An Introduction to MINC

John G. Sled

MINC
Medical Image NetCDF

REPRIC Regroupement Provincial de
Recherche en Imagerie Cérébrale

What is MINC?

- A medical image file format based on NetCDF
- A core set tools and libraries for image processing
- A collection of applications for advanced (neuro) medical image analysis and data management

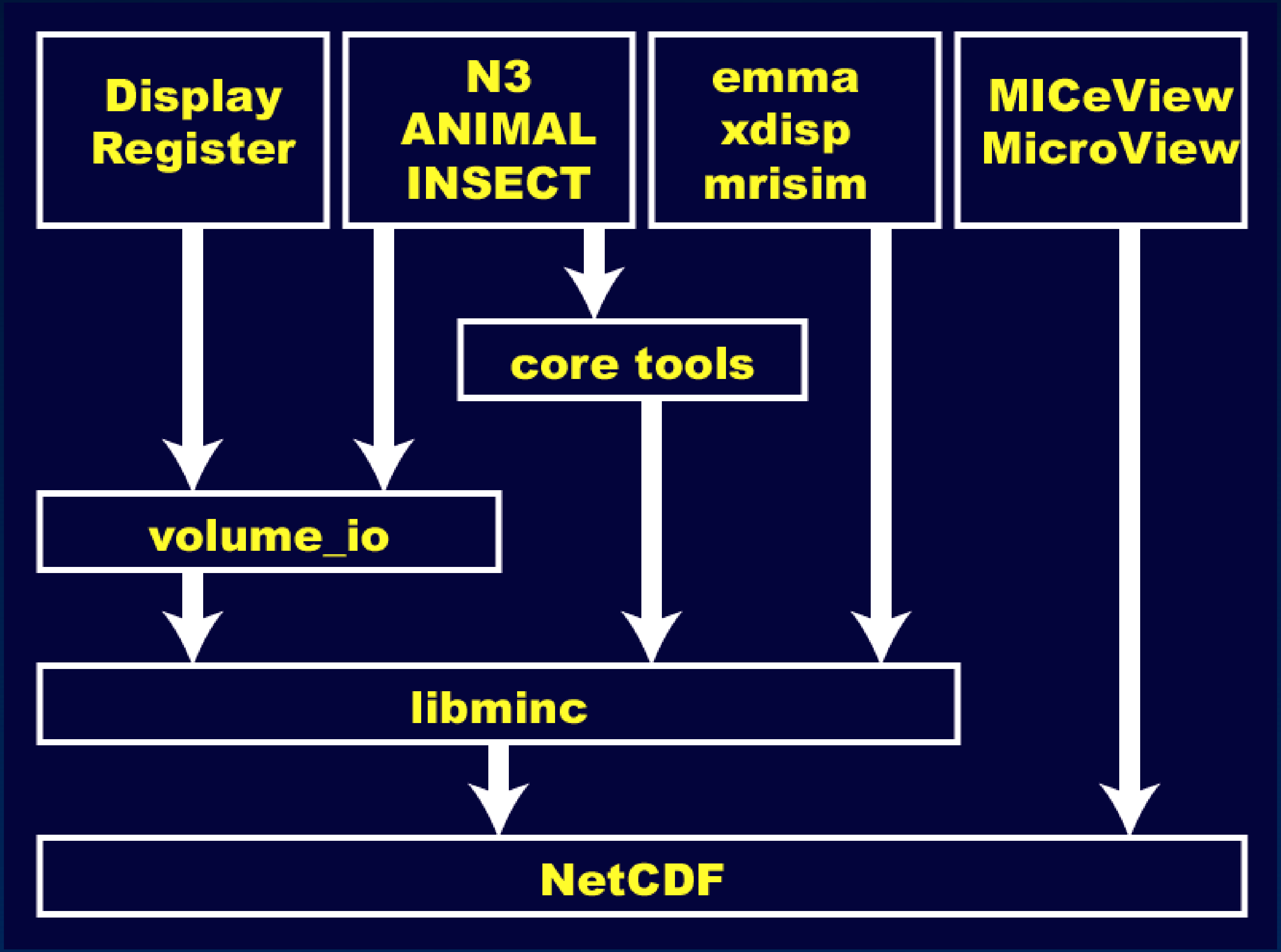
History of MINC

- Created by Peter Neelin in 1992 at the Montreal Neurological Institute to provide a powerful, modality-independent data format for the medical imaging research community
- Release 1.0 in 2002 – stewardship transferred to community of developers
- Release 1.2 – most recent to-date with improved build procedure and binary distributions for numerous platforms

Why MINC?

