iso2mesh: an image-based mesh generation toolbox

**Automatic surface and volumetric mesh generation from 3D binary or gray-scale images**

http://iso2mesh.sourceforge.net

Qianqian Fang and David Boas

**Motivation:** To generate triangular surface or tetrahedral volumetric meshes is typically an important pre-processing step in many popular numerical modeling methods, such as the boundary element (BE) or finite element (FE) methods. One can find abundant meshing tools, freely or commercially available, to process simple geometries. However, there is an increasing need to produce meshes from complex geometries represented by 3D volumetric images, for example, the tissue structures in a segmental or raw medical MRI or CT images. This need becomes critical for many emerging multi-modality medical imaging approaches where the structural images fuse with functional data. iso2mesh is a free mesh generation tool and is capable of producing high quality surface and volumetric meshes directly from 3D binary or grayscale images. This toolbox was written in Matlab language and is compatible with Matlab(7.0+) and GNU Octave (3.0+) on multiple platforms.

**iso2mesh: an image-based mesh generation toolbox**

- **Matlab:** iso2mesh is free and open-source under the terms of GPL.
- **Repos:** iso2mesh repository containing installation and how-to guides.
- **Testing:** iso2mesh can be run on Linux, Windows (2008) and Mac OS X to both PowerPC or Intel processors.
- **Motivation:** iso2mesh is highly structured and one can cascade the atomic meshing operations to achieve more complicated meshing tasks.

**iso2mesh features:**
- **Surface extraction:** iso2mesh can be used to mesh binary, grayscale or surface patches. This can exactly meet the needs for mesh generation.
- **Fast:** iso2mesh routines were optimized by vectorization and profiling. Many of the underlying tools were compiled with optimization flags.
- **Meshing multiple regions:** iso2mesh extraction containing multiple regions with different connectivity.
- **Multiple mesh format support:** iso2mesh supports writing output files in mesh format.

**iso2mesh workflow**

**Step 1:** (preparation of the binary image)
- Create a 3D binary image from brain images.
- `s2m(node,face,1,100);`

**Step 2:** (convert the binary image to volumetric mesh in 1 line)
- `v2m(node,elem,value,plane);`

**Step 3:** (preparation of the binary image)
- Create a 3D binary image from brain images.
- `s2m(node,face,1,100);`

**Step 4:** (convert the surface to tetrahedral mesh)
- `v2m(node,elem,value,plane);`

**What iso2mesh can do?**
- Surface extraction
- Surface simplification in meshing
- Surface visualization and repair
- Volumetric mesh generation
- Mesh slicing
- Adaptive meshing
- Meshing internal compartments
- Support for irregularly shaped domains
- Mesh quality

**iso2mesh one-liner demonstration**

```matlab
% iso2mesh one-liner demonstration
fig = surfm(surfedge); % extract individual loops from an edge table of a loop collection
surf(edgeimg,fig,...
```

**Web sites and Open Source Baseline Tools**

- CGAL
- Tetgen
- JMeshLib

**iso2mesh one-liner demonstration**

```matlab
% iso2mesh one-liner demonstration
fig = surfm(surfedge); % extract individual loops from an edge table of a loop collection
surf(edgeimg,fig,...
```

**(do more with iso2mesh)**

Mesh generation from open surfaces

isoface is able to generate a weighted surface from a open surface. There are many surface triangulators in the literature. However, the majority of them is not able to deal with open surfaces. For this reason, we decided to implement an open surface meshing algorithm. This algorithm is based on the Marching Cubes algorithm and the classical surface triangulation algorithm. The result is a triangular surface mesh.

1. `surf(edgeimg,fig,...`
2. `surf(edgeimg,fig,...`
3. `surf(edgeimg,fig,...`
4. `surf(edgeimg,fig,...`
5. `surf(edgeimg,fig,...`
6. `surf(edgeimg,fig,...`
7. `surf(edgeimg,fig,...`

Mesh generation from gray-scale images

To process a grayscale image is as straightforward as binary images. The only thing that changes is the image segmentation and the mesh generation. As we want to extract a surface mesh from a grayscale image, we need to define a threshold. Once we have defined a threshold, we can use iso2mesh to generate a surface mesh. The result is a triangular surface mesh.

1. `surf(edgeimg,fig,...`
2. `surf(edgeimg,fig,...`
3. `surf(edgeimg,fig,...`
4. `surf(edgeimg,fig,...`
5. `surf(edgeimg,fig,...`
6. `surf(edgeimg,fig,...`
7. `surf(edgeimg,fig,...`