

SVEUČILIŠTE U ZAGREBU
RUDARSKO-GEOLOŠKO-NAFTNI FAKULTET
ZAVOD ZA RUDARSTVO I GEOTEHNIKU

KREŠIMIR MALETIĆ
MATIJA PALIJAN
SILVIO ŠUBIĆ

RAZVOJ ALGORITMA ZA ODREĐIVANJE KUTA SIPANJA IZ
TRODIMENZIONALNIH MODELA

Zagreb, 2021.

Ovaj rad je izrađen na Zavodu za rudarstvo i geotehniku Rudarsko-geološko-naftnog fakulteta Sveučilišta u Zagrebu pod vodstvom doc. dr. sc. Marija Klanfara dipl. ing. i doc. dr. sc. Dubravka Domitrovića dipl. ing. predan je na natječaj za dodjelu Rektorove nagrade u akademskoj godini 2020./2021.

Izjava o postojanju povezanih radova

Ideja za ovaj rad začeta je na temelju iskustava u ispitivanjima provedenim unutar završnih radova istih studenata te su tako radovi tematski povezani. Za potrebe izrade rektorove nagrade provedena su potpuno nova ispitivanja, osmišljen je i napisan poseban algoritam te su provedene nove analize i izvedeni novi zaključci. Na temelju toga smatramo da ne postoji sličnost s povezanim radovima u temeljnim znanstvenim poglavljima.

Autori:

Krešimir Maletić

Matija Palijan

Silvio Šubić

Mentori:

Doc. dr. sc. Dubravko Domitrović

Doc. dr. sc. Mario Klanfar

POPIS I OBJAŠNJENJE KRATICA

Stl - StereoLithography

IDLE - Integrated Development and Learning Environment

POPIS KORIŠTENIH OZNAKA I JEDINICA

Oznaka	Jedinica	Opis
α	-	kut sipanja
R	m	radijus baze korigiranog stošca
r	m	gornji radijus korigiranog stošca
h	m	visina između baze i vrha korigiranog stošca

SADRŽAJ

1. UVOD	1
2. KUT SIPANJA	2
2.1. Metode mjerenja kuta sipanja	3
2.1.1. Metoda kutomjera	3
2.1.2. Metoda promjera i visine	4
2.1.3. Metoda 2D modela.....	5
2.1.4. Metoda 3D modela.....	7
2.2. Utjecaj oblika hrpe na izmjerene vrijednosti	10
3. PYTHON.....	12
3.1. Općenite informacije o Python-u	12
3.2. Sučelje programa Python (IDLE)	12
3.3. Moduli rada unutar Python-a	14
4. ALGORITAM.....	16
4.1. Općenito o algoritmu	16
4.2. Blok dijagram	17
4.3. Opis algoritma	19
5. PROVEDENA ISPITIVANJA.....	25
5.1. Opis ispitivanja	25
5.2. Materijali korišteni u istraživanjima	27
6. ANALIZA REZULTATA.....	29
6.1. Izrada 3D modela.....	29
6.2. Obrada <i>stl</i> datoteka u Pythonu	30
7. ZAKLJUČAK	36
8. POPIS LITERATURE	37
SAŽETAK.....	39
SUMMARY	40

POPIS SLIKA

Slika 2-1 Kut sipanja materijala.....	2
Slika 2-2 Kutomjer.....	3
Slika 2-3 Digitalni kutomjer.....	4
Slika 2-4 Mjerenje visine hrpe materijala digitalnim uređajem (Mistry i dr., 2017).....	5
Slika 2-5 Sučelje programa <i>ImageJ</i> sa učitanoj fotografijom.....	6
Slika 2-6 Obrada podataka u Microsoft Excelu	6
Slika 2-7 Princip stereofotogrametrije (Kostrić, 2018).....	7
Slika 2-8 Model stošca (Wójcik i dr., 2017).....	8
Slika 2-9 Korekcija stošca (Wójcik i dr., 2017).....	9
Slika 2-10 Korigirani stožac (Wójcik i dr., 2017).....	9
Slika 2-11 Ovisnost vrijednosti kuta sipanja o broju ispitivanja (Wójcik i dr., 2017).....	10
Slika 2-12 Realna hrpa materijala	11
Slika 2-13 Pravilna (referentna) stožasta hrpa	11
Slika 3-1 IDLE sučelje programa Python	12
Slika 3-2 Komunikacija s IDLE-om korištenjem <i>print</i> naredbe	13
Slika 3-3 Prikaz načina rada naredbe <i>Restart Shell</i>	14
Slika 3-4 Grafičko sučelje nakon pozivanja modula Tkinter.....	15
Slika 4-1 <i>Stl</i> datoteka (ITS, 2021).....	17
Slika 4-2 Blok dijagram formiranog algoritma.....	18
Slika 4-3 Algoritam 1/6.....	19
Slika 4-4 Algoritam 2/6.....	20
Slika 4-5 Algoritam 3/6.....	21
Slika 4-6 Algoritam 4/6.....	22
Slika 4-7 Algoritam 5/6.....	23
Slika 4-8 Algoritam 6/6.....	24
Slika 5-1 Dio Laboratorija za rudarsku mehanizaciju i arhitektonsko-građevni kamen.....	25
Slika 5-2 Shema ispitivanja metodom cilindra (Kostrić, 2018).....	26
Slika 5-3 Stožasta hrpa materijala sačinjena od suhih kukuruznih zrna (Rackl i Grötsch, 2018).....	28
Slika 5-4 Stožasta hrpa materijala sačinjena od drvene piljevine (Rackl i Grötsch, 2018)	28
Slika 6-1 <i>Agisoft PhotoScan</i> sa učitanim fotografijama	29
Slika 6-2 3D model izrađen u programu <i>Agisoft PhotoScan</i>	30

Slika 6-3 Grafički prikaz ovisnosti iznosa kuta sipanja o broju zaglađivanja i rezanja stožaste hrpe na primjeru šljunka	31
Slika 6-4 Grafički prikaz ovisnosti iznosa kuta sipanja o broju zaglađivanja i rezanja stožaste hrpe na primjeru sitnog dolomita	31
Slika 6-5 Grafički prikaz ovisnosti iznosa kuta sipanja o broju zaglađivanja i rezanja stožaste hrpe na primjeru kukuruznih zrna	32
Slika 6-6 Grafički prikaz ovisnosti iznosa kuta sipanja o broju zaglađivanja i rezanja stožaste hrpe na primjeru drvene piljevine.....	33
Slika 6-7 Izrada referentne hrpe (Kostrić, 2018)	34
Slika 6-8 Grafički prikaz ovisnosti iznosa kuta sipanja o broju zaglađivanja i rezanja na primjeru referentne hrpe	35

1. UVOD

U svijetu računarstva jedan od važnijih pojmova je algoritam. Algoritmi predstavljaju temelj za razumijevanje računarstva, ali nisu strogo ograničeni na tehničke aktivnosti, već su prožeti kroz svakodnevne situacije čovjekova života. Jedan od prvih algoritama, većini ljudi poznat, naziva se Euklidov algoritam. Služi za određivanje najveće zajedničke mjere dvaju cijelih brojeva, dužina ili polinoma. Euklidov algoritam je i danas vrlo koristan te se i dalje uspješno primjenjuje (Hrvatska enciklopedija, mrežno izdanje, 2021).

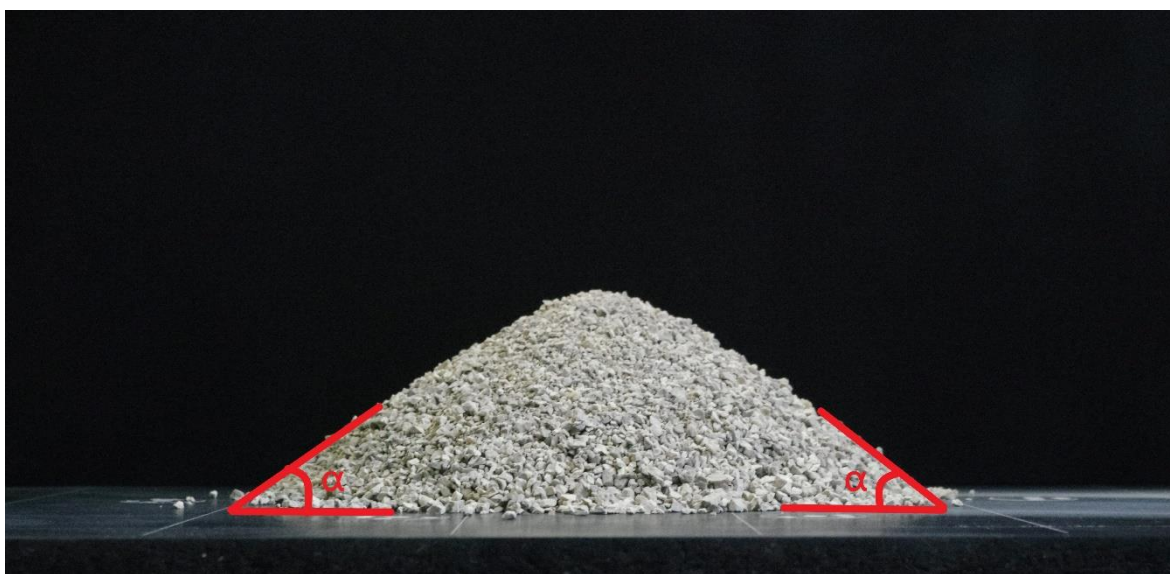
Algoritam je uputa kako obaviti neki zadatak, odnosno njime se opisuje točan redoslijed kojim želimo obaviti taj zadatak. Čine ga točno određene riječi pomoću kojih pišemo program. Napisani program sastavljen je od niza naredbi koje moraju biti razumljive računalu. Da bi program na kraju bio funkcionalan, moramo unaprijed predvidjeti što želimo dobiti kao konačan ishod. Kako bismo napisali neki program moramo odabrati programski jezik.

Za potrebe razvoja algoritma za određivanje kuta sipanja materijala iz trodimenzionalnih modela odabran je programski jezik Python. Međutim, da bi navedeni algoritam imao svoju funkciju, prethodno je potrebno obaviti laboratorijska ispitivanja na uzorcima različitih vrsta materijala kako bi, u konačnici, dobili trodimenzionalne modele hrpa. Kao što je već spomenuto, osnovni zadatak ovog rada je razviti algoritam koji će nam služiti kako bi odredili kut sipanja na temelju trodimenzionalnih modela. U cilju je dokazati da je navedeni algoritam primjenjiv za sve vrste materijala te će se kao takav moći koristiti u raznim područjima industrije. Korištenjem algoritma značajno bi trebala porasti točnost određivanja kuta sipanja materijala jer se radi o računalnoj metodi, za razliku od ručnih metoda gdje rezultat ovisi o promatraču.

2. KUT SIPANJA

Prirodni kut sipanja materijala (eng. Angle of repose) je onaj kut koji materijal tvori s horizontalnom podlogom (Kujundžić, 2010). Na slici 2-1 simbolom α označen je kut sipanja. Materijal poprima oblik stošca, a kut sipanja može iznositi od 0° do 90° . Morfologija znatno utječe na različitost kuta sipanja pa će tako kut sipanja ovisiti o hrapavosti podloge, veličini zrna, obliku zrna, vlažnosti, temperaturi, sastavu i stanju materijala te o načinu sipanja na podlogu.

Česta je primjena kuta sipanja u različitim područjima industrije. Kut sipanja možemo koristiti kod proračuna obujma prebacivanja materijala, dimenzioniranja transportnih traka, određivanja koeficijenta punjenja lopate utovarača, dimenzioniranja deponija, odabira sanduka kamiona, ispitivanja nasipnosti betona, projektiranja i dimenzioniranja silosa, spremnika ili rotacijskih peći, problema stabilnosti kosina, određivanja sipkosti granuliranih prahova koji se koriste u proizvodnji i premazivanju lijekova, ali i u mnogim drugim slučajevima (Beakawi i Baghabra, 2018). Potrebno je razlikovati statički kut sipanja od dinamičkog kuta sipanja materijala. Kada određeni materijal sipamo s veće visine doći će do klizanja zrna materijala po obodu stošca pa tako nastali kut nazivamo kut sipanja u trenutku gibanja ili dinamički kut sipanja. Ako materijal sipamo s male visine pri čemu ne dolazi do klizanja zrna materijala po obodu stošca, već se zbog trenja između zrna formira hrpa kod koje zrna miruju, u tom slučaju govorimo o maksimalnom kutu sipanja ili statičkom kutu sipanja materijala.



Slika 2-1 Kut sipanja materijala

2.1. Metode mjerenja kuta sipanja

Postoji niz metoda za određivanje kuta sipanja, međutim ni jedna od metoda nije univerzalna, a razlog tome su brojni čimbenici i parametri koji utječu na način formiranja hrpe materijala. Neki od njih su veličina i oblik zrna materijala, vlažnost i temperatura materijala, glatkoća podloge na koju se sipa materijal, zatim radi li se o klasiranom ili neklasiranom materijalu, ali i metoda kojom se formira hrpa stožastog oblika (Kopunić, 2019).

2.1.1. Metoda kutomjera

Kutomjer (Slika 2-2) je osnovni instrument za mjerenje kutova, a najčešće se primjenjuje u području geometrije. Da bi kutomjerom izmjerili kut između dvaju pravaca, potrebno je njegovo središte postaviti u sjecište tih pravaca i paralelno sa jednim od pravaca. Kut se očitava sa skale kutomjera gdje drugi pravac dodiruje određenu vrijednost na skali. Napretkom tehnologije razvile su se razne vrste kutomjera, a jedan od njih je digitalni kutomjer koji se često primjenjuje u građevinarstvu, geodeziji i sličnim strukama (Slika 2-3). Kod digitalnih kutomjera kut očitavamo na displayu jednog od dva kraka od kojih se sastoji digitalni kutomjer (Kostrić, 2018).



Slika 2-2 Kutomjer



Slika 2-3 Digitalni kutomjer

2.1.2. Metoda promjera i visine

Indirektan način dobivanja kuta sipanja postiže se korištenjem trigonometrije uz prethodno mjerenje vrijednosti promjera i visine netom prije dobivene hrpe materijala. Za to je potrebno postaviti primjerice podnu šablonu na ravnu podlogu pomoću koje se mjerenjem mogu odrediti promjeri baze stošca u dva međusobno okomita smjera. Za određivanje visine stošca koristi se pomično mjerilo. Nakon mjerenja tih dvaju parametara, iz presjeka ravnine okomite na podlogu po stošcu dobiju se dva pravokutna trokuta uz pretpostavku idealizacije da je hipotenuza trokuta pravac koji najbolje opisuje plašt stožaste hrpe. Dobivena dva trokuta, odnosno kutovi koje hipotenuze zatvaraju sa katetama koje su paralelne podlozi predstavljaju dva kuta sipanja. Oni u pravilu nisu jednaki upravo zbog nesavršenosti hrpe, dakle jedna kateta koja predstavlja visinu zajednička je za oba trokuta dok druga kateta pojedinog trokuta predstavlja radijus baze stošca. U tom slučaju kut sipanja računamo prema slijedećoj jednadžbi:

$$\alpha = \tan^{-1} \left(\frac{h}{r} \right) \quad (2-1)$$

gdje su: α - kut sipanja
 h - visina stošca
 r - radijus stošca

Mistry i dr. (2017) na ovakav način proveli su istraživanja s ciljem određivanja kuta sipanja na farmaceutskim materijalima što govori o širokoj i raznolikoj primjeni navedenog fizikalnog svojstva, a čijim je poznavanjem moguće riješiti širok spektar problema koji su ljudska svakodnevnica. Na slici 2-4 prikazan je digitalni uređaj pomoću kojega je moguće odrediti visinu hrpe materijala.

