NEMA, Suite 1752
1300 North 17th Street
Rosslyn, VA 22209
Ph: (703) 841-3285
http://dicom.nema.org

DICOM
(for MRI images)

David Atkinson

D.Atkinson@ucl.ac.uk

References

- David Clunie's web site and links
  http://www.dclunie.com/
- Reference data and presentations
  http://dicom.nema.org/
  ftp://medical.nema.org/medical/Dicom/Multiframe/

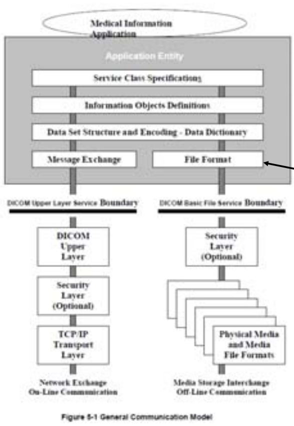
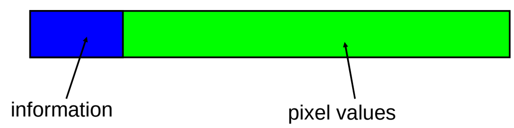


Figure 5-1 General Communication Model

DICOM is a vast set of standards
Concentrate here on file format for MRI

[NEMA documentation]

DICOM File Structure



PatientName
ImageOrientationPatient
ImagePositionPatient
PixelSpacing
…
PrivateFields

DICOM old vs enhanced

- Old style
  - one file per slice – huge numbers of files.
  - Important parameters e.g. diffusion weighting hidden in non-standard Private Fields.
- Enhanced DICOM
  - multi-frame,
  - better information about 3D and time,
  - many more parameters in Public Fields (was 2, now 94)
  - raw data archive possible.

Enhanced MR SOP Class attribute types

- Separate gradient and RF echo train lengths
- Out-of-plane phase encoding steps
- Flow compensation
- Spectrally selective excitation & suppression
- Blood signal nulling
- Tagging
- Diffusion values and direction
- Spatial saturation slabs
- Velocity encoding
- Chemical shift imaging (metabolite maps)

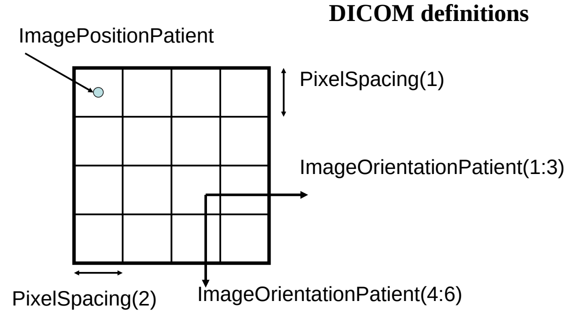
[medical.nema.org]

Geometry Information in DICOM

- DICOM uses a right handed LPH coordinate system.
- Relates to patient, not scanner.
- Origin is arbitrary (not isocentre) but fixed.
- Nifti uses RAH (also right handed)
- Analyze uses LAH (left handed!!)

- DICOM provides public fields that relate a 2D image to 3D patient space.

DICOM definitions

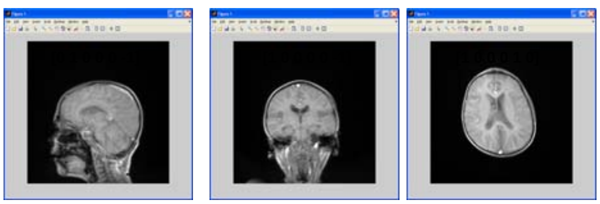


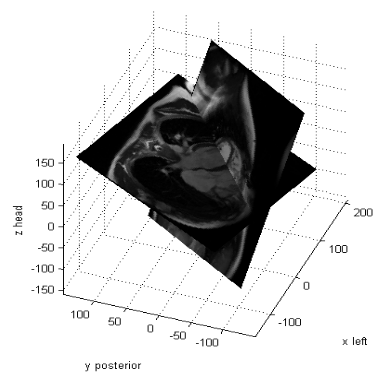
PixelSpacing and ImagePositionPatient are in mm
ImageOrientationPatient are two unit vectors (direction cosines)

Height and Width give number of rows and columns.