- Large scale and sophisticated medical image analysis inevitably means coping with a wide diversity of image data
- MINC provides the flexibility and generality for this task by capturing the data, its organization, and elements of its interpretation
- MINC aware tools can take over much of the tedious bookkeeping that goes with data processing

Structure of the MINC software



Display
Register

N3
ANIMAL
INSECT

emma
xdisp
mrisim

MICeView
MicroView

core tools

volume_io

libminc

NetCDF

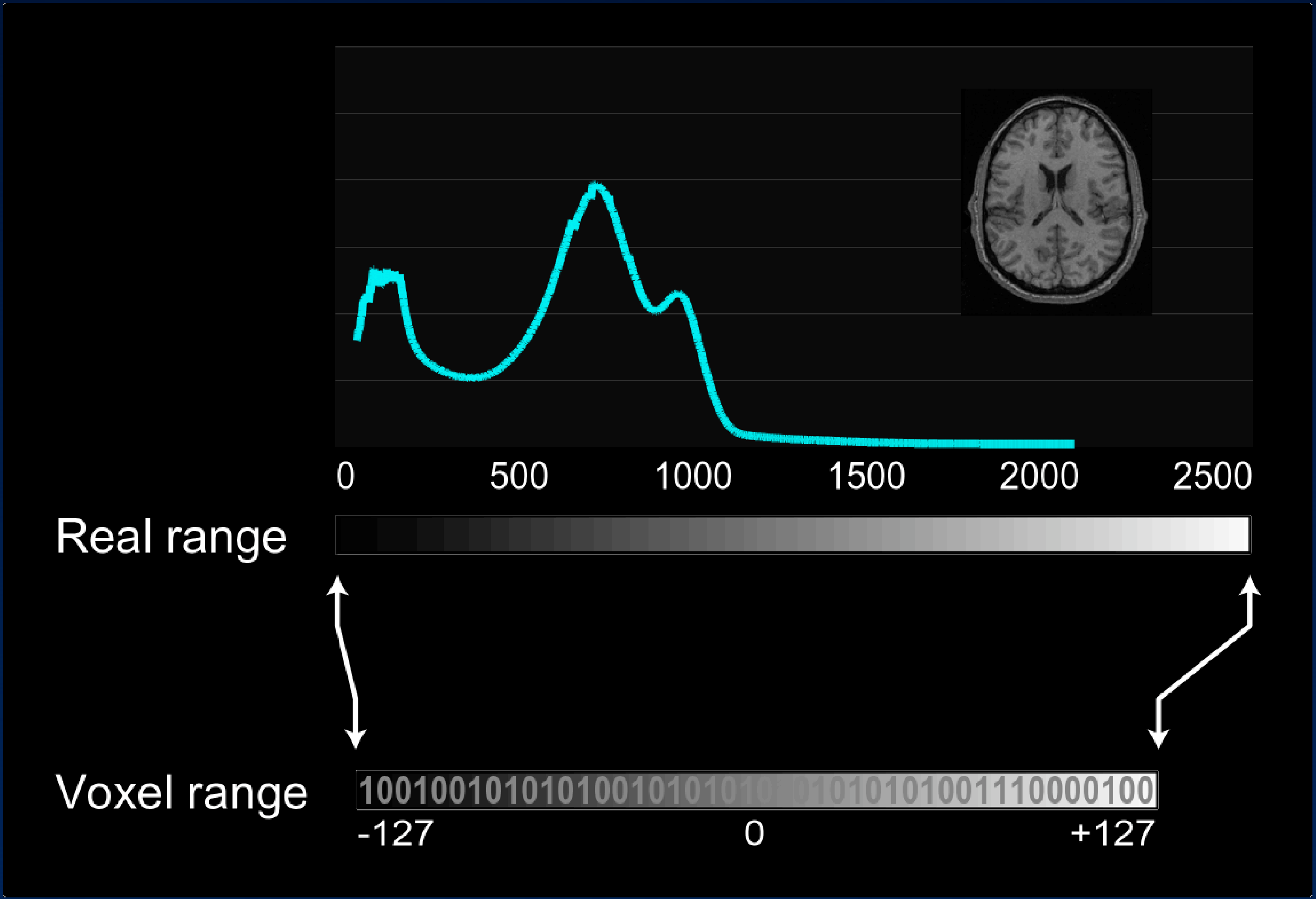
The file format

- **Platform independent format based on NetCDF standard**
- **Flexible N-dimensional modality independent representation of image data**
- **Arbitrary dimension ordering with standard and user defined dimension names**
- **Self documenting (human readable) extensible file header**
  - Includes standard terms for acquisition parameters, study information, patient information, units, processing history
  - Capacity to include entire DICOM header

