrázku projde přes celý čárový kód a při dostatečném počtu průchodů ho bude možné dekodovat.

Obrázek 6.3: Ukázky natočených a barevných kódů



Tabulka 6.3: Úspěšnost dekodování pro natočené a barevné kódy

typ kódu	EAN-8	EAN-13	UPC-A
počet vzorků	23	50	11
úspěšnost	60 %	72 %	90 %

6.1.4 Poškozené kódy

Z důvodu nedostatku kódů UPC-A a EAN-8, byla vytvořena testovací sada pouze pro EAN-13, čítající 50 obrázků. Obrázky jsou různě poškozené, např. ustřížené, zalepené, překryté, zmačkané, atd. (obrázek 6.4).

Obrázek 6.4: Ukázky poškozených kódů



Díky tomu, že algoritmus provádí průchod čtení několikrát přes celý obraz, je ustřížený kód přečten správně. Stejně platí i pro odlesky a nerovnosti v kódu, pokud jsou jenom na nějaké z jeho částí. Úspěšnost v tomto případě je 73 %.

6.1.5 Porovnání

Pro úplnost uvedme porovnání úspěchu dekodování všech testovacích sad (tabulka 6.4). Z tabulky je zřejmé, že s klesající kvalitou kódů rapidně klesá i úspěšnost. Tyto výsledky nelze brát dramaticky, vzhledem k nerovnoměrným testovacím sadám, slouží opravdu pouze k testovacím účelům a ukázání, na jakých typech kódů algoritmus funguje. Rovnoměrné testy jsou popsány v následující sekci.

6.2 Robustnost

Aby bylo možné určit robustnost programu, byly vytvořeny sady po 10 obrázcích pro generované kódy a 10 obrázcích pro nafocené kódy. Byl zvolen kód EAN-13, který má nejpropra-

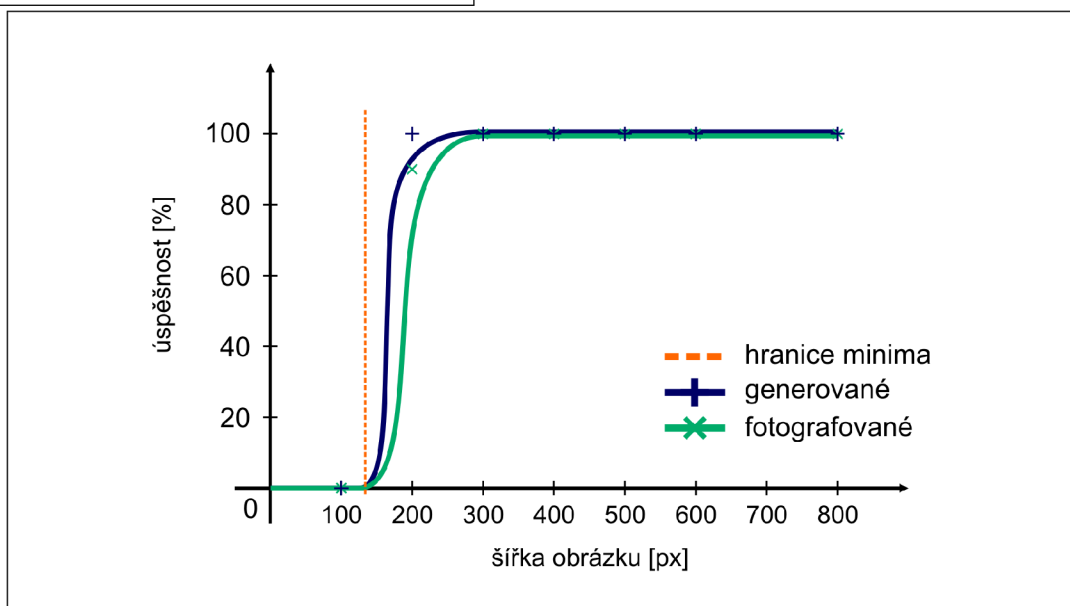
Tabulka 6.4: Úspěšnost dekodování

typ kódu	generované	fotografované	deformované	barevné
EAN-8	90 %	93 %	60 %	–
EAN-13	86 %	96 %	72 %	73 %
UPC-A	83 %	73 %	90 %	–

covanější způsob dekodování. Na sadu byly aplikovány opakovaně transformace deformující obraz, jako je velikost rozlišení, rotace a rozostření.