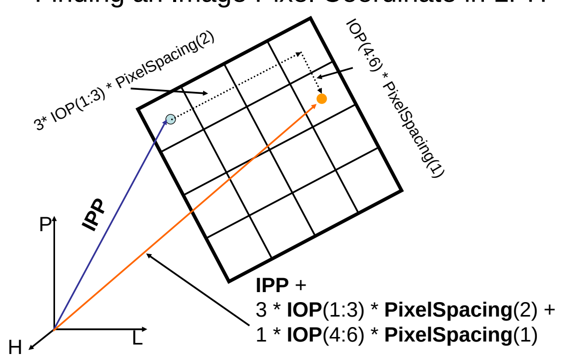
Quiz.

- What are the ImageOrientationPatient vectors for these images (radiological presentation)?





Finding an Image Pixel Coordinate in LPH



IPP = ImagePositionPatient
IOP = ImageOrientationPatient

Stacking Slices

Problem: Multiple 2D slices, each as a separate DICOM file – how do you assemble into a 3D matrix?

- Do not rely on file naming.
- Find the through-slice direction using the vector product **n** = **IOP**(1:3) **x IOP**(4:6)
- For each file, compute the component of **IPP** in this through-slice direction (**n.IPP**) and sort.

Enhanced DICOM

Organizational Features

- Multi-frame pixel data
- Shared and per-frame functional groups
  - Each functional group contains attributes that likely vary as a group, e.g. Pixel Measures, Plane Orientation, Velocity Encoding, etc.
  - Compact & makes explicit what doesn't change
- Dimensions
  - *a priori* hints as to how the frames are organized
  - Specify intended order of traversal, such as space, then time (e.g., for cardiac cine loops)
- Stacks
  - Groups of spatially-related slices, repeatable
- Temporal positions

[medical.nema.org]

Organization of Data

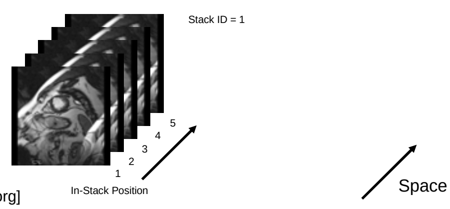
- Goal is to reduce the work that the receiving application has to do to "figure out"
  - How the data is organized
  - Why it is organized that way
- Without preventing use of the data in unanticipated ways
  - E.g. 3D on a dataset not intended as a volume
- Two levels
  - The detailed shared & per-frame attributes
  - The overall dimensions, stacks and temporal positions

[medical.nema.org]

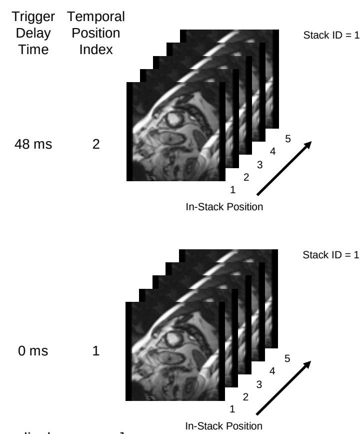
Dimensions

Start with a dimension of space.

A set of contiguous slices through the heart.

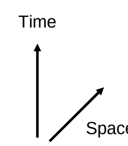


[medical.nema.org]

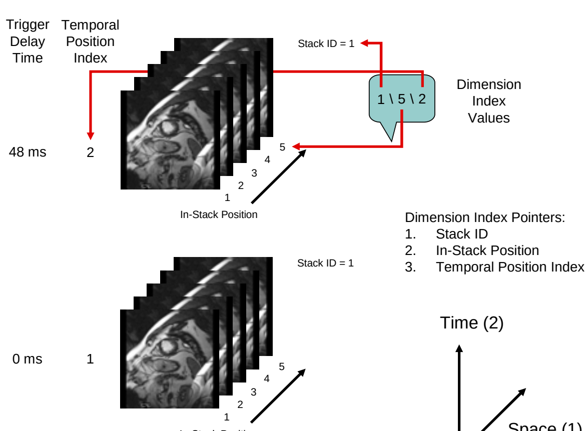


Add dimension of time (delay time from R-wave).

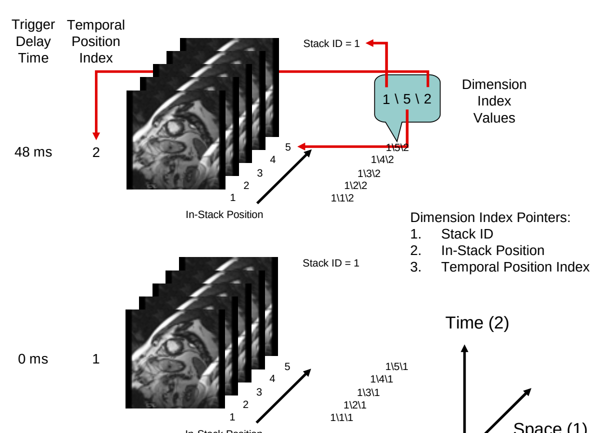
Sets of contiguous slices throughout cardiac cycle.



