Tutorial 34: 2D Projection Images and Camera Coordinate System Reconstruction

2017-05-15: Tim C. Lueth, MIMED - Technische Universität München, Germany (URL: http://www.mimed.de) - Last Change: 2017-07-25

Contents

Complete List of all Tutorials with Publishable MATLAB Files of this Solid-Geoemtries Toolbox

The following topics are covered an explained in the specific tutorials:

- Tutorial 01: First Steps Using the VLFL-Toolbox for Solid Object Design
- Tutorial 02: Using the VLFL-Toolbox for STL-File Export and Import
- Tutorial 03: Closed 2D Contours and Boolean Operations in 2D
- Tutorial 04: 2½D Design Using Boolean Operators on Closed Polygon Lists (CPL)
- Tutorial 05: Creation, Relative Positioning and Merging of Solid Geometries (SG)
- Tutorial 06: Relative Positioning and Alignment of Solid Geometries (SG)
- Tutorial 07: Rotation of Closed Polygon Lists for Solid Geometry Design
- Tutorial 08: Slicing, Closing, Cutting and Separation of Solid Geometries
- Tutorial 09: Boolean Operations with Solid Geometries
- Tutorial 10: Packaging of Sets of Solid Geometries (SG)
- Tutorial 11: Attaching Coordinates Frames to Create Kinematik Models
- Tutorial 12: Define Robot Kinematics and Detect Collisions
- Tutorial 13: Mounting Faces and Conversion of Blocks into Leightweight-structures
- Tutorial 14: Manipulation Functions for Closed Polygons and Laser Cutting (SVG)
- Tutorial 15: Create a Solid by 2 Closed Polygons
- Tutorial 16: Create Tube-Style Solids by Succeeding Polygons
- Tutorial 17: Filling and Bending of Polygons and Solids
- Tutorial 18: Analyzing and modifying STL files from CSG modeler (Catia)
- Tutorial 19: Creating drawing templates and dimensioning from polygon lines
- Tutorial 20: Programmatically Interface to SimMechanics Multi-Body Toolbox
- Tutorial 21: Programmatically Convert Joints into Drives (SimMechanics)
- Tutorial 22: Adding Simulink Signals to Record Frame Movements
- Tutorial 23: Automatic Creation of a Missing Link and 3D Print of a Complete Model
- Tutorial 24: Automatic Creation of a Joint Limitations
- Tutorial 25: Automatic Creation of Video Titels, Endtitels and Textpages
- Tutorial 26: Create Mechanisms using Universal Planar Links
- Tutorial 27: Fourbar-Linkage: 2 Pose Syntheses and Linkage Export for 3D Printing
- Tutorial 28: Fourbar-Linkage: 3 Pose Syntheses and Linkage Export for 3D Printing
- Tutorial 29: Create a multi body simulation using several mass points
- Tutorial 30: Creating graphical drawings using point, lines, surfaces, frames etc.
- Tutorial 31: Importing 3D Medical DICOM Image Data and converting into 3D Solids
- Tutorial 32: Exchanging Data with a FileMaker Database
- Tutorial 33: Using a Round-Robin realtime multi-tasking system
- Tutorial 34: 2D Projection Images and Camera Coordinate System Reconstruction
- Tutorial 35: Creation of Kinematic Chains and Robot Structures
- Tutorial 36: Creating a Patient-Individual Arm-Skin Protector-Shell
- Tutorial 37: Dimensioning of STL Files and Surface Data
- Tutorial 38: Some more solid geometry modelling function

Motivation for this tutorial: (Originally SolidGeometry 3.8 required)

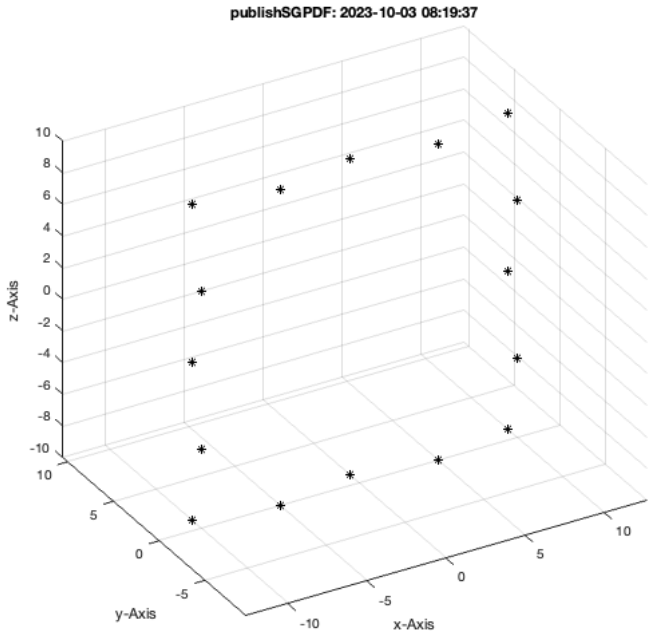
Many surgical procedures in orthopedics are not based on three-dimensional CT or MRI image data but on C-arm images. These C-arm images are 2D projection images of a spatial region of the patient. In this, the most important strategies for the conversion of volume images to projection images are presented. It is also explained how the position of the X-ray camera can be calculated from projection images, if one knows the exact location of objects in space and the 2D image. The research area is also called Camera Calibration.

