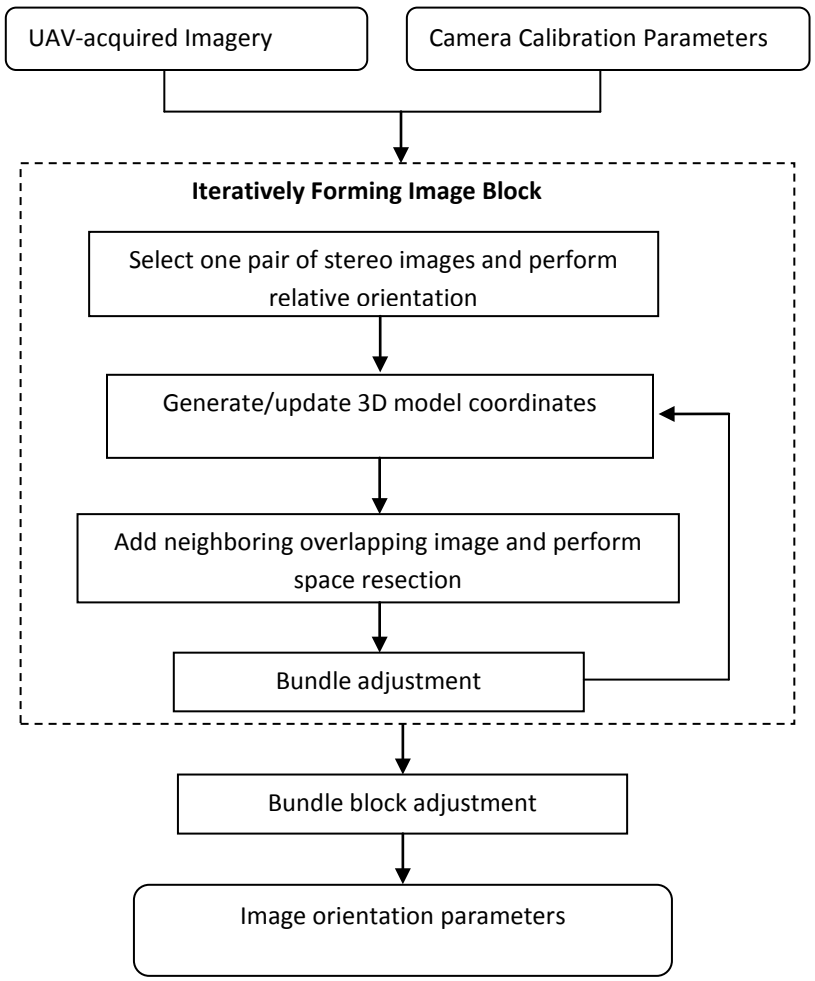
Manual

This simple manual briefly describes the usage of the programs developed in the project. The programs orientate the UAV-acquired imagery, and generate 3D model of road surface. They are developed in Matlab and C language.

Image Orientation

Image orientation is to determine the orientation of the images collected along the roads. This is necessary for georeferencing images and precise 3D point location. The complete procedure has been described in the technical report, and is shown in the flowchart below.



- Program names:
  - Relative Orientation: Ro_dep_x_l1_mtchs.m
  - Space Intersection: intersect_n1_new.m

- Bundle Adjustment: bundle_free_unf_fnl.m
- Resection: resect_new_l1.m (including SIFT matching)

- Procedure:
  - Start with RO (relative orientation) of two images and get there relative EO (Ro_dep_x_l1_mtchs.m)
  - Do SI (space intersection) to get the 3D point cloud that will be used in the next step of resection (intersect_n1_new.m)
  - Do BA (Bundle Adjustment) to compute adjusted 3D coordinates of points and EO's (bundle_free_unf_fnl.m)
  - Now add the third image by doing SR (Space Resection) to compute the EO's of the third image (resect_new_l1.m)
  - Redo SI with 3 images to compute the 3D coordinates of the point cloud (intersect_n1_new.m)
  - Redo BA with 3 images to compute the adjusted 3D coordinates of points and EO's (bundle_free_unf_fnl.m)
  - Now add the forth image by doing SR (Space Resection) to compute the EO's of the third image
  - Redo SI with 4 images to compute the 3D coordinates of the point cloud
  - Redo BA with 4 images to compute the adjusted 3D coordinates of points and EO's.
  - **Repeat for any new image until done**