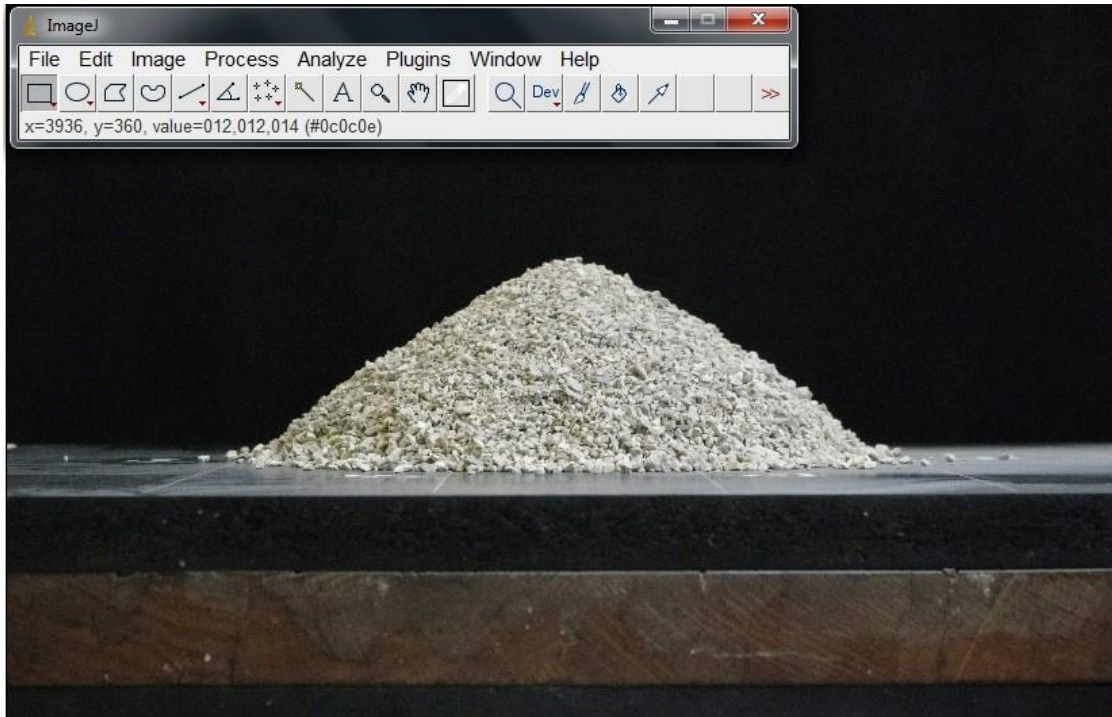


Slika 2-4 Mjerenje visine hrpe materijala digitalnim uređajem (Mistry i dr., 2017)

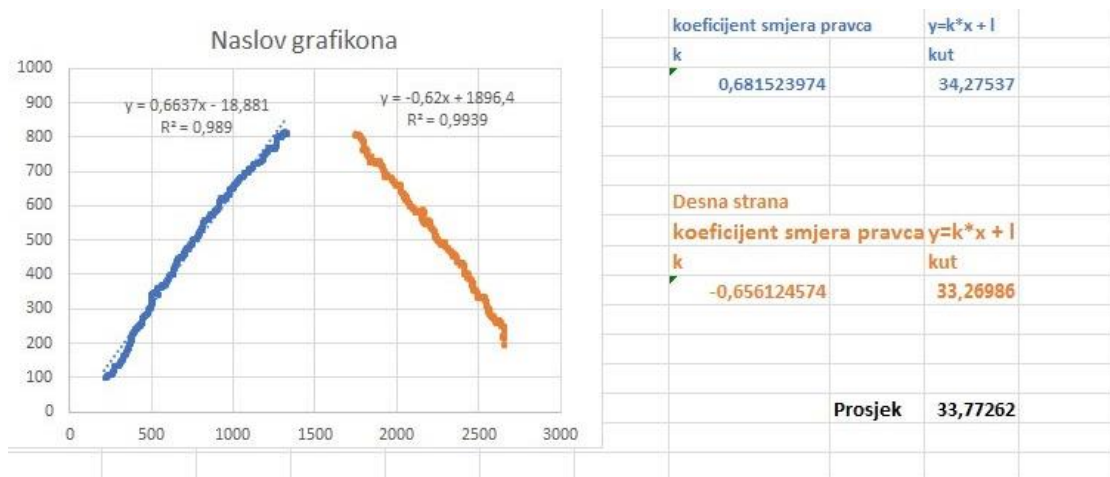
2.1.3. Metoda 2D modela

Metoda 2D modela spada u nešto novije metode određivanja kuta sipanja pa zahtjeva određena znanja iz područja obrade fotografija i podataka u računalnim programima. Metoda je zasnovana na digitalnim fotografijama. Da bi stvorili digitalnu fotografiju potrebno je koristiti digitalni fotoaparati koji zabilježene digitalne fotografije sprema u memoriju u obliku podataka te se one kao takve prenose na računalo gdje se odvija obrada fotografija u jednom od računalnih programa. Neki od računalnih programa za obradu fotografija su *GIMP*, *Photoescape*, *Photoshop CC*, *Corel PaintShop* i mnogi drugi (Urličić, 2017.). Za potrebe određivanja kuta sipanja češća je primjena jednostavnijih računalnih programa kao što je *ImageJ*. Sučelje programa *ImageJ* sa alatnim trakama nakon učitane fotografije prikazano je na slici 2-5. Kod metode 2D modela postavljenu hrpu materijala potrebno je poslikati iz najmanje dva različita kuta koja su međusobno okomita te se obradom fotografije i podataka koje ona pruža (npr. x-y koordinate točaka) u nekom računalnom programu za tablično računanje može dobiti kut sipanja, međutim za takvo što potrebno je uložiti dosta vremena. Nakon što se računalnim programom *ImageJ* odrede x i y koordinate hrpe, one se eksportiraju u jedan od programa za obradu podataka (npr. *Microsoft Excel*) gdje se iz koordinata određuju nagibi pravaca koji najbolje naliježu na rub hrpe, te se zatim odredi kut sipanja. Ovakav način određivanja kuta sipanja prikazan je na

slici 2-6. Mistry i dr. (2017) koristili su se takvim načelima u vlastitim istraživanjima određivanja kuta sipanja gdje su umjesto digitalnog fotoaparata koristili web kameru kako bi zabilježili hrpu u obliku stošca, a kao program za vizualizaciju koristili su LabVIEW.



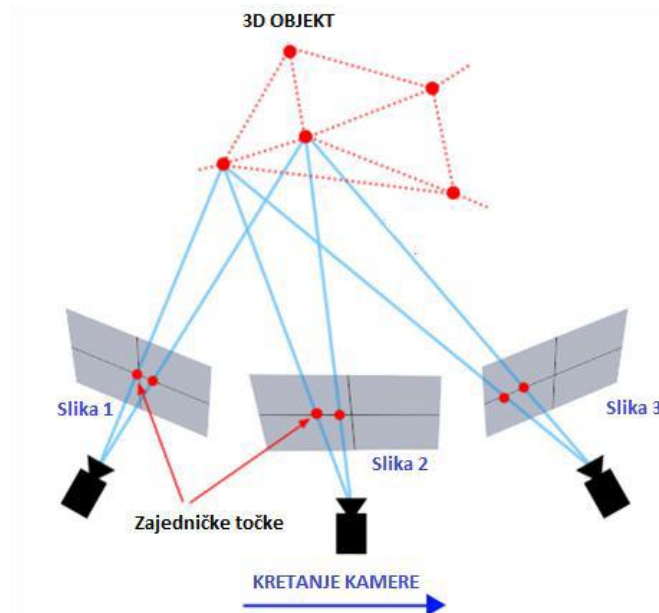
Slika 2-5 Sučelje programa *ImageJ* sa učitanim fotografijom



Slika 2-6 Obrada podataka u Microsoft Excelu

2.1.4. Metoda 3D modela

Metoda 3D modela temeljena je na stereofotogrametriji, odnosno obliku fotogrametrije koja predstavlja tehniku dobivanja prostornih informacija nekog objekta na temelju snimanja i obrade stereoparova koji su dobiveni na različitim mjestima snimanja istog objekta (Hrvatska enciklopedija, 2018). Kod fotografiranja sa više različitih stajališta, cilj je da se na svakoj fotografiji prepoznaju zajedničke točke, nakon čega se triangulacijom pomoću projekcijskih linija procjenjuju 3D koordinate točaka, odnosno njihova lokacija (Slika 2-7). Prilikom fotografiranja objekta potrebno je biti veoma oprezan i obuhvatiti željeni objekt na sljedećoj slici tako da određeni detalji budu prisutni na ostalim fotografijama u iznosu 60-80%, a da se pritom izbjegnu moguće refleksije te objekti u pokretu (Radiona Wiki, 2018).



Slika 2-7 Princip stereofotogrametrije (Kostrić, 2018)

Wójcik i dr. (2017) odlučili su predložiti novu metodu za precizno mjerenje prirodnog kuta sipanja materijala pomoću fotogrametrije. Fotogrametrija snimaka postala je najčešće korištena metoda za određivanje koordinata u prostoru zbog mogućnosti automatske obrade slike i održavanja točnosti piksela. Mjereni objekt od kojeg je napravljeno nekoliko fotografija promatra se kao 3D model koji se kasnije analizira na temelju preciznog određivanja koordinata. Wójcik i dr. (2017) postavili su na tvrdu podlogu grafički papir na koji su ucrtane točke koje su definirane koordinatama kako bi odredili orijentaciju pojedinih fotografija. Kamera se postavlja na fiksnoj udaljenosti od hrpe materijala koja je u obliku konusa, a rotirana je za 18° unutar 20 različitih pozicija. Načinjene fotografije importirane

su u računalni program *Agisoft PhotoScan Professional* u kojem se odvija obrada te se procjenjuje njihova orijentacija. Generirani parovi slika (stereoparovi) georeferencirani su pomoću markera sa slika na temelju čega je postavljen koordinatni sustav. Nakon toga, na temelju fotografija formira se oblak točaka koji predstavlja ispitivani objekt. Za određivanje geometrijskih parametara 3D oblaka točaka korišten je program *Leica Cyclone* koji ima mogućnost aproksimacije oblaka točaka određenim geometrijskim tijelom, u ovom slučaju konusom. Realna hrpa materijala ima tendenciju rasipanja na dnu hrpe, dok je na vrhu zaobljena (slika 2-8). Na temelju tog saznanja, Wójcik i dr. (2017) odlučili su zanemariti navedene dijelove hrpe kako bi dobili što precizniji kut sipanja. S donje strane zanemaruje se 10-20% hrpe, dok se od vrha hrpe zanemaruje 20-30%, što je vidljivo na slici 2-9. Kut sipanja se prema slici 2-9 računa pomoću izraza (Wójcik i dr., 2017):

$$\alpha_{n,m} = \tan^{-1} \frac{h_{n,m}}{R_{n,m} - r_{n,m}} \quad (2-2)$$

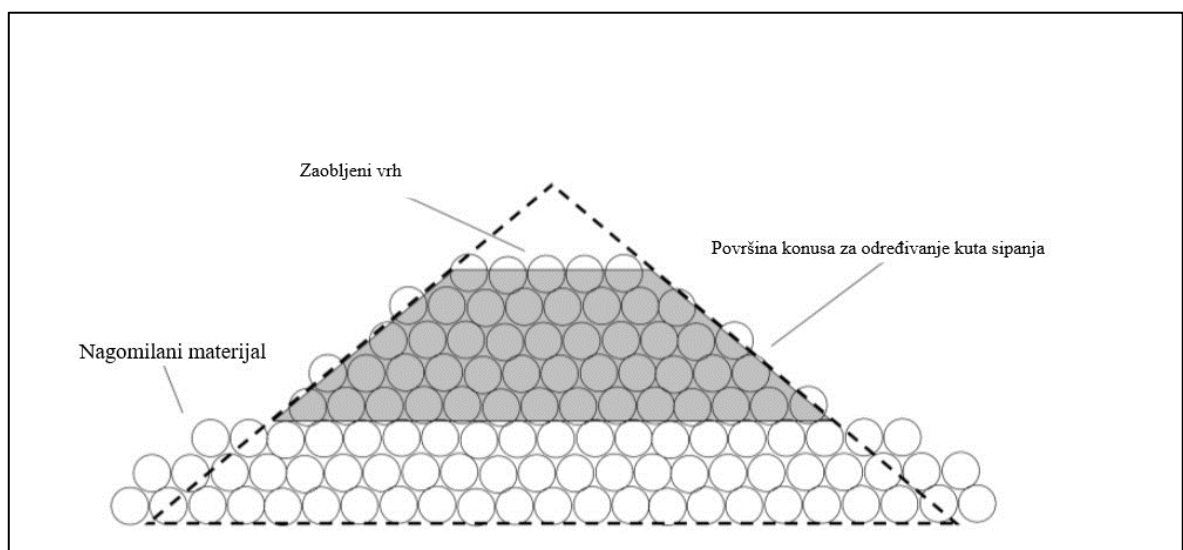
gdje je:

$\alpha_{n,m}$ – kut sipanja

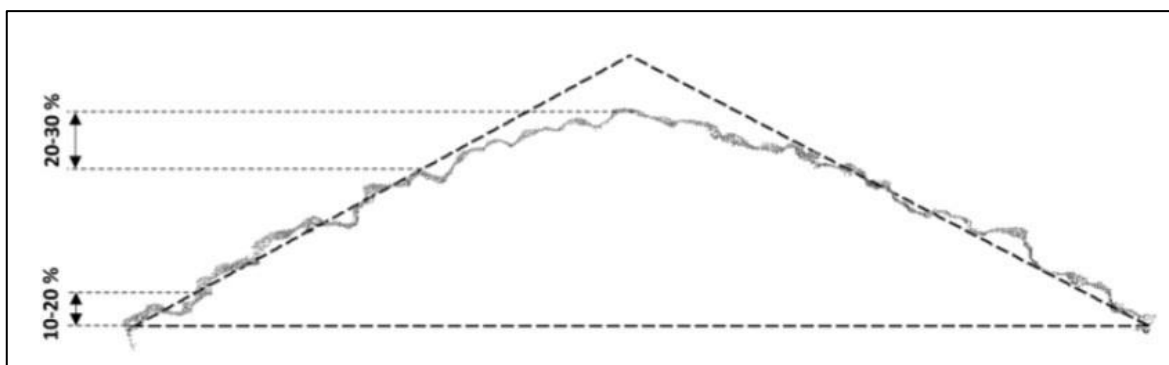
$R_{n,m}$ – radijus baze korigiranog stošca

