Tutorial 51: Creating Parallel Tasks for batch processing

2020-08-25: Tim C. Lueth, Professor at Technische Universität München, Germany (URL: http://www.SG-Lib.org) - Last Change: 2020-08-28

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

Motivation for this tutorial: (Originally SolidGeometry 5.0 required)

Motivation for this tutorial: (Originally SolidGeometry 4.0 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

```
clear all      % For Scripts make shure that no variables are misdeclared
```

1. Create a pool of parallel processes

```
% Clear the current parallel pool if one already exists
if ~isempty(gcp('nocreate')); delete(gcp); end;
% Create a local hardware cluster instance of class parcluster named mc
mc = parcluster('local');
% Check if the local cluster if big enough
nworkers = 2;
if mc.NumWorkers < nworkers
    nworkers = mc.NumWorkers;
%     disp(['The computer can start only ' num2str(nworkers) ' workers.'])
%     fprintf (2,'%s\n',['The computer can start only ' num2str(nworkers) ' workers.']);
    warning(['The computer can start only ' num2str(nworkers) ' workers.']);
end
```

```
% Start the pool by creating nworkers copies of matlab
% The expected starting time on a 2.9 GHz Intel Core i7 with 16 GB Ram is
% about 16 seconds plus 2 sec per core
st=tic; parpool(nworkers); tim=toc(st)
fprintf('Expected matlab to core loading time is about %.2f seconds\n',tim/nworkers);
```

```
Starting parallel pool (parpool) using the 'Processes' profile ...
Connected to parallel pool with 2 workers.
tim =
   30.0391
Expected matlab to core loading time is about 15.02 seconds
```

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 08:51:12!
Executed 03-Oct-2023 08:51:14 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
optimization_toolbox
pde_toolbox
simmechanics
simscape
simulink
=========================================================================================================
```

Published with MATLAB® R2023a