6.2.1 Velikost rozlišení

Program byl testován dvakrát na 10 sadách, kde obrázky měly zprvu šířku 2400 pixelů a v dalších krocích se jejich velikost vždy zmenšovala až na 100 pixelů. První test proběhl na generovaných kódech, druhý test na kódech focených (obrázky jsou umístěny v adresářích `barcodes/test5-velikost`). V grafu 6.5 jsou znázorněny výsledky pro šířku do 800 pixelů, při větší šířce kódu je výsledek konstantní (100 %) a nemá velký význam ho analyzovat.

Obrázek 6.5: Závislost na šířce kódu

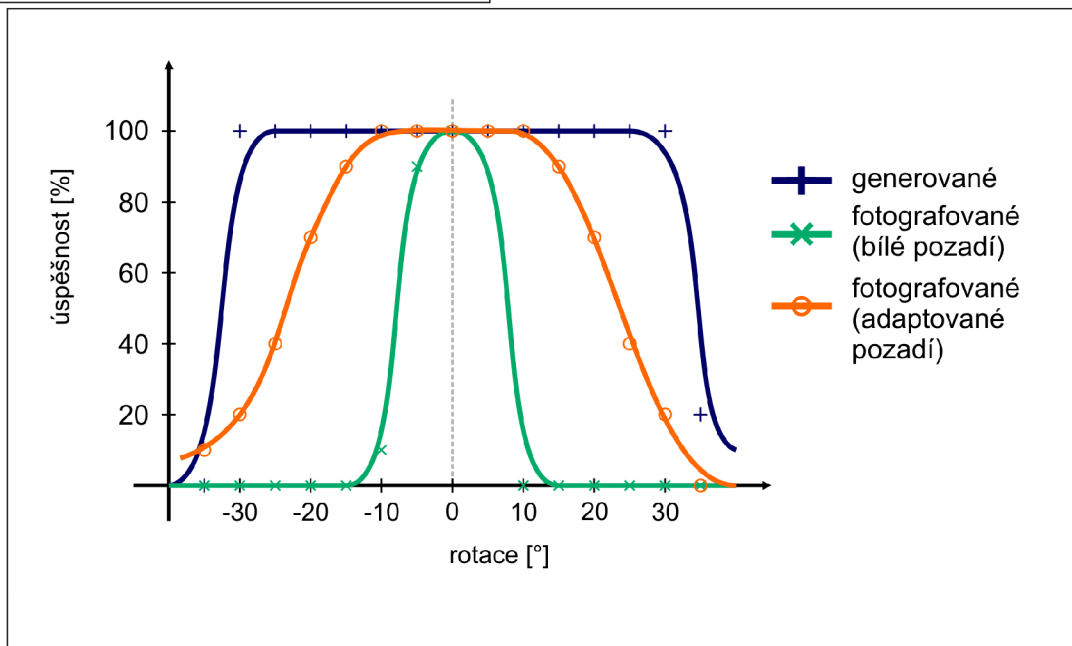
Vypočteme si minimální šířku kódu pro správné rozpoznání, víme že to je 7 modulů na číslo, přičemž EAN-13 má 13 čísel, tedy 91 modulů. Je třeba ještě připočítat 3 moduly pro znak začátku, 3 pro znak konce a 5 pro dělicí znak. Celkem tedy 102 modulů, což odpovídá minimálně 102 pixelům, přičemž je třeba počítat s ochranou zónou kódu, která se vždy v obrázku vyskytuje. Tedy obrázek menší než cca 150 pixelů nelze rozpoznat.

Algoritmus je pro generované kódy robustní. Poradí si s kódy o šířce 200 pixelů. Velké rozlišení obrázku nečiní algoritmu nejmenší potíže.

