

Meshlab PHOV

Žiga Stopinšek

26. september 2014

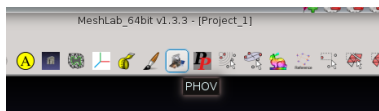
1 Opis orodja

Meshlab je programska oprema za ogledovanje, urejanje, manipuliranje in izvažanje 3D modelov. Sam po sebi ne omogoča fotogrametrije, vendar pa že vsebuje orodje Arc3D, ki omogoča ogledovanje in urejanje fotogrametričnih oblakov točk, pridobljenih s spletno storitvijo Arc3D. Problem, ki ga rešuje orodje Meshlab PHOV je nalaganje in obdelava fotografij izključno v orodju Meshlab.

Arheologom po svetu je programska oprema Meshlab blizu, organiziranih pa je tudi vrsto delavnic za spoznavanje s programom, tudi v Sloveniji. Zato bi integracija takšnega orodja v že nek poznan program uporabniku olajšala proces spoznavanja in učenja novega orodja.

2 Uporaba orodja

Uporabniški vmesnik za uporabo PHOV orodja v Meshlabu je izjemno preprost, zaradi česar pa je težje sistem prilagajati. Vsa komunikacija s storitvijo poteka preko gumba za orodje, ki ima v različnih okoliščinah različen pomen.

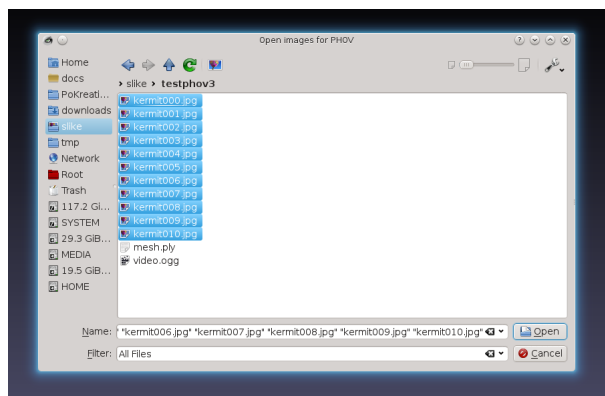


Slika 1: Gumb PHOV

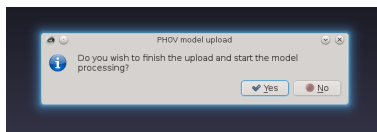
2.1 Nalaganje modela

Zaradi preprečevanja napadov je možno procesirati le en set fotografij na enkrat. To storimo s pritiskom gumba PHOV. Vmesnik, če dobi od streznika

dovoljenje, omogoči nalaganje fotografij. Po vsakem nalaganju nas vmesnik vpraša, če želimo končati. Če odgovorimo z NE, je mogoče ponovno nalagati sete fotografij, sicer pa vmesnik sporoči strežniku ukaz za procesiranje.



Slika 2: Nalaganje slik



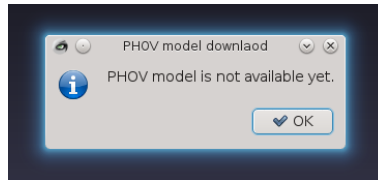
Slika 3: Dodatno nalaganje

2.2 Čakanje na rezultat

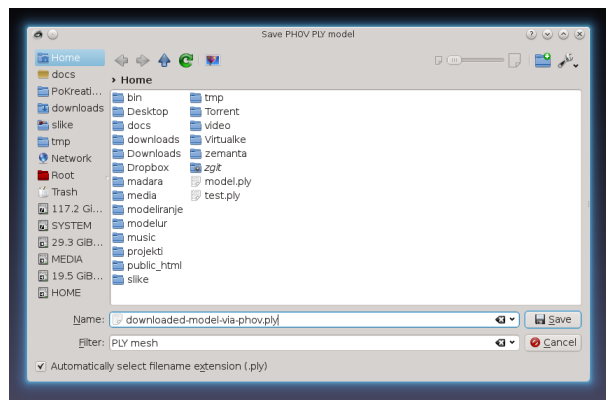
Zaradi omejitev pri razvoju v okolju Meshlab mora uporabnik sam zahtevati pregled rezultata. S klikom na tipko PHOV zahteva poizvedovanje na strežniku. Možne so tri različne posledice:

- sporočilo "model je še v obdelavi"
- sporočilo "obdelava ni bila uspešna", vse operacije se prekličejo, uporabnik mora poskusiti znova
- dialog za shranjevanje rezultata 3D modela

Ko prenesemo 3D model na računalnik, ga lahko po utečeni metodi uvozimo v projekt.



Slika 4: Čakanje na rezultat



Slika 5: Prenos modela



Slika 6: Pridobljeni 3D model

3 Namestitev

Ker orodje še ni na seznamu podprtih orodij, je potrebo za uporabo orodja program prevesti ročno na računalniku. Za prevajanje kode na računalniku so priložena navodila že znotraj izvirne kode programa Meshlab.

Za vključitev orodja je potrebno:

- v datoteko *meshlab_full.pro* dodati za vrstico, ki vsebuje *edit_arc3D*, dodati vrstico z vsebino: *meshlabplugins/edit_phov *
- v direktorij *meshlabplugins* je potrebno skopirati izvirno kodo orodja

4 Dokumentacija kode

Vsa logika programa se nahaja v datoteki *edit_phov.cpp*. Datoteka *edit_phov_factory.cpp* je nujno potrebna le za inicializacijo orodja. Vsebinske *header* datotek:

- *edit_phov_api.h.template* vsebuje iskalne nize (regex) in URL naslove do API-ja. Za delovanje jo je potrebno preimenovati in vnesti prave URL naslove.
- *edit_phov_factory.h* vsebuje meta podatke za nalaganje orodja.
- *edit_phov.h* vsebuje meta podatke za izvajanje kode.

4.1 Dokumentacija najpomembnejših funkcij

Najpomembnejše funkcije so naslednje:

- *StartEdit* - potek orodja
- *loadSettings* - nalaganje nastavitev iz INI datoteke
- *saveSettings* - shranjevanje nastavitev v INI datoteko
- *getPhovId* - pridobitev ID 3D modela
- *checkDownloadAvailable* - preveri, ali je 3D model že na voljo oz. če je prišlo do napake
- *downloadModel* - iz spleta prenese 3D model
- *uploadImages* - nalaganje fotografij
- *finishUpload* - zaključi nalaganje fotografij

- *deleteModel* - brisanje modela, kar pa trenutno še ni podprto

Vsi zahtevki se delajo na roke. Primer zahtevka za pridobitev ID:

```
void EditPhovPlugin::getPhovId() {
    QNetworkAccessManager mm(this); QEventLoop eventLoop;
    QUrl url(apiUrlGetId);
    QByteArray postData;
    postData.append("notify=mailto:example@email.com");
    QNetworkRequest request(url);
    request.setHeader(QNetworkRequest::ContentTypeHeader,
                     "application/x-www-form-urlencoded");
    QObject::connect(&mm, SIGNAL(finished(QNetworkReply*)),
                     &eventLoop, SLOT(quit()));
    QNetworkReply* reply = mm.post(request, postData);
    eventLoop.exec();
    if (reply->error() == QNetworkReply::NoError) {
        QRegExp rgx(rxId);
        int pos = rgx.indexIn(reply->readAll());
        if (pos > -1) {
            phovID = rgx.cap(1);
            saveSettings();
        } else {}
    } else {}
    delete reply;
}
```