

# **f3d**

## **Format for Storage of 3D Grids**

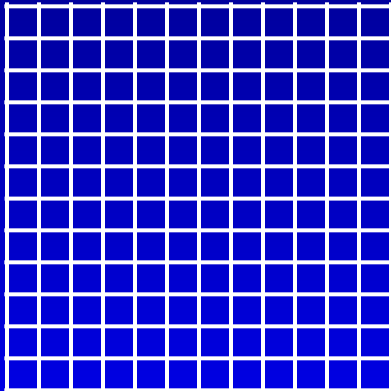
---

**Miloš Šrámek**

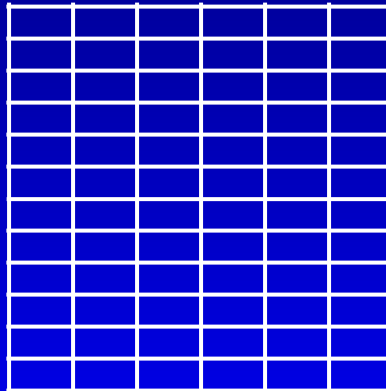
**Austrian Academy of Sciences**

# What would be nice...

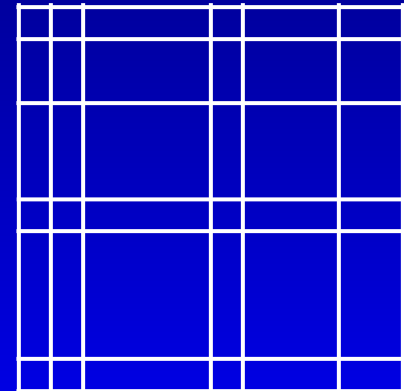
- ◆ **Support for different kinds of voxels**
  - ◆ Scalar and color voxels
  - ◆ 8, 16, 32 bit integers, signed and unsigned, floats
- ◆ **Different kinds of grids**



cartesian



regular



rectilinear

# What would be nice...

---

- ◆ **Self descriptive**

- ◆ Stores all necessary data attributes

- ◆ **Application specific comments**

- ◆ Provide means to transfer information between different applications

- ◆ **Preview image**

- ◆ An icon image, giving an idea about the contents of the file.

- ◆ **Data compression**

# What would be nice...

---

- ◆ **Cross platform compatibility**
  - ◆ Little/Big endian
  - ◆ Size of voxel representation
- ◆ **Availability**
  - ◆ Covered by a non-restrictive license
- ◆ **Ease of usage**

# 3D Formats and Approaches

---

- ◆ **Sequence of slices**

- ◆ Preview is possible, otherwise awkward

- ◆ **Raw 3D Data**

- ◆ Easy to use (load), hard to manipulate
- ◆ Everything else is bad

- ◆ **Package specific formats (AVS, DE, VTK):**

- ◆ Too general (structured/unstructured, geometry, topology, attributes)
- ◆ Not available outside the package

# Existing 3D Formats

---

- ◆ **HDF and netCDF**
  - ◆ Too complicated
  - ◆ No preview
- ◆ **DICOM (ACR-NEMA)**
  - ◆ Complex, only 2D format, (usually) no compression
- ◆ **ANALYZE**
  - ◆ No preview, no compression
- ◆ **ETC...**

# The 'f3d' Format

A 3D extension of the PNM format

- ◆ **Keeps the basic simple structure of PNM:**
  - ◆ Magic (P5, P6)
  - ◆ # Comments
  - ◆ Dimensions (2D)
  - ◆ 2D image data (preview)
- ◆ **3D extension**
  - ◆ 3D data is appended as compressed 2D slices (zlib compression)
  - ◆ Parameters stored as PNM comments

# Features

- ◆ **Preview** (data reprojected along an axis)
  - ◆ Any image viewer
  - ◆ Simplifies manipulation with files (what the hell is in this file?)
- ◆ **ASCII header and comments**
  - ◆ Viewable by image viewers (xv)
  - ◆ Viewable by vi (can be even edited!)
- ◆ **Compression**
  - ◆ Ratio up to 50 for synthetic data or 2 for noisy measured data



# PNM - f3d Header Comments

- ◆ Regular PNM comments, e.g.:
  - ◆ `# http://www.viskom.oeaw.ac.at/~milos`
- ◆ f3d format attributes, e.g.:
  - ◆ `#!f3d 1.2`
  - ◆ `#!vdim 128 128 128`
- ◆ f3d format comments
  - ◆ `#!comment` File created by 'vxtGrid3D'
- ◆ Application specific parameters, e.g.:
  - ◆ `#!comment vxt density 0.0 254.0`