ATTENTION >>> The Publishermode changes the aspect ratio of figures, therefor it is strongly recommended to copy lines from this tutorial instead of just executing the publishabe example

1. Create a number of random points around the center

The following commands could be typed in absolute in the same way as part

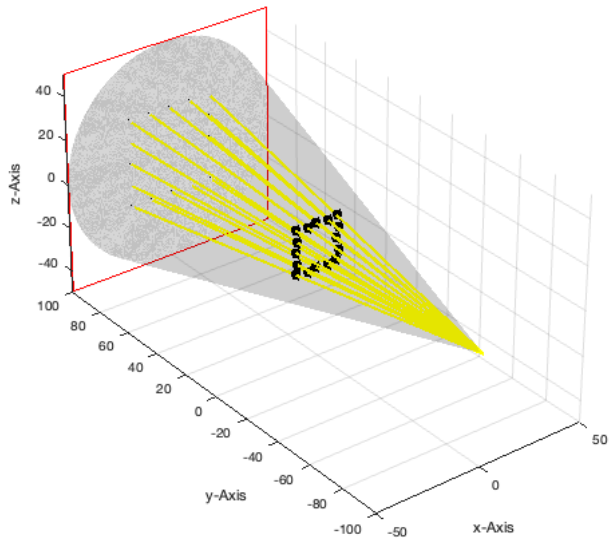
```
SGfigure; view(-30,30); xlabel 'x-Axis', ylabel 'y-Axis', zlabel 'z-Axis';
VL=50*rand(10,3)-25; VL(:,2)=1*rand(10,1)';
% VL=VLsample(9)
VL=VLsample(11); VL=VLtransT(VL,TofR(rot(-pi/20,0,pi/20)));
VL=VLsample(12);
VLplot(VL,'k*');
```



2. Create an X-ray image by using the camera parameter of Matlab

The x-ray source is at position [0 100 0]; The target is at [0 +100 0] The screen has a size of 100x100 The scaling factor is 4, i.e. the pixel size if 025 x 0.25 mm

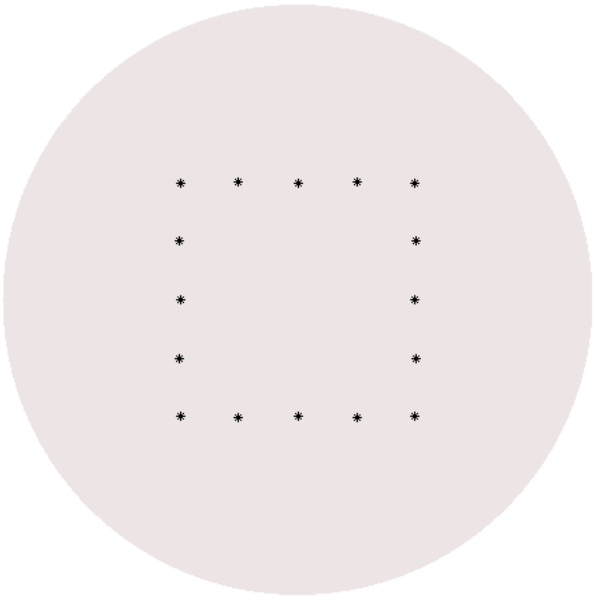
```
imageofVLprojection(VL,[100 100],[0 -100 0],[0 100 0],4);
set(gca,'Projection','perspective');    % this line is only required because of publishing function
```



show the image

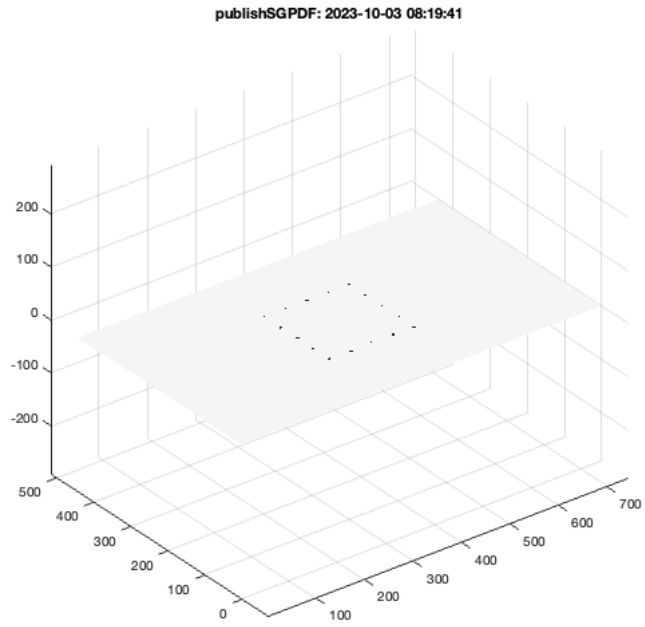
```
I=imageofVLprojection(VL,[100 100],[0 -100 0],[0 100 0],4);
set(gca,'Projection','perspective');    % this line is only required because of publishing function
whos I                                  % this line is only required because of publishing function
```

```
  Name        Size               Bytes  Class     Attributes

  I         440x744            2618880  double
```



```
SGfigure
imwarpT(I);
```

