HARDWARE MULTISCALE DETECTION BASED ON ADABOOST CLASSIFIER

Martin Musil, Petr Musil
Master Degree Programme (2), FIT BUT
E-mail: {xmusil34, xmusil36}@stud.fit.vutbr.cz

Supervised by: Pavel Zemčík
E-mail: zemcik@fit.vutbr.cz

Abstract: This article describes a hardware multiscale detection objects in an image. It focuses on evaluation of existing approaches and proposies new ones, with much smaller memory requirement. Benefits of the new approaches are reflected in the design of the new hardware object detector which is powerful enough to detect several object classes in real-time.

Keywords: Object detection, AdaBoost, Image features, Image scaling, FPGA

1 ÚVOD

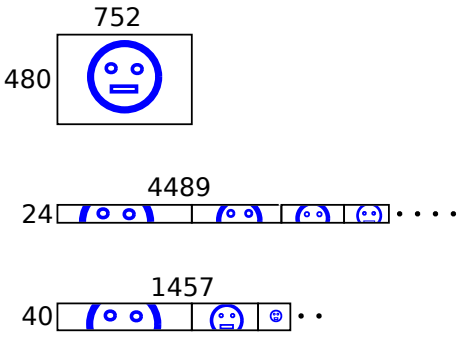
Existuje mnoho aplikací v průmyslu, dopravě i bezpečnosti, které mohou využít úsporné a malé zařízení, umožňující v reálném čase automaticky klasifikovat obrázky a hledat v nich zvolené objekty zájmu. Očekávanou vlastností takovýchto zařízení je jistě schopnost vyhledávat objekty v různé vzdálenosti od kamery. Vzdálenost se obecně projevuje jako monokulární podnět a ovlivňuje velikost hledaného objektu (bližší objekty se jeví větší než vzdálenější). Tato práce je zaměřena právě na problém detekce objektů různé velikosti v obraze na hardwarových detektorech.

2 HARDWAROVÁ DETEKCE OBJEKTŮ

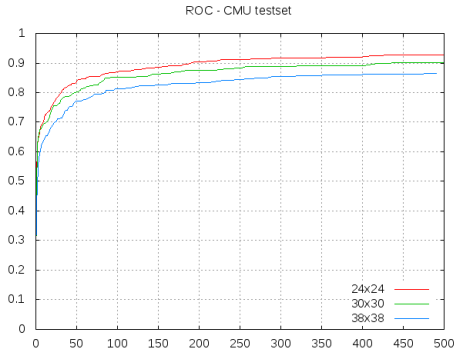
Pro samotnou detekci objektů existuje mnoho metod. Jako nejefektivnější se ukazují metody založené na strojovém učení využívající klasifikátorů. Jako první použili klasifikátory pro detekci objektů v obraze Viola a Jones [2]. Využili algoritmus AdaBoost pro natrénování silného klasifikátoru, který je složen z mnoha slabých klasifikátorů, což jsou jednoduché a snadno vyhodnotitelné funkce vypočtené z obrazu. Detekce objektů pomocí klasifikátorů poskytuje velmi dobré výsledky s nízkým procentem špatně detekovaných objektů a zároveň také dobrou generalizaci v rámci hledané třídy. Je však stále výpočetně velmi náročná a v reálném čase proveditelná pouze na výkonných počítačích. Proto se záhy objevily myšlenky pro její hardwarovou akceleraci. Huang a Vahid [1] vytvořili klasifikátor na programovatelném hradlovém poli (FPGA). Jejich řešení umožňovalo vyhledávat objekty v různé velikosti. Toho dosáhli použitím velké paměti pro uloženi celého obrázku (framebuffer). Jiné řešení představili například Zemčík a Žádník [3].Vytvořili engine obsahující FPGA klasifikátor a DSP procesor pro předzpracování obrazu. Výhodou jejich řešení je ukládání jenom malého pruhu obrazu do framebufferu, což umožňuje ušetřit distribuovaný paměťový prostor a použít menší a levnější čip. Samotné řešení bez dalšího rozšíření však umožňuje pouze vyhledávání objektů v jedné velikosti.