6.2.2 Rotace kódu

Další testování bylo prováděno na kódech, které byly postupně rotovány o $\pm 5^\circ$ na každou stranu v rozmezí 0° až 35° od vodorovné osy. Byly použity 3 sady pro generované kódy, fotografované kódy s bílým pozadím a fotografované kódy s pozadím stejným jako světlá barva čar (obrázky jsou umístěny v adresářích `barcodes/test4-rotace`). Velikost obrázků je 1000 pixelů na šířku. Úspěšnost je znázorněna v grafu 6.6.

Obrázek 6.6: Závislost na rotaci kódu



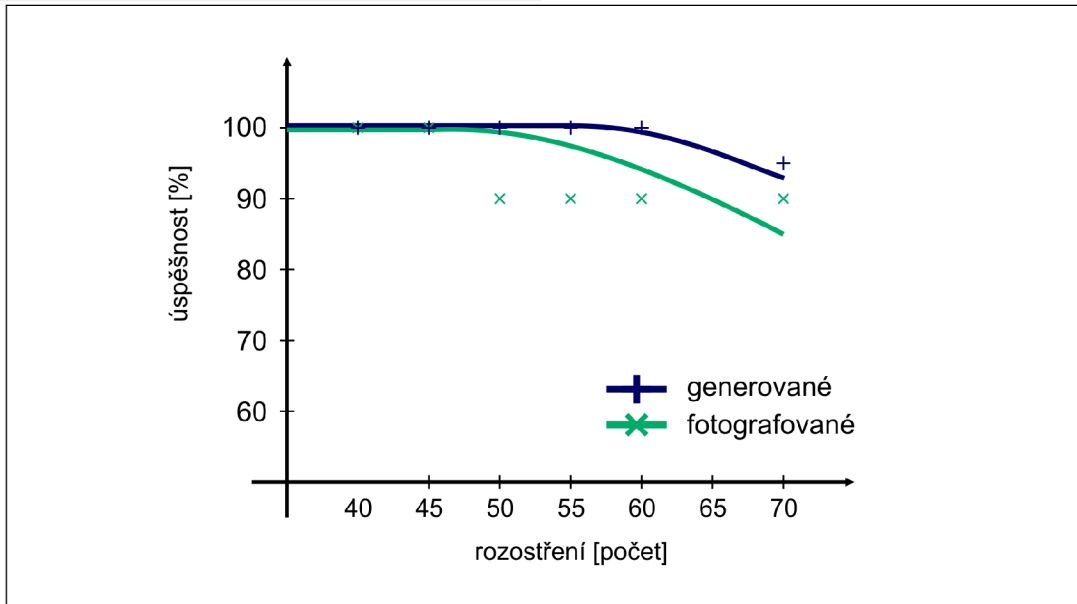
Na generovaných kódech je úspěšnost až do 25° natočení stoprocentní. Generované kódy mají vždy standardní velikost (viz kapitola 2), oproti tomu fotografované kódy mají již velikost malou nebo standardní, proto je úspěšnost na fotografovaných nižší. Z toho vyplývá, že natočení nesmí přesáhnout úhel α , který je dán tangensem výšky kódu k jeho šířce. To zaručí, že alespoň jeden řádek obrázku projde přes celý čárový kód.

Fotografované kódy s bílým pozadím, které vzniklo rotací obrazu v grafickém editoru, způsobuje problémy při binarizaci, protože je do obrázku vneseno syntetické osvětlení. Při nahrazení bílé barvy pozadí barvou světlých čar se úspěšnost okamžitě zvýšila a je uspokojivá.

6.2.3 Rozostření

Jako poslední bylo vyzkoušeno, jak je algoritmus odolný vůči rozostření kódu. Na obrázky generované i foceně bylo aplikováno rozostření vždy $5\times$ po sobě a to ve 13 krocích (obrázky jsou umístěny v adresářích `barcodes/test6-ostrost`). Velikost obrázků je 1000 pixelů na šířku.

Obrázek 6.7: Závislost na rozostření kódu



Úspěšnost dekódování na generovaných kódech nebyla rozostřením téměř ovlivněna. Některé z fotografovaných kódů byly již rozmazané, přidání další neostrosti způsobilo nemožnost dekódování. Výraznější rozostření způsobí změnu barvy světlých čar více k tmavým a dojde již k chybě určení prahu a následné binarizaci.