# Basic C Library (libf3dformat)

---

## ◆ Reading:

- ◆ f3dReadHeader

- ◆ f3dReadSlice

- ◆ f3dReadGrid, f3dReadCubGrid

## ◆ Writing: analogous

## ◆ Other:

- ◆ f3dSetHdrComment, f3dDelHdrComment

- ◆ f3dHostType

- ◆ f3dVoxelSize

# C++ Class Library (libf3dclass)

---

- ◆ **Basic access to data voxels**
- ◆ **File storage and retrieval**
- ◆ **3D data processing**
  - ◆ **filters, segmentation, rendering, interpolation, transformations, gradient, ...**
- ◆ **Typeless reading of 3D data**

# f3d Tools

- ◆ **f3d functionality implemented in the form of UNIX filters:**
  - ◆ `f3dprog [sw] [input.f3d] > output.f3d`
  - ◆ **Filter concatenation is possible:**  
`tool1 [sw] in.f3d |...| toolN [sw] > out.f3d`
- ◆ **f3dview – simple slice viewer**

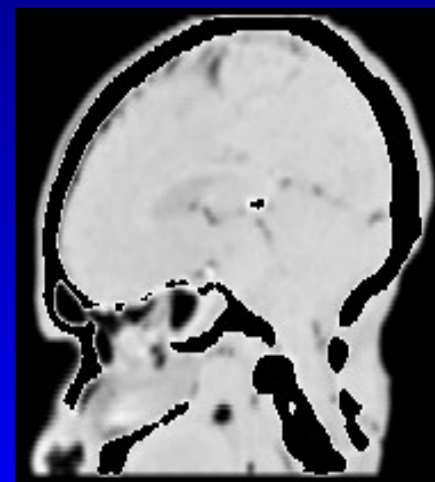
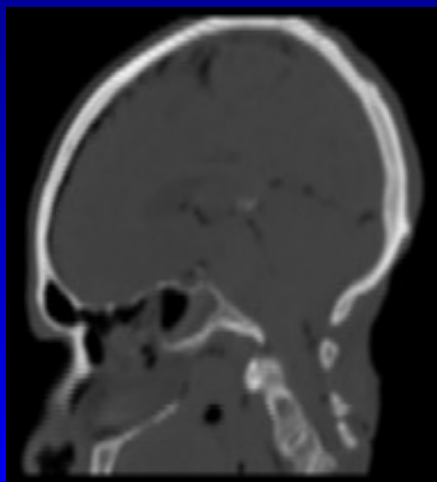
# f3d Tools: Format Conversion

- ◆ **Readers: raw data, stack of 8 and 16 bit slices, VTK, DICOM**
- ◆ **Example: convert a raw volume with 32 bit voxels to f3d:**  
`raw2f3d -i in.raw -r -l 5 -s 256 256 128 > out.f3d`
- ◆ **Command usage:** `f3dcmd -h`

# f3d Tools: Point Operators

- ◆ f3d2f3d (**voxel type conversion**), f3dinvert, f3dthresh, f3darith, f3d2rith2, f3dmask
- ◆ **Example: Delete bone from CT data:**

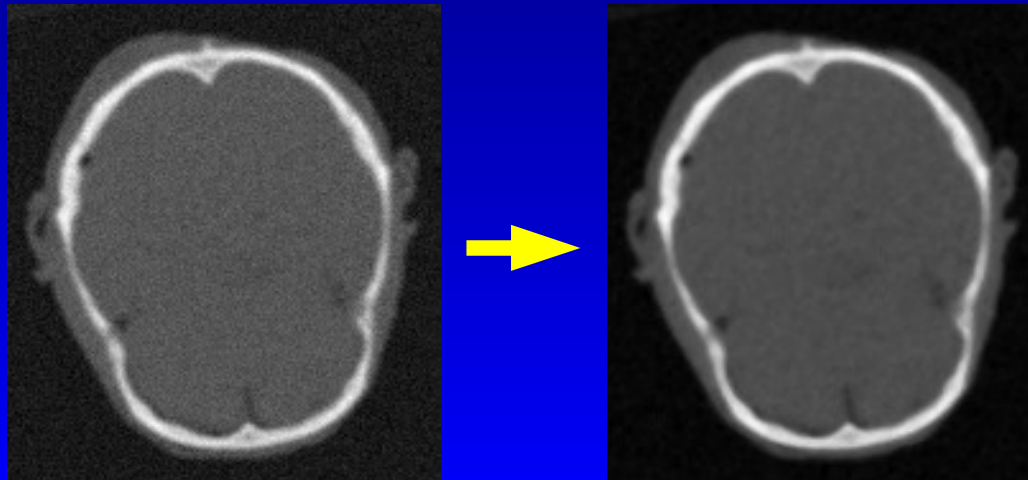
```
f3dthresh -lo 90 in.f3d | f3dmask -i in.f3d > out.f3d
```