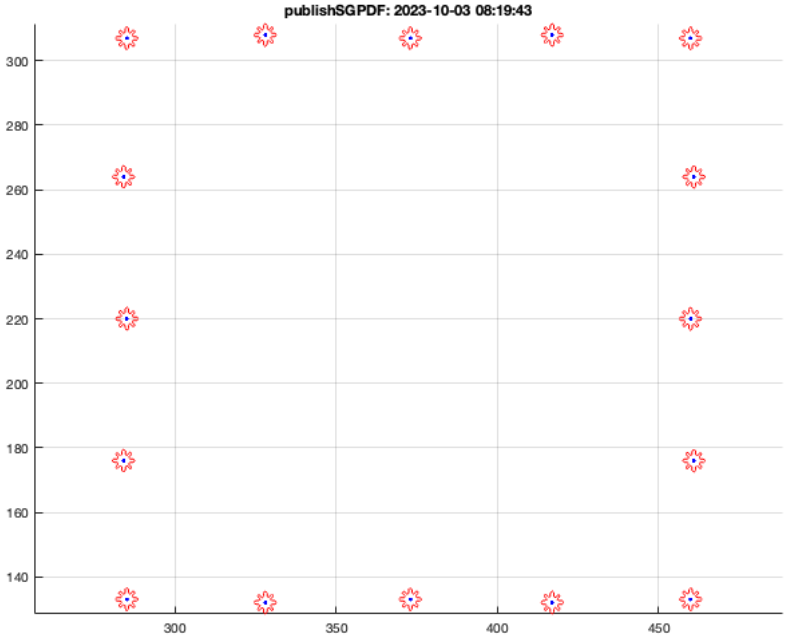
```
ans =
  Figure (1: AOI Matlab Solid Modeler app_2012_11_09) with properties:

      Number: 1
        Name: 'AOI Matlab Solid Modeler app_2012_11_09'
       Color: [1 1 0.9000]
    Position: [31 803 960 540]
       Units: 'pixels'

  Use GET to show all properties
```



3. Find the marker points in the image

```
SGfigure;
CPL=CPLcontourc(I,1); % Contour segmentation on image base
CPLplot(CPL,'r-');
PL=centerCPL(CPL)
PLplot(PL,'b.',4);
size(I,2)                              % this line is only required because of publishing function
```

```
PL =
   284   264
   284   176
   461   264
   461   176
   285   307
   285   220
   285   133
   460   307
   460   220
   460   133
   328   308
   328   132
   417   308
   417   132
   373   307
   373   133
ans =
   744
```

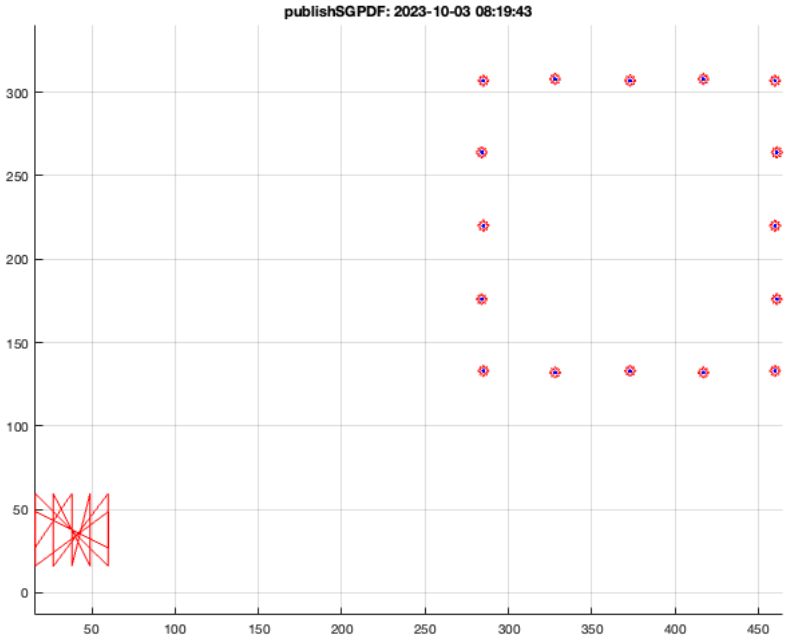


turn the coordinate

```
PL(:,2)=-PL(:,2)+size(I,2)   % flip up and down (y-axis)
PL=(PL-1)-size(I)/2          % Move coordinate into center
PL=PL/4                      % Scale using pixle size
CPLplot(PL,'r-');
```

```
PL =
   284   480
   284   568
   461   480
   461   568
   285   437
   285   524
   285   611
   460   437
   460   524
   460   611
   328   436
   328   612
   417   436
   417   612
   373   437
   373   611
PL =
    63   107
    63   195
```