Kapitola 7

Závěr

V předchozích kapitolách je popsáno dekódování čárového kódu v obraze. Stav práce dle autorčina názoru splnil všechny požadované body zadání.

Statistiky hovoří za vše. Při dobrém rozlišení obrázků (šíře minimálně 200 px) program nemá potíže kód rozpoznat, pokud je symbol čárového kódu rozmazaný, tak při dobrém rozlišení není problém s rozpoznáním. Natočení kódu v obraze je možné pouze do definovaného úhlu, který lze vypočítat z rozměrů čárového kódu. Kód poškozený po celé šířce či silně rozmazaný, program většinou nerozpozná.

Program umí dekódovat čárový kód typu EAN-8, EAN-13 a UPC-A, stejně tak by bylo možné dekódovat kód UPC-E, Kódy EAN a UPC se liší pouze zápisem numerických znaků pod a vedle symbolu čárového kódu. Prvně dekódovaný byl EAN-8, který je nejjednodušší, následovalo rozšíření na EAN-13, u kterého se vyskytla komplikace při čtení prvního znaku, který není součástí datového symbolu čárového kódu. Tento problém byl vyřešen jednoduše, zkoušením všech kombinací první číslice. Současně byl implementován algoritmus pro čtení kódu UPC-A. Úspěšnost algoritmu byla zvýšena několikanásobným pokusem čtení kódu v jenom obraze, odhadem se jednalo o 15 %. Dekódování EAN-13 bylo stále neúspěšné, proto byl algoritmus vylepšen o výpočet minimální chyby při porovnávání přečtených vzorků s tabulkou. Tímto úspěšnost vzrostla o 20 %. Jako poslední bylo implementováno adaptivní prahování, které zvýšilo úspěšnost na fotografovaných kódech o 15 %.

Předzpracování obrazu je provedeno relativně jednoduchým, ale účinným způsobem. Rozšíření algoritmu o předzpracování obrazů s nerovnoměrně osvětlenými kódy, případně jakkoli otočenými by zvýšilo jeho efektivitu a bylo by možné ho porovnávat s jinými již publikovanými algoritmy.

Jednoduchým rozšířením práce může být implementace dekódování ISBN a ISSN sestrojených pomocí dodatkového symbolu připojeného ke kódu EAN-13. Zde by se musela vytvořit další třída, která by implementovala nový druh kódu, a především další segmentace pro určení dodatkového symbolu.

Dále je možné implementovat vyhledávání čárového kódu v obraze, čímž by algoritmus

nabyl na robustnosti. Zde by bylo potřeba se zaměřit na časovou náročnost algoritmu a dobře ho optimalizovat.

Jako poslední je možné diskutovat dodržování norem při výrobě čárových kódů na obalech výrobků. Ze zkušeností plyne, že ne všechny výrobky splňují normu a proto je program nemusí dekodovat. Jedná se především o barevnost čárových kódů (např. inverzní barvy, které nejsou dovolené) a dodržování klidové zóny okolo symbolu čárového kódu.

I přes to, že je program koncipován bez lokalizace kódu v obraze, byla ověřena správnost návrhu aplikace a její prototyp funguje velmi dobře.

