Contents

Chapter 1

Akhdefo:

Computer Vision for Slope Stability: Land Deformation Monitoring Using Optical Satellite Imagery

Chapter 2

Background of Akh-Defo:

AKh-Defo is combination of two different words 1) Akh in Kurdish language means land, earth or soil (origion of the word is from Kurdish badini dailect) 2) Defo is short of English word deformation.

Chapter 3

Recommended Citation:

Muhammad M, Williams-Jones G, Stead D, Tortini R, Falorni G and Donati D (2022) Applications of ImageBased Computer Vision for Remote Surveillance of Slope Instability.Front. Earth Sci. 10:909078. doi: 10.3389/feart.2022.909078

Chapter 4

Updates:

- Akhdefo version one is deprecated please use Akhdefo version 2.
- Akhdefo now can run on the cloud for real-time processing
- Akhdefo now consist of 12 Modules that performs end to end python-based GIS and Image Processing and Custumized Figure generation

Chapter 5

Usage:

- download anaconda environment file

Please visit the GITHUB homepage to download the file

- download python package requirement text

Please visit the GITHUB homepage to download the file

- Create new python Anaconda environment using the below command

```
conda env create -f akhdefov2.yml
```

- Install required python packages using below command

```
pip install -r pip_req.txt
```

- Now install Akhdefo using the below command

```
pip install akhdefo-functions
```

Chapter 6

User Guide

- Under construction will be released soon!

Chapter 7

Akhdefo Functions Summary

7.1 Preprocessing functions to unzip and sort images

```
from akhdefo_functions import copyImage_Data, copyUDM2_Mask_Data , unzip
```

7.2 Import Module to mosaic and crop raster to Area of Interest using shapefile

```
from akhdefo_functions import Mosaic, Crop_to_AOI
```

7.3 Import Module to filter rasters

```
from akhdefo_functions import Filter_PreProcess
```

7.4 Import Module to calculate triplet raster velocities

```
 from akhdefo_functions import DynamicChangeDetection
```

7.5 Import Modules for plotting

```
from akhdefo_functions import akhdefo_viewer
from akhdefo_functions import plot_stackNetwork
from akhdefo_functions import Akhdefo_resample
from akhdefo_functions import Akhdefo_inversion
```

7.6 Import Module to process stacked triplet velocities and collect velocity candidate points

```
from akhdefo_functions import stackprep'
```

7.7 Python module to coregister satallite images

```
from akhdefo_functions  import  Coregistration
```

7.8 python module to process and calculate timestamp linear deformation rates

```
from akhdefo_functions import  Time_Series
```

7.9 python module to plot timeseries profile for selected points

```
from  akhdefo_functions import  akhdefo_ts_plot

```

akhdefo_functions.unzip (page 17)(zipdir, dst_dir)	This program unzips all the zip products into one folder
akhdefo_functions.copyImage_Data (page 19)([...])	This program copy all the raster images.
akhdefo_functions.copyUDM2_Mask_Data (page 21)([...])	This program copy all raster masks.
akhdefo_functions.Filter_PreProcess (page 23)([...])	This program prepare and uses filters to balanace raster image brightness
akhdefo_functions.Crop_to_AOI (page 25)([...])	This program used to clip multiple raster files
akhdefo_functions.Mosaic (page 27)([...])	This program mosiacs raster images in geotif format as well grab dates the satellite image taken for further processing.
akhdefo_functions.Coregistration (page 29)([...])	This program coregisteres multiple rasters using both structural similarity index and feature matching techniqws.
akhdefo_functions.DynamicChangeDetection (page 31)([...])	This program calculates optical flow velocity from triplets of daily optical satellite images.
akhdefo_functions.plot_stackNetwork (page 33)([...])	This Program plots temporal network of triplets to be stacked for calculating Annual Mean Velocity from stacked optical images.
akhdefo_functions.stackprep (page 35)([...])	This program collects velocity candiate points for time-series analysis.
akhdefo_functions.Time_Series (page 37)([...])	This program uses candiate velocity points from stackprep function and performs linear interpolation in time-domain to calibrate stacked velocity.
akhdefo_functions.akhdefo_ts_plot (page 39)([...])	This program used for analysis time-series velocity profiles
akhdefo_functions.rasterClip (page 41)(rasterpath, ...)	This program used to clip single raster file.
akhdefo_functions.akhdefo_viewer (page 43)([...])	This program used for plotting raster products.
akhdefo_functions.Akhdefo_resample (page 45)([...])	This program performs raster resampling for rasters
akhdefo_functions.Akhdefo_inversion (page 47)([...])	This program calculates 3D displacement velocity (East-West,North-South and vertical) using combined optical and InSAR products
akhdefo_functions.utm_to_latlon (page 49)(easting, ...)	This program converts geographic projection of shapefiles from UTM to LATLONG
akhdefo_functions.MeanProducts_plot_ts (page 51)([...])	This program used to plot shapefile data
akhdefo_functions.Auto_Variogram (page 53)([...])	This program automatically selects best variogram model which later can be used to interpolate datapoints.
akhdefo_functions.binary_mask (page 55)(raster_path, ...)	Function that generates a binary mask from a vector file (shp)