# f3d Tools: Local Operators (Filters)

- ◆ Data processing in a local voxel neighborhood
- ◆ Order filters: f3dmax, f3dmin, f3dmedian
- ◆ Example: Noise removal by the median filter

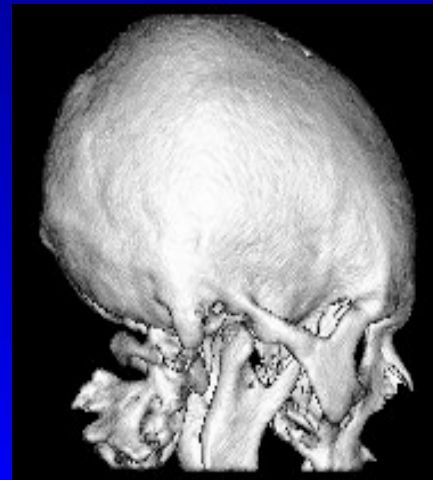
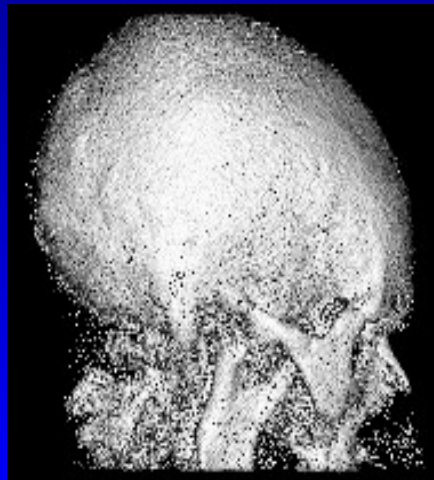
```
f3dmedian -k 3 in.f3d > out.f3d
```



# f3d Tools: Local Operators (Filters)

- ◆ Data processing in a local voxel neighborhood
- ◆ Order filters: f3dmax, f3dmin, f3dmedian
- ◆ Example: Noise removal by the median filter

```
f3dmedian -k 3 in.f3d | f3drender -lo 90 -surfg > out.png
```



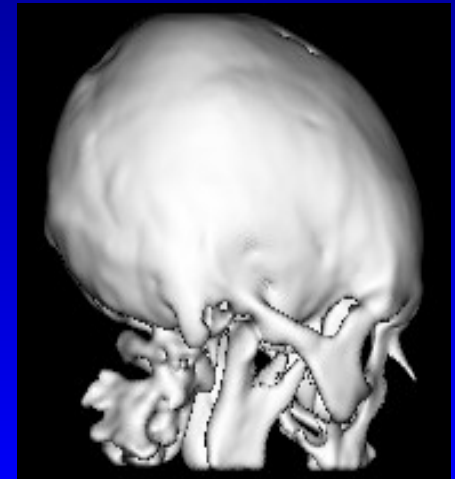
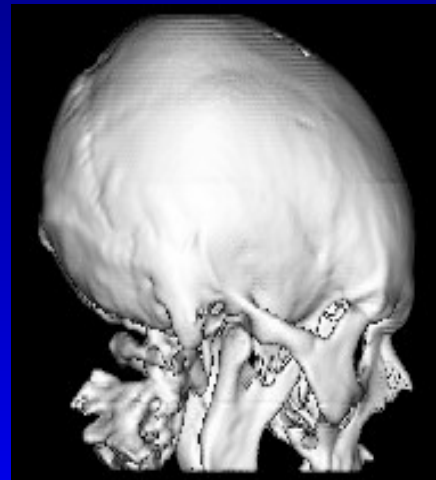
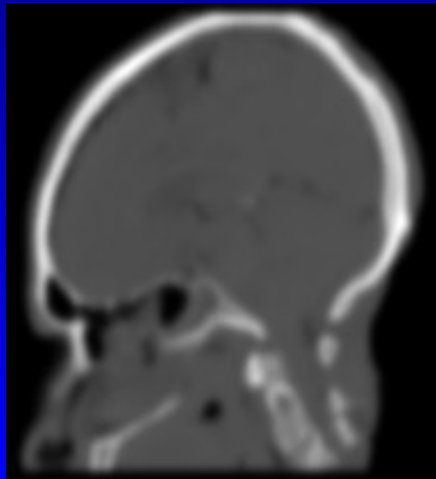
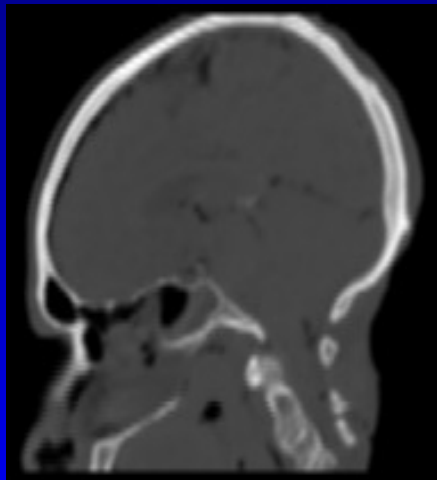