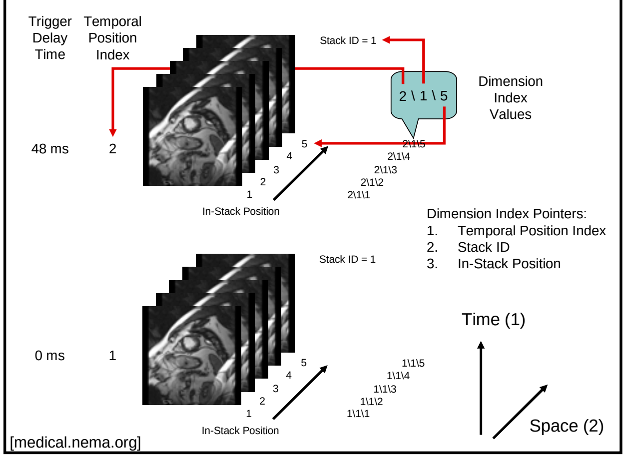
[medical.nema.org]



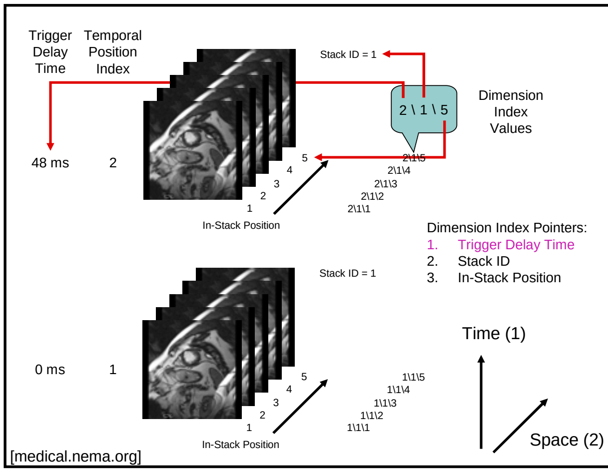
[medical.nema.org]



[medical.nema.org]



[medical.nema.org]



[medical.nema.org]

Dimension features

- Description of dimensions separate from their indices
  - Dimensions are described once
  - Indices within dimensions are encoded per-frame
- May be multiple sets of dimensions in one object
  - E.g., Set 1: space then time, Set 2: time then space
- Receiving application only needs to follow the index values
  - Does NOT need to select or sort by attribute value
  - Dimensions can be entire functional groups
  - Dimensions can be private attributes or functional groups

[medical.nema.org]

Dimension applications

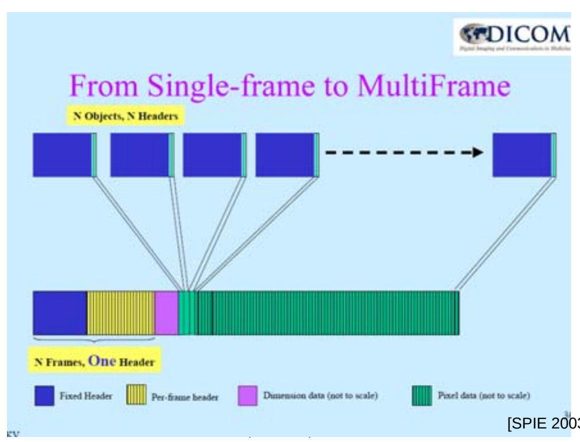
- Selection of sort order for simple viewing
- Partitioning of frames for hanging
- Selection of frames that constitute a
  - volume in space
  - temporal sequence
  - contrast administration phase
  - physiological parameter, e.g. diffusion b value

[medical.nema.org]

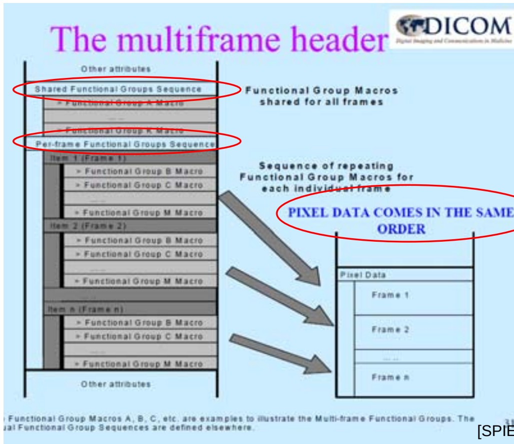
Diffusion

Appears to be structure for recording either the diffusion gradient direction, or the full b-matrix in the patient coordinate system.