Chapter 8

akhdefo_functions.unzip

`akhdefo_functions.unzip(`*zipdir*, *dst_dir*`)`

This program unzips all the zip products into one folder

8.1 Parameters

zipdir

[str] path to directory contains all the zipfiles

dst_dir

[str] path to destination folder to copy all unzipped products.

8.2 Returns

unzip folder

Chapter 9

akhdefo_functions.copyImage_Data

`akhdefo_functions.copyImage_Data`(*path_to_unzipped_folders=''*, *Path_to_raster_tifs=''*)

This program copy all the raster images.

9.1 Parameters

path_to_unzipped_folders : str

Path_to_raster_tifs : str

9.2 Returns

rasters

Chapter 10

akhdefo_functions.copyUDM2_Mask_Data

`akhdefo_functions.copyUDM2_Mask_Data`(*path_to_unzipped_folders=''*, *Path_to_UDM2raster_tifs=''*)

This program copy all raster masks.

10.1 Parameters

path_to_unzipped_folders

[str] file extension must end with udm2_clip.tif

Path_to_UDM2raster_tifs : str

10.2 Returns

rasters

Chapter 11

akhdefo_functions.Filter_PreProcess

`akhdefo_functions.Filter_PreProcess`(*unfiltered_folderPath=''*, *UDM2_maskfolderPath=''*, *outpath_dir=''*, *Udm_Mask_Option=False*)

This program prepare and uses filters to balanace raster image brightness

11.1 Parameters

unfiltered_folderPath : str

UDM2_maskfolderPath : str

outpath_dir : str

Udm_Mask_Option
[bool] False if True the program uses planetlabs imagery unusable pixel mask to ignore and mask bad image pixels

11.2 Returns

geotif rasters
Filtered geotif rasters

Figures
plotted filtered rasters and mask for bad pixels

Chapter 12

akhdefo_functions.Crop_to_AOI

`akhdefo_functions.Crop_to_AOI(`*Path_to_WorkingDir=''*, *Path_to_AOI_shapefile=''*, *output_CroppedDir=''*`)`

This program used to clip multiple raster files

12.1 Parameters

Path_to_WorkingDir
[str] path to raster working directory

Path_to_AOI_shapefile
[str] path to Area of interest in shapefile format

output_CroppedDir
[str] path to save cropped raster files

12.2 Returns

cropped raster files

Chapter 13

akhdefo_functions.Mosaic

`akhdefo_functions.Mosaic`(*Path_to_WorkingDir=''*, *output_MosaicDir=''*, *img_mode=0*)

This program mosiacs raster images in geotif format as well grab dates the satellite image taken for further processing. The current version only supports Planet Labs SurfaceReflectance products.

13.1 Parameters

Path_to_WorkingDir : str

output_MosaicDir : str

img_mode

[int] if img_mode=0 the the programs mosaics only the raster images. if img_mode=1 the program mosiacs only mask rasters

13.2 Returns

Mosaiced raster images

Chapter 14

akhdefo_functions.Coregistration

`akhdefo_functions.Coregistration`(*input_Folder='', output_folder='', grid_res=20, min_reliability=60, window_size=(64, 64), path_figures='', showFig=False, no_data=[0, 0], single_ref_path=''*)

This program coregisteres multiple rasters using both structural similarity index and feature matching techniqws. This program is written based on arosics python library.

