

Technická správa

Program pre zobrazovanie vlastností obrazu

23. marca 2017

Autor: Martin Matejčík, xmatej46@stud.fit.vutbr.cz
Fakulta Informačních Technologí
Vysoké Učení Technické v Brně

Obsah

1	Zloženie tímu	1
2	Zadanie	1
3	Náš cieľ	1
4	Štúdium problematiky	1
4.1	JavaCv	1
4.2	OpenCv for Java	1
4.3	Testovacie dáta	1
5	Použité technológie	2
6	Rozdelenie práce v tíme	2
7	Aktuálny stav	2
7.1	Screenshoty z aplikácie	3
7.2	Problémy, na ktoré sme stihli naraziť	4
	Literatúra	5

1 Zloženie tímu

- Tibor Mikita, xmikit01@stud.fit.vutbr.cz
- Martin Matejčík, xmatej46@stud.fit.vutbr.cz

2 Zadanie

Zadaním je vypracovať program, ktorý bude schopný zobrazovať rôzne údaje o načítanom obrázku.

3 Náš cieľ

Výsledkom tohto projektu bude aplikácia s grafickým užívateľským rozhraním, do ktorej bude možné nahráť ľubovoľný obrázok. Po načítaní obrázka do aplikácie sa, v užívateľsky prívetivom prostredí, zobrazia rôzne informácie. Budú to informácie týkajúce sa súboru ako takého, ale aj informácie súvisiace so samotným obrázkom, prípadne ďalšie informácie, ako je napríklad informácia o pixeli v mieste kurzora myši, histogram obrázka a iné.

4 Štúdium problematiky

Ako implementačný jazyk bola zvolená java, ktorá so spojením gradle build systému poskytuje jednoduché používanie knižníc tretích strán. Jedna z najdostupnejších knižníc pre spracovanie obrazu je OpenCv. Pre javu existuje viacero implementácií.

4.1 JavaCv

Rozsiahla knižnica postavená na JavaCpp natívnych implementáciach. Zabaľuje v sebe široké možnosti spracovania obrazu cez OpenCv a FFmpeg. Nepoužíva ale úplne rovnaké rozhranie pre OpenCv a jej dokumentácia nie je úplne kompletná[1].

4.2 OpenCv for Java

Java knižnica využívajúca rovnaké rozhranie na OpenCv ako originálna implementácia v C++. Jej hlavnou výhodou je možnosť využitia oficiálnej dokumentácie k OpenCv[2].

4.3 Testovacie dáta

Testovacie dáta budú prirodzene pozostávať z rôznych obrázkov rôzneho formátu(JPG, PNG, BMP) a veľkosti. Obrázky budú mať rôzny počet kanálov(RGB, RGBA, GrayScale). Súčasťou testovacej sady budú napríklad aj súbory bez prípony, alebo súbory s nesprávnou príponou, aby bolo možné otestovať MIME typ obrázka nezávisle na prípone v názve súboru. Testovacie dáta budú stiahnuté z nejakej voľne prístupnej databázy.

5 Použité technológie

- Java 8
- JavaFX - GUI
- OpenCV 3.2.0 (packaged by OpenPnP) - práca s obrázkami
- Apache Tika Core - MIME typ
- JFoenix - material design
- Gradle - build system

6 Rozdelenie práce v tíme

Martin:

- základná štruktúra aplikácie podľa MVC modelu
- prepojenie použitých technológií
- rozdelenie obrázku na jednotlivé kanály a ich vzájomné zobrazovanie

Tibor:

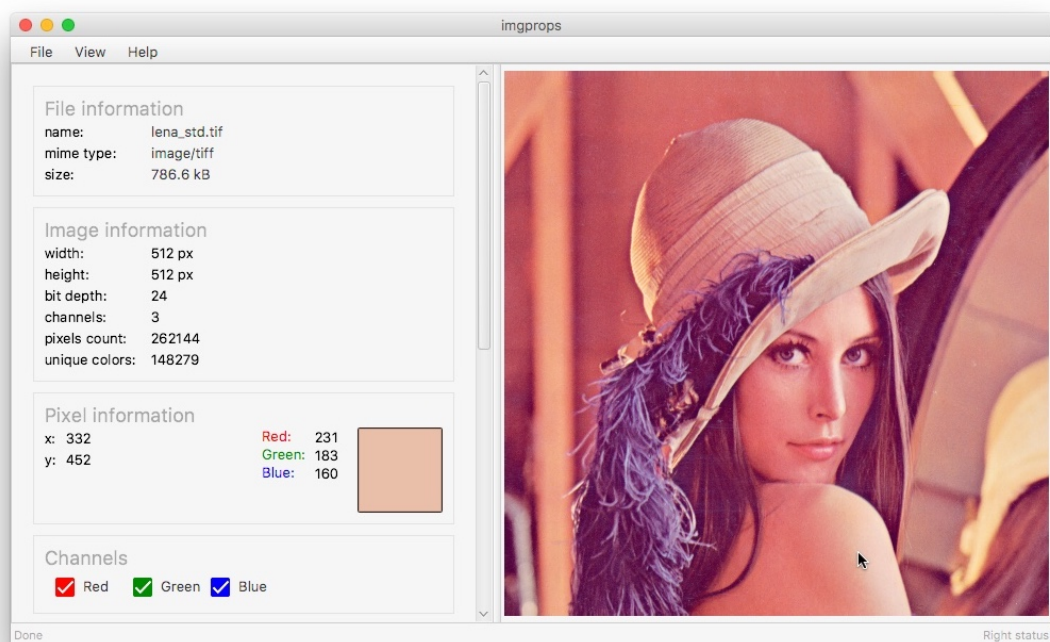
- načítanie obrázku zo súboru a získanie základných informácií o obrázku a o súbore ako takom
- histogram
- zobrazovanie informácií o pixeli pod kurzorom myši
- zhromaždenie testovacej sady obrázkov