$r_{n,m}$ – gornji radijus korigiranog stošca

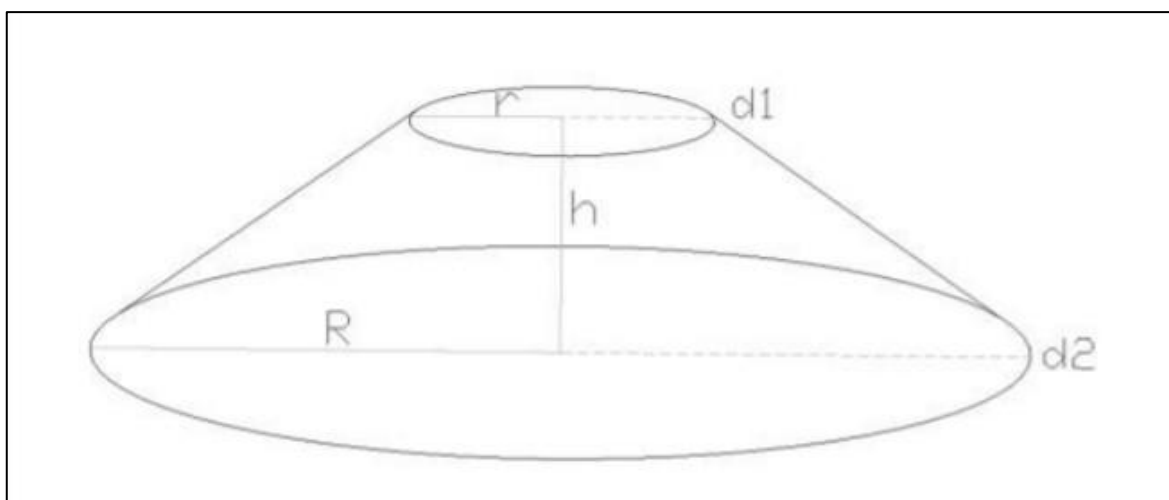
$h_{n,m}$ – visina između baze i vrha korigiranog stošca



Slika 2-8 Model stošca (Wójcik i dr., 2017)

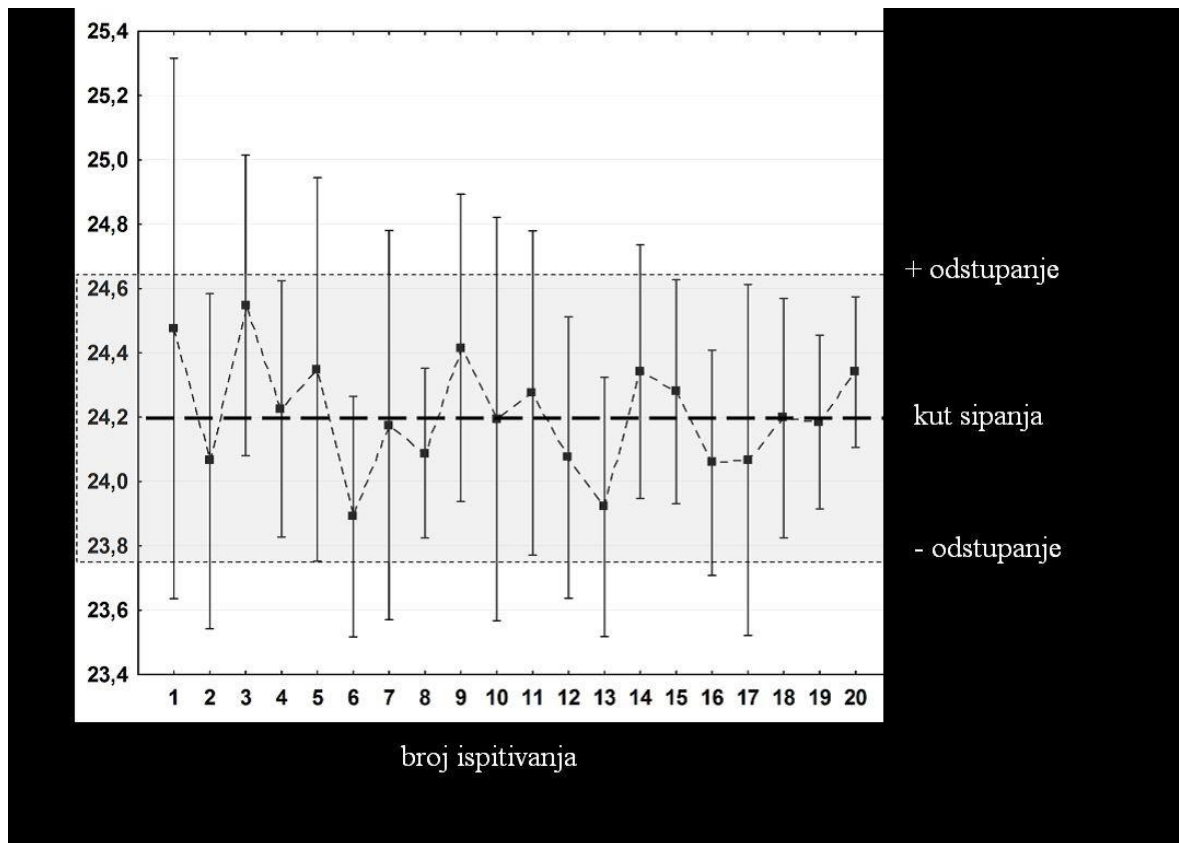


Slika 2-9 Korekcija stošca (Wójcik i dr., 2017)



Slika 2-10 Korigirani stožac (Wójcik i dr., 2017)

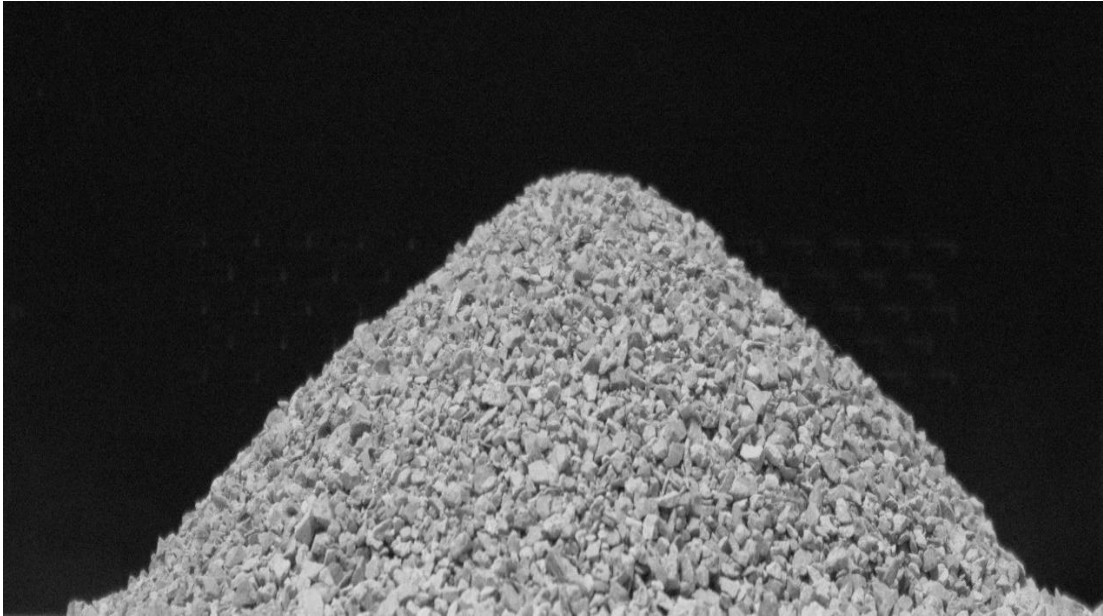
Wójcik i dr. (2017) proveli su istraživanja na više od 600 nezavisnih mjerenja fotogrametrijskim metodama kako bi došli do novih saznanja u vezi određivanja kuta sipanja čime se došlo do zaključka kako je metoda pogodna za različite materijale koji posjeduju određen granulometrijski sastav, tj. nisu koherentni. Utvrdili su i ovisnost odstupanja vrijednosti kuta sipanja o broju ispitivanja što je prikazano na slici 2-11. Sa slike je vidljivo kako se odstupanja od prave vrijednosti smanjuju sa povećanjem broja ispitivanja, a to se načelo koristilo i pri laboratorijskim ispitivanjima koja su provedena.



Slika 2-11 Ovisnost vrijednosti kuta sipanja o broju ispitivanja (Wójcik i dr., 2017)

2.2. Utjecaj oblika hrpe na izmjerene vrijednosti

Kod određivanja kuta sipanja velik utjecaj ima oblik hrpe materijala. Kad materijal sipamo na podlogu s određene visine, ovisno o određenim parametrima (veličina i oblik zrna, kohezija i stupanj prekonsolidacije), on će na podlozi poprimiti oblik koji je najsličniji stošcu (Rackl i dr., 2017). Međutim, realna hrpa materijala ima tendenciju rasipanja na dnu dok se na vrhu napravi određeno zaobljenje kao što prikazuje slika 2-12. Ukoliko bi kut sipanja mjerili na takvoj hrpi, iznos kuta bio bi manji u odnosu na pravilan stožac jer je prilikom mjerenja potrebno obuhvatiti cijelu hrpu. Zbog toga, kut sipanja određujemo na hrpi gdje se od vrha i dna hrpe oduzme određeni postotak hrpe kako bi dobili realan iznos kuta sipanja. Na pravilnim hrpama (stožac) kut sipanja jednak je u bilo kojoj točki na obodu stošca (slika 2-13).



Slika 2-12 Realna hrpa materijala



Slika 2-13 Pravilna (referentna) stožasta hrpa

3. PYTHON

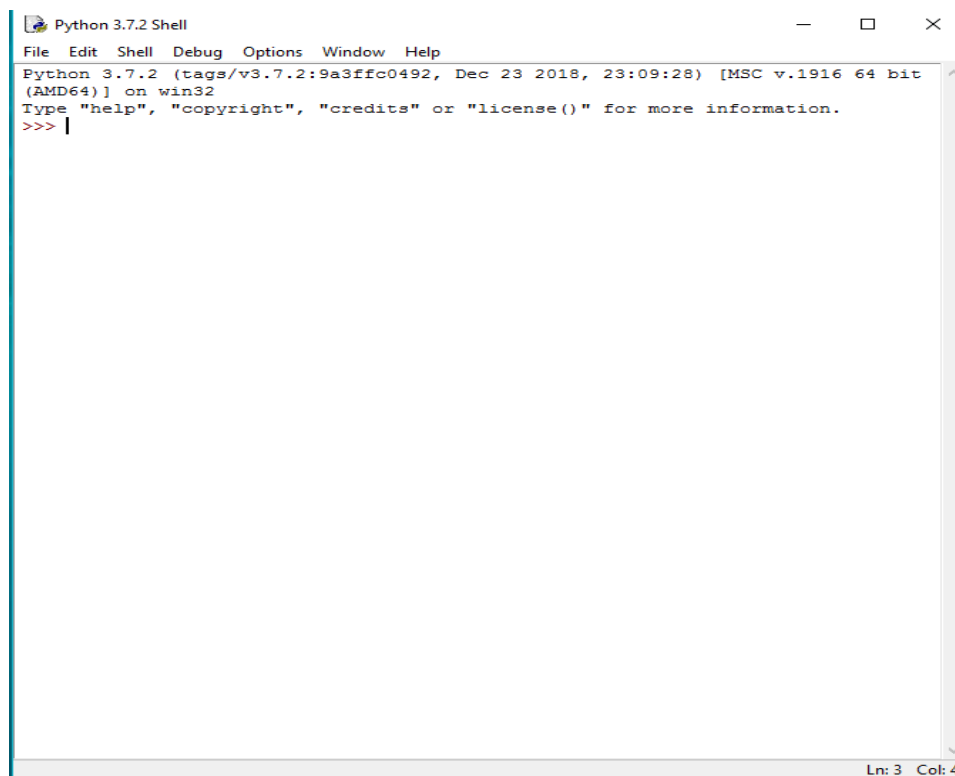
3.1. Općenite informacije o Python-u

Python je objektno orijentiran programski jezik koji se primarno koristi u podatkovnoj znanosti, razvoju aplikacija te kao jezik kojim se često povezuju već napravljene komponente programa (Hercigonja, 2018).

Programski jezik stvorio je Guido van Rossum 1990. godine i zamišljen je kao nasljednik jezika pod nazivom ABC (van Rossum, 2018). Python se smatra jednim od najlakših programa za učenje zbog svoje jednostavne sintakse. Velika prednost ovog programskog jezika je lagano otklanjanje „errora“, tj. proces editiranja-testiranja-otklanjanja grešaka je vrlo pojednostavljen i ubrzan s obzirom da nema kompilacijskog koraka.


3.2. Sučelje programa Python (IDLE)

IDLE je skraćenica za „*Integrated Development and Learning Environment*“, što prevedeno znači integrirano razvojno okruženje. Koristi se za izvršavanje primarno jednostavnijih naredbi te za modifikaciju koda. Slika 3-1 prikazuje otvoreno IDLE sučelje.



Slika 3-1 IDLE sučelje programa Python

„Shell“ je standardni operacijski modul za Python IDLE i on se otvara klikom na ikonu IDLE-a. *Shell* nudi trenutno izvršenje upisane naredbe te ispisuje rezultat na ekran. Nakon toga kreće sa izvršavanjem sljedeće napisane naredbe i postupak se ponavlja do posljednje napisane naredbe. Slika 3-2 prikazuje najprimitivniji oblik komunikacije s IDLE-om preko *print* naredbe.