Literatura

- [1] Leonardo technology: Čárové kódy, 2D kódy a RFID [online]. http://www.lt.cz/index.php?option=com_content&task=view&id=156&Itemid=148, květen 2010.
- [2] Barcodes, S.: EAN Barcode Labels [online]. <http://www.officialeancode.com/>, květen 2010.
- [3] Benadiková, A.; Weinlich, S.; Mada, S.: *Čárové kódy – automatická identifikace*. Praha: GRADA, 1994, ISBN 80-85623-66-8, 272 s., <http://www.volny.cz/alef-cz/ck.htm>.
- [4] Blahut, R. E.: *Theory and practice of error control codes*. Addison-Wesley Pub. Co., 1983, ISBN 0201101025.
- [5] Dočekal, D.: Budoucnost a současnost čárových kódů pro mobily [online]. <http://www.lupa.cz/clanky/budoucnost-a-soucasnost-carovych-kodu-pro-mobily/>, červenec 2008.
- [6] Fang, X.; Wu, F.; Luo, B.; aj.: Automatic Recognition of Noisy Code-39 Barcode. In *Proceedings of the 16th International Conference on Artificial Reality and Telexistence-Workshops*, China: IEEE Computer Society, 2006, 0-7695-2754-X/06.
- [7] GS1: Čárové kódy a identifikace [online]. <http://www.gs1cz.org>, leden 2009.
- [8] IDAutomation.com: IDAutomation Barcode FAQ and Tutorial [online]. <http://www.idautomation.com>, listopad 2009.
- [9] Český normalizační institut: *ČSN EN 1556: Čárové kódy – Terminologie*. Praha, duben 1999, 44 s.
- [10] Microsoft: High Capacity Color Barcodes [online]. <http://research.microsoft.com/en-us/projects/hccb/>, květen 2010.
- [11] Ohbuchi, E.; Hanaizumi, H.; Hock, L. A.: Barcode Readers using the Camera Device in Mobile Phones. In *Proceedings of the 2004 Internatioanl Conference on Cyberworlds*, Japan: IEEE Computer Society, 2004, 0-7695-2140-1/04.

- [12] Oktem, R.; Oktem, L.: A Superresolution Approach for Bar Code Reading. Ankara, Turkey.
- [13] Ottaviani, E.; Pavan, A.; Bottazzi, M.; aj.: A Common Image Processing Framework for 2D Barcode Reading. In *Image Processing and its Applications, Conference Publication No. 465*, Italy: IEE, 1999, s. 652–655.
- [14] Sedláčková, K.; Vaja, Z.: Využití systému čárových kódů v oblasti logistiky. <http://www.cvis.cz/hlavni.php?stranka=novinky/clanek.php&id=209>, leden 2005.
- [15] Tekin, E.; Coughlan, J.: A Bayesian Algorithm fo Reading 1D Barcodes. San Francisco, CA: The Smith-Kettlewell Eye Research Institute, <http://www.ski.org/>.
- [16] Wachenfeld, S.; Terlunen, S.; Jiang, X.: Robust Recognition of 1-D Barcodes Using Camera Phones. Computer Vision and Pattern Recognition Group, Depatment of Computer Science, University of Münster, Germany: IEEE, 2008.
- [17] Wand, K.; Zou, Y.; Wang, H.: 1D Bar Code Reading on Camera Phones. In *International Journal of Image and Graphics*, ročník 7, China: World Scientific Publishing Company, 2007, s. 529–550.

Dodatek A

Obsah CD

Zde je uvedena struktura adresářů na CD a popisy nejdůležitějších souborů.

A.1 Obsah datového nosiče

doc	dokumentace.pdf	text semestrálního projektu v PDF
doc	programova	programová dokumentace v HTML
doc	tex	adresář se zdroji dokumentace v L ^A T _E X
doc	plakat	plakát prezentující výsledky, velikost A2
barcodes	testX-ean8	adresář s testovacími daty
include	opencv	knihovna OpenCV
papers		adresář s odbornými články v pdf
src	bc-reader.vcproj	projekt s programem ve Visual Studio 2008
src	*.cpp, *.h	zdrojové a hlavičkové soubory
test	test.py	testovací skript v Python
test	*.test	výsledky testovacích skriptů
www		zdrojové soubory webového rozhraní
Makefile		hlavní Makefile (použití viz dále)

Dodatek B

Jednoduchá nápověda

Jsou uvedeny příkazy pro spuštění překladač programu, dokumentace a testů.

B.1 Spuštění na OS Linux

<code>make</code>	přeložení programu
<code>make clean</code>	smazání všech binárních souborů
<code>make doc</code>	generování projektové dokumentace
<code>make programova</code>	generování programové dokumentace
<code>make norm</code>	výpočet normostran textu
<code>make test</code>	generování výstupů všech testů – trvá několik minut!

B.1.1 Spuštění testů na OS Linux

Spuštění testů probíhá pro všechny sady stejně. U testu je vždy vypsáno, jakých typů kódu se test týká.