Ro_dep_x_l1_mtchs.m

- Input

  Two images & extracted feature points. Example:

```
load im39_05.key;keys1=im39_05;im1=imread('IMG_1939.tif');

load im40_05.key;keys2=im40_05;im2=imread('IMG_1940.tif');
```

  Approximate EO's. Example:

```
Xl=0;Yl=0;Zl=45;wl=0.0;pl=0.0;kl=0;
Xr=0;Yr=-6;Zr=45;wr=0.0;pr=0.0;kr=0;
```

- Output

```
Relative Eo's for the two images, written (appears) on the
screen.
```

intersect_n1_new.m

- Input (program read the input files automatically from hard disk)
  The Eo's from the relative orientation that outcome from the last process (Ro_dep_x_l1_mtchs), just copy it from the screen and paste it in the file with no changes. Example:

```
infile=fopen('phts_pts14_int_gps.dat','r');
```

here, 'phts_pts14_int_gps.dat' is the file copied from the screen. The file contents and format are shown in following.

phts_pts14_int_gps.dat

```
Number of photos
2
Total number of points (from sift feature extraction) in the intersection
100 (n points)
Focal length, image width, image height, pixel size, xp, yp k1
k2 k3
52.6159 (f)
3888 2592 (w,h)
0.0057 (pxl_size)
0.0961 -0.0912 (xp,yp)
5.6136e-5 -4.7328e-08 2.3968e-10 (k1,k2,k3)
The total number of points in the photos(point/image)
200 (n*2)(number of points in image 1+ number of points in image 2)
The std of the image coordinates(m)
0.000015
The exposure station parameters and std
Photo#      X     Y     Z     W     Q     K
You get this from the RO
1.000000000000000                 0                  0  44.890000000000001
0.010087715386685  -0.005216232549410  -0.018690714774266
First imagery

2.000000000000000   0.139770000000000  -5.091300000000000  44.829999999999998
0.019588274871893  -0.006375799442946  -0.012347986233835
Second imagery

parameters to be calibrated foc, xp, yp, k1,k2,k3 if yes 1if no
0
0
0 0
0 0 0
parameters to be adjusted X,Y,Z,W,Q,K  if yes 1 if no 0
1 1 1
1 1 1

The image coordinates(m) (automatically generated from RO)
```

```
point#      X           Y           Index
   1 2136.8000 1479.7100 1 84
   2 582.3700 1226.1100 1 326
   3 1750.8600 1383.4600 1 359
   4 548.3300 1447.5900 1 378
   5 548.3300 1447.5900 1 379
```

(point 1 is in image 1, and its correspondence in image 2 will be in the same file in the format as: 1 xxxx.0000 yyyy.7100 2 zz). The index is actually not used.)

Output (3D point cloud with ID & least squares result properties)

- File name: gr_pts_out_intrsct14_gps.dat
- Example:

```
 1.0000000e+000  2.9231716e+000 -1.5220952e-001 -1.0903476e+000
 2.0000000e+000  6.3997297e+001  2.0458220e-008
```