# f3d Tools: Local Operators

## Gaussian filtering

- ◆ Convolution by a Gaussian with different widths
- ◆ Example (data smoothing):

```
f3dgauss -w 2 in.f3d > out.f3d
```

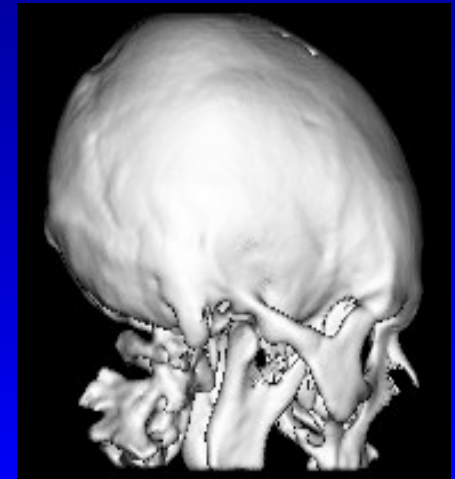
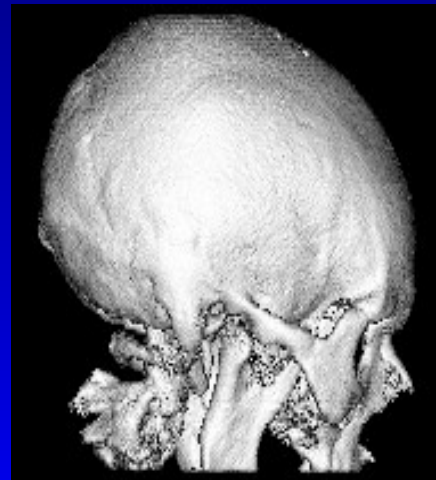
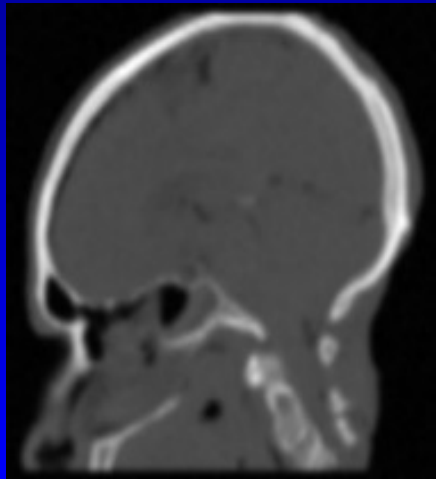
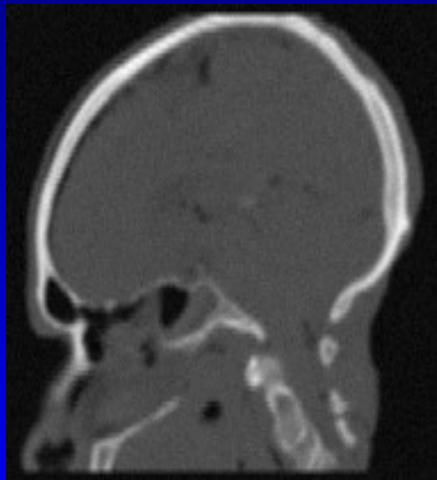