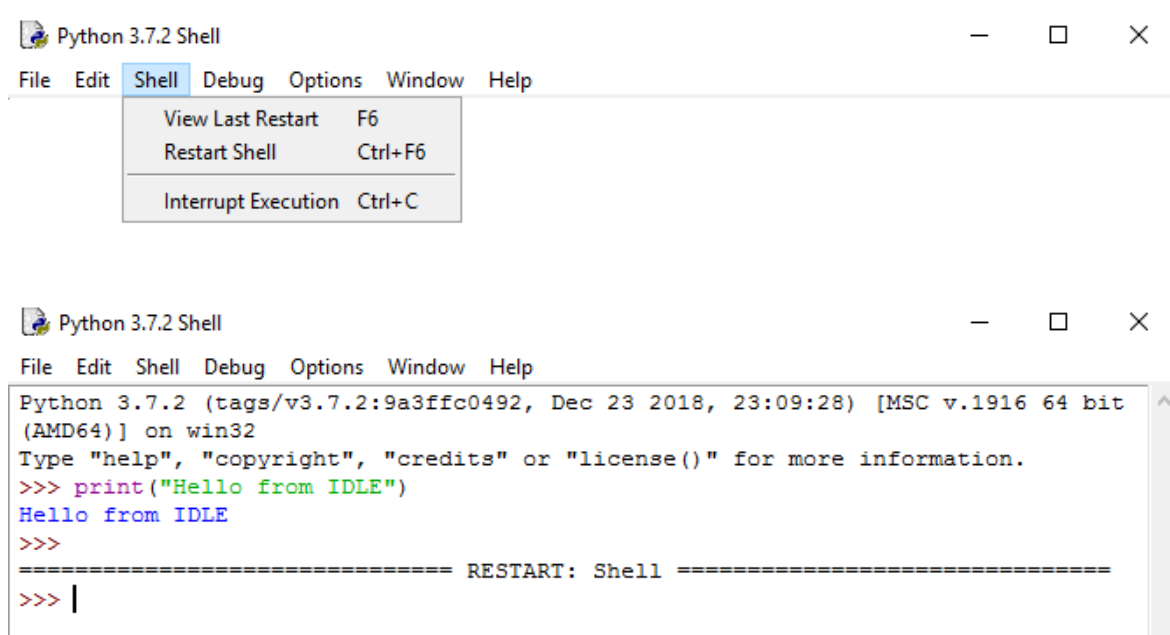


```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello from IDLE")
Hello from IDLE
>>> |
```

Ln: 5 Col: 4

Slika 3-2 Komunikacija s IDLE-om korištenjem *print* naredbe

Preko naredbe „*Restart Shell*“ poništava se trenutno stanje *Shella* i program se ponaša kao da je ponovno pokrenut. Slika 3-3 prikazuje rezultat opisanog postupka.



Slika 3-3 Prikaz načina rada naredbe *Restart Shell*

3.3. Moduli rada unutar Python-a

Modul unutar Pythona definiran je kao skup funkcija, konstanti i razreda koji se može pozvati u program u bilo kojem trenutku (van Rossum, 2018). Postoje prethodno definirani moduli koji dolaze u paketu s Pythonom, ali moguće je i samostalno definiranje modula. Module je moguće koristiti na dva načina: najavljuvanjem korištenja i uključivanjem u program. Najavljuvanje korištenja se ostvaruje pomoću ključne riječi *import*. Kod ovakvog oblika korištenja modula pozivaju se sve funkcije, konstante i razredi koje sadrži pozvani modul. Mana ovakvog načina korištenja jest potreba da se prije svakog korištenja modul ponovo poziva. Uključivanje u program se provodi uz pomoć dvije ključne riječi, a to su *import* i *from*. Naredbom *from* se poziva modul koji će se koristiti, a *import* se koristi odabir funkcija iz odabranog modula. Kod ovakvog načina korištenja nije potrebno naknadno pozivati modul. U svrhu izrade ovog rada korištena su četiri modula, a to su: OS, TKINTER, NUMPY, MATH i MATHPLOTLIB.

OS – ovaj modul omogućava interakciju s operativnim sustavom računala na kojem se koristi. Pomoću naredbe *open()* program može pročitati ili ispisati traženu datoteku, nudi se mogućnost stvaranja privremenih, temporary, datoteka i još mnoge opcije s kojima je moguće manipulirati datotekama na računalu.

TKINTER – modul koji se koristi za jednostavno i lagano kreiranje grafičkog korisničkog sučelja. Nakon poziva Tkinter modula prikazuje se grafičko sučelje sa slike 3-4. Tkinter nudi 15-ak widgeta preko kojih je moguća komunikacija s korisnikom. Neke od mogućnosti interakcije su: crtanje grafičkih oblika, ispisivanje jednog ili više redova teksta, interaktivni gumbi i slično (Amos, 2020).



Slika 3-4 Grafičko sučelje nakon pozivanja modula Tkinter

NUMPY – koristi se za znanstveno računanje u Python-u, a primarna upotreba je u radu s nizovima. Modul omogućuje provođenje raznih operacija s nizovima uz njihovo stvaranje, numeriranje i sortiranje. U Numpy modulu je moguća i integracija koda iz drugih programskih jezika, primarno C i C++.

MATH – vjerojatno najčešće korišten modul u Python-u. Modul nudi rad s trigonometrijskim, eksponencijalnim, logaritamskim i mnogim drugim funkcijama. Pozivom ovog modula program poznaje i vrijednosti dvaju konstanti, a to su π i Eulerova konstanta.

MATPLOTLIB – modul najčešće korišten za stvaranje statičkih, animiranih i interaktivnih vizualizacija. U svrhu izrade ovog rada, modul je korišten za crtanje grafikona prikazanih u poglavlju 6.

4. ALGORITAM

4.1. Općenito o algoritmu

Formirani algoritam računa prosjek nasipnog kuta iz trianguliranih modela učitanih u obliku *stl* datoteka. Program provodi i proces zaglađivanja modela, *smoothing*, onoliko puta koliko se navede pri pisanju naredbe. Još jedna karakteristika algoritma jest da automatski reže vrh i dno hrpe za odabrani postotak. Taj postupak se provodi s ciljem poboljšanja rezultata uzimajući u obzir da su to dijelovi hrpe koji najčešće negativno djeluju na rezultat.

Od iznimne je važnosti da su modeli u spomenutom *stl* formatu jer je algoritam napisan na način da je jedino takve formate u mogućnosti ispravno interpretirati. X, Y i Z koordinate razlikuje prema njihovom položaju u samoj datoteci (Tan i dr., 2020). Primjer jedne datoteke u *stl* formatu je prikazan na slici 4-1.

```

solid DataSet1
  facet normal 0.000000e+000 0.000000e+000 1.000000e+000
    outer loop
      vertex 2.114699e+002 1.089402e+002 0.000000e+000
      vertex 2.352096e+002 1.437085e+002 0.000000e+000
      vertex 1.798614e+002 1.305223e+002 0.000000e+000
    endloop
  endfacet
  facet normal 0.000000e+000 0.000000e+000 1.000000e+000
    outer loop
      vertex 1.798614e+002 1.305223e+002 0.000000e+000
      vertex 2.352096e+002 1.437085e+002 0.000000e+000
      vertex 2.000674e+002 1.837107e+002 0.000000e+000
    endloop
  endfacet
  facet normal 0.000000e+000 0.000000e+000 1.000000e+000
    outer loop
      vertex 1.798614e+002 1.305223e+002 0.000000e+000
      vertex 2.000674e+002 1.837107e+002 0.000000e+000
      vertex 1.617475e+002 1.711867e+002 0.000000e+000
    endloop
  endfacet
  facet normal 0.000000e+000 0.000000e+000 1.000000e+000

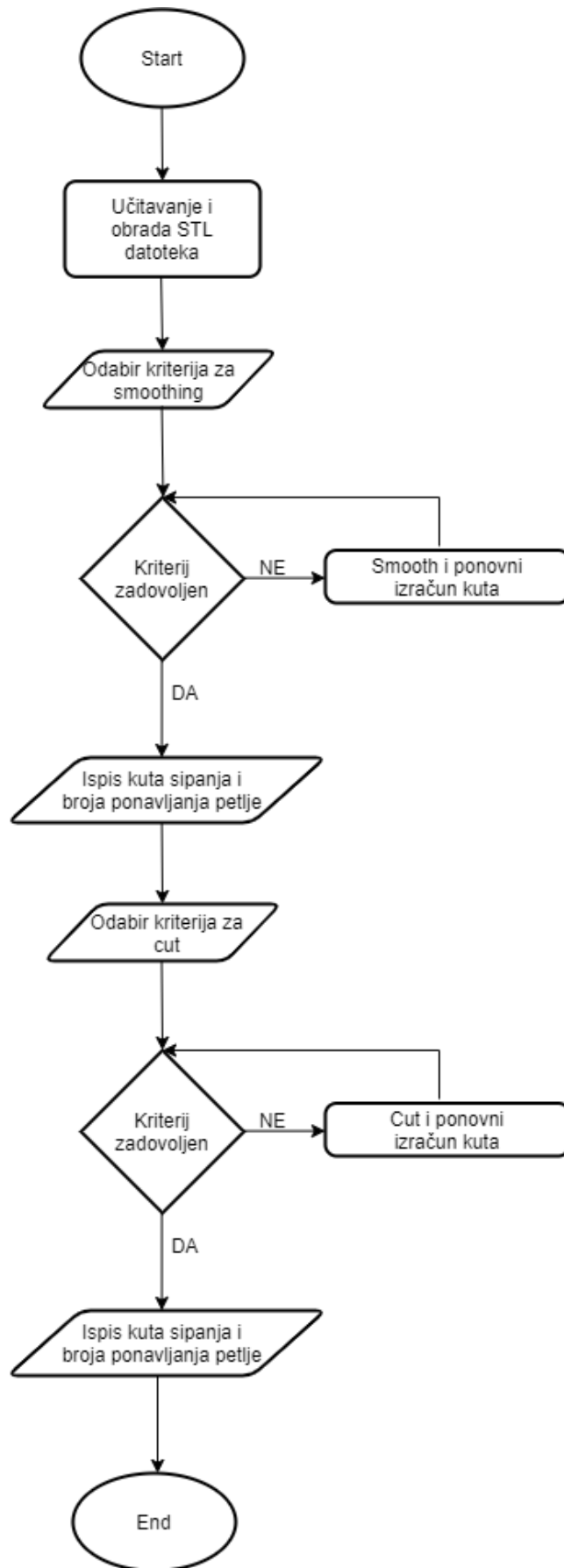
```

Slika 4-1 *Stl* datoteka (ITS, 2021)

Uz zaglađivanje i rezanje dijelova modela, najbitnija komponenta algoritma jest izračun nasipnog kuta za koji se koristi metoda težinskog prosjeka površina trianguliranih trokuta. Nakon izračuna kuta, program nudi opciju vizualizacije dobivenih rezultata u obliku dijagrama ovisnosti kuta sipanja o postotku odrezanog modela ili stupnja zaglađivanja.

4.2. Blok dijagram

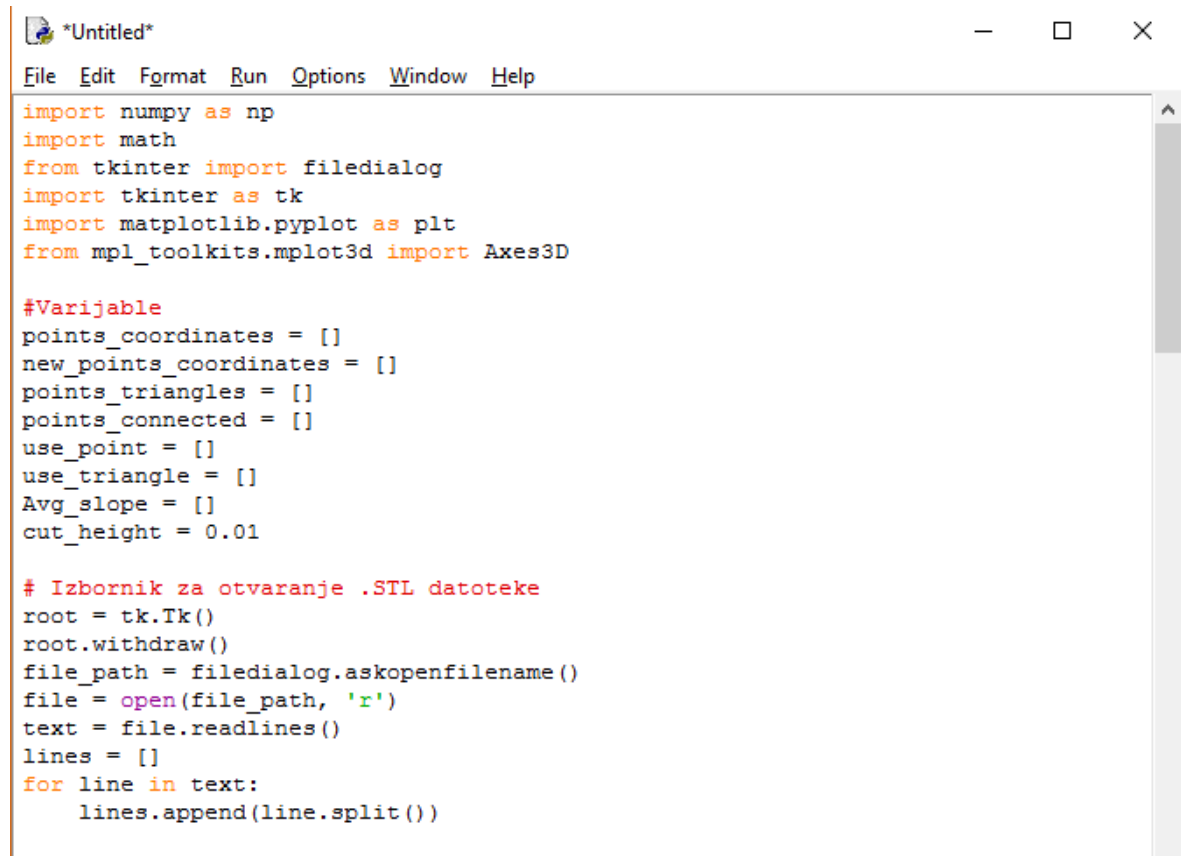
Na slici 4-2 je prikazan blok dijagram koji prikazuje princip prema kojem napravljeni algoritam računa kut sipanja. Program nudi opciju da se proizvoljno odabere kriterij (točnost) na kojem algoritam prekida provođenje *smooth* i *cut* funkcija te kao povratnu informaciju ispisuje dobivenu vrijednost kuta i broj koraka proveden unutar petlje.