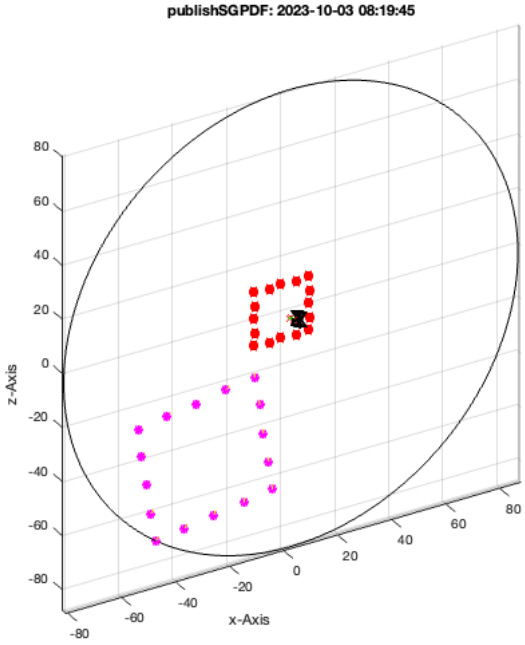
```
   240   107
   240   195
    64    64
    64   151
    64   238
   239    64
   239   151
   239   238
   107    63
   107   239
   196    63
   196   239
   152    64
   152   238
PL =
   15.7500   26.7500
   15.7500   48.7500
   60.0000   26.7500
   60.0000   48.7500
   16.0000   16.0000
   16.0000   37.7500
   16.0000   59.5000
   59.7500   16.0000
   59.7500   37.7500
   59.7500   59.5000
   26.7500   15.7500
   26.7500   59.7500
   49.0000   15.7500
   49.0000   59.7500
   38.0000   16.0000
   38.0000   59.5000
```



Some knowledge on corresponding axis

```
TofcamVLPL(sortrows(VL,[1 3]),sortrows(PL,[1 2]))
```

```
K =
   -1.1927   -0.1958   26.3421
    0.0000    0.8946   26.0685
    0.0000   -0.0000    0.5765
s =
    1.7345
ans =
    0.9888    0.1491   -0.0042    3.4747
    0.0048   -0.0036    1.0000   -0.9897
    0.1491   -0.9888   -0.0042   -3.5332
         0         0         0    1.0000
```



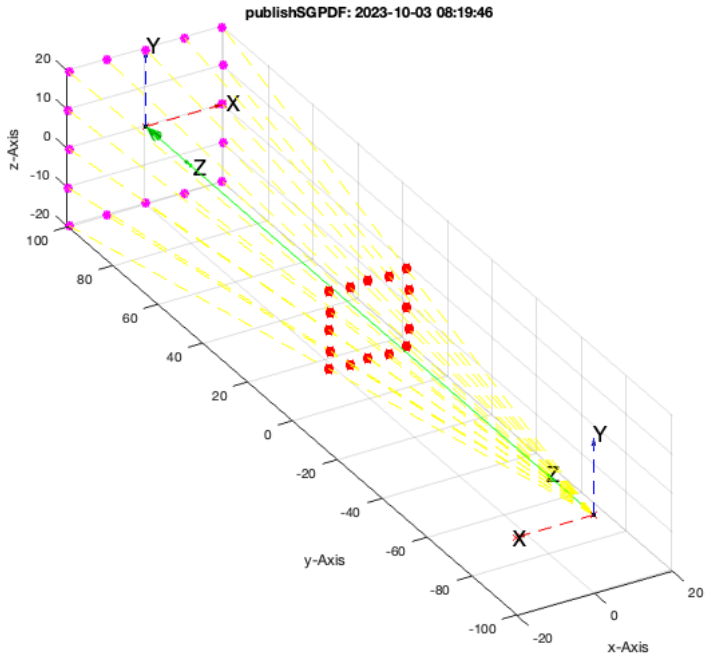
3. Calculate the Point Position of a X-Ray Camera