Numeric data

- **The values in a MINC file are always interpreted as real numbers (real values) which are represented internally by an arbitrary floating point or fixed point type**
- **Floating point formats**
  - float (32 bit), double (64 bit)
- **Integer (fixed point) formats (signed or unsigned)**
  - Byte (8 bit), short (16 bit), integer (32 bit), long (32 bit)
- **MINC tools operate on the real values, independent of the internal numeric format**

Intensity mapping



Intensity mapping

- **Internally the translation between voxel and real values are determined by voxel-min, voxel-max, image-min, and image-max**
- **voxel-min and voxel-max determine the subrange of the internal data type that can be used, eg. 0 to 4095 for 12 bit MRI data**

$$(real\ \ value) = \frac{(image_{\mathrm{max}} - image_{\mathrm{min}})}{(voxel_{\mathrm{max}} - voxel_{\mathrm{min}})}(voxel\ \ value - voxel_{\mathrm{min}}) + image_{\mathrm{min}}$$

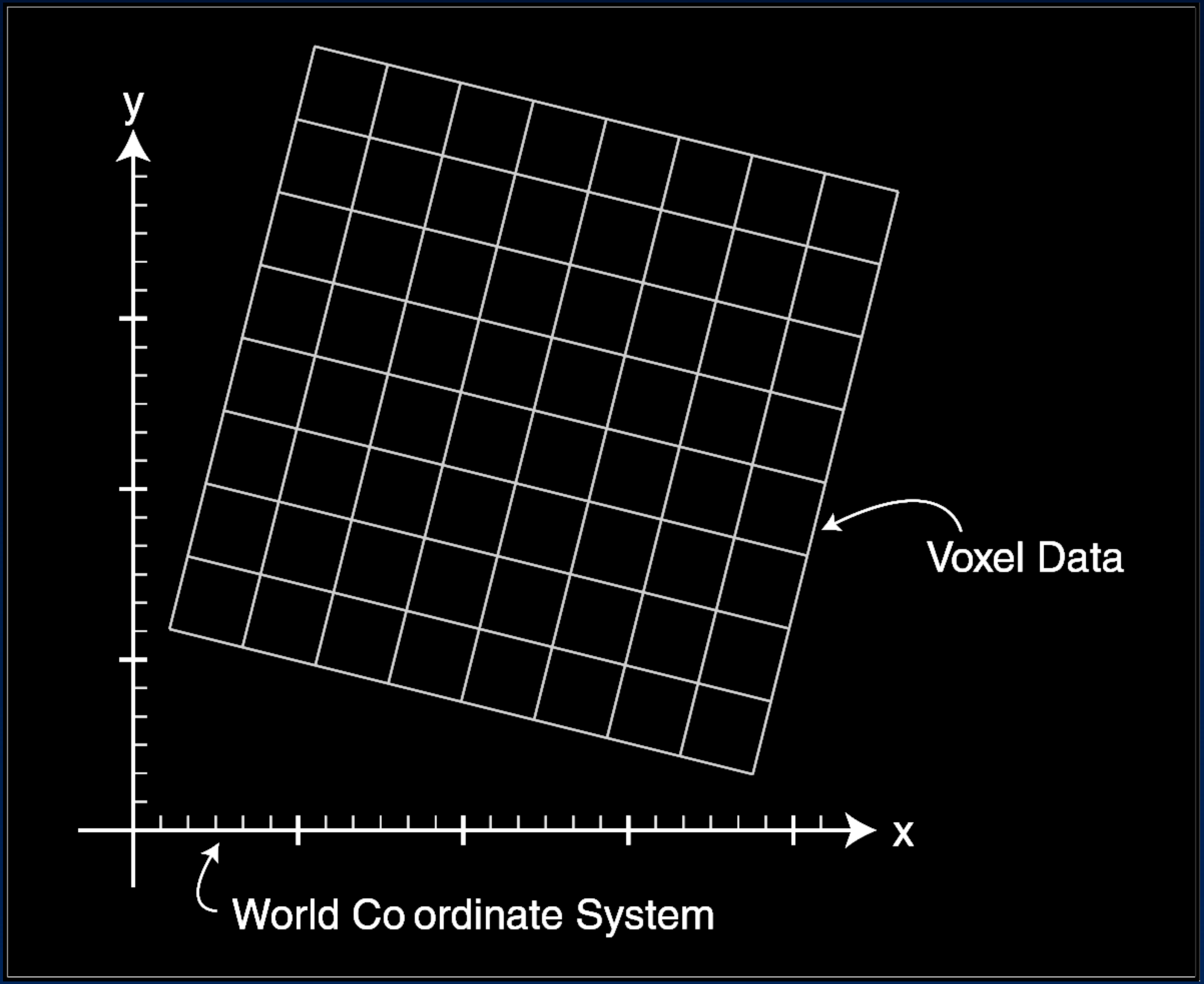
Intensity mapping

- **voxel-min and voxel-max apply to the volume as a whole**
- **image-min and image-max can be defined for the whole volume or independently for each slice**
- **when converting to and from MINC format, use real values**
  - the voxel values in each slice are often scaled independently
- **when the voxel and real range are the same, the mapping is 1 to 1, eg. label data sets**

Image dimensions

- **Image dimensions are named**
  - Standard dimension names include: xspace, yspace, zspace, time, and frequency
  - Standard dimension orders such as transverse, coronal, and sagittal have clear meanings
    - Eg. transverse order is zspace, yspace, xspace
- **Dimensions are documented in the header**
  - The interpretation is included, for example xspace is from patient left to patient right
  - Units where appropriate
  - The sampling is specified: uniform or non-uniform
  - For spatial dimensions, the orientation in world space is defined

World coordinate system



World coordinate system

- **Standard dimensions have a defined interpretation, eg. xspace increases from patient left to patient right**
- **The orientation of a given spatial dimension is specified by three direction cosines: the cosines of the angles between the direction of the dimension and the x, y, and z axes respectively**
  - eg. the default direction cosines for xspace are (1, 0, 0)

Voxel coordinate system

- Voxel coordinates refer to the index of a given voxel
- The first voxel along a dimension is 0, the next 1, etc.
- Voxel coordinates always refer to the voxel centre