Slika 4-2 Blok dijagram formiranog algoritma

4.3. Opis algoritma

U ovom podpoglavlju napravljen je opis postojećeg algoritma po segmentima.



```
*Untitled*
File Edit Format Run Options Window Help
import numpy as np
import math
from tkinter import filedialog
import tkinter as tk
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

#Varijable
points_coordinates = []
new_points_coordinates = []
points_triangles = []
points_connected = []
use_point = []
use_triangle = []
Avg_slope = []
cut_height = 0.01

# Izbornik za otvaranje .STL datoteke
root = tk.Tk()
root.withdraw()
file_path = filedialog.askopenfilename()
file = open(file_path, 'r')
text = file.readlines()
lines = []
for line in text:
    lines.append(line.split())
```

Slika 4-3 Algoritam 1/6

Na slici 4-3 je prikazan početni dio koda napravljen u Pythonu. Naredbom *import* su uvedeni svi moduli potrebni za izradu algoritma. Nakon modula, uvedene su sve korištene varijable u formatu lista. Redom korištene varijable su: koordinate točaka, koordinate točaka nakon zaglađivanja, lista trokuta koji sadrži točku, lista susjednih točaka, lista točaka koje pripadaju trokutu, upotreba točke pri zaglađivanju, upotreba točke pri računanju kuta i lista izračunatih kutova koja se obnavlja svakim novim računanjem i unosom naredbe *slope()*. Nakon jednog provođenja naredbe *cut()*, program reže model za 1% s gornje i donje strane. Nakon unosa varijabli, slijedi učitavanje određene *stl* datoteke.

```
*Untitled*
File Edit Format Run Options Window Help
# Ucitavanje i stvaranje lista točaka, trokuta i pripadnosti
triangle_index = -1
print_progress_control = 0
for index, line in enumerate(lines):
    if line[0] == 'facet':
        triangle_index = triangle_index + 1

        triangles.append([])
        use_triangle.append(True)
        point_already_in = False
        x = float(lines[index+2][1])
        y = float(lines[index+2][2])
        z = float(lines[index+2][3])
        for point_index, point in enumerate(points_coordinates):
            if point[0] == x:
                if point[1] == y:
                    if point[2] == z:
                        point_already_in = True
                        points_triangles[point_index].append(triangle_index)
                        triangles[triangle_index].append(point_index)
                        a_index = point_index

        if point_already_in == False:
            points_coordinates.append([x,y,z])
            points_triangles.append([triangle_index])
            a_index = len(points_coordinates)-1
            triangles[triangle_index].append(a_index)
            points_connected.append([])
            use_point.append(True)

        point_already_in = False
        x = float(lines[index+3][1])
        y = float(lines[index+3][2])
        z = float(lines[index+3][3])
        for point_index, point in enumerate(points_coordinates):
            if point[0] == x:
                if point[1] == y:
                    if point[2] == z:
                        point_already_in = True
                        points_triangles[point_index].append(triangle_index)
                        triangles[triangle_index].append(point_index)
                        b_index = point_index

Ln: 122 Col: 69
```

Slika 4-4 Algoritam 2/6

Na slikama 4-4 i 4-5 prikazan je dio koda koji interpretira učitaneu *stl* datoteku i dobivene podatke razvrstava u liste. Kako je i prethodno spomenuto, program čita koordinate iz *stl* datoteke s obzirom na njihovu lokaciju u samom dokumentu.

```
*Untitled*
File Edit Format Run Options Window Help

    if point_already_in == False:
        points_coordinates.append([x,y,z])
        points_triangles.append([triangle_index])
        b_index = len(points_coordinates)-1
        triangles[triangle_index].append(b_index)
        points_connected.append([])
        use_point.append(True)

point_already_in = False
x = float(lines[index+4][1])
y = float(lines[index+4][2])
z = float(lines[index+4][3])
for point_index,point in enumerate(points_coordinates):
    if point[0] == x:
        if point[1] == y:
            if point[2] == z:
                point_already_in = True
                points_triangles[point_index].append(triangle_index)
                triangles[triangle_index].append(point_index)
                c_index = point_index

    if point_already_in == False:
        points_coordinates.append([x,y,z])
        points_triangles.append([triangle_index])
        c_index = len(points_coordinates)-1
        triangles[triangle_index].append(c_index)
        points_connected.append([])
        use_point.append(True)

    if a_index not in points_connected[b_index]:
        points_connected[b_index].append(a_index)
    if a_index not in points_connected[c_index]:
        points_connected[c_index].append(a_index)
    if b_index not in points_connected[a_index]:
        points_connected[a_index].append(b_index)
    if b_index not in points_connected[c_index]:
        points_connected[c_index].append(b_index)
    if c_index not in points_connected[a_index]:
        points_connected[a_index].append(c_index)
    if c_index not in points_connected[b_index]:
        points_connected[b_index].append(c_index)

    progress = int(triangle_index/(len(text)/7)*100)
    #print(progress)
    #print(progress%10)
    if progress >= print_progress_control*10:
        print('učitavanje ',progress, '%')
        print_progress_control = print_progress_control + 1

Ln: 113 Col: 0
```

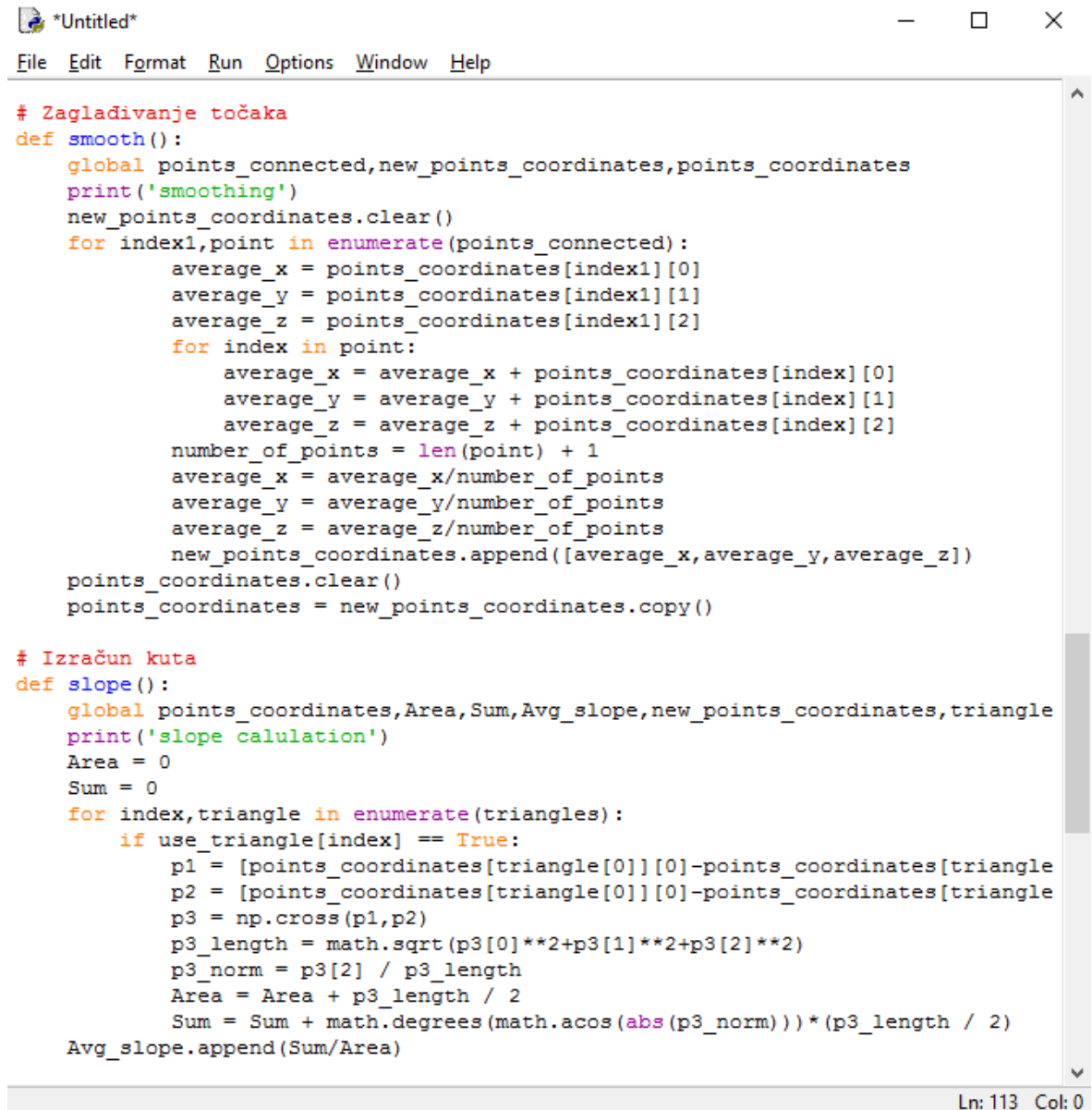
Slika 4-5 Algoritam 3/6

Nakon interpretacije učitanih podataka, definirane su funkcije korištene u ispitivanjima trodimenzionalnih modela. Zaglađivanje modela je provedeno Laplacianovom metodom zaglađivanja. Metoda se provodi na način da se za svaku točku računa prosječna vrijednost susjednih točaka. Kada se ta vrijednost izračuna, uzima se kao koordinata početne točke.

Postupak se provodi za svaku točku na modelu i na taj način je ostvaren faktor zaglađivanja ili *smoothing factor* iznosa 1. Svakim ponavljanjem postupka faktor zaglađivanja raste za 1.

Izračun kuta se provodi, kako je i prethodno spomenuto, metodom težinskog prosjeka.

Slika 4-6 prikazuje dio koda zadužen za zaglađivanje modela i izračun kuta sipanja.



```
*Untitled*
File Edit Format Run Options Window Help

# Zaglađivanje točaka
def smooth():
    global points_connected,new_points_coordinates,points_coordinates
    print('smoothing')
    new_points_coordinates.clear()
    for index1,point in enumerate(points_connected):
        average_x = points_coordinates[index1][0]
        average_y = points_coordinates[index1][1]
        average_z = points_coordinates[index1][2]
        for index in point:
            average_x = average_x + points_coordinates[index][0]
            average_y = average_y + points_coordinates[index][1]
            average_z = average_z + points_coordinates[index][2]
            number_of_points = len(point) + 1
            average_x = average_x/number_of_points
            average_y = average_y/number_of_points
            average_z = average_z/number_of_points
        new_points_coordinates.append([average_x,average_y,average_z])
    points_coordinates.clear()
    points_coordinates = new_points_coordinates.copy()

# Izračun kuta
def slope():
    global points_coordinates,Area,Sum,Avg_slope,new_points_coordinates,triangle
    print('slope calculation')
    Area = 0
    Sum = 0
    for index,triangle in enumerate(triangles):
        if use_triangle[index] == True:
            p1 = [points_coordinates[triangle[0]][0]-points_coordinates[triangle
            p2 = [points_coordinates[triangle[0]][0]-points_coordinates[triangle
            p3 = np.cross(p1,p2)
            p3_length = math.sqrt(p3[0]**2+p3[1]**2+p3[2]**2)
            p3_norm = p3[2] / p3_length
            Area = Area + p3_length / 2
            Sum = Sum + math.degrees(math.acos(abs(p3_norm)))*(p3_length / 2)
    Avg_slope.append(Sum/Area)

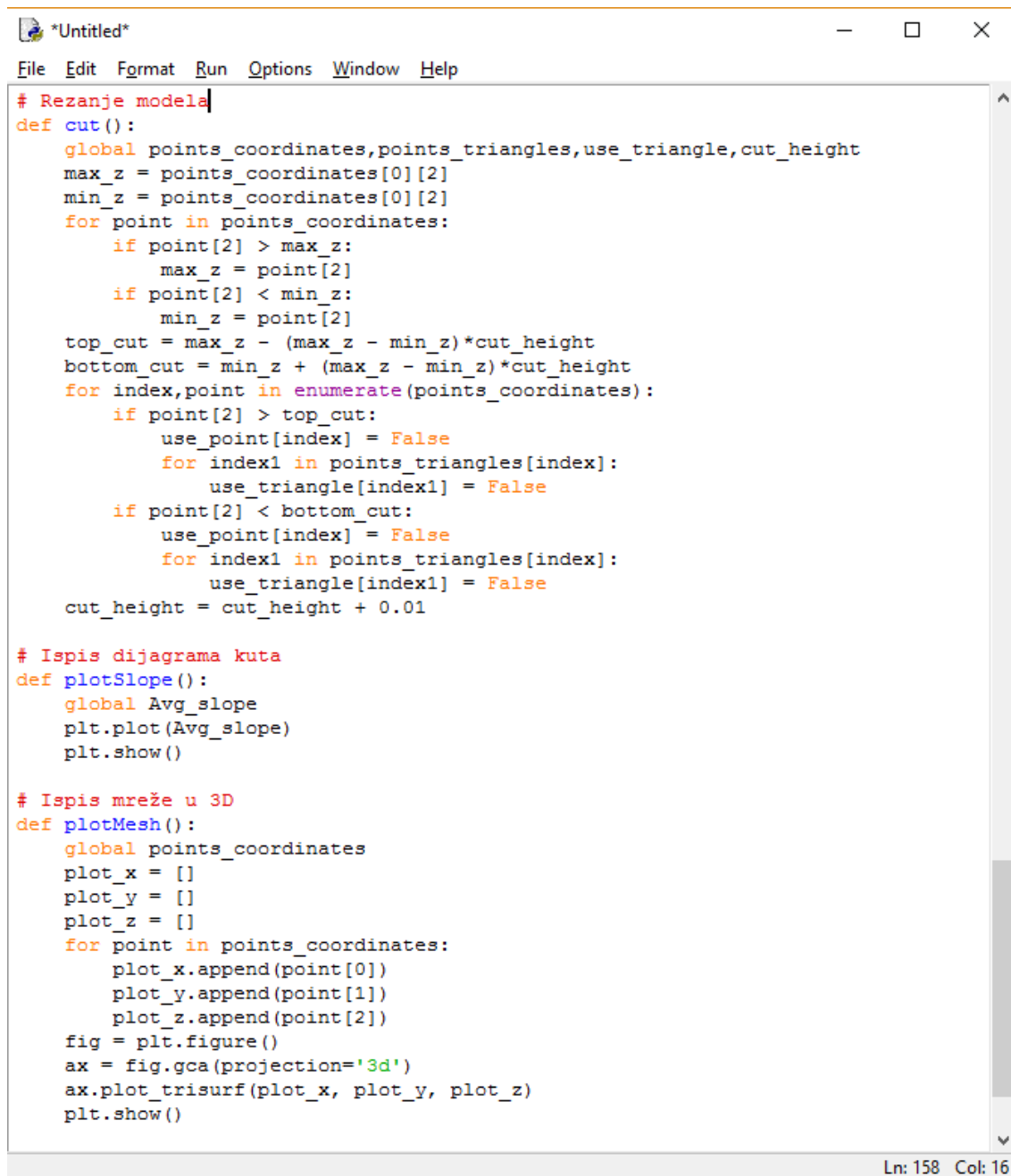
Ln: 113 Col: 0
```

Slika 4-6 Algoritam 4/6

Rezanje modela je provedeno na način da se brišu sve točke koje su u odabranom postotku gornjeg ili donjeg dijela modela. Konkretno, ako se pozove naredba *cut(2)*, izbrisat će se 2% modela s gornje i donje strane. Unosom naredbe *plotSlope()* program će prikazati dijagram u kojem je vidljivo kretanje kuta sipanja s obzirom na provedene modifikacije (zaglađivanje

i rezanje). Ispis 3D mreže je postignut uvođenjem treće, z, koordinate u kod te se poziva naredbom *plotMesh()*.