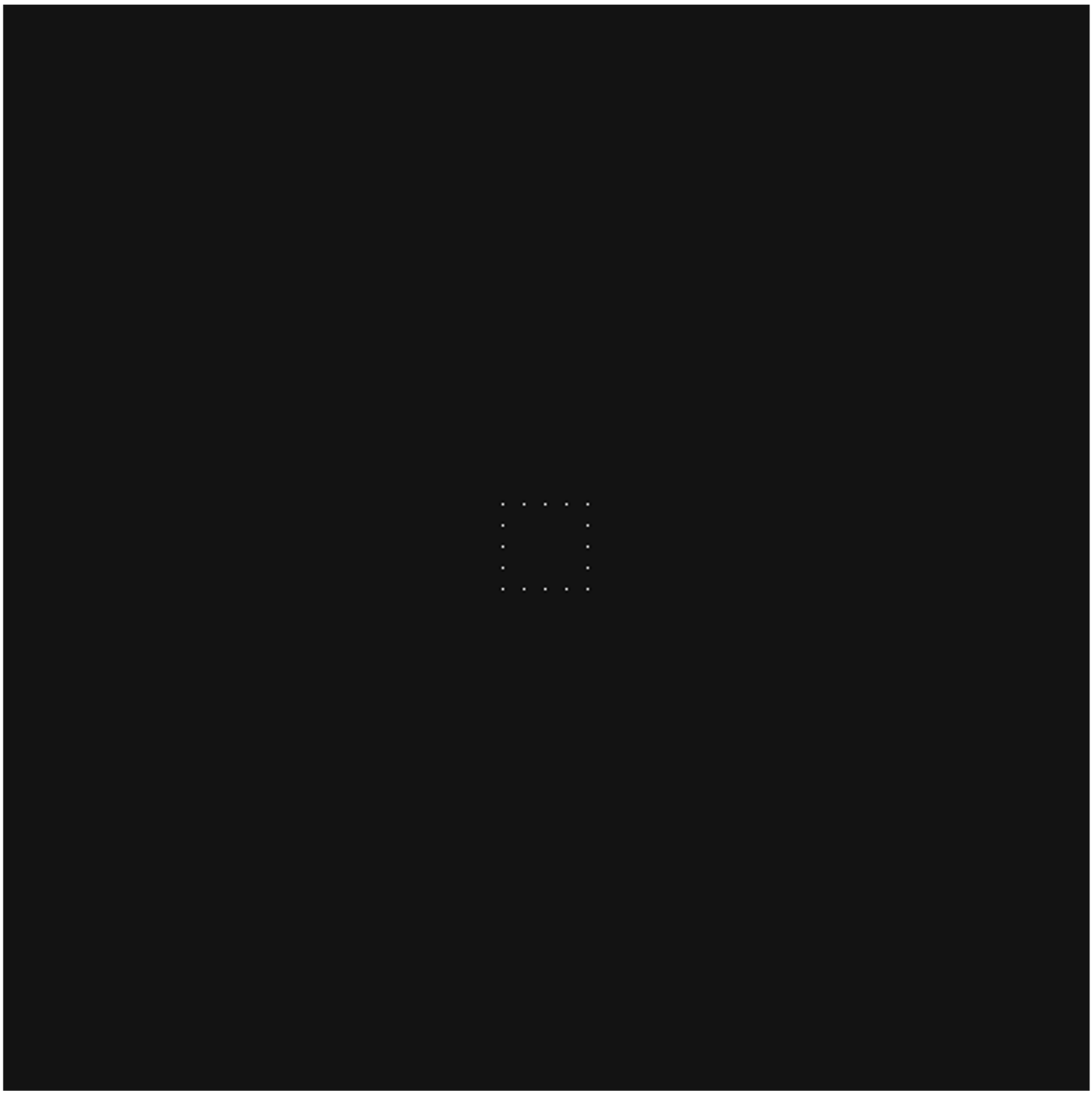
The x-ray source is at position [0 100 0]; The target is at [0 +100 0]

```
PLofVLprojection(VL,[0 -100 0],[0 100 0]);
PL=PLofVLprojection(VL,[0 -100 0],[0 100 0])
```

```
  Name        Size              Bytes  Class     Attributes

  I         512x512           2097152  double

  Name        Size              Bytes  Class     Attributes

  I         512x512           2097152  double

PL =
  -19.8020  -19.8020
  -10.0000  -20.0000
         0  -19.8020
   10.0000  -20.0000
   19.8020  -19.8020
   20.0000  -10.0000
   19.8020         0
   20.0000   10.0000
   19.8020   19.8020
   10.0000   20.0000
```

```
         0   19.8020
  -10.0000   20.0000
  -19.8020   19.8020
  -20.0000   10.0000
  -19.8020         0
  -20.0000  -10.0000
```