Point number, x, y, z, number of images the point appeared in, sigma of the point, max value of delta from the least squares for the point

bundle_free_unf_fnl.m

This program takes the output from Ro_dep_x_l1_mtchs.m and intersect_n1_new.m. following above examples, the two files are

phts_pts14_int_gps.dat

and gr_pts_out_intrsct14_gps.dat

The program outputs the updated 3D ground coordinates

Example: gr_pts_out_bndl_14_gps.dat

```
Point number,        x,              y,         z

  1.2000000e+001  1.4573826e+000 -3.8022580e+000  9.4034950e-001
  1.8000000e+001  4.9105224e+000 -4.9684622e+000  1.0912460e+000
  3.8000000e+001 -6.1064009e+000 -1.8027132e+000  1.1459677e+000
  4.9000000e+001 -6.6908994e+000 -3.0361596e+000  1.1857089e+000
  5.2000000e+001 -5.2244586e+000 -3.1724927e+000  1.2240998e+000
  5.4000000e+001 -5.2232748e+000 -3.1726768e+000  1.2334125e+000
```

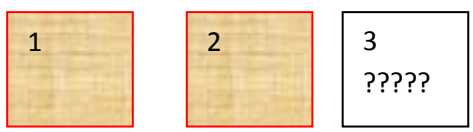
resect_new_l1.m

This program uses the computed 3D coordinates of the ground points to estimate the exterior orientation of a new image.

- Input
  - `exterior=[-2 +18 45 0 0 3.14]; this is the approximate parameters for the new image.`
  - `prmtrs=[52.6159 3888 2592 0.0057 0.0961 -0.0912 5.6136e-5 -4.7328e-08 2.3968e-10]; this is the camea interior orientation parameter. Since the setting of camera is not changed in the project, these parameters should not be changed.`
  - `3D points from bundle adjustment above. In this example, the file name is` gr_pts_out_bndl_14_gps.dat.
  - `Two images (the one you would like to compute EO's & the last one from the previous process). Example:`

    ```
    load im39_05.key;keys1=im39_05;im1=imread('IMG_1939.tif');

    load im40_05.key;keys2=im40_05;im2=imread('IMG_1940.tif');
    ```



Suppose 1 & 2 have been computed and we want to add 3. Then use the images & key points for 2 & 3

During resection, image 2 & 3 will be matched with SIFT matching. We then run "my_mtchs.m" to sort out the matched points. Afterwards, the matched points in image 3 will be manually identified, selected and pasted to "**phts_pts14_int_gps.dat". This file is very important for following multi-image intersection and bundle adjustment.**

- Output
  - `Written on the screen (the computed EO's for the new image).`

<u>Now we intersect 3 images and then bundle again with the 3 images</u>

LEAST SQUARES MATCHING

Preparing for image matching

Code: C:\work1\intersect

We run the program intersect_pts.m and it will generate us the grid for input in the image matching.
The input to this program is exactly the same input to the intersect_n1_new.m.

Output: coordinates of the 3D points . `gr_out_1_in.dat (as X Y Z grid)`

Example: (see below `gr_out_1_in.dat in the` image_data_uav)

X Y Z

```
-4.0000000e+000 -4.0000000e+000 -2.1113274e+000
 -4.0000000e+000 -3.9500000e+000 -2.1113274e+000
 -4.0000000e+000 -3.9000000e+000 -2.0535673e+000
 -4.0000000e+000 -3.8500000e+000 -2.0535673e+000
 -4.0000000e+000 -3.8000000e+000 -2.0535673e+000
 -4.0000000e+000 -3.7500000e+000 -2.0535673e+000
 -4.0000000e+000 -3.7000000e+000 -2.0535673e+000
 -4.0000000e+000 -3.6500000e+000 -2.0535673e+000
 -4.0000000e+000 -3.6000000e+000 -2.0535673e+000
 -4.0000000e+000 -3.5500000e+000 -2.0535673e+000
 -4.0000000e+000 -3.5000000e+000 -2.0535673e+000
 -4.0000000e+000 -3.4500000e+000 -2.0535673e+000
```

Parameters in the file intersect_pts.m:

```
xmin=-8; // min X on the ground
xmax=0; // max X on the ground
ymin=-14.000; // min Y on the ground
ymax=+2.000; // max Y on the ground
intrvl=0.01; // interval on the ground
```

Image matching

Code: C:\work1\REF

The input file is `gr_out_1_in.dat` generated above.

The output file is `gr_out_1_ot.dat.` see image_data_uav In side below.