Slika 4-7 prikazuje posljednji dio koda koji se bavi rezanjem modela i izradom dijagrama i 3D mreže.



```
File Edit Format Run Options Window Help
# Rezanje modela
def cut():
    global points_coordinates, points_triangles, use_triangle, cut_height
    max_z = points_coordinates[0][2]
    min_z = points_coordinates[0][2]
    for point in points_coordinates:
        if point[2] > max_z:
            max_z = point[2]
        if point[2] < min_z:
            min_z = point[2]
    top_cut = max_z - (max_z - min_z)*cut_height
    bottom_cut = min_z + (max_z - min_z)*cut_height
    for index, point in enumerate(points_coordinates):
        if point[2] > top_cut:
            use_point[index] = False
            for index1 in points_triangles[index]:
                use_triangle[index1] = False
        if point[2] < bottom_cut:
            use_point[index] = False
            for index1 in points_triangles[index]:
                use_triangle[index1] = False
    cut_height = cut_height + 0.01

# Ispis dijagrama kuta
def plotSlope():
    global Avg_slope
    plt.plot(Avg_slope)
    plt.show()

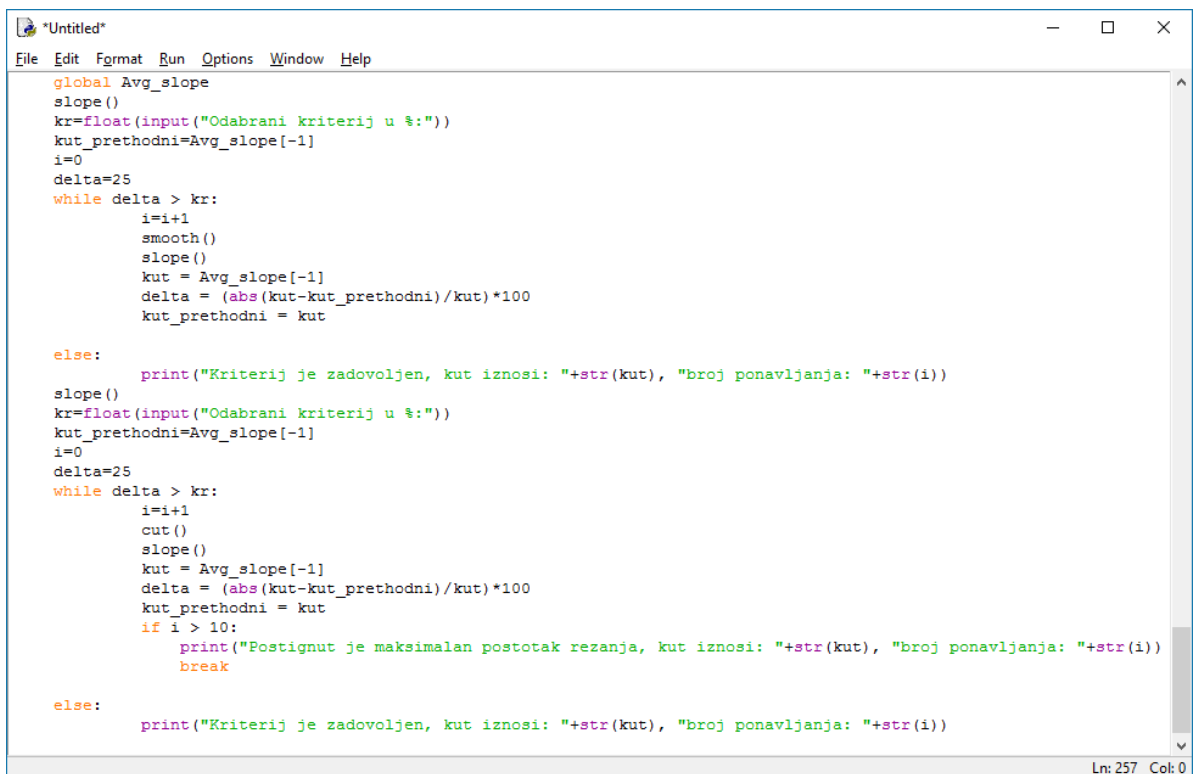
# Ispis mreže u 3D
def plotMesh():
    global points_coordinates
    plot_x = []
    plot_y = []
    plot_z = []
    for point in points_coordinates:
        plot_x.append(point[0])
        plot_y.append(point[1])
        plot_z.append(point[2])
    fig = plt.figure()
    ax = fig.gca(projection='3d')
    ax.plot_trisurf(plot_x, plot_y, plot_z)
    plt.show()
```

Ln: 158 Col: 16

Slika 4-7 Algoritam 5/6

Nakon definiranja svih potrebnih funkcija, napravljen je algoritam koji računa kut sipanja do zadane točnosti. Program samostalno obavlja zaglađivanje i rezanje modela sve dok promjena kuta ne zadovolji traženi uvjet koji je definiran kao kriterij zadan od strane korisnika. Nakon postizanja željene točnosti, program ispisuje izračunati kut sipanja i broj ponavljanja petlje koji simbolizira faktor provedenog zaglađivanja i postotak za koji je model odrezan s gornje i donje strane.

Slika 4-8 prikazuje dio koda zadužen za provođenje opisanog algoritma.



```
global Avg_slope
slope()
kr=float(input("Odabrani kriterij u %:"))
kut_prethodni=Avg_slope[-1]
i=0
delta=25
while delta > kr:
    i=i+1
    smooth()
    slope()
    kut = Avg_slope[-1]
    delta = (abs(kut-kut_prethodni)/kut)*100
    kut_prethodni = kut

else:
    print("Kriterij je zadovoljen, kut iznosi: "+str(kut), "broj ponavljanja: "+str(i))
slope()
kr=float(input("Odabrani kriterij u %:"))
kut_prethodni=Avg_slope[-1]
i=0
delta=25
while delta > kr:
    i=i+1
    cut()
    slope()
    kut = Avg_slope[-1]
    delta = (abs(kut-kut_prethodni)/kut)*100
    kut_prethodni = kut
    if i > 10:
        print("Postignut je maksimalan postotak rezanja, kut iznosi: "+str(kut), "broj ponavljanja: "+str(i))
        break

else:
    print("Kriterij je zadovoljen, kut iznosi: "+str(kut), "broj ponavljanja: "+str(i))
```

Slika 4-8 Algoritam 6/6

5. PROVEDENA ISPITIVANJA

5.1. Opis ispitivanja

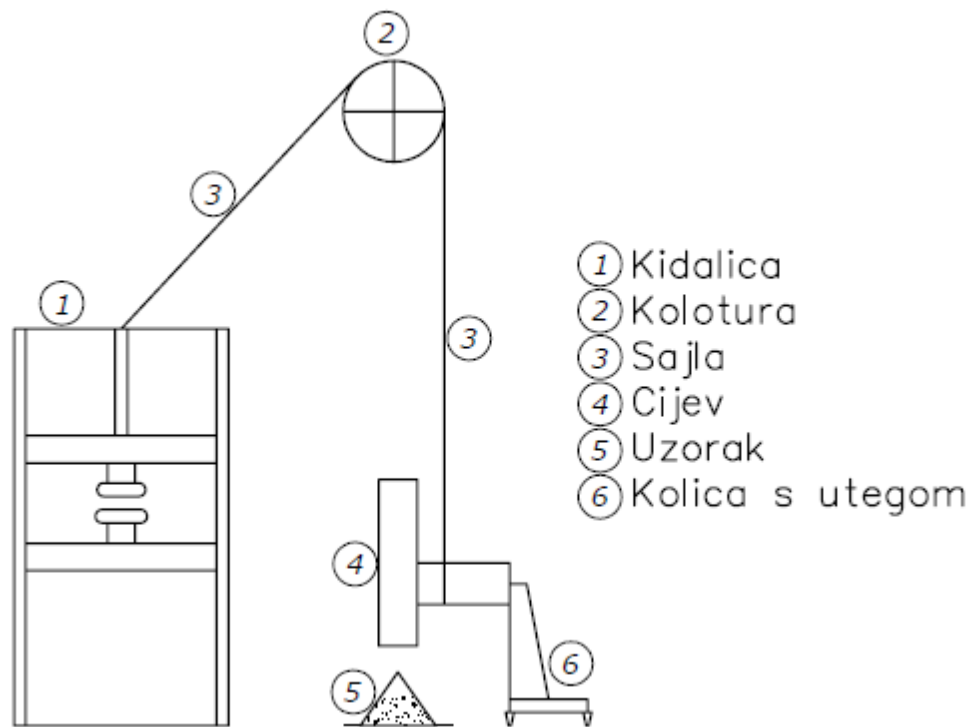
Ispitivanja kao neizbježan dio provjere pretpostavki obavljena su u Laboratoriju za rudarsku mehanizaciju i arhitektonsko-građevni kamen Rudarsko-geološko-naftnog fakulteta u Zagrebu (Slika 5-1).



Slika 5-1 Dio Laboratorija za rudarsku mehanizaciju i arhitektonsko-građevni kamen

Oprema vidljiva na slici 5-1 korištena je u svrhu provedbe metode cilindra kako bi se dobila približno stožasta hrpa materijala koja kao takva služi za određivanje kuta sipanja uz pomoć računalne opreme i dobivenih trodimenzionalnih modela. Naime, samo ispitivanje započinje kada je cilindar u najnižem položaju, a materijal prethodno usipan u njega te se tada cilindar podigne konstantnom brzinom od 4,03 mm/s i tvori se već spomenuta stožasta

hrpa. Sam proces podizanja ostvaruje se kidalicom koja je posebno preuređena kako bi uz pomoć užeta (sajle) i koloture cilindar bio pomičan u vertikalnom smjeru. Cilindar je bio od plastičnog materijala, unutarnjeg promjera od 110 mm. Pojednostavljena shema postupka ispitivanja prikazana je na slici 5-2.



Slika 5-2 Shema ispitivanja metodom cilindra (Kostrić, 2018)

Ono što je prethodilo ispitivanju, bila je izrada i postavljanje tzv. markera na rotirajuću podlogu koji su nužni za računalnu obradu dobivenih fotografija, a predstavljaju svojevrsne orijentire. Osim toga, potrebnu opremu čini fotoaparat ili kamera, a konkretno u ovom radu korišten je fotoaparat Pentax K-5 koji je za svako pojedinačno podizanje cilindra bio fiksiran pomoću tronožnog stalka na istoj poziciji kako bi se u konačnici smanjile moguće pogreške i eventualna odstupanja u rezultatima kuta sipanja.

Sva ispitivanja kuta sipanja provedena su isti dan kako bi se umanjili mogući nepovoljni utjecaji promjene temperature, vlažnosti zraka i slično na rezultate ispitivanja. Za sve materijale hrpa je formirana metodom cilindra po tri puta. Svaka hrpa materijala fotografirana je 18 puta tako da je rotirajuća podloga pomicala za 20° pri svakoj sljedećoj fotografiji obzirom da je fotoaparat fiksiran.

5.2. Materijali korišteni u istraživanjima

Konkretno, istraživanja su provedena na 5 različitih materijala kojima se razlikuju mineraloško-petrografska svojstva, granulometrijski sastav, veličina i oblik zrna. Na ovaj način istraživana je primjenjivost algoritma za triangulirane modele različitih materijala koji sa sobom nose specifičnosti čije je postojanje bilo nužno uklopiti u okviru programskog jezika Python.

Kako je riječ o laboratoriju pretežno ustrojenom za rudarsku struku, tako su i korišteni materijali bili kameni agregati kao dolomit (krupni te sitni) i mramor, ali i šljunak te kvarcni pijesak. Za svako ispitivanje korištena je ista masa materijala (3 kg). Veličina čestica pojedinih uzoraka definirana je klasama, sitni dolomit 4/8 mm, krupni dolomit 8/16 mm, mramor 8/16 mm, a šljunak 4/8 mm. Najsitniji materijal bio je kvarcni pijesak čija zrna pripadaju klasi 0,7/1,2 mm.

Kako bi se uporabljivost algoritma za određivanje kuta sipanja promatrala i izvan područja rudarstva i graditeljstva, uz već spomenute materijale, u razmatranje i računalnu obradu uzeti su i materijali kao što su mlijeko u prahu, suha kukuruzna zrna (Slika 5-3) te piljevina (Slika 5-4). Triangulirani modeli ovih materijala preuzeti su iz dostupnih podataka koje su objavili Rackl i Grötsch (2018). Oni su svako ispitivanje ponovili 4 puta, slikali, napravili triangulirane modele hrpa u *stl* formatu koji su neizmjerljivo pomogli u izradi ovog rada.



Slika 5-3 Stožasta hrpa materijala sačinjena od suhих kukuruznih zrna (Rackl i Grötsch, 2018)



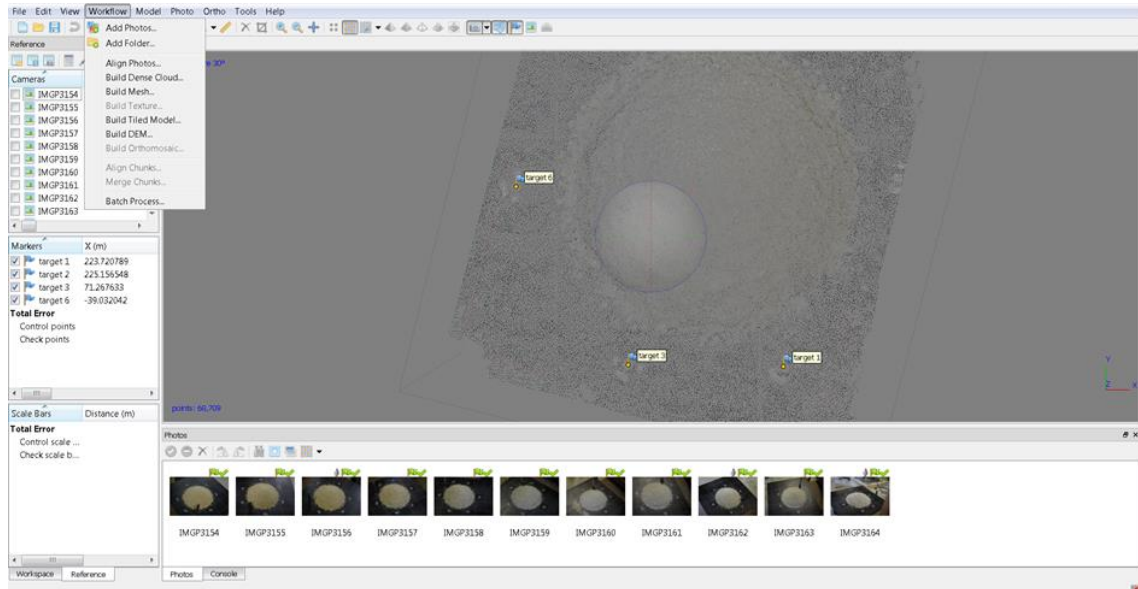
Slika 5-4 Stožasta hrpa materijala sačinjena od drvene piljevine (Rackl i Grötsch, 2018)

Računalnom obradom trodimenzionalnih modela dobivenih u nekim drugačijim uvjetima i sa materijalima potpuno različitih svojstava moguće je dobiti širu sliku o nužnim parametrima računalnog koda kako bi rezultat toga bio algoritam koji bi mogao biti primjenjiv za što širi spektar materijala.