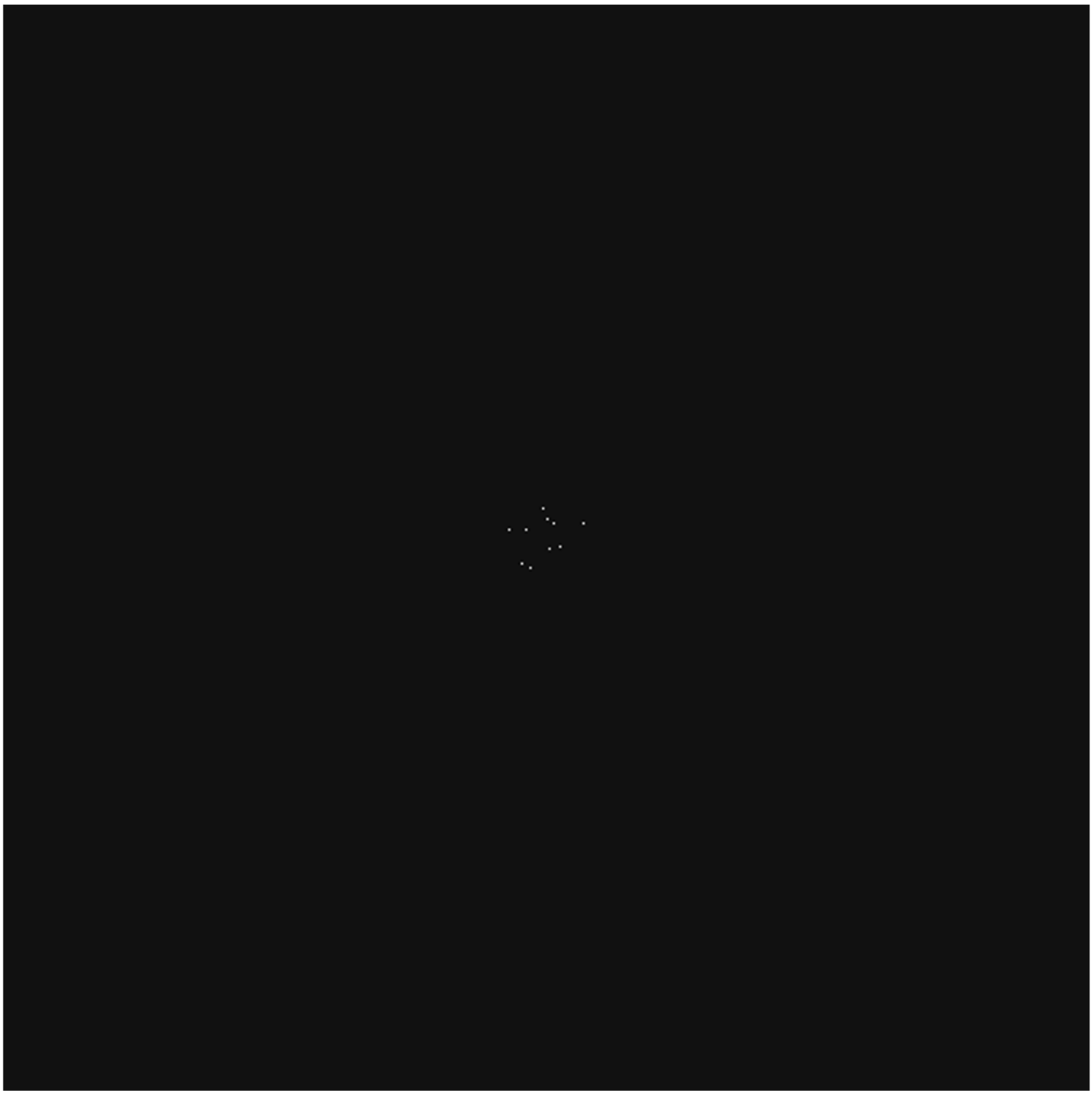




5. Comparision of point lists created by numerical projection or projection image reconstruction

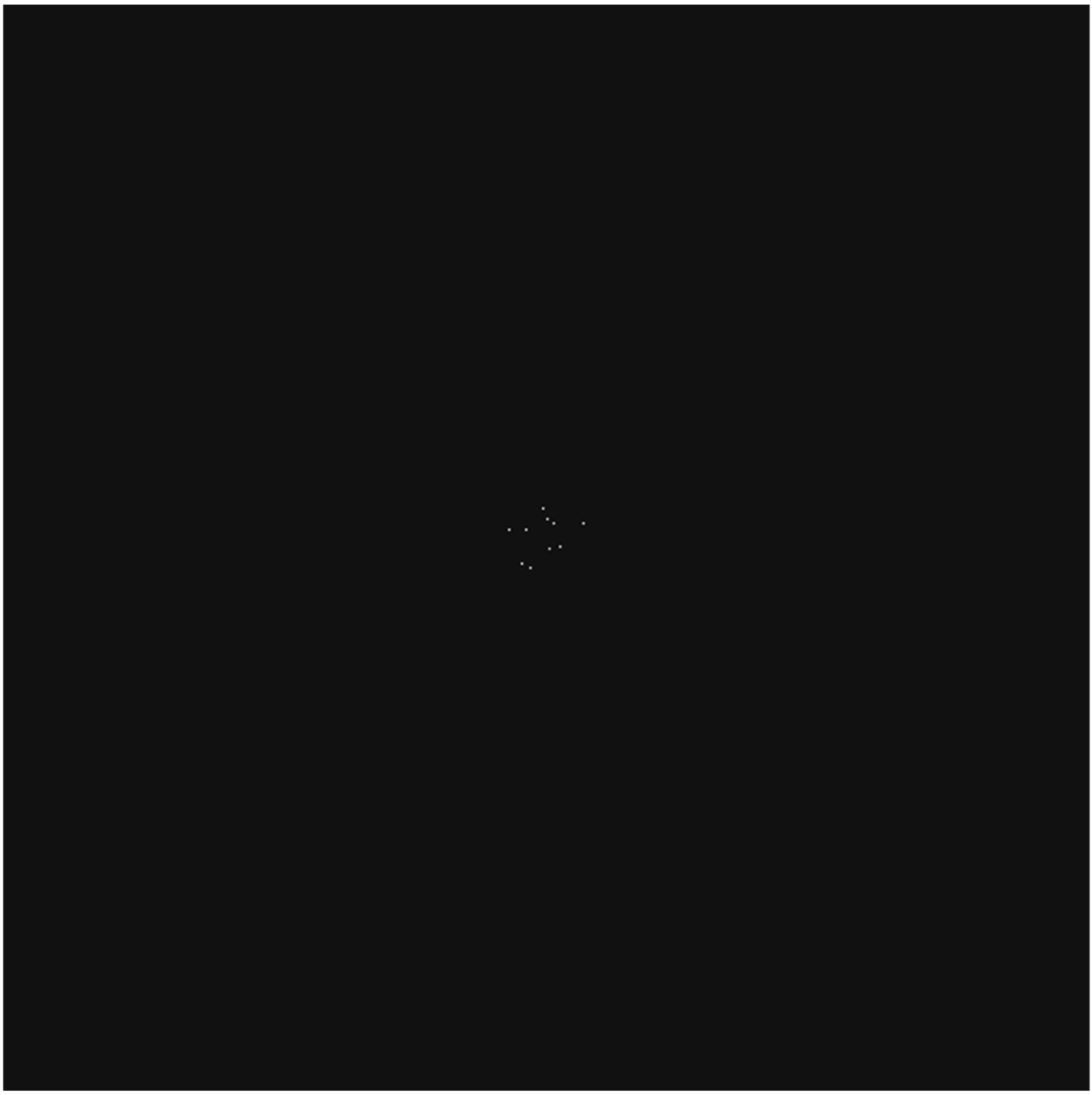
```
VL=20*rand(10,3)-10; VL(:,2)=5*rand(10,1)';
I=imageofVLprojection(VL,[100 100],[0 -100 0],[0 100 0],4);
[sortrows(PLofimcontourc(I,true,1/4)) sortrows(PLofVLprojection(VL,[0 -100 0],[0 100 0]))]
```