(0018,9087)	Diffusion b-value
(0018,9089)	Diffusion Gradient Orientation
(0018,9601)	Diffusion b-matrix Sequence
(0018,9602)	Diffusion b-value XX
(0018,9603)	Diffusion b-value XY
(0018,9604)	Diffusion b-value XZ
(0018,9605)	Diffusion b-value YY
(0018,9606)	Diffusion b-value YZ
(0018,9607)	Diffusion b-value ZZ



[SPIE 2003]



[SPIE 2003]

Geometry Fields in Multi-frame DICOM

ImageOrientationPatient

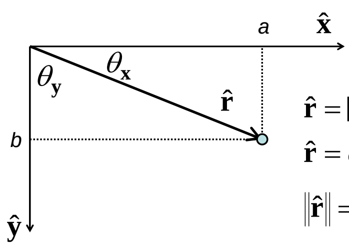
SharedFunctionalGroupsSequence.Item_1.PlaneOrientationSequence.Item_1.ImageOrientationPatient

ImagePositionPatient

PerFrameFunctionalGroupsSequence.Item_168.PlanePositionSequence.Item_1.ImagePositionPatient

(ignore specific Item numbers here)

Direction Cosines

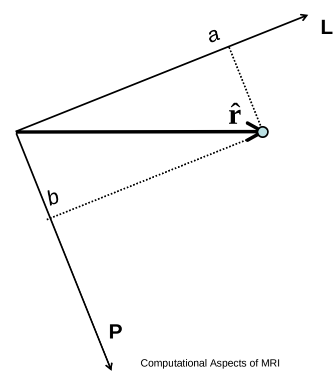


$$\hat{\mathbf{r}} = [a\ b]$$
$$\hat{\mathbf{r}} = a\hat{\mathbf{x}} + b\hat{\mathbf{y}}$$
$$\|\hat{\mathbf{r}}\| = \sqrt{a^2 + b^2} = 1$$
$$\hat{\mathbf{r}} = \cos(\theta_{\mathbf{x}})\hat{\mathbf{x}} + \cos(\theta_{\mathbf{y}})\hat{\mathbf{y}}$$

The components of a unit vector (*a* & *b* here) are the cosines of the angles the vector makes with the basis directions

Computational Aspects of MRI

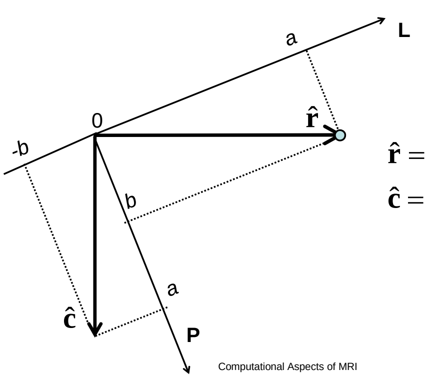
Direction Cosines



Rotate
Re-label

Computational Aspects of MRI

Direction Cosines and DICOM

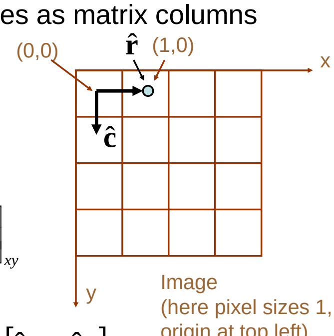


$$\hat{\mathbf{r}} = [a\ b]$$
$$\hat{\mathbf{c}} = [-b\ a]$$

Computational Aspects of MRI

Image to Patient Transform: direction cosines as matrix columns

$$\hat{\mathbf{r}} = [a\ b]$$
$$\hat{\mathbf{c}} = [-b\ a]$$
$$\hat{\mathbf{r}}: \begin{bmatrix} a \\ b \end{bmatrix}_{LP} = \begin{bmatrix} a & \cdot \\ b & \cdot \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}_{xy}$$
$$\hat{\mathbf{c}}: \begin{bmatrix} -b \\ a \end{bmatrix}_{LP} = \begin{bmatrix} \cdot & -b \\ \cdot & a \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}_{xy}$$
$$\begin{bmatrix} a & -b \\ b & a \end{bmatrix} = [\hat{\mathbf{r}}:\ \ \hat{\mathbf{c}}:]$$