3 DETEKCE OBJEKTŮ RUZNÉ VELIKOSTI

Detekce objektů probíhá nad částmi obrazu. Klasifikátor se pohybuje po obraze a na každé jeho pozici určí, zda se na ní nachází nebo nenachází hledaný objekt. Aktuální zpracovávaná část obrazu je dána takzvaným "skenovacím oknem", jehož rozměry odpovídají velikosti hledaného objektu. V



Obrázek 1: Architektura pamětí pro detekci objektů. Nahoře je klasické řešení ukládající celý obraz, uprostřed uložení pruhu obrazu se všemi měřítky a dole je pruh obrazu jen s polovičními měřítky.



Obrázek 2: Porovnání klasifikátorů s LBP příznaky pro různou velikost okna. Natrénováno nad datasetem Schneiderman, testováno nad datasetem CMU.

praxi se používají nejčastěji čtvercová okna o velikosti hrany 16 až 31 obrazových bodů. To znamená, že například při použití okna o velikosti hrany 24 pixelů jsme schopni detekovat v obraze pouze objekty zájmů, které mají přibližně velikost 24×24 pixelů. Pokud tedy chceme detekovat například větší objekty (bližší) musíme zvětšit rozměry skenovacího okna nebo provést detekci nad zmenšeninou původního obrazu.

3.1 Klasický postup

Pro detekci objektů na počítači se nejčastěji používá zmenšování původního obrázku. Pro nalezení všech objektů dané třídy je zapotřebí takových zmenšení udělat několik a nad všemi vyhodnotit klasifikátor. Vytváří se tak pyramida zmenšených obrázků s krokem přibližně 1,2. Výhodou je, že zmenšený obrázek obsahuje méně pozic než předchozí, a proto je také jeho vyhodnocení statisticky rychlejší. Nevýhodou tohoto řešení je potřeba dostatečně velké paměti, alespoň pro uložení celého původního obrázku, což limituje použití v hardware (viz Tabulka1, architektura 1). Pokud má vstupní obraz malé rozlišení (např. 320x200px [1]), je možné jej na větším (dražším) FPGA uložit přímo na čipu a provádět zde i zmenšování. V případě větších rozlišení však budeme potřebovat rychlou externí paměť nebo přímo procesor (typicky DSP), který bude vytvářet obrázky v menším měřítku. Obě tato řešení velmi komplikují a prodražují výsledný detektor.

3.2 Navrhovaný postup

Budeme vycházet z přístupu použitého v [3]. Do paměti se neukládá celý obraz, ale jen jeho pruh o šířce obrazu a výšce o málo větší než rozměr skenovacího okna. Jiná varianta, která by neukládala celou šířku obrazu neni možná a to z důvodu "streamové" povahy vstupních dat, jejichž zdrojem je typicky kamera. Samotné toto řešení však neumožňuje detekci objektů různé velikosti. Proto použijeme paměť o větší šířce, do které si budeme ukládat také pruh se zmenšenými verzemi obrazu. Šířka paměti se pro náš systém (kamera s rozlišením 752×480 pixelů) zvětšila téměř šestkrát, a to z 752 na 4484 sloupců. Pro tvorbu zmenšených pruhů je s výhodou použit druhý port pamětí, který je na FPGA zpravidla k dispozici, a proto je možné provádět zmenšování současně s detekcí bez omezení rychlosti. Po implementaci (viz Tabulka 1, architektura 2) bylo zjištěno, že množství potřebných hradel na čipu je jen o 10% větší než při klasickém přístupu. Největší počet hradel (skoro 70%) zabíraly jednotky pro změnu měřítka o 1,2 násobek (scale 6/5). Z tohoto důvodu jsme vytvořili další přístup, který provádí zmenšování obrazu vždy jen o polovinu. Zmenšení o polovinu je velice dobře realizovatelné v hardwaru. Velikosti mezi jsou detekovány pomocí skenovacího okna s jinou velikostí. V našem případě jsme použili okna o velikosti hrany 24, 28, 33 a 40. Šířka potřebné paměti

