

USING OPEN SOURCE OPENCV LIBRARY FOR PRACTICAL COURSES OF COMPUTER VISION

Pavel JETENSKÝ

Abstract: The article shows how to use free software library OpenCV as a main software tool for Computer Vision courses at secondary schools and universities. Sample lectures of Computer Vision are covered with respect to how to demonstrate algorithms' behavior using OpenCV. Comparison of OpenCV with alternative software tool MATLAB is presented. Working C# code samples are part of the article.

Key words: Computer Vision, Machine Perception, teaching, course, OpenCV, software, EmguCV, MATLAB, tool, Open Source.

VYUŽITÍ OPEN SOURCE KNIHOVNY OPENCV PRO PRAKTICKÁ CVIČENÍ PŘEDMĚTU POČÍTAČOVÉ VIDĚNÍ

Resumé: Příspěvek poskytuje informace o tom, jak použít volně dostupnou knihovnu OpenCV jako hlavní softwarový nástroj pro výuku cvičení předmětu Počítačové vidění na středních a vysokých školách. V článku jsou popsány vzorové úlohy počítačového vidění s ohledem na to, jak demonstrovat vlastnosti algoritmů pomocí knihovny OpenCV. Je uvedeno srovnání s alternativním nástrojem pro výuku, komerčním softwarem MATLAB. Součástí textu jsou funkční ukázky zdrojového kódu v jazyce C#.

Klíčová slova: Počítačové vidění, výuka, cvičení, OpenCV, software, EmguCV, MATLAB, nástroj, otevřený software.

1 Úvod

Procvičování algoritmů počítačového vidění se neobejde bez potřebného software, pomocí něhož si studenti budou osvojovat nově nabývané znalosti. Jednou z možností je komerční software MATLAB (matrix laboratory). Ten si mnoho vědců i inženýrů vybírá pro svou práci, protože se jedná o interaktivní systém zapouzdřující v jednom celku matematické výpočty, vědecké vizualizace i symbolické výpočty (1).

Jednou z alternativ k používání softwaru MATLAB je využít knihovnu OpenCV. OpenCV je otevřená multi platformní knihovna pro manipulaci s obrazem, distribuovaná zdarma.

Oba tyto nástroje spolu s řadou dalších možných softwarových nástrojů vhodných k výuce počítačového vidění zmiňuje G. Bebis v článku Review of computer vision education (2).

V této práci autoři také zmiňují důležitost praktických demonstrací, pomocí níž si studenti mohou ověřit získané teoretické znalosti: *"Because of the visual nature of computer vision, an important consideration for computer vision educators is to develop inter- active materials and demonstrations that will allow students to have active 'hands-on' learning experiences."*

2 O programu MATLAB

MATLAB je profesionální produkt vhodný pro vědeckou práci. K výuce předmětu počítačové vidění se používá např. na fakultě elektrotechniky a komunikačních technologií VÚT v Brně. Jeho využití na konkrétní škole však může bránit několik faktorů:

- MATLAB není dostupný zdarma, pro výuku je nutné zakoupit buď akademickou licenci nebo licenci pro jednoho studenta. Licence pro jednoho studenta stojí \$99.
- Pro práci v MATLAB se musí studenti naučit syntaxi jazyka MATLAB.
- MATLAB projekt nelze přímo použít v projektu naprogramovaném v nějakém jiném jazyce (např. Java nebo C#), lze pouze vygenerovat C/C++ zdrojový kód pomocí MATLAB coder. I MATLAB coder je nutné zakoupit.

3 Knihovna OpenCV

Knihovna OpenCV je již od počátku zaměřena především na počítačové vidění a zpracování obrazu v reálném čase. Knihovna je napsaná v jazyce C. Oproti MATLAB nabízí tyto výhody:

- Je zdarma jak pro akademické, tak komerční využití (pod BSD licenci).
- Má větší portabilitu výsledné práce; knihovnu OpenCV lze využít z projektů napsaných v jazyce C, díky obalujícím knihovnám je možné je zavolat i z jiných jazyků (viz tabulka 1). Studenti tak mohou v rámci cvičení počítačové grafiky rozvíjet i schopnosti programovat ve vybraném programovacím jazyce. Alespoň v 32-bitové verzi lze knihovnu spustit na Windows, Linux, i MacOS operačních systémech (3).
- Algoritmy běží rychleji proto, že je knihovna napsaná v jazyce C; pro úlohy počítačového vidění je rychlejší než MATLAB, který interpretuje svůj vlastní abstraktní jazyk (4).
- Obsahuje širší množinu podporovaných funkcí; díky specializaci pouze na oblast počítačového vidění obsahuje knihovna OpenCV algoritmy, které se v MATLAB nevyskytují. Příkladem mohou být funkce pro detekci obličeje v obraze, kalibraci dvou kamer pro stereo snímání nebo výpočet transformační matice perspektivy ze čtyř zadaných bodů.

Andrea Cavallaro použil knihovnu OpenCV jako základ pro vytvoření vlastního frameworku pro výuku počítačového vidění, napsaného v jazyce C++ (5). Jeho výuková platforma umožňuje studentům snížit čas potřebný k instalaci prostředí potřebného ke zkoušení základních algoritmů počítačového vidění z jednoho týdne na pouhé čtyři hodiny. V kapitole 4 popíšeme alternativní možnost, jak snadno využít OpenCV, a tou je využití knihovny EmguCV.

Používaná nadstavba OpenCV	Programovací jazyk v rámci cvičení
bez nadstavby	C, C++
EmguCV	C#
JavaCV	Java
PyOpenCV	Python

Tab 1: Programovací jazyky, z nichž je možné využít OpenCV.

4 Ukázkové zdrojové kódy

Následující kapitoly obsahují ukázky zdrojového kódu pro řešení jednoduchých úloh počítačového vidění. Úlohy jsou převzaty ze skript předmětu Počítačové vidění, který se vyučuje na VÚT v Brně (6). Smyslem prezentovaných úloh je ukázat, že zápis algoritmu počítačového vidění s využitím

OpenCV je pro studenty jednoduchý a srozumitelný.

Samotnou knihovnu OpenCV je možné využívat přímo z programovacího jazyka C++. V takovém případě však studenti musí sami řešit správu přidělené paměti. Jednodušší je tedy využít nějaký programovací jazyk s vlastní správou paměti, jako je C# nebo Java. Ukázkové zdrojové kódy (7) jsou tedy psány v jazyce C# s využitím nadstavby EmguCV. EmguCV využívá duální licenci a pro open source projekty je zdarma. Jako vývojové prostředí byl použit program Microsoft Visual Studio 2010 Ultimate.

4.1 Základní nastavení vývojového prostředí pro cvičení

Následující kroky vedou k vytvoření prostředí, které je potřeba pro spuštění ukázkových kódů:

1. V programu Microsoft Visual Studio vytvořte nový projekt typu Windows Forms Application.
2. Z Internetu stáhněte a do referencí projektu přilinkujte knihovny Emgu.CV.UI.dll, Emgu.CV.dll a Emgu.Util.dll.
3. Do projektu pomocí funkce [Add] => [Existing Item] přidejte soubor opencv_core231.dll a v jeho vlastnostech nastavte [Copy to Output Directory] na hodnotu [Copy if newer].
4. Totéž proveďte pro soubory opencv_imgproc231.dll a ZedGraph.dll (tyto knihovny jsou potřebné pro zobrazení histogramu).
5. Na formulář vložte ovládací prvek typu Emgu.CV.UI.ImageBox a nazvěte ho imageBox.

4.2 Úloha 1 - vytvoření šedotónového obrazu

Zadání: Vytvořte matici představující obrazovou rovinu šedotónového obrazu o rozměrech 640x480 bodů s nulovými prvky. V obraze vytvořte nahodile se překrývající obdélníky o různých hodnotách intenzity.

Cílem tohoto úkolu je seznámit studenty s maticovým způsobem reprezentace obrazových dat a nechat je uvědomit si, jakým způsobem souvisí číselná hodnota v matici s výslednou světlostí odpovídajícího pixelu v šedotónovém obraze.

