The future of MINC

Robert D. Vincent
bert@bic.mni.mcgill.ca

MINC
Medical Image NetCDF

REPRIC Regroupement Provincial de Recherche en Imagerie Cérébrale

Outline

- Background
- New features and functions
- Compatibility
- Programming and other nerdy stuff

Background

- **MINC is built upon a general format**
  - netCDF: Network Common Data Form
  - MINC 1.0 is a specialization of this format
- **NetCDF Disadvantages:**
  - Not hierarchical
  - Limited data types
  - 32-bit file size
  - Simple, contiguous data storage

Motivation for change

- **File sizes are increasing**
  - Functional imaging, high-resolution anatomical imaging
  - May easily exceed 2 gigabytes
- **Efficient online viewing**
  - Storage of multiple pre-computed resolutions
- **Richer data types**
  - Vector, tensor, complex voxels
  - Labeled voxels

Hierarchical Data Format 5

- **NetCDF Disadvantages**
  - Not hierarchical
  - Limited data types
  - 32-bit file size
  - Simple, contiguous data storage model
- **HDF5 Advantages**
  - Fully hierarchical
  - Rich set of data types
  - 64-bit file size
  - Complex, powerful data storage model

MINC 2.0 Features

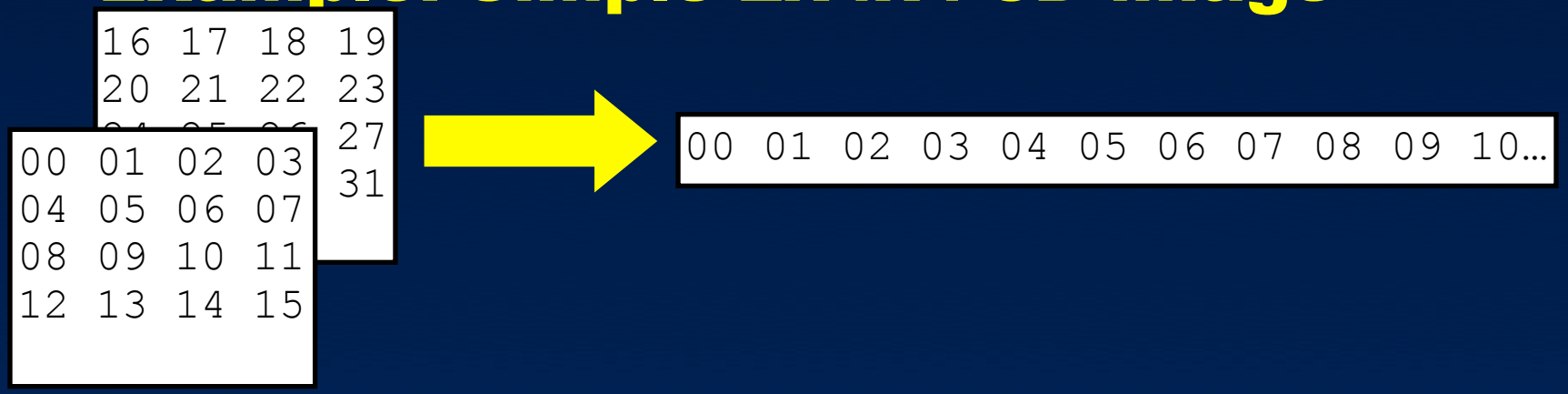
- **Block-structured storage**
- **Labeled volumes**
- **Internal compression**
- **Multi-resolution images**

Block-structured storage

- Stores image in a series of N-dimensional blocks
- May improve performance in some cases
- Enables internal compression