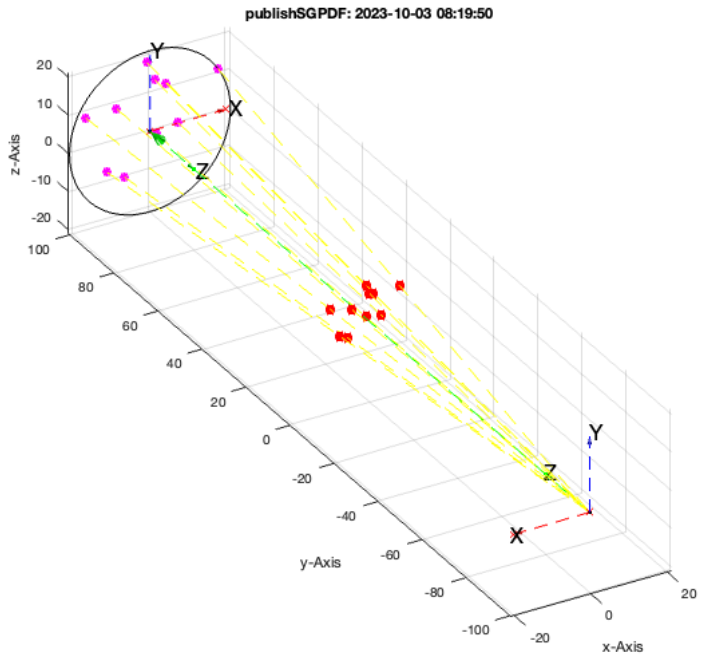
```
  Name        Size               Bytes  Class     Attributes

  I         512x512            2097152  double

ans =
  -17.1250    8.3750  -16.8123    8.2529
  -11.6250   -7.6250  -11.2556   -7.5212
   -8.8750    8.3750   -8.6804    8.2927
   -6.8750  -10.1250   -6.6762  -10.1249
   -1.1250   18.1250   -0.7593   18.2097
    0.8750   13.1250    1.2352   13.0986
    1.3750   -1.1250    1.6112   -1.0634
    3.8750   11.3750    4.2393   11.3588
    7.1250    0.1250    7.2909    0.2278
   17.6250   11.3750   17.8217   11.3458
```



```
TofcamVLPL(sortrows(VL,[1 3]),sortrows(PLofVLprojection(VL,[0 -100 0],[0 100 0]),[1 2]))
```