Output:

Pt # X Y Z …….

```
1 -4.000000 -16.000000 -2.617382
```

Input:

image_data_uav.dat (inside the code `char *input_fn="image_data_uav.dat"; DON'T NEED TO OPEN CODE`)

- the num of images , dz, num pts
- 4 0.0001 3627344(number of grid points in the gr_out_1_in.dat for CC and LS)
-
- gr_out_1_in.dat gr_out_1_ot.dat (output file you take to surfer)
-
- the image data
- im_2929org_wallis.pgm image name
- 0.0961 -0.0912 52.6159 0.0057 xp,yp,f,pxl_sz
- -0.043940818971103 -0.143133869992773 45.017785467769421
- 0.003067878907014 -0.000347619022755 -0.000257916795351
- X Y Z
- Omega Phi Kapa
- 2592 3888 image width & hieght
-
- im_2930org_wallis.pgm
- 0.0961 -0.0912 52.6159 0.0057
- -0.110934818530809 -6.113275316287858 44.416465630397539
- 0.027650972662792 0.007158579737142 0.025073443599470
- 2592 3888
-
- im_2931org_wallis.pgm
- 0.0961 -0.0912 52.6159 0.0057
- -0.222767600939832 -12.319879799098016 43.724751321463671
- 0.042431442317582 0.015868017387046 0.039108832563617
- 2592 3888
-
- im_2932org_wallis.pgm
- 0.0961 -0.0912 52.6159 0.0057
- 0.164535168179316 -17.394153684836517 43.034997635333255
- 0.042904793304082 0.027929064407301 0.074375859049356
- 2592 3888

Input image

gr_out_1_ot.dat:
Example: pt num, X, Y, Z
i,x,y,DEM[i][3]

```
1 -4.000000 -13.000000 -2.427627 -2.473033 3844.096500 1187.956520
2911.888365 1217.233652 1815.425872 1331.343370 777.207981 1415.595467

2 -4.000000 -12.950000 -2.399720 -2.473033 3835.821167 1187.503116
2902.754474 1217.025189 1805.474219 1331.310227 766.588640 1415.911605

3 -4.000000 -12.900000 -2.416866 -2.473033 3825.171784 1187.781722
2892.331468 1217.592286 1795.427417 1332.019165 756.912863 1416.988481
```

Parameters:

In code:

~~**#define MAXIMG 12**~~

#define winsz 19 (number of columns and rows in the ground window)

~~**#define win_sz_xy *0.005 (image pixel size on the ground in meters)***~~

In input file (image_data_uav.dat) (YOU DON'T WANY TO CHANGE THE dz)

```
the num of images , dz,    num pts
        5          0.0001  451351
```

CORRELATION MATCHING

Preparing for image matching

Code: C:\work1\intersect

We run the program intersect_pts.m and it will generate us the grid for input in the image matching.

The input to this program is exactly the same input to the intersect_n1_new.m.

Output: coordinates of the 3D points . `gr_out_1_in.dat`

Example: (see below `gr_out_1_in.dat in the` image_data_uav)

X Y Z

Parameters in the file intersect_pts.m:

```
xmin=-8; // min X on the ground
xmax=0; // max X on the ground
ymin=-14.000; // min Y on the ground
ymax=+2.000; // max Y on the ground
intrvl=0.01; // interval on the ground
```

Image matching

Code: C:\work1\DEM_GEN

The input file is `gr_out_1_in.dat` generated above.

The output file is `gr_out_1_ot.dat.` see image_data_uav In side below.