Řešení v EmguCV

```
var image =
```

```

new Image<Gray, Byte>(640,480);
image.Draw(
    new Rectangle(20, 60, 100, 25),
    // Rectangle intensity
    new Gray(128),
    // Fill rectangle (no contour)
    0);

// kreslení dalších obdélníků
// vynecháno

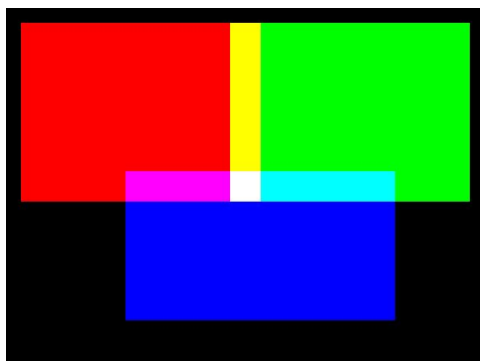
imgBox.Image = image;

```

4.3 Úloha 2 – složení barevného obrázku

Zadání: Vygenerujte barevný obraz složením 3 šedotónových obrazů představující složky červené, zelené a modré barvy. Obrazy navrhnete tak, aby ve výsledném barevném obrazu vznikly všechny kombinace tří základních barev. Pro srovnání vykreslete současně všechny tři složkové obrazy a obraz barevný do jednoho okna.

Cílem tohoto úkolu je seznámit studenty s rozdílem mezi šedotónovým a barevným obrázkem a dovést je k pochopení, jakým způsobem se pracuje s aditivním barevným modelem RGB a jak lze skládat tři základních



Obr 2: Výsledný obrázek pro úlohu 2

barevných složek (červená, zelená, modrá) vytvořit ostatní barvy.

Řešení v EmguCV:

```

var img =
    new Image<Bgr, Byte>(640,480);

var redChannel =
    new Image<Gray, byte>(640,480);

var greenChannel =
    new Image<Gray, byte>(640,480);

```

```

var blueChannel =
    new Image<Gray, byte>(640,480);

var rect1 =
    new Rectangle(20, 20, 320, 240);
var rect2 =
    new Rectangle(300, 20, 320, 240);
var rect3 =
    new Rectangle(160, 220, 360, 200);

var white = new Gray(255);
redChannel .Draw(rect1, white, 0);
greenChannel .Draw(rect2, white, 0);
blueChannel .Draw(rect3, white, 0);

image[2] = redChannel;
image[1] = greenChannel;
image[0] = blueChannel;
imgBox.Image = image;

```

4.4 Úloha 3 – složení barevného obrázku

Zadání: Načtete obraz z disku, proveďte nulování jeho prostřední třetiny, zobrazte ho a opět uložte na disk.

Smyslem tohoto úkolu je seznámit studenty s často používanou technikou počítačového vidění, kterou je práce pouze s částí obrazu (tzv. ROI - Region Of Interest). Nad takto vybranou oblastí lze poté provádět operace pro manipulace s



Obr 1: Výsledný obrázek pro úlohu 3

obrazem, např. nastavení všech složek barev na nulu.

Řešení v EmguCV:

```

var image = new Image<Bgra, Byte>
    ("C:\\test.jpg");
image.ROI = new Rectangle(
    image.Width/3, image.Height/3,
    image.Width/3, image.Height/3);
image.SetZero();
image.ROI = Rectangle.Empty;
image.Save("C:\\test.jpg");

```

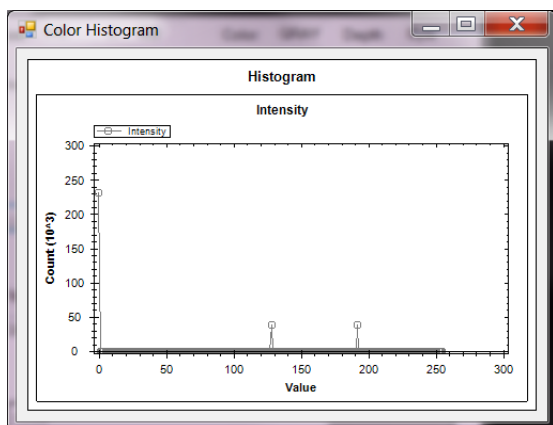
5 Interaktivní vlastnosti ovládacího prvku

ImageBox využitelné při výuce

Formulářový ovládací prvek ImageBox, který je dostupný v knihovně EmguCV, neslouží pouze pro zobrazování obrázků. Lze ho také použít jako mocný nástroj pro ladění problémů a testování jednotlivých grafických algoritmů bez nutnosti měnit kód. Obsahuje totiž interaktivní kontextové menu, které se aktivuje klepnutím pravého tlačítka myši. Tak je možné např. na obrázek aplikovat různé filtry, podívat se na hodnotu intenzity konkrétního pixelu (obdoba funkce kapátka) nebo zobrazit histogram.