7 Aktuálny stav

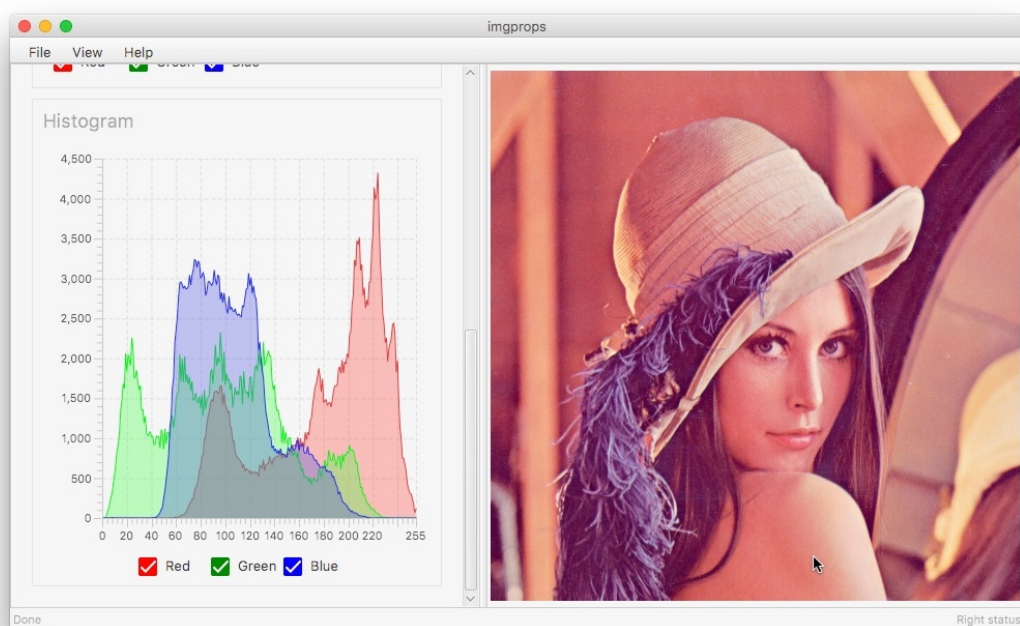
Toto sme už stihli implementovať:

- zobrazovanie informácií
 - o súbore(názov, MIME typ, veľkosť v bytoch)
 - o obrázku(šírka a výška v pixeloch, bitová hĺbka, počet kanálov, celkový počet pixelov)
 - o pixeli v mieste kurzora myši(x-ová a y-ová súradnica, farba v RGB, resp. intenzita)
- histogram rozdelený podľa RGB zložiek(jednotlivé zložky je možné schovávať a znova zobrazovať)
- zobrazovanie a schovávanie jednotlivých farebných zložiek obrázku
- ďalšie možnosti prístupné cez menu
 - znovuotvorenie nedávno otvorených obrázkov
 - schovávanie a zobrazovanie jednotlivých sekcií s informáciami

7.1 Screenshoty z aplikácie



Obr. 1: Základné informácie



Obr. 2: Ukážka histogramu

7.2 Problémy, na ktoré sme stihli naraziť

Funkcia `imread()` s príznakom `IMREAD_UNCHANGED` zhodí celý program, ak vstupom je 16-bitový `tiff` súbor. Preto sme sa rozhodli vymeniť príznak `IMREAD_UNCHANGED` za dvojicu príznakov `CV_LOAD_IMAGE_ANYCOLOR` a `CV_LOAD_IMAGE_ANYDEPTH`. Tým ale strácame informáciu o pôvodnom počte kanálov, pretože OpenCV pôvodný obrázok prekonvertuje do greyscale alebo RGB modelu a teda vždy pracujeme s jedným alebo tromi 8-bitovými kanálmi. Pôvodná informácia o počte kanálov a bitovej hĺbke pre určité obrázky (napr. spomínaný 16-bitový `tiff`) je teda nedostupná.

Literatúra

- [1] Bytedeco: bytedeco/javacv: Java interface to OpenCV and more. <https://github.com/bytedeco/javacv>, (cit. 23.3.2017).
- [2] OpenPnP: openpnp/opencv: OpenCV Java bindings packaged with native libraries, seamlessly delivered as a turn-key Maven dependency. <https://github.com/openpnp/opencv>, (cit. 23.3.2017).