14.1 Parameters

input_Folder: str
Path to input raster folders

grid_res: int

min_reliability: int
structural simialrity index threshold to differentaite deformation from raster shift (min=20, max=100)

window_size: tuple
window size for pixel search

showFig: bool
True to display results or False to not displat results

no_data: list
No data values to be ignored for both reference and target image

single_ref_path: str
provide path to raster if interested to coregister all rasters to a single reference, ignore this option the program uses subsequent rasters as refernce.

output_folder: str
returns coregistred and georeferenced raster in geotif format

path_figures: str
returns figure with plotted displaced pixels in raster coordinate system units

14.2 Returns

coregistred rasters

Chapter 15

akhdefo_functions.DynamicChangeDetection

`akhdefo_functions.DynamicChangeDetection(`*Path_working_Directory='', Path_UDM2_folder='', Path_to_DEMFile='', Coh_Thresh=0.75, vel_thresh=0.063, udm_mask_option=False, cmap='jet', Median_Filter=False, Set_fig_MinMax=False, show_figure=False, plot_option='origional', xres=10, yres=10*)

This program calculates optical flow velocity from triplets of daily optical satellite images. Final Timeseris products will be a shapefile format using Time_Series function after stackprep step.

15.1 Parameters

Path_working_Directory : str

Path_UDM2_folder : str

Path_to_DEMFile : str

Coh_Thresh : float

vel_thresh : float

udm_mask_option : bool

cmap : str

Median_Filter : bool

Set_fig_MinMax : bool

show_figure : bool

plot_option
[str] origional, resampled

xres : int

yres : int

15.2 Returns

Rasters

velocity in X direction(EW) Velocity in Y direction(NS)

Figures

Initial Timesereis Figures (those figures are only intermediate products needs calibration)

Chapter 16

akhdefo_functions.plot_stackNetwork

`akhdefo_functions.plot_stackNetwork`(*src_folder=''*, *output_folder=''*, *cmap='tab20'*, *date_plot_interval=(5, 30)*, *marker_size=15*)

This Program plots temporal network of triplets to be stacked for calculating Annual Mean Velocity from stacked optical images.

16.1 Parameters

src_folder
[str] path to georeferenced_folder

output_folder
[str] path to output folder to save output Figure plot

cmap
[str] colormap for the plot default is tab20

date_plot_interval
[list] minumum and maximum plot x axis interval dates for the plot

marker_size
[float] size of plotted points default is 15

16.2 Returns

Figure

Chapter 17

akhdefo_functions.stackprep

`akhdefo_functions.stackprep`(*path_to_flowxnFolder='', path_toFlowynFolder='', dem='', print_list=False, start_date='YYYYMMDD', end_date='YYYYMMDD', output_stackedFolder='', VEL_scale=('month', 'year'), xres=3.125, yres=3.125, Velocity_shapeFile=False, Resampling=True*)

This program collects velocity candiate points for time-series analysis.

17.1 Parameters

path_to_flowxnFolder
[str] path to folder include east-west velocity files

path_toFlowynFolder
[str] path to folder include north-south velocity files

dem
[str] path to digital elevation model file will be used to geocode the products

print_list
[bool] print list of temporal proceesed dates default is False

start_date
[str] YYYYMMDD

end_date
[str] YYYYMMDD

output_stackedFolder : str

VEL_scale
[str] month or year) at this stage you can ignore this option; will be removed from future versions

xres : float

yres : float

Velocity_shapeFile: bool

set to True if need to generate points for temporal deformation analysis

Resampling : bool

if True reduce number of measurement points but faster processing

17.2 Returns

ESRI Shapefile

This file include candiate velocity points for timeseries analysis

Chapter 18

akhdefo_functions.Time_Series

`akhdefo_functions.Time_Series`(*stacked_raster_EW=''*, *stacked_raster_NS=''*, *velocity_points=''*, *dates_name=''*, *output_folder=''*, *outputFilename=''*, *rasteriz_mean_products=True*, *std=1*, *VEL_Scale='year'*, *velocity_mode='mean'*, *master_reference=False*)

This program uses candiate velocity points from stackprep function and performs linear interpolation in time-domain to calibrate stacked velocity. Additionally produces corrected timeseries velocity(daily) in a shapefile.