# f3d Tools: Local Operators

## Gaussian filtering

- ◆ Convolution by a Gaussian with different widths
- ◆ Example (noise removal):

```
f3dgauss -w 2 in.f3d > out.f3d
```

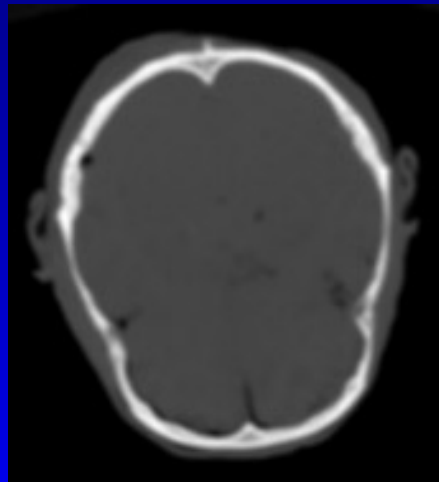


# f3d Tools: Local Operators

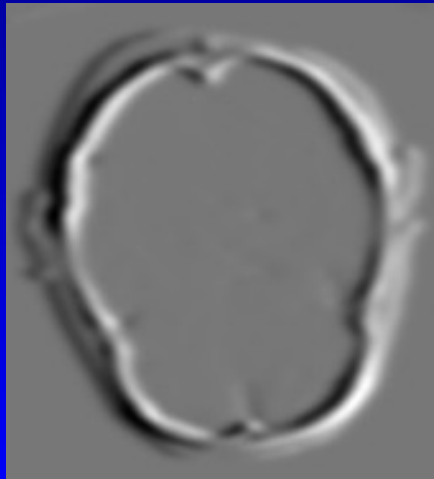
## Gradient Estimation

- ◆ Convolution by a Gabor filter with different widths
- ◆ Example :

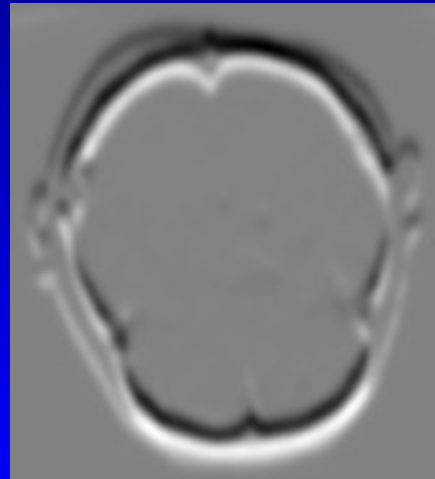
```
f3dgabor -x -w 3 in.f3d > out.f3d
```