Output:

Pt # X Y Z …….

```
1 -4.000000 -16.000000 -2.617382
```

Input:

image_data_uav.dat (inside the code `char *input_fn="image_data_uav.dat"; DON'T NEED TO OPEN CODE`)

- the num of images , minf_level, num pts
- 4 ~~1~~ 3627344 (the second number is not used in CC, but used in LS)
-
- gr_out_1_in.dat gr_out_1_ot.dat (output file you take to surfer)
-
- the image data
- im_2929org_wallis.pgm image name
- 0.0961 -0.0912 52.6159 0.0057 xp,yp,f,pxl_sz
- -0.043940818971103 -0.143133869992773 45.017785467769421
- 0.003067878907014 -0.000347619022755 -0.000257916795351
- X Y Z
- Omega Phi Kapa
- 2592 3888 image width & hieght
-
- im_2930org_wallis.pgm
- 0.0961 -0.0912 52.6159 0.0057
- -0.110934818530809 -6.113275316287858 44.416465630397539
- 0.027650972662792 0.007158579737142 0.025073443599470
- 2592 3888
-
- im_2931org_wallis.pgm
- 0.0961 -0.0912 52.6159 0.0057
- -0.222767600939832 -12.319879799098016 43.724751321463671
- 0.042431442317582 0.015868017387046 0.039108832563617
- 2592 3888
-
- im_2932org_wallis.pgm
- 0.0961 -0.0912 52.6159 0.0057
- 0.164535168179316 -17.394153684836517 43.034997635333255
- 0.042904793304082 0.027929064407301 0.074375859049356
- 2592 3888

Input image

gr_out_1_ot.dat:
Example: pt num, X, Y, Z
i,x,y,DEM[i][3]

1 -4.000000 -13.000000 -2.427627 -2.473033 3844.096500 1187.956520
2911.888365 1217.233652 1815.425872 1331.343370 777.207981 1415.595467

2 -4.000000 -12.950000 -2.399720 -2.473033 3835.821167 1187.503116
2902.754474 1217.025189 1805.474219 1331.310227 766.588640 1415.911605

3 -4.000000 -12.900000 -2.416866 -2.473033 3825.171784 1187.781722
2892.331468 1217.592286 1795.427417 1332.019165 756.912863 1416.988481

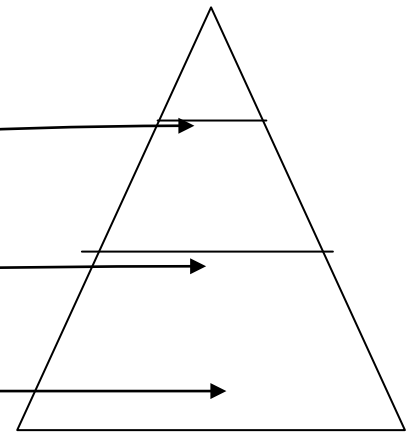
Parameters:

In code: **line 696**

```
Dz=0.10;L4=10;winsz=35;
DEM_gen(4,L4,DEMx,DEMy,DEMz);

Dz=0.05;L2=10;winsz=37;
DEM_gen(2,L2,DEMx,DEMy,DEMz);

Dz=0.020;L1=10;     winsz=39;
DEM_gen(1,L1,DEMx,DEMy,DEMz);
```



Dz= step interval
L4, L2, L1=total number of steps;
winsz=35; window size

Producing the 3D image

With the produced 3D points, 3D surface model and 3D images can be generated and visualized. Metric measurements can also be conducted. This is realized in commercial software package availables in GIScCE, SDSU.

SURFER:

First we have the output from LS which is a grid file of gr_out_1_ot.dat:

Example: pt num, X, Y, Z

i,x,y,DEM[i][3]

1 -4.0 -13.0 -2.4 -2.473 3844.096 1187.950 2911.88 1217.2 1815.4 1331.3 777.20 1415.5

In surfer:

Go grid->data->open your file

Map-> new> 3D surface:

We need to convert to ascii

Select GRD Surfer 6 Text Grid *.grd

ERDAS:

First import the grid from surfer ascii to .img

After generating of orthoimages we mosaic them from data preparation in ERDAS main menu. The full procedures of this process can be found in ERDAS documents.

ARCSCENE:

The generated orthoimages and 3D models can be loaded in ARCSCENE where tools are available to measure features of interest, including dimension, area, and volume etc.