

MoZOP 1.0
Software pro monitoring zaplnění ochranných plotů

Technická dokumentace

12/2019

Václav Nežerka a Michael Somr
Katedra mechaniky, Fakulta stavební, ČVUT v Praze
somr.michael@gmail.com

1. Úvod

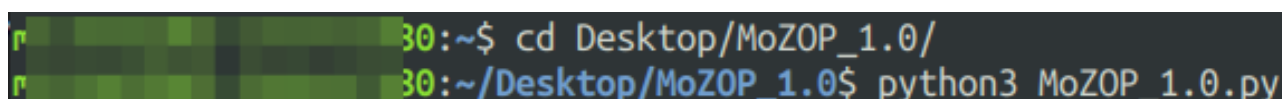
Software MoZOP 1.0 (Monitoring Zaplnění Ochranných Plotů) je nástroj pro optické vyhodnocení míry zaplnění ochranných plotů z fotografií vyvinutý v jazyce Python 3.6. Míru zaplnění program určuje díky změně indexu strukturální podobnosti (structural similarity index) zasypávaných signálních tyčí, které jsou umístěny poblíž sítí ochranného plotu nebo přímo na nich. Algoritmy jsou vytvořeny tak, aby bylo možné analyzovat libovolné množství signálních tyčí, které jsou vyfotografovány s dostatečným rozlišením (šířka nejvzdálenější tyče cca 20 px). Zdrojový soubor a technická dokumentace jsou k dispozici také na [GitHub](#) repositáři.

Index strukturální podobnosti

Algoritmus pro vyhodnocení míry zasypaní signálních tyčí je založen na indexu strukturální podobnosti, který představili především Zhang a Bovik [1, 2]. Jedná se o metodu srovnání, která bere v potaz fakt, že lidské vidění (vnímání scény) je vysoce přizpůsobeno k extrahování strukturální informace, a která uvažuje degradaci obrazu jako vnímanou změnu ve strukturálních informacích a zároveň zahrnuje důležité jevy vidění, včetně maskování jasu a maskování kontrastu. Podstata strukturální informace je založena na myšlence, že pixely mají silné vzájemné závislosti, zejména pokud jsou prostorově blízké. Tyto závislosti nesou důležité informace o struktuře objektů ve vnímané scéně. Maskování jasu je jev, při kterém je zkreslení obrazu (v tomto kontextu) ve světlých regionech méně viditelné, zatímco maskování kontrastu je jev, při kterém se zkreslení stává méně viditelným tam, kde je v obraze významná aktivita nebo „textura“.

2. Struktura programu

Program se spouští z příkazové řádky (Windows) / terminálu (Mac, Linux) pomocí příkazu `python3` (zadaného ve složce, ve které se nachází samotný program `MoZOP_1.0.py`) následovaného názvem programu, v tomto případě tedy `MoZOP_1.0.py`, jak je ilustrováno na Obr. 1.



```
30:~$ cd Desktop/MoZOP_1.0/
30:~/Desktop/MoZOP_1.0$ python3 MoZOP_1.0.py
```

Obr. 1: Navigace do složky s programem (první řádek) a spuštění programu (druhý řádek).

Pro správný chod programu musí být nainstalovaný [Python](#) verze 3.6 nebo vyšší a knihovny `os`, `matplotlib`, `numpy` a `scikit-image`. Návod jak instalovat knihovny je k dispozici např. [zde](#).

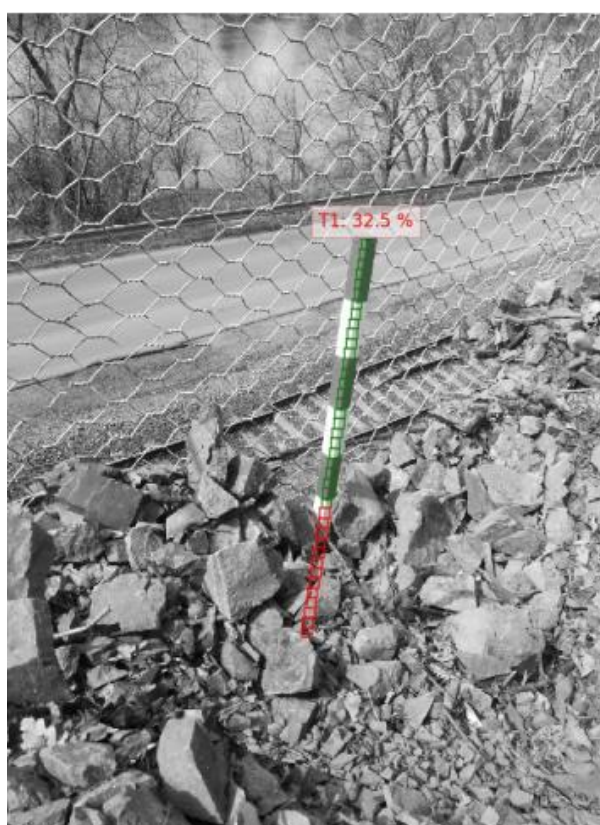
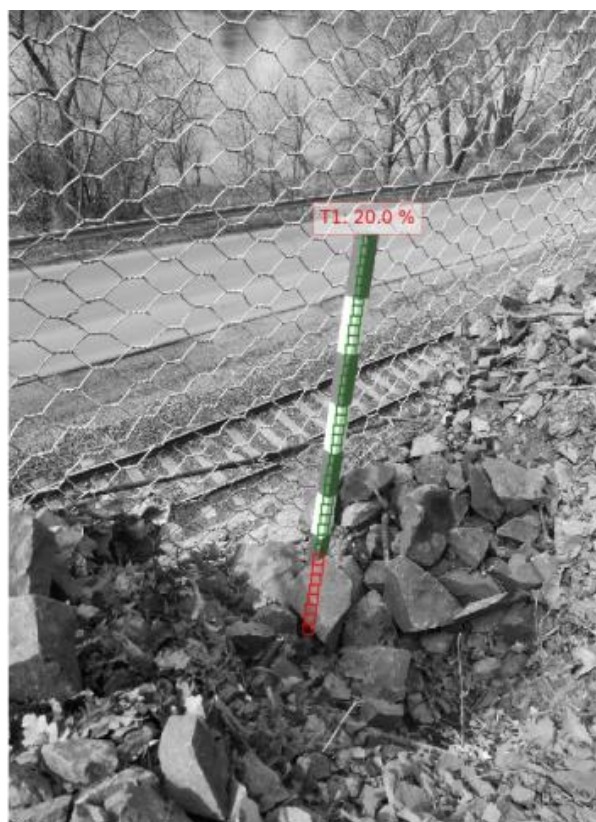
Výchozí struktura je taková, že ve složce s programem se nachází ještě soubor *roiData.dat*, obsahující souřadnice signálních tyčí a také složka se snímky, které budou analyzovány. Složka se snímky zároveň slouží jako adresa kde bude vytvořena složka *output* obsahující výstupy analýzy. Tato struktura však není striktně dána a všechny soubory a složky je možné umístit na libovolné adresy; tyto však musí být následně uvedeny v samotném skriptu.