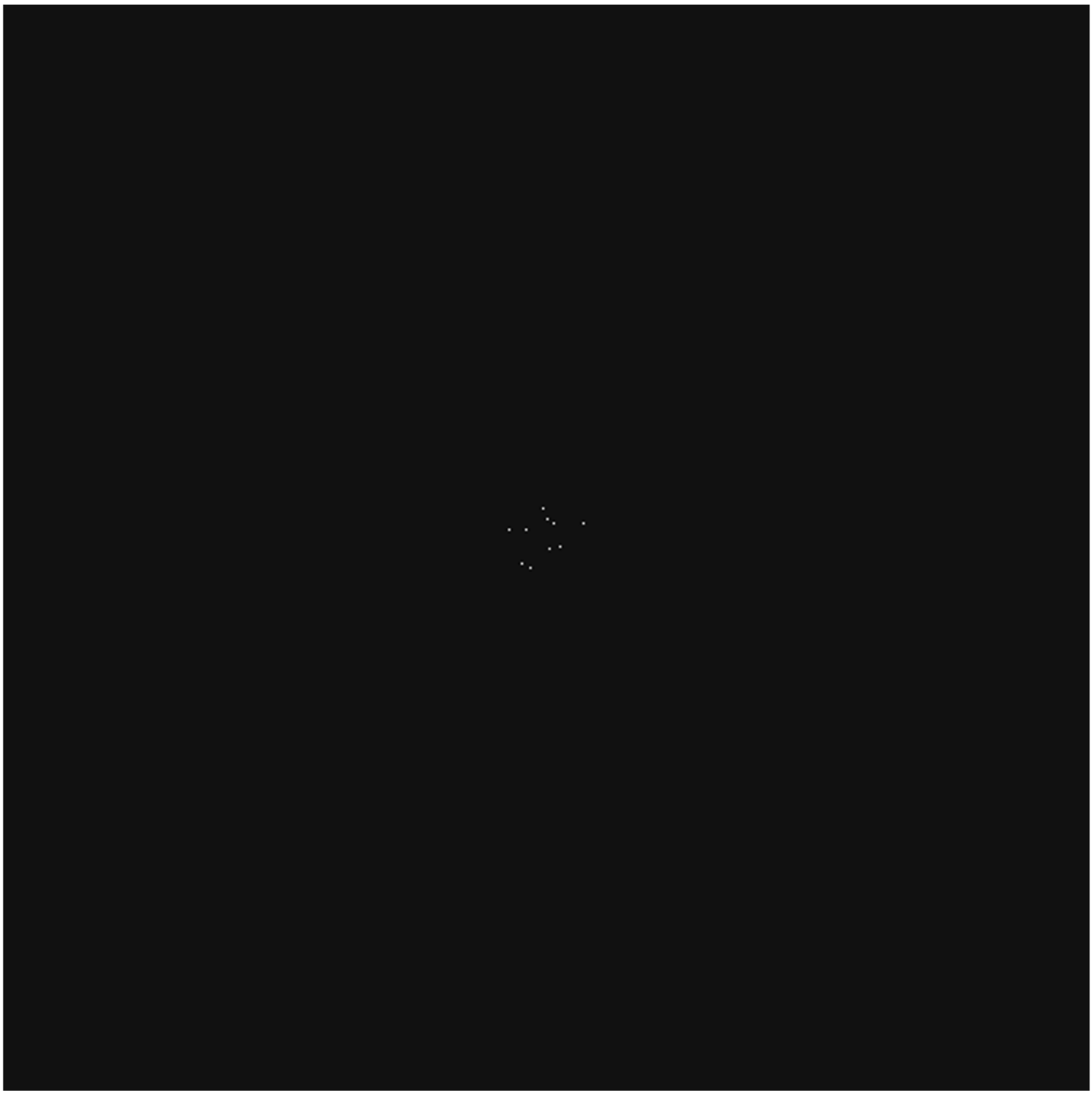
```
  Name        Size              Bytes  Class     Attributes

  I         512x512           2097152  double

K =
    1.0000   -0.0000    0.0000
         0    1.0000    0.0000
   -0.0000   -0.0000    0.0050
s =
   200
ans =
   -1.0000         0         0    0.0000
         0         0    1.0000 -100.0000
         0    1.0000         0    0.0000
         0         0         0    1.0000
```

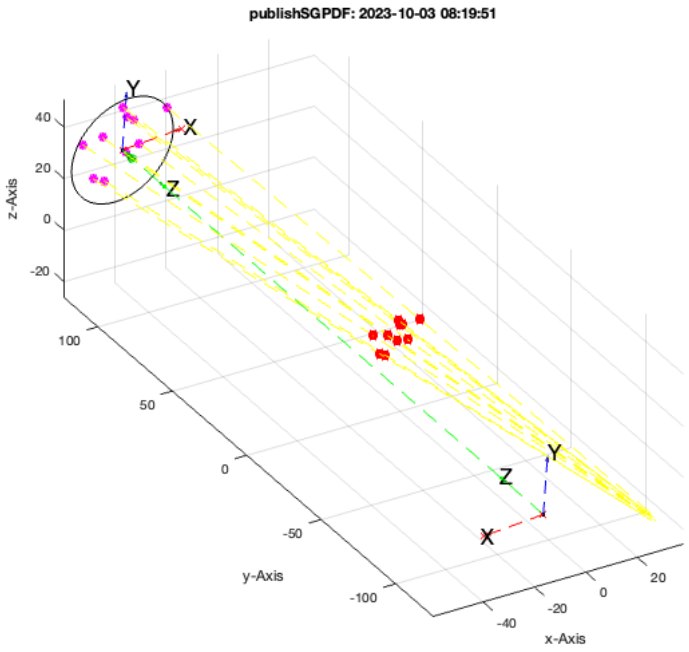





```
TofcamVLPL(sortrows(VL,[1 3]),sortrows(PLofimcontourc(I,true,1/4),[1 2]))
```

```
K =
   -0.9896   -0.0043   -0.1179
    0.0000    0.9930    0.1179
   -0.0000   -0.0000    0.0041
s =
  245.0885
ans =
   -0.9953    0.0047   -0.0971  -14.9715
   -0.0970   -0.1112    0.9890 -122.5733
   -0.0061    0.9938    0.1112   -0.6167
         0         0         0    1.0000
```





Final Remarks

```
close all
VLFLlicense
```

```
This VLFL-Lib, Rel.  (2023-Oct-03), is for limited non commercial educational use only!
Licensee: Tim Lueth (Development Version)!
Please contact Tim Lueth, Professor at TU Munich, Germany!
WARNING: This VLFL-Lib (Rel. ) license will exceed at 06-Jul-2078 08:19:53!
Executed 03-Oct-2023 08:19:55 by 'timlueth' on a MACI64 using Mac OSX 13.6 | R2023a Update 5 | SG-Lib 5.4
======================================== Used Matlab products: ========================================
database_toolbox
distrib_computing_toolbox
fixed_point_toolbox
image_toolbox
map_toolbox
matlab
simmechanics
simscape
simulink
================================================================================================================
```

Published with MATLAB® R2023a