18.1 Parameters

stacked_raster_EW : str

stacked_raster_NS : str

velocity_points
[str] Velcity Candidate points

dates_name
[str] text file include name of each date in format YYYYMMDD

output_folder : str

outputFilename : str

VEL_Scale
[str] year , month or empty to calculate velocity within provided dataset date range

velocity_mode
[str] mean or linear

master_reference
[bool] True if calculate TS to a single reference date, False if calculate TS to subsequent Reference dates

18.2 Returns

Time-series shape file of velocity and direction EW, NS, and 2D(resultant Velocity and direction)

Chapter 19

akhdefo_functions.akhdefo_ts_plot

`akhdefo_functions.akhdefo_ts_plot`(*path_to_shapefile=''*, *dem_path=''*, *point_size=1.0*, *opacity=0.75*, *cmap='turbo'*, *Set_fig_MinMax=True*, *MinMaxRange=[-50, 50]*, *color_field='VEL'*, *user_data_points=''*, *path_saveData_points=''*, *save_plot=False*, *Fig_outputDir=''*, *VEL_Scale='year'*)

This program used for analysis time-series velocity profiles

19.1 Parameters

user_data_points
[str] provide path to csv. file contains x and y coordinate for points of interest you can generate this file by providing path to path_saveData_points (POI.csv). This is useful to save mouse click positions to repeat the plots for different datasets for example if you plot several TS profiles for EW velocity product, you can recreate TS for the same exact position by saving POI.csv with path_saveData_points and then use that as input for the another plot such as NS velocity product via setting user_datapoints=POI.csv

path_to_shapefile
[str] type path to timeseries shapefile in stack_data/TS folder

dem_path
[str] path to dem raster in geotif fromat

point_size
[float] size of the sactter plot points

opacity
[float] transparency of the scater overlay

cmap
[str] Matplotlib colormap options example RdYlBu_r, jet, turbo, hsv, etc

Set_fig_MinMax
[bool] True or False

MinMaxRange
[list] [-50,50] Normalize plot colormap range if Set_fig_MinMax=True

color_field
[str] VEL ,VEL_2D, VEL_N, VEL_E, VELDir_MEA

path_saveData_points
[str] optional, provide directory path if you want to save profile data. the data will be saved under POI.csv file

save_plot: bool
True or False

Fig_outputDir
[str]

if save_plot=True then you save your profile plots in interactive html file and jpg image

VEL_Scale
[str] year or month projects the velocity into provided time-scale

19.2 Returns

Interactive Figures

Chapter 20

akhdefo_functions.rasterClip

`akhdefo_functions.rasterClip(`*rasterpath*, *aoi*, *outfilename*`)`

This program used to clip single raster file.

20.1 Parameters

rasterpath

[str] path to raster file in geotif format

aoi

[str] path to Area of interest in shapefile format

outfilename

[str] path to output raster file in geotif format .tif

20.2 Returns

clipped raster

Chapter 21

akhdefo_functions.akhdefo_viewer

`akhdefo_functions.akhdefo_viewer(`*Path_to_DEMFile='', rasterfile='', cbar_label='Velocity(mm/year)', title='Akhdefo-Viewer', pixel_resolution_meter=3.125, outputfolder='', alpha=0.8, unit=1, cmap='jet', noDATA_Mask=False, Normalize=True, SetDates_Filename=False, Set_fig_MinMax=False*`)`

This program used for plotting raster products.