<code>make testean8</code>	všechny EAN-8
<code>make testean13</code>	všechny EAN-13
<code>make testupca</code>	všechny UPC-A
<code>make test0</code>	generované EAN-8, EAN-13, UPC-A
<code>make test1</code>	fotografované EAN-8, EAN-13, UPC-A
<code>make test2</code>	poškozené EAN-8, EAN-13, UPC-A
<code>make test3</code>	barevné EAN-13

Jako další následuje sada testů pro robustnost, kde se vyskytuje pouze kód EAN-13.

```
make test40 generované, rotace
make test41 fotografované, rotace, bílé pozadí
make test41 fotografované, rotace, adaptované pozadí
make test50 generované, velikost
make test51 fotografované, velikost
make test60 generované, rozostření
make test61 fotografované, rozostření
```

B.2 Spuštění na OS Windows

Spuštění programu pod OS Windows je doporučeno ve Visual Studio 2008. Stačí otevřít projekt v daném vývojovém prostředí a je možné ho začít ladit. Jak již bylo zmíněno v textu, spuštění testovacích skriptů na tomto systému není možné.

Dodatek C

Seznam kódů zemí pro EAN-UCC

Každá země má dle GS1 přidělen kód, který je využíván jako první sekvence čísel v čárovém kódu pro označení země, kde byl výrobek vyroben [7]. Seznam kódů zemí EAN-UCC pro celý svět je uveden v následující tabulce.

kód	název země	kód	název země
530	Albánie	599	Maďarsko
613	Alžírsko	611	Maroko
779	Argentina	484	Moldavsko
485	Arménie	400–440	Německo
950	Aruba	618	Pobřeží slonoviny
476	Ázerbájdžán	900–919	Rakousko
950	Bangladěš	520	Řecko
540–549	Belgie	629	Spojené arabské emiráty
602	Benin	840–849	Španělsko
481	Bělorusko	760–769	Švýcarsko
387	Bosna a Hercegovina	389	Prefix Ean Int.
789–790	Brazílie	480	Filipíny
950	Brunej	950	Katar
380	Bulharsko	950	Madagaskar
950	Burundi	531	Makedonie
780	Chile	955	Malajsie
385	Chorvatsko	950	Maledivy
859	Česká republika	535	Malta
570–579	Dánsko	609	Mauricius

kód	název země	kód	název země
746	Dominikánská republika	750	Mexiko
474	Estonsko	865	Mongolsko
570–579	Faerské ostrovy	950	Myanmar
640–649	Finsko	615	Nigérie
300–379	Francie	743	Nikaragua
950	Gabon	870–879	Nizozemsko
603	Ghana	700–709	Norsko
950	Gibraltar	940–949	Nový Zéland
570–579	Grónsko	950	Omán
486	Gruzie	745	Panama
740	Guatemala	896	Pákistán
950	Guyana	775	Peru
742	Honduras	590	Polsko
489	Hongkong	560	Portugalsko
890	Indie	594	Rumunsko
899	Indonésie	860	Prefix Ean Int.
539	Irsko	950	Britské Panenské ostrovy
569	Island	950	Jemen
800–839	Itálie	600–601	Jihoafrická republika
626	Írán	460–469	Rusko
950	Jamajka	741	Salvador
450–459;490–499	Japonsko	950	Senegal
625	Jordánsko	888	Singapur
950	Kajmanské ostrovy	383	Slovinsko
884	Kambodža	000–139	Spojené státy americké
950	Kamerun	950	Surinam
000–139	Kanada	950	Súdán
487	Kazachstán	621	Sýrie
616	Keňa	730–739	Švédsko
770	Kolumbie	488	Tádžikistán
950	Kongo	885	Thajsko
744	Kostarika	471	Tchaj-wan

kód	název země	kód	název země
529	Kypr	950	Trinidad a Tobago
470	Kyrgyzstán	619	Tunisko
528	Libanon	869	Turecko
624	Libye	482	Ukrajina
760–769	Lichtenštejnsko	773	Uruguay
477	Litva	500–509	Velká Británie
475	Lotyšsko	759	Venezuela
540–549	Lucembursko	950	Velká Británie