6. ANALIZA REZULTATA

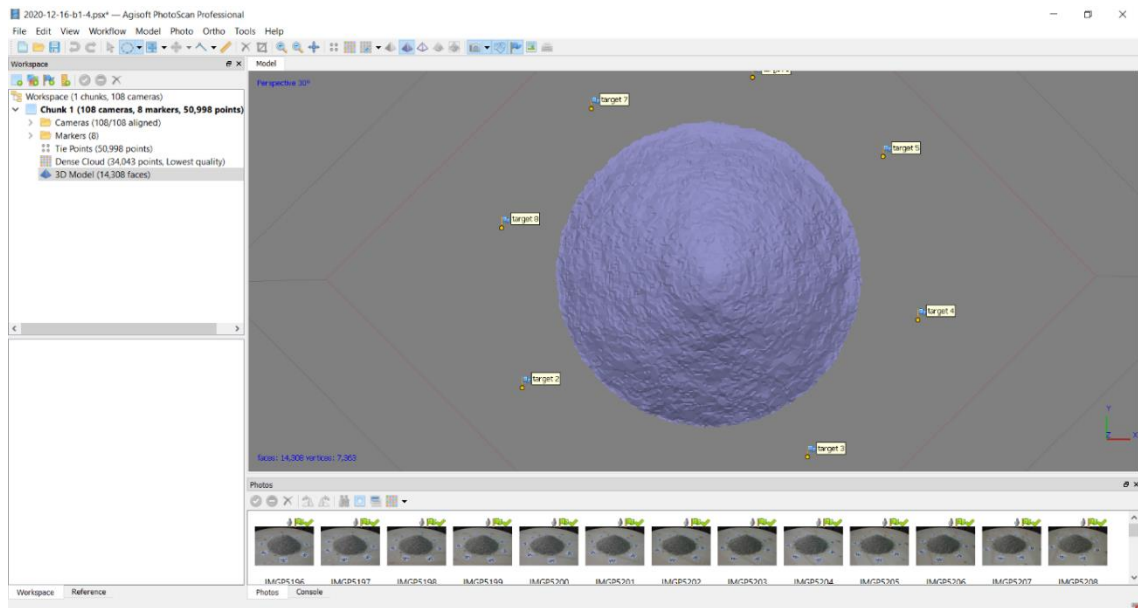
6.1. Izrada 3D modela

Budući da algoritam radi sa *stl* datotekama, iz fotografija materijala potrebno je izraditi 3D modele. Modeli su izrađeni u programu *Agisoft PhotoScan*. Prvi korak kod izrade modela sastoji se od učitavanja prethodno spomenutih fotografija. Sučelje programa *Agisoft PhotoScan* sa učitanim fotografijama prikazano je na slici 6-1.



Slika 6-1 *Agisoft PhotoScan* sa učitanim fotografijama

Nakon toga, slijedi poravnanje fotografija, učitavanje markera i detektiranje markera što je napravljeno pomoću tzv. *Batch* procesa. Kao produkt tog procesa dobiven je oblak točaka u obliku hrpe sipanog materijala. Iz dobivenih točaka potrebno je napraviti trianguliranu mrežu koja se sastoji od velikog broja trokuta, a u konačnici iz tih trokuta dobije se 3D model koji se eksportira u obliku *stl* datoteke te se kao takav može koristiti za proračun kuta sipanja pomoću programa *Python*. Dobiveni 3D model u programu *Agisoft PhotoScan* prikazan je na slici 6-2.

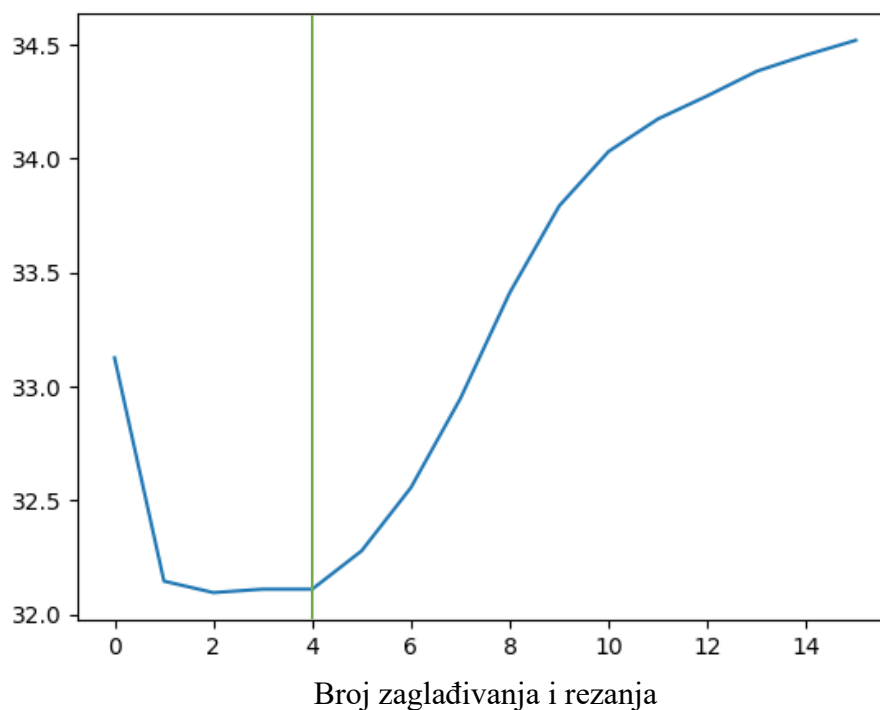


Slika 6-2 3D model izrađen u programu *Agisoft PhotoScan*

6.2. Obrada *stl* datoteka u Pythonu

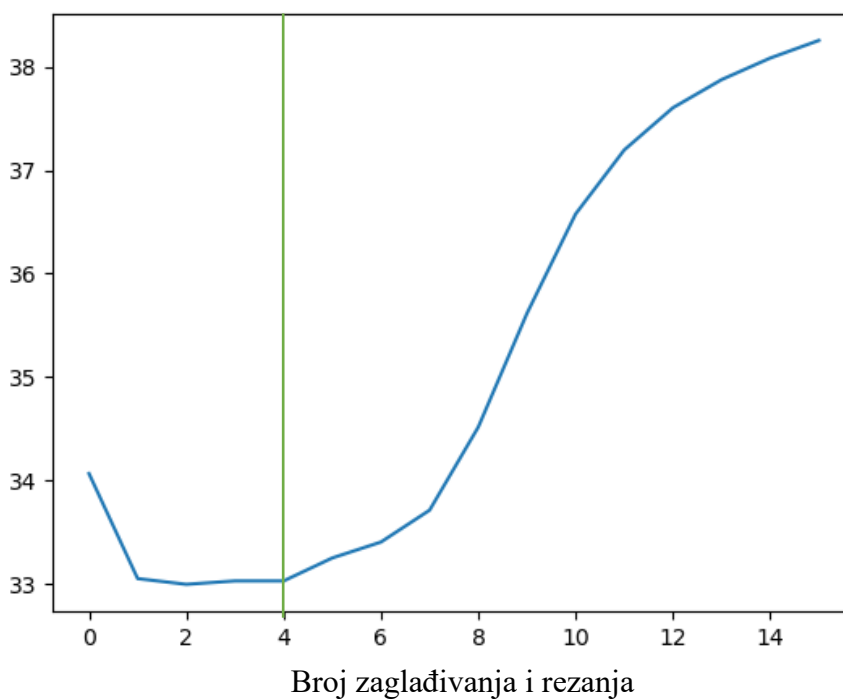
Nastavno na obradu 3D modela, sljedeći korak bio je učitavanje istih u već pripremljeni Python algoritam odabirući iz izbornika *Run->Run Module*. Tada se otvara dodatni radni prozor gdje se omogućava pozivanje određenih naredbi ili izrada petlji koje se žele izvršiti u nekom trenutku. Primjenjivost algoritma za određivanje kuta sipanja različitih vrsta materijala promatrana je pomoću tzv. *smoothing* faktora, tj. intenziteta zaglađivanja plašta stožaste hrpe materijala te o položaju na plaštu gdje algoritam provodi rezanje gornjeg i donjeg dijela hrpe. Pri tome su korištene naredbe *smooth* i *cut* u jednostavnoj *for* petlji. Kao ključno bilo je odrediti broj ponavljanja obje naredbe koje bi kod različitih materijala izazvale jednako ponašanje što se provjeravalo iscrtavanjem grafa iz *MATPLOTLIB* modula. Maksimalan broj promatranih ponavljanja ove dvije naredbe bio je 10. U slučaju većeg broja ponavljanja bio bi odrezan prevelik dio hrpe, dok se pri manjim brojevima ne bi mogla uočiti stabilizacija vrijednosti kuta sipanja oko nekog iznosa. Primjerice, za šljunak koji je karakteristično zaobljenih zrna te za sitni dolomit koji je karakteristični predstavnik uglatih zrna zamijećen je sličan trend što je prikazano slikama 6-3 te 6-4.

Kut sipanja (°)



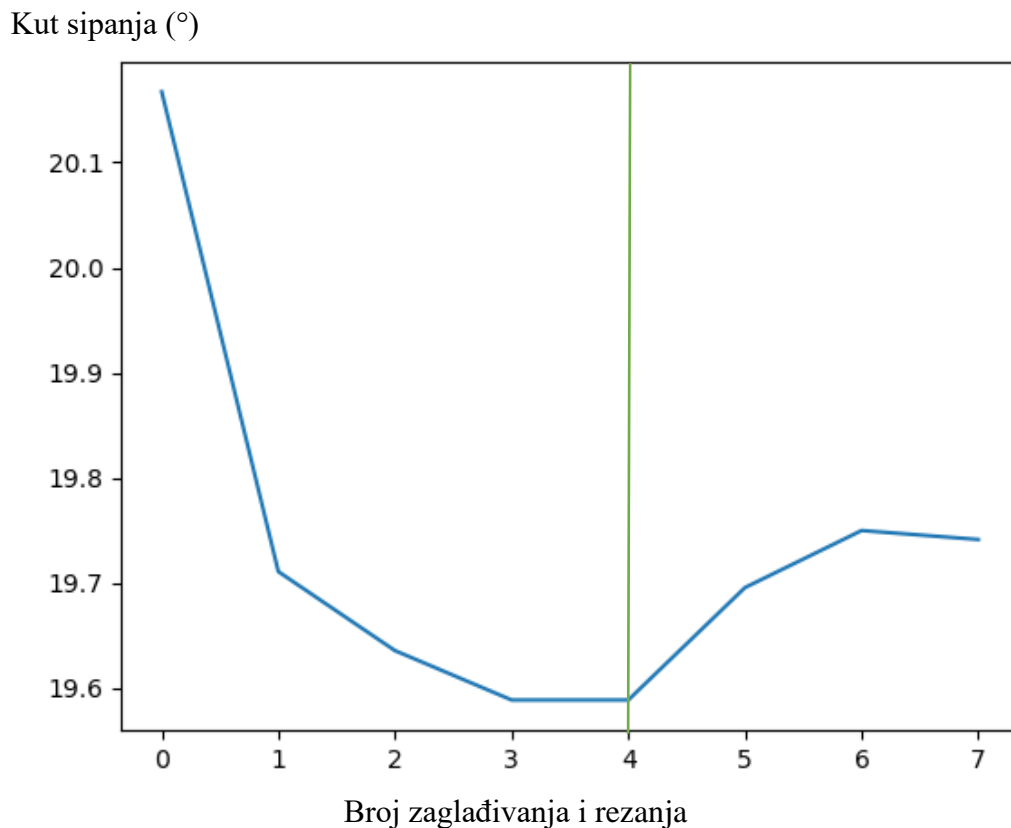
Slika 6-3 Grafički prikaz ovisnosti iznosa kuta sipanja o broju zaglađivanja i rezanja stožaste hrpe na primjeru šljunka

Kut sipanja (°)



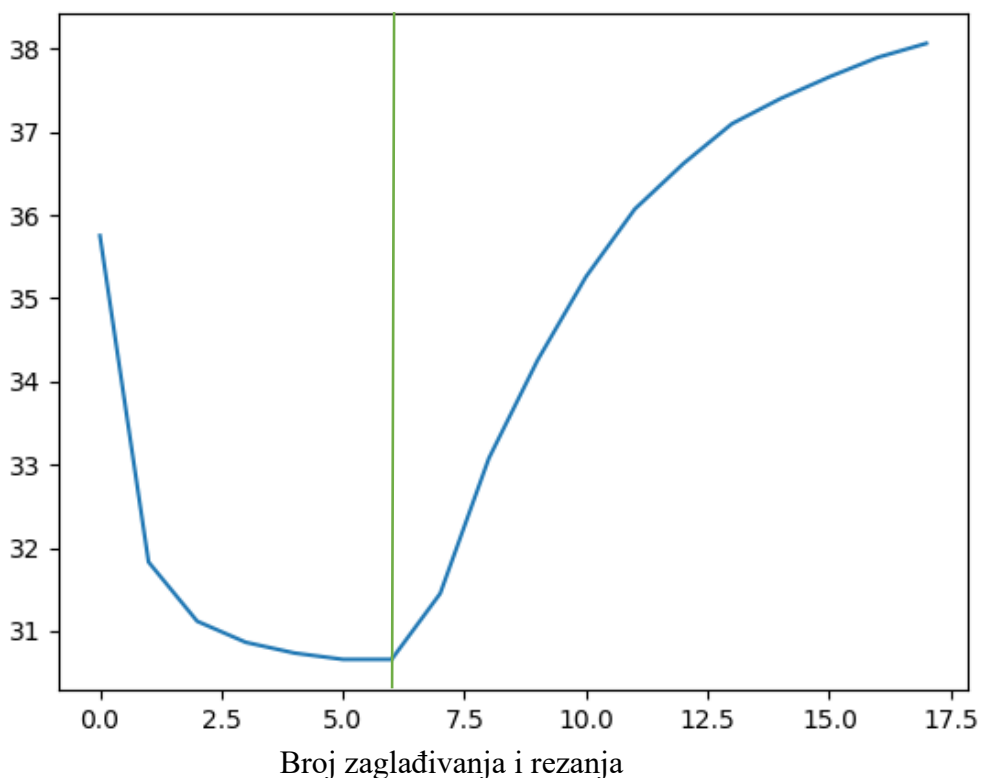
Slika 6-4 Grafički prikaz ovisnosti iznosa kuta sipanja o broju zaglađivanja i rezanja stožaste hrpe na primjeru sitnog dolomita

Iz obje je slike vidljivo kako u slučaju zaglađivanja iznos kuta sipanja opada, dok je kod rezanja plašta stožaste hrpe trend suprotne prirode, tj. ima tendenciju rasta što se objašnjava činjenicom da prilikom zaglađivanja sve moguće vertikalne površine budu poravnate te se kut sipanja time smanji, a rezanjem gornjeg i donjeg dijela stožaste hrpe čime realne hrpe odudaraju od pravilne, taj kut sipanja raste. Da bi se generalno moglo utvrditi takvo ponašanje, isti postupak proveden je za materijale kukuruznih zrna te drvene piljevine čiji izgled je vidljiv na slikama 5-3 i 5-4. Osim toga, te su hrpe materijala dobivene u drugačijim uvjetima, ali istom metodom. No, konačan rezultat bio je gotovo identičan i potvrdio slične trendove kretanja vrijednosti kuta sipanja ovisno o primijenjenom broju koraka zaglađivanja i rezanja, što se može vidjeti i na slikama 6-5 te 6-6.