Kontextové menu obsahuje tyto funkce:

- File (Load Image, Save As),
- Filters,
 - Gradient, Edges (Canny, Laplace, Sobel),
 - Logic (Cmp, Or, Xor, InRange, Max, Min, Not),
 - Math (AbsDiff, Add, Exp, Log, Mul, Pow, Sub, SubR),
 - Morphology (Dilate, Erode),
 - Pyramids (PyrDown, PyrUp),
 - Smoothing (Bilateral, Blur, Gaussian, Median),
 - Threshold (Adaptive, Binary, BinaryInf, ToZero, ToZeroInv, Trunc),
 - Transform (LogPolar, Rotate, Flip),
 - Various (Resize, EqualizeHist, GammaCorrect, SetRandNormal, SetRandUniform),
- Zoom (In, Out, Un-Zoom),
- Property – zobrazí se nové okno s rozšířenými funkcemi (funkce kapátka, informace o barevné hloubce, zdrojový kód použitých filtrů, možnost zobrazení histogramu).



Obr 3: Histogram dostupný z kontextového menu

6 Závěr

Článek shrnul možnosti knihovny OpenCV a jejích výhod oproti platformě MATLAB. Na základě uvedených informací je zřejmé, že OpenCV je pro svoji flexibilitu a specializaci na oblast počítačového vidění životaschopnou alternativou. Nejen, že zdarma umožní studentům pracovat i na těch nejsložitějších úlohách z oblasti počítačového vidění, ale umožní jim i snadno použít výsledky vlastní práce v rámci dalších softwarových projektů. Učitelům předmětů počítačové grafiky a počítačového vidění přeji s touto jedinečnou knihovnou řadu úspěchů a radosti.

7 Literatura

- [1] ETTER, D., KUNCICKY, D., HULL, D. Introduction To Matlab 6. 1. vyd. Prentice Hall, 2001, s. 1. ISBN 978-01-3032-845-8.
- [2] BEBIS, G., D. EGBERT a M. SHAH. Review of computer vision education. IEEE JOURNALS & MAGAZINES. 2003, č. 46, s. 2-21.
- [3] BRADSKI, G., KAEHLER, A. Learning OpenCV: Computer Vision with the OpenCV Library. 1. vyd. O'Reilly Media, 2008, s. 14. ISBN 978-05-9651-613-0.
- [4] MATUSKA, S., HUDEC, R. and BENCO, M. The comparison of CPU time consumption for image processing algorithm in Matlab and OpenCV. In: Rajeck Teplice: Dept. of Telecommun. & Multimedia, Univ. of Zilina, Zilina, Slovakia, 2012, s. 75-78. ISBN 978-1-4673-1180-9
- [5] CAVALLARO, A. Image analysis and computer vision for undergraduates. In: 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing. Philadelphia, Pennsylvania, USA: The Institute of Electrical and Electronics Engineers Signal Processing Society, 2005, s. 577.

- [6] HORÁK, K., KALOVÁ, I. Počítačové vidění - Počítačová cvičení (64087) - VUT v Brně (online). [cit. 2012-10-13]. URL : <http://www.vutbr.cz/vyzkum-a-vyvoj/publikace?action=detail&pub_id=64087>.
- [7] JETENSKÝ, P. Zdrojové kódy - Computer vision course with OpenCV (online). [cit. 2012-10-13]. URL : <<http://www.jetensky.net/download/files/ComputerVisionCourseWithOpenCV.zip>>.

Ing. Pavel Jetenský
Katedra informačních technologií
Fakulta Elektrotechniky a Informatiky UPCE
Studentská 62,
Pardubice, 532 10, ČR
Tel: +420 466 037 108
E-mail: pavel.jetensky@student.upce.cz,
pavel.jetensky@seznam.cz