21.1 Parameters

Path_to_DEMFile
[str] provide path to digital elevation raster file to be used as shaded base Map

rasterfile
[str] provide path to raster file to be plotted

title
[str] provide your desired title for the plot

pixel_resolution_meter
[float] provide pixel resolution of the digital elevation raster to draw proper figure scalebar

outputfolder
[str] provide path to ouput folder to save the plot

outputfileName
[str] provide name for the output plot including the desired extension such as .jpg, .pmg, .pdf, etc..

alpha
[float] transparency level for the plotted raster relative to hillshaded basemap

unit
[int] conversion unit default is 1 (no conversion) if your data is in meter such as velocity in meter/year set unit to 2 to convert to mm/year

noDATA_MAsk
[bool] set to True if you do not want to plot zero values of your dataset

21.2 Returns

Figure

Chapter 22

akhdefo_functions.Akhdefo_resample

akhdefo_functions.Akhdefo_resample(*input_raster=''*, *output_raster=''*, *xres=3.125*, *yres=3.125*, *SavFig=False*, *convert_units=False*)

This program performs raster resampling for rasters

22.1 Parameters

input_raster
[str] path to input raster

output_raster
[str] path to output raster

xres
[float] horizontal resolution

yres
[float] vertical resolution

SavFig
[bool] True to save output plot False to ignore exporting plot

convert_units
[bool] if True converts raster value units from m to mm

22.2 Returns

Raster geotif

Chapter 23

akhdefo_functions.Akhdefo_inversion

`akhdefo_functions.Akhdefo_inversion`(*horizontal_InSAR=''*, *Vertical_InSAR=''*, *EW_Akhdefo=''*, *NS_Akhdefo=''*, *demFile=''*, *output_folder=''*)

This program calculates 3D displacement velocity (East-West,North-South and vertical) using combined optical and InSAR products

23.1 Parameters

horizontal_InSAR
[str] path to East Velocity InSAR product in geotif format

Vertical_InSAR
[str] path to Vertical Velocity InSAR product in geotif format

EW_Akhdefo
[str] path to east-west velocity akhdefo(optical) product in geotif format

NS_Akhdefo
[str] path to north-south velocity akhdefo(optical) product in geotif format

demFile
[str] path to DEM raster in geotif format

output_folder
[str] path to save raster products

23.2 Returns

Three geotif rasters
3D-Velocity (D3D in mm/year) raster Plunge raster in degrees Trend raster in degress

Chapter 24

akhdefo_functions.utm_to_latlon

`akhdefo_functions.utm_to_latlon(`*easting*, *northing*, *zone_number*, *zone_letter*`)`

This program converts geographic projection of shapefiles from UTM to LATLONG

24.1 Parameters

easting: Geopandas column with Easting

northing: Geopandas column with Northing

zone_number: int

zone_letter: N or S

24.2 Returns

[lon , lat]: List

Chapter 25

akhdefo_functions.MeanProducts_plot_ts

`akhdefo_functions.MeanProducts_plot_ts`(*path_to_shapefile='', dem_path='', out_folder='Figs_analysis', color_field='', Set_fig_MinMax=False, MinMaxRange=[-100, 100], opacity=0.5, cmap='jet', point_size=1, cbar_label='mm/year', batch_plot=False, dates_list=''*)

This program used to plot shapefile data

25.1 Parameters

path_to_shapefile : str

dem_path : str

out_folder : str

color_field
[str] geopandas column name

Set_fig_MinMax : bool

MinMaxRange : list

opacity : float

cmap : str

point_size : str

cbar_label
[str] mm/year or degrees, etc.. based on unit of the data column name in the color_field

25.2 Returns

Figure

Chapter 26

akhdefo_functions.Auto_Variogram

`akhdefo_functions.Auto_Variogram`(*path_to_shapefile=''*, *column_attribute=''*, *latlon=False*)

This program automatically selects best variogram model which later can be used to interpolate datapoints.

26.1 Parameters

path_to_shapefile : str type path to shapefile to include data (point data) the shapefile attribute must have x, y or lat, lon columns

column_attribute

[str] Name of shapefile field attribute include data

26.2 Returns

str

name of best variogram model also figure for plotted variogram models

Chapter 27

akhdefo_functions.binary_mask

`akhdefo_functions.binary_mask`(*raster_path*, *shape_path*, *output_path*, *file_name*)

Function that generates a binary mask from a vector file (shp)

27.1 Parameters

raster_path = path to the .tif;

shape_path = path to the shapefile or GeoJson.

output_path = Path to save the binary mask.

file_name = Name of the file.

27.2 Returns

Raster Binary Mask in tif format

Chapter 28

Indices and tables

- genindex
- modindex
- search