World coordinates

- **The offset of the zero’th voxel along a given dimension in world units is the dimension start**
- **The separation between voxels for uniformly sampled dimensions in world units is the dimension step or separation**
    - Steps can be negative indicating that the order of the data in the file is counter to that described for the given dimension

Voxel to world translation

The relationship between voxel coordinates and world coordinates is

$$\begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} \alpha_{xi} & \alpha_{xj} & \alpha_{xk} \\ \alpha_{yi} & \alpha_{yj} & \alpha_{yk} \\ \alpha_{zi} & \alpha_{zj} & \alpha_{zk} \end{bmatrix} \begin{bmatrix} v_i \times step_i + start_i \\ v_j \times step_j + start_j \\ v_k \times step_k + start_k \end{bmatrix}$$

Coordinate spaces

- **Dimensions include a space type attribute that indicates which of several interpretations apply for the standard dimension names**
  - Native space: the spatial dimensions are defined with respect to the data acquisition hardware eg. zspace is through the bore of the MRI magnet
  - Talairach space: the spatial dimensions are defined with respect to the brain anatomy eg. yspace is posterior to anterior
- **World space aware tools maintain the spatial relationship between datasets in the same space**
- **A MINC file can be transformed from one space to another using a transform file (xfm) and mincresample**

Processing history

- MINC files include a history attribute
- Successive processing steps append the a copy of the command line invocation or a description of the processing to the history attribute

Writing new MINC tools

- **Libminc (C library)**
  - Provides full access to all aspects of MINC
  - Deals strictly with the on-disk management of the data
  - Requires a significant initial time investment to learn
- **Voxel_loop (C library)**
  - An easy way to write tools that iterate through all voxels in one or more volumes

Writing new MINC tools

- **Volume_io (C library)**
  - Provides are large subset of MINC functionality
  - Provides memory management in addition to reading and writing of MINC files
  - Includes a caching mechanism to allow out-of-core processing
  - Includes a large number of supporting functions for volume management
  - Requires a small initial time investment to learn

Getting more from MINC

- Users of MINC tools and applications

  Mailing list:

  minc-users@bic.mni.mcgill.ca

- Developers of MINC tools and the new MINC 2.0 format

  Mailing list:

  minc-development@bic.mni.mcgill.ca