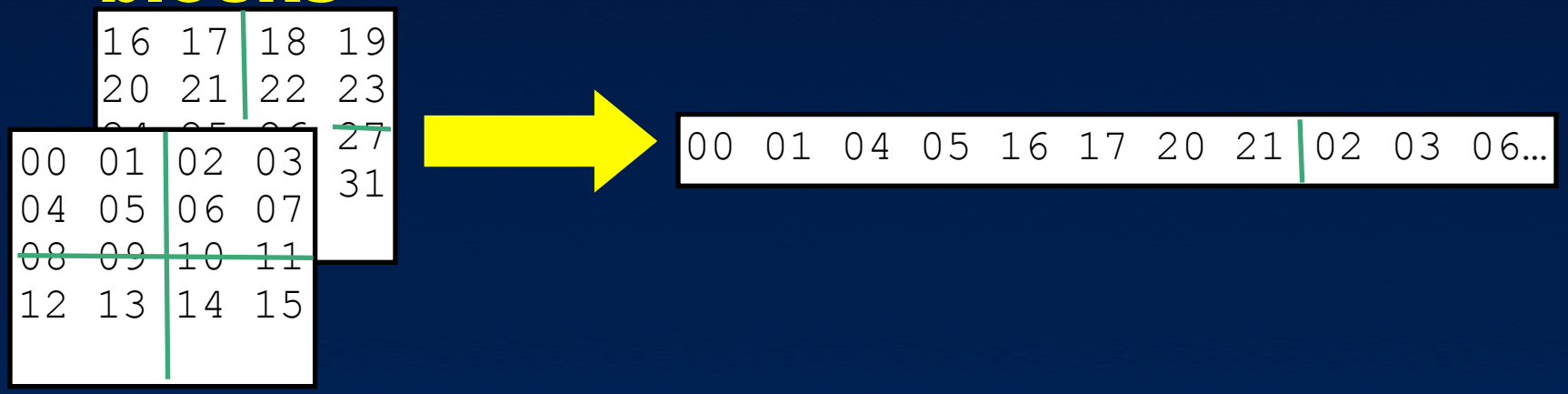
Block-structured storage

- **In an ordinary MINC volume, data is stored in a simple linear arrangement**
- **Example: Simple 2x4x4 3D image**



Block-structured storage

- In block structured file, nearby voxels are grouped together
- Example: 2x4x4 image with 2x2x2 blocks



Labeled volumes

- **Associates text labels with voxel values**
  - Appropriate for classifier output, for example
  - 0 = “Background”, 1 = “White matter”, 2 = “Grey matter”, and so forth
- **Furthers the MINC goal of “self-describing” data files**

Internal compression

- **MINC 1.0 External compression**
  - External program must decompress an entire file
  - File must be fully decompressed to read a single voxel
  - Uses temporary disk space, bandwidth, and time
- **MINC 2.0 Internal compression**
  - Data is compressed on a block-by-block basis
  - Invisible to user, compression is internal to library
  - Nearly random access
  - Parts of the file may be left uncompressed

Multi-resolution images

- **Applications**
  - Visualization over networks
  - Multi-resolution algorithms
- **Design – Multiple image objects**
  - Normal data stored at full resolution
  - Additional images stored at 1/2, 1/4, 1/8, etc.
    - Initially will use a simple voxel averaging algorithm
    - May be enhanced/replaced by us or third parties
  - Adds at most 1/7 of full-resolution size for 3D images

Compatibility

- **MINC programs still support NetCDF**
  - Input – Tools automatically “do the right thing”
  - Output – User must explicitly specify “-2” flag for HDF5
- **MINC 1.0 “libminc” still supported**
  - Minor enhancements, new flags
  - Some new functions
- **Very few changes to MINC 1.0 code**
  - It might actually keep working!

MINC 2.0 Programming

- **Philosophy**
  - Formalize MINC 1.0 programming idioms
  - Provide a complete set of functions for manipulating image data
  - Backward compatibility
- **Design**
  - http://www.bic.mni.mcgill.ca/software/minc2/api_doc-2003-10-31
  - Initial implementation in “C”, with additional language bindings to follow…

MINC 2.0 Programming

- **New concepts**
  - volume type – image objects
    - Voxel data
    - Attributes
    - Dimensions
  - dimension type – individual coordinate axes
    - Orientation
    - Spacing

MINC 2.0 Programming

- **MINC 1.0**
  - Functions tend to be very general
  - Many of functions serve multiple purposes
  - Programmer must understand both netCDF and MINC concepts
- **MINC 2.0**
  - Functions tend to be more specialized
  - Completely hides details of HDF5/netCDF
  - MINC 2.0 must define many more functions (129 and counting vs. less than 50)

MINC 2.0 Programming

- **Example – get one “real” valued voxel**
- **MINC 1.0**

```
icv = miicv_create();
miicv_setint(icv, MI_ICV_DO_RANGE, TRUE);
…
miicv_attach(icv, fd, var_id);
miicv_get(icv, coords, lengths, &value);
```

- **MINC 2.0**

```
miget_real_value(volume,coords,3,&value);
```

MINC 2.0 Programming

- **Example – direction cosines**
- **MINC 1.0**

```
miattget(fd, varid, Midirection_cosines,
         NC_DOUBLE, 3, cosines, &length);
```

- **MINC 2.0**

```
miget_direction_cosines(dimension,cosines);
```

The future of the future

- **Language support**
  - Perl, Python, Java, C++
- **New data classes**
  - Geometric objects
  - Transform information
- **Additional compression methods**
  - bzip2
  - “Lossy” image compression

Credits

- **Leila Baghdadi**
  - Design, documentation, and programming
- **John Sled**
  - Design, documentation, and sanity checking
- **Many others who have offered advice and opinions!**
- **netCDF created by UCAR**
- **HDF5 created by NCSA**
- **Financial support from NIfTI**