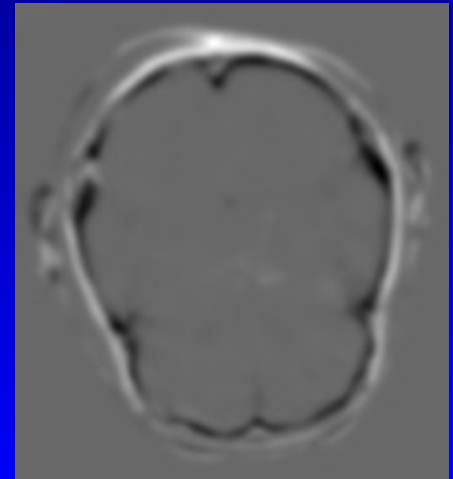
-x



-y



-z

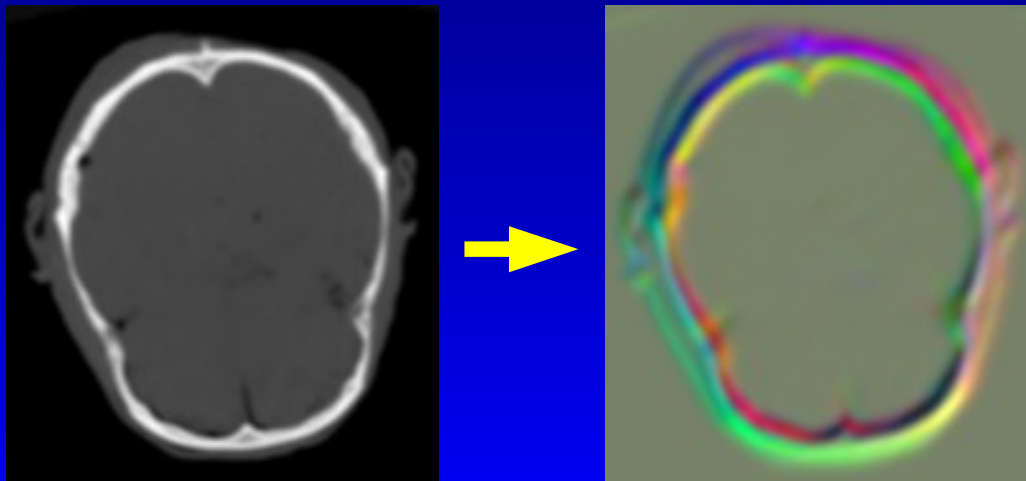


# f3d Tools: Local Operators

## Gradient Estimation

- ◆ Convolution by a Gabor with different widths in all 3 directions
- ◆ Example :

```
f3dgrad -w 3 in.f3d > out.f3d
```

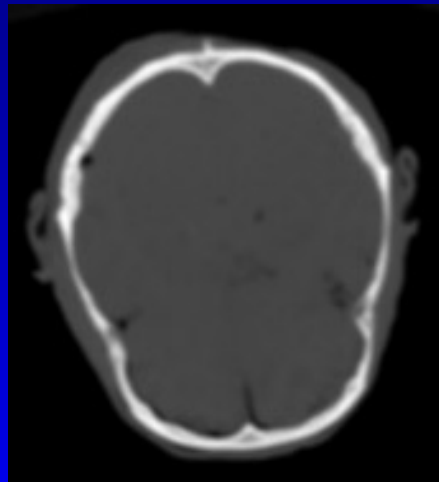


# f3d Tools: Local Operators

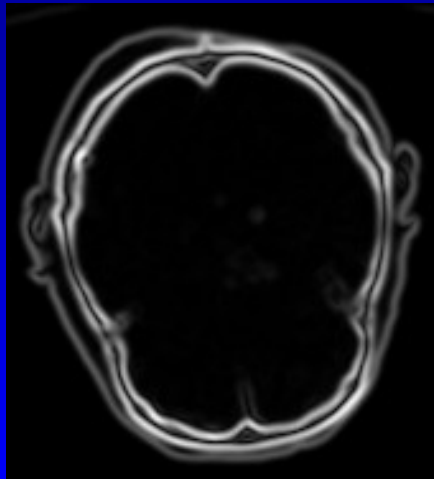
## Gradient Magnitude Estimation

- ◆ Magnitude of the Gabor filter applied to all 3 directions, edge detection
- ◆ Example :

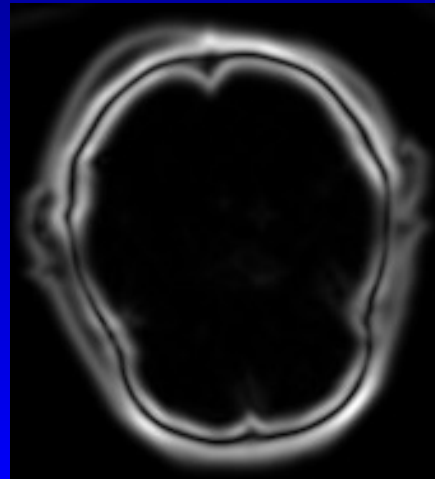
```
f3dgradmag -w 3 in.f3d > out.f3d
```



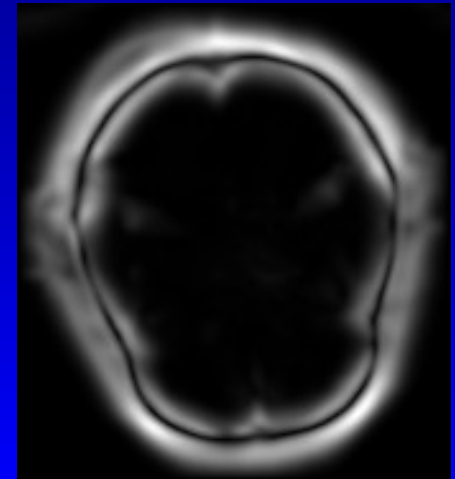
-w 1



-w 3



-w 5



# More about f3d

---

<http://bender.dam.fmph.uniba.sk/f3dtrac>