tak klesla na 1457 sloupců, naopak požadovaná výška stoupla z 24 na 40 řádků. Implementace tohoto systému (viz Tabulka 1, architektura 3) si vyžádala nejméně zdrojů ze všech řešení. Tím že se neprovádí tolik zmenšení ale vzrostl počet pozic, které musí detektor vyhodnotit a proto klesá rychlost detekce.

3.3 VYHODNOCENÍ

V tabulce 1 jsou porovnány výsledky navržených systémů pro vstup z kamery o rozlišení 752×480 pixelů. Pro porovnání je zde uveden i systém který umožňuje detekci objektů jen v jedné velikosti.

	BRAM [počet]	BRAM (n/arch1)	hradel (n/arch1)	scale 5/6 [operací]	scale 1/2 [operací]	pozic pro vyhodnocení	pozic (n/arch1)
bez zmenšování	6	5 %	30 %	0	0	332 968	33 %
architektura 1	118	100 %	100 %	32536	0	1013 053	100 %
architektura 2	36	30 %	110 %	32536	0	1013 053	100 %
architektura 3	20	17 %	40 %	0	13316	1645 222	162 %

Tabulka 1: Porovnání navržených architektur pro detekci objektu na FPGA řady Spartan6/X45T. Porovnání je vztažené k architektuře 1, která spotřebuje 101 procent paměti a 38 procent LUT Slice.

Pro ověření vlastností navrhovaného řešení byl proveden experiment kvality detekce objektů v závislosti na velikosti skenovacího okna. Pro ověření bylo natrénováno několik klasifikátorů s různou velikostí skenovacího okna. V praktické realizaci je pro ušetření paměti použit pouze jeden klasifikátor, jehož koeficienty se přepočítávají v závislosti na požadovaném měřítku. Na obrázku 2 je zobrazena jedna z mnoha naměřených ROC křivek. Je patrné, že se kvalita detekce s rostoucí velikostí skenovacího okna mírně snižuje.

4 ZÁVĚR

Tento článek hodnotí existující postup pro detekci objektů různé velikosti v obraze. Navrhuje jeho úpravy pro nasazení v hardwarových detektorech. Jednotlivé postupy byly implementovány na architektuře FPGA. První navrhovaný systém snižuje potřebu paměti na třetinu, výrazně nezvyšuje spotřebu zdrojů na čipu a neovlivňuje výkon a kvalitu detektoru. Druhý systém snižuje spotřebu paměti ještě více, a to až na šestinu a odstraňuje nutnost použití složitých jednotek pro změnu měřítka. Na druhou stranu snižuje výkon detektoru o 62% a mírně zhoršuje kvalitu detekce.

Navržené systémy a jejich vlastnosti jsou zohledněny v hardwarovém detektoru objektu ve videosekvencích vyvíjeném na UPGM. Je zde použit poslední navržený systém. Důvodem je právě nejmenší paměťová náročnost, která se ukázala jako kritická. Snížení rychlosti detekce nás naopak neomezovalo díky dostatečné výkonové rezervě detektoru.

REFERENCE

[1] Huang, C.; Vahid, F.: Scalable object detection accelerators on FPGAs using custom design space exploration. In *IEEE Symposium on Application Specific Processors (SASP)*, IEEE Computer Society, 2011, ISBN 978-1-4577-1212-8, s. 115–121.

[2] Viola, P.; Jones, M.: Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, 2001, ISBN 0-7695-1272-0, s. 511–518.

[3] Zemčík, P.; Žádník, M.: Adaboost Engine. In *Proceedings of FPL 2007*, IEEE Computer Society, 2007, ISBN 1-4244-1060-6, s. 656–660.