Computational Aspects of MRI

Direction Cosines in 3D with homogeneous coordinates

Rotation matrix composed from row, col. and slice direction cosines as columns

$$\hat{\mathbf{s}} = \hat{\mathbf{r}} \times \hat{\mathbf{c}}$$

Image coordinate

Patient system coordinate

$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = \begin{bmatrix} rdcx & cdcx & sdcx & 0 \\ rdcy & cdcy & sdcy & 0 \\ rdcz & cdcz & sdcz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{im} \\ y_{im} \\ z_{im} \\ 1 \end{bmatrix}$$

[See http://www.electromagnetics.biz/DirectionCosines.htm]

Putting it all together

- ImageOrientationPatient
  - rotation
- ImagePositionPatient
  - translation
- PixelSpacing
  - scaling

Composing the overall transform from Image to Patient

$$\mathbf{M} = \mathbf{T}_{\mathrm{IPP}}\ \mathbf{R}\ \mathbf{S}\ \mathbf{T}_{\mathbf{0}}$$

Shift image to make top left voxel centre at (0,0,0)

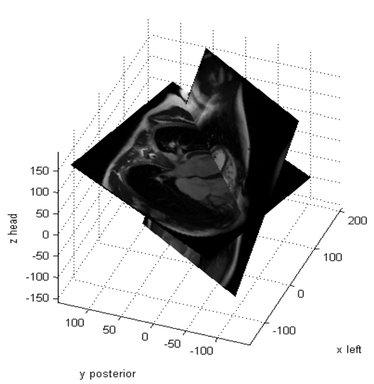
Scale using PixelSpacing

Rotate into Patient coordinate system using Direction Cosines from ImageOrientationPatient

Translate to put top left pixel at ImagePositonPatient

Applying the transform to multiple coordinates "at once"

$$\begin{bmatrix} l_1 & l_2 & \cdots \\ p_1 & p_2 & \cdots \\ h_1 & h_2 & \cdots \\ 1 & 1 & \end{bmatrix} = \mathbf{M} \begin{bmatrix} x_1 & x_2 & \cdots \\ y_1 & y_2 & \cdots \\ z_1 & z_2 & \cdots \\ 1 & 1 & \end{bmatrix}$$



MATLAB Default Image Coordinates

Row numbers increase going DOWN

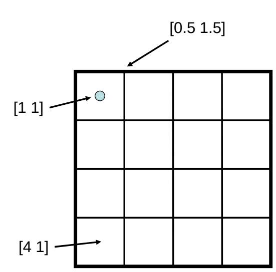
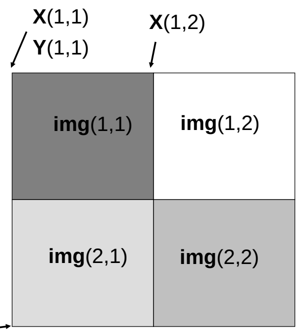


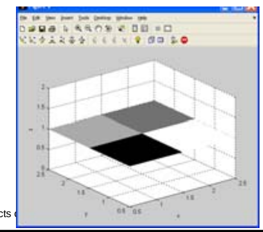
Image coordinates in [row column] order

Displaying a 2D image in 3D using **surf(X,Y,Z,img)**

- 2D matrices **X**,**Y** and **Z** contain patient coordinates of the vertices of the patches.
- 2D matrix **img** contains patch “colours”.
- Sizes of **X**,**Y** and **Z** are one greater than **img** in each dimension




```
>>
>> img = [0.1 1 ; 0.7  0.5]
img =
   0.100000000000000   1.000000000000000
   0.700000000000000   0.500000000000000
>> [X,Y,Z] = meshgrid([0:2]+0.5, [0:2]+0.5, 1)
X =
   0.500000000000000   1.500000000000000   2.500000000000000
   0.500000000000000   1.500000000000000   2.500000000000000
   0.500000000000000   1.500000000000000   2.500000000000000
Y =
   0.500000000000000   0.500000000000000   0.500000000000000
   1.500000000000000   1.500000000000000   1.500000000000000
   2.500000000000000   2.500000000000000   2.500000000000000
Z =
     1     1     1
     1     1     1
     1     1     1
>> surf(X,Y,Z,img,'EdgeColor','None')
>> colormap gray
>> xlabel('x'), ylabel('y'), zlabel('z')
>>
```



Allows interactive spinning of 3D plots. (Camera toolbar)