Slika 6-5 Grafički prikaz ovisnosti iznosa kuta sipanja o broju zaglađivanja i rezanja stožaste hrpe na primjeru kukuruznih zrna

Kut sipanja (°)



Slika 6-6 Grafički prikaz ovisnosti iznosa kuta sipanja o broju zaglađivanja i rezanja stožaste hrpe na primjeru drvene piljevine

Sljedeće pitanje bilo je na temelju grafičkih prikaza odrediti koji je iznos kuta sipanja onaj koji zbilja predstavlja stvarnu vrijednost kuta sipanja hrpe materijala. Naime, iznos kuta sipanja varira (smanjuje se i raste) ovisno o broju koraka zaglađivanja i rezanja hrpe, a u trenutku gdje se može vidjeti da kut sipanja teži „smirivanju“ oko neke vrijednosti, može se smatrati reprezentativnom vrijednošću kuta sipanja. Upravo je taj smjer razmišljanja bio vodilja za utvrđivanje kriterija koji bi neovisno o vrsti materijala i pretpostavci kako je 10 puta potrebno zagladiti te odrezati 10% stožaste hrpe, bio u stanju prepoznati trend „smirivanja“ iznosa kuta sipanja oko neke vrijednosti što je učinjeno i za proces zaglađivanja i rezanja jer se radi o međusobno neovisnim naredbama. Dodatak kriteriju bilo je ograničenje od maksimalno 10% odrezane stožaste hrpe materijala odozgo te odozdo što znači da više od toga nije dopušteno, dok je kod zaglađivanja ipak taj dio izostavljen jer ne utječe značajno na brzinu provođenja postupka izračuna, a rezultat ostaje reprezentativan. Osim toga, u kriteriju je uvedena mogućnost da se za svaki pojedinačni materijal dopusti korisniku individualno određivanje nakon kolike promjene iznosa kuta sipanja se smatra „smirivanje“

oko neke vrijednosti kako bi sam algoritam bio prilagodljiv korisniku ovisno o traženoj točnosti i preciznosti istoga. Navedeno se ostvaruje jednostavnim pozivanjem naredbe, u ovom slučaju pod nazivom *auto_sc*.

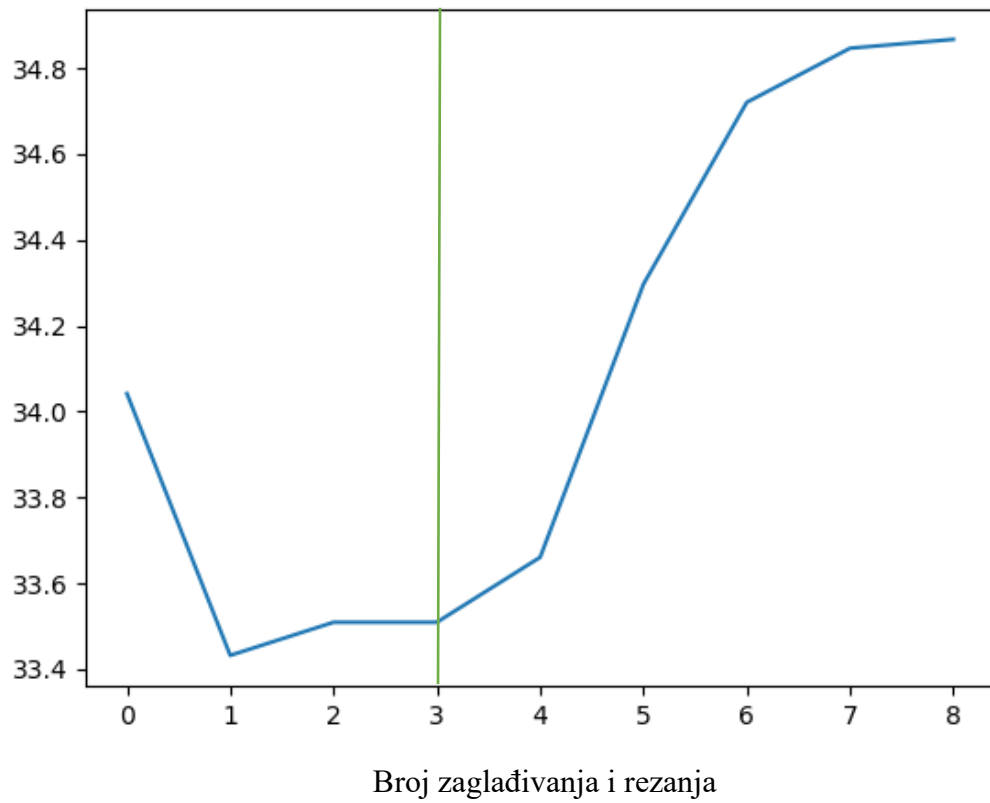
S ciljem verifikacije navedene metode određivanja kuta sipanja, odnosno verifikacije novorazvijenog algoritma napisanog u programskom jeziku Python, provedeno je ispitivanje kuta sipanja referentne hrpe koja je prikazana na slici 6-7. Referentna hrpa izrađena je kao pravilni stožac unaprijed definiranog nagiba stranica od 35° , što bi trebalo odgovarati traženom kutu sipanja. Ovakav referentni stožac prekriven je ljepilom i posipan granulacijom materijala veličine zrna od 2 do 4 mm.



Slika 6-7 Izrada referentne hrpe (Kostrić, 2018)

Postupak obrade i analize bio je identičan onima provedenima na realnim hrpama materijala, a rezultat je potvrdio iste trendove koji su također bili prisutni kod realnih hrpa. Dakle, primjenom koraka zaglađivanja kod referentne hrpe kut sipanja je blago opadao sve do „smirivanja“, a potom pri rezanju iste hrpe odozgo i odozdo iznos kuta sipanja porastao je također do trenutka kada se stabilizirala oko vrijednosti 35° čime je dobivena povratna informacija kako algoritam uveden u programskom jeziku Python daje veoma precizne rezultate. Grafički je to prikazano na slici 6-8.

Kut sipanja (°)



Slika 6-8 Grafički prikaz ovisnosti iznosa kuta sipanja o broju zaglađivanja i rezanja na primjeru referentne hrpe

7. ZAKLJUČAK

U okviru ovog rada obavljena su laboratorijska ispitivanja metodom cilindra različitih vrsta materijala s različitom veličinom i oblikom zrna kako bi se odredio kut sipanja kao jedno od važnijih fizikalnih svojstava materijala ne samo u području rudarstva, nego i šire. Samo mjerenje kuta sipanja može se provesti na više načina: običnim kutomjerom, 2D metodom zasnovanom na digitalnim fotografijama čiji je broj veoma mali te sve prisutnijom 3D metodom čiji temelj predstavljaju također digitalne fotografije (najčešće veći broj čime se povećava točnost) te njihova obrada kako bi se dobio trodimenzionalni model hrpe. Upravo na posljednjoj metodi određivanja kuta sipanja bazira se ovaj rad. Za potrebe ovog rada navedena metoda je upotpunjena novorazvijenim algoritmom kojemu je cilj automatizacija proračuna kutova sipanja iz prethodno izrađenih trodimenzionalnih modela. Automatizacijom se smanjuje utjecaj opažača kod mjerenja kuta sipanja, odnosno povećava točnost samog mjerenja. Kao alat, upotrijebljen je programski jezik *Python*, a za obradu digitalnih fotografija *Agisoft PhotoScan*.

Novorazvijeni algoritam pokazao je svoju univerzalnost obzirom da je na različitim materijalima bio jednako primjenjiv te su dobivene vrijednosti kuta sipanja bile reprezentativne što je verificirano mjerenjima na referentnoj hrpi. Algoritam je napisan tako da je maksimalno prilagodljiv korisniku koji bi poznavajući mineraloško-petrografska svojstva, granulometrijski sastav, veličinu i oblik zrna ispitivanog materijala mogao odrediti željene parametre u vidu tražene preciznosti (reda veličine) kuta sipanja. U skladu s time, detaljno je razrađen kriterij algoritma kojeg se lako poziva naredbom *auto_sc* gdje se odrađuju procesi zaglađivanja i rezanja stožaste hrpe slijedeći potrebe korisnika.

Konačno, naznaka uspješnosti algoritma bila je činjenica kako su se različiti ispitivani materijali ponašali praktički po identičnom obrascu slijedeći veoma slične trendove za pojedine procese – zaglađivanje te rezanje. Sama potvrda uslijedila je kroz provođenje istog postupka za referentnu (pravilnu) stožastu hrpu s poznatom vrijednosti kuta sipanja od 35° gdje se algoritam pokazao veoma pouzdanim i preciznim izračunajući kut sipanja koji se očitava sa grafičkog prikaza te je iznosio $34,9^\circ$.

8. POPIS LITERATURE

AMOS D., 2020. *Python GUI Programming With Tkinter*. URL: <https://realpython.com/python-gui-tkinter/#author> 28.4.2021.

BEAKAWI AL-HASHEMI H.M., BAGHABRA AL-AMOUDI O.S., 2018. *A review on the angle of repose of granular materials*. pdf, Powder technology, 330 (2008), str. 397-417.

HERCIGONJA Z., 2018. UVOD U PROGRAMSKI JEZIK PYTHON. URL: <https://pythonosnoveblog.wordpress.com/> 28.4.2021.

HRVATSKA ENCIKLOPEDIJA, mrežno izdanje, 2018. *Stereofotogrametrija*, URL: <https://www.enciklopedija.hr/natuknica.aspx?ID=58027> 20.4.2021.

HRVATSKA ENCIKLOPEDIJA, mrežno izdanje, 2021. *Euklidov algoritam*. URL: <http://www.enciklopedija.hr/Natuknica.aspx?ID=18595> 14. 4. 2021.

INFORMATION TRANSFORMATION SERVICES, 2021. *7 Best 3D File Formats Used In Industries*. URL: <https://it-s.com/wp-content/uploads/2021/05/3-sample-stl-file-of-a-workspace-in-ascii-format.png> 15.5.2021.

KOPUNIĆ M., 2019. *Metode ispitivanja kuta sipanja i ponovljivost rezultata metodom cilindra*, diplomski rad. Rudarsko-geološko-naftni fakultet.

KOSTRIĆ S., 2018. *Određivanje kuta stošca pri ispitivanju kuta sipanja metodom cilindra*, diplomski rad, Zagreb: Rudarsko-geološko-naftni fakultet.

KUJUNDŽIĆ T., 2010. *Fizikalna svojstva materijala*, predavanja Transport i izvoz, Zagreb: Rudarsko-geološko-naftni fakultet.

MISTRY R., DALAL C., SONI T., 2017. *Determination of angle of repose of pharmaceutical materials based on Image processing using LabVIEW*. International journal of advanced research in electrical, electronics and instrumentation engineering, 7, str. 3-5.

RACKL M., GRÖTSCH, F.E., RUSCH M., FOTTNER J., 2017. *Qualitative and quantitative assessment of 3D-scanned bulk solid heap data*. pdf, Powder Technology, str. 105-118.

RACKL M., GRÖTSCH, F.E., 2018: *3D Scans, Angles of Repose and Bulk Densities of 108 Bulk Material Heaps*. Figshare. Collection.

RADIONA WIKI, 2018. *DIY 3D skeniranje (1/3) Stereofotogrametrija*. URL: http://radiona.org/wiki/project/3d_scanning_photogrammetry 20.4.2021.

TAN Y., FOTTNER J., KESSLER S., 2020. *An efficient and reliable method for determining the angle of repose of biomass by using 3D scan*. pdf, Biomass and Bioenergy.

URLIČIĆ S., 2017. *Najbolji programi za uređivanje slika i fotografija*. URL: <https://vidi-vishe.com/programi-za-uredivanje-slika/> 20.4.2021.

VAN ROSSUM G. 2018. *Python tutorial*. URL: https://bugs.python.org/file47781/Tutorial_EDIT.pdf 28.4.2021.

SAŽETAK

Krešimir Maletić

Matija Palijan

Silvio Šubić

RAZVOJ ALGORITMA ZA ODREĐIVANJE KUTA SIPANJA IZ TRODIMENZIONALNIH MODELA

Algoritam kao svojevrsna uputa za obavljanje nekog zadatka koja opisuje točan redoslijed za rješavanje istog nužno zahtijeva prilagodbu računalu kako bi uopće razumjelo potrebe korisnika. Algoritmi mogu biti različite prirode, no ovdje je riječ o računalnom koji je implementiran u programski jezik Python. U današnje vrijeme gdje se svaka aktivnost nastoji digitalizirati i uključiti je u različite baze podataka, razvila se potreba i za metodom računalnog određivanja kuta sipanja koja isključuje moguće ljudske greške te znatno ubrzava postupak. Iz tog razloga, zadatak specifičnog algoritma bio je izračun kuta sipanja iz trodimenzionalnih modela, a potom i njegova razrada kroz optimizaciju kriterija koji treba dati reprezentativnu vrijednost kuta sipanja, a da pritom bude prilagodljiv i jednostavan za korištenje svima. Nakon što je algoritam sastavljen, neophodno je bilo provesti stvarna ispitivanja u laboratoriju na stožastim hrapama materijala te potom izraditi trodimenzionalne modele pomoću programa *Agisoft PhotoScan* koji se koriste u fazi rada samog algoritma. Na taj način provjerio se rad algoritma koji omogućuje ne samo određivanje kuta sipanja, već i jedinstvenu mogućnost prilagodbe korisniku ovisno o njegovim zahtjevima za preciznošću kuta sipanja. U svrhu dodatne provjere točnosti algoritma, na referentnoj (pravilnoj) hrpi materijala, povratna informacija je bila izuzetno dobra uzevši u obzir odstupanje u vrijednosti kuta sipanja od svega $0,1^\circ$.

Ključne riječi: algoritam, programski jezik Python, trodimenzionalni modeli, kut sipanja

SUMMARY

Krešimir Maletić

Matija Palijan

Silvio Šubić

DEVELOPMENT OF ALGORITHM FOR DETERMINATION OF ANGLE OF REPOSE FROM THREE-DIMENSIONAL MODELS

An algorithm, as a kind of instruction for doing a task that describes the exact sequence for solving it, necessarily requires adaptation to the computer in order to understand the needs of users in general. Algorithms can be of different natures, but this is a computer that is implemented in the Python programming language. Nowadays, where every activity is sought to be digitized and included in various databases, the need has developed for a method of computer determination of the angle of repose that eliminates possible human errors and significantly speeds up the process. For this reason, the task of the specific algorithm was to calculate the angle of repose from three-dimensional models, and then its elaboration through the optimization of criteria that should give a representative value of the angle of repose, while being flexible and easy to use for everyone. After the algorithm was assembled, it was necessary to conduct actual laboratory tests on conical piles of materials and then create three-dimensional models using Agisoft PhotoScan software used in the operation phase of the algorithm itself. In this way, the operation of the algorithm was checked, which enables not only the determination of the angle of repose, but also the unique possibility of adjustment to the user depending on his requirements for the accuracy of the angle of repose. For the purpose of further checking the accuracy of the algorithm, on the reference (correct) pile of materials, the feedback was extremely good considering the deviation in the value of the angle of repose of only $0,1^\circ$.

Key words: algorithm, programming language Python, three-dimensional models, angle of repose