3. Workflow programu

Program je vytvořen tak, aby vyžadoval co nejméně (ideálně žádné) vstupů od uživatele. Po spuštění programu jsou ze souboru *roiData.dat* načteny souřadnice signálních tyčí definující liniové segmenty (nezáleží na pořadí souřadnic), na kterých jsou vytvořeny "kontrolní podoblasti" - subsety (viz Obr. 2), v nichž je vyhodnocována změna indexu strukturální podobnosti. Následně je ve složce se snímky určenými k analýze vytvořena složka *output*, do které se nahrávají výstupní data analýzy. Poté jsou načteny samotné snímky, je u nich vyhodnocen index strukturální podobnosti a postupně jsou všechny porovnávány s prvním (výchozím) snímkem. Míra zasypání signálních tyčí je vyjádřena procenty z délky analyzovaného liniového segmentu a je zapsána do prostého textového souboru ve složce *output*, který je vhodný pro další zpracování např. v databázi. Volitelnou možností je pak grafický výstup, který je demonstrován na Obr. 2. Celý tento proces se odehrává automaticky bez potřeby zásahů uživatele.

Program dále pracuje s následujícími 3 proměnnými, které mohou být editovány přímo ve skriptu:

- *tolerance* - udává procentuální hodnotu tolerované změny v indexu strukturální podobnosti v rámci kontrolní podoblasti; výchozí hodnota je 5 %, tedy pokud dojde k nejvýše 5% změně indexu ve sledované oblasti, není to vyhodnoceno jako její zasypání; touto proměnnou jsou eliminovány např. vlivy zhoršené viditelnosti při pořizování snímků, které také mohou vést ke změně indexu a následnému falešnému vyhodnocení.
- *subset_size* - určuje velikost kontrolních podoblastí, které jsou rozmístěny na liniových segmentech reprezentujících signální tyče; čím větší podoblasti, tím spolehlivější vyhodnocení, ale také větší nároky na rozlišení fotografie; výchozí hodnota je 15 px, tedy i nejvzdálenější kontrolní tyč musí mít na snímku šířku minimálně 15 px (spíše 20 px), jinak by kontrolní podoblast zasahovala mimo tyč a vyhodnocovaly by



Obr. 2: Grafický výstup vyhodnocování procentuální míry zasypání signálních tyčí pomocí indexu strukturální podobnosti v rámci "kontrolních podoblastí" (čtverečky podél tyčí) - zelené podoblasti značí nezasypanou a červené zasypanou část.

se i změny indexu v jejím okolí a na jejím pozadí, které mohou být časté a značné.

- *subset_spacing* - určuje podélné (ve smyslu orientace podélné osy liniového segmentu) rozestupy mezi středy kontrolních podoblastí; výchozí hodnota je 15 px, tedy při velikosti podoblastí rovněž 15 px na sebe budou jednotlivé podoblasti přesně navazovat svými hranami, v případě menších rozestupů než jsou velikosti podoblastí by se podoblasti překrývaly a v případě větších rozestupů než jsou kontrolované podoblasti by mezi podoblastmi vznikaly mezery a tedy nekontrolované části.

Poděkování

Tato práce byla podpořena grantem Technologické agentury České republiky TJ01000263.

Zdroje

[1] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, Apr. 2004.

[2] Z. Wang, A. C. Bovik, "Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures," Signal Processing Magazine, IEEE, vol. 26, no. 1, pp. 98-117, Jan. 2009.