DICOM

Digital Imaging and Communications in Medicine

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/

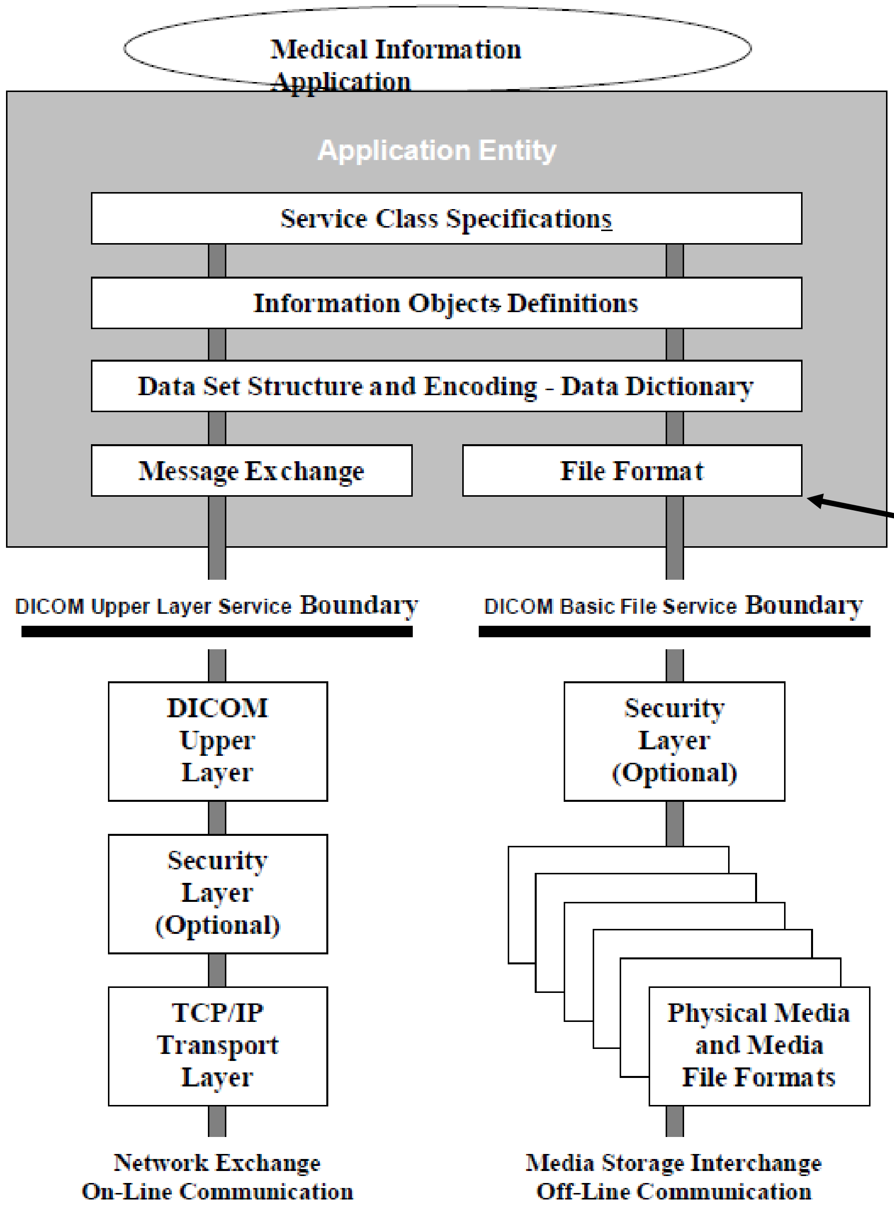
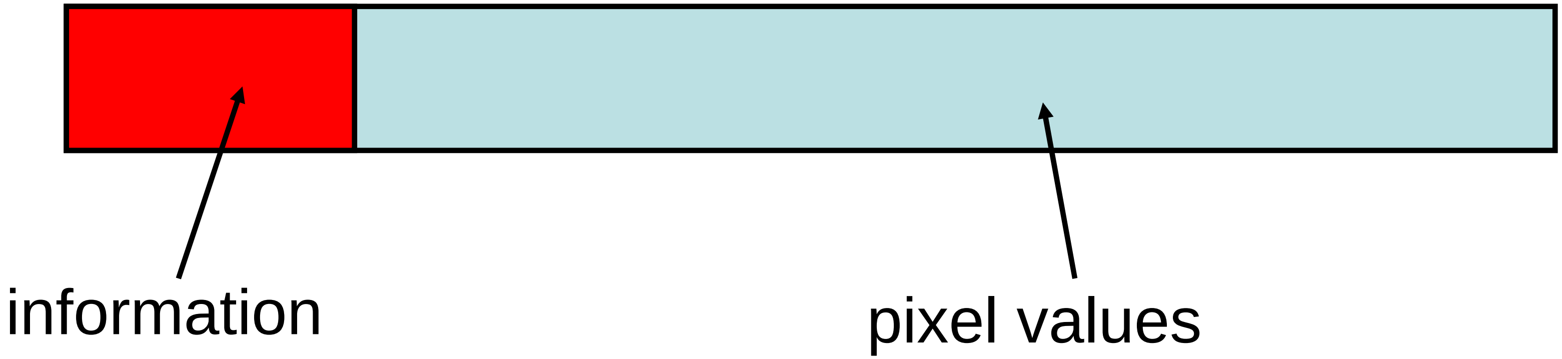


Figure 5-1 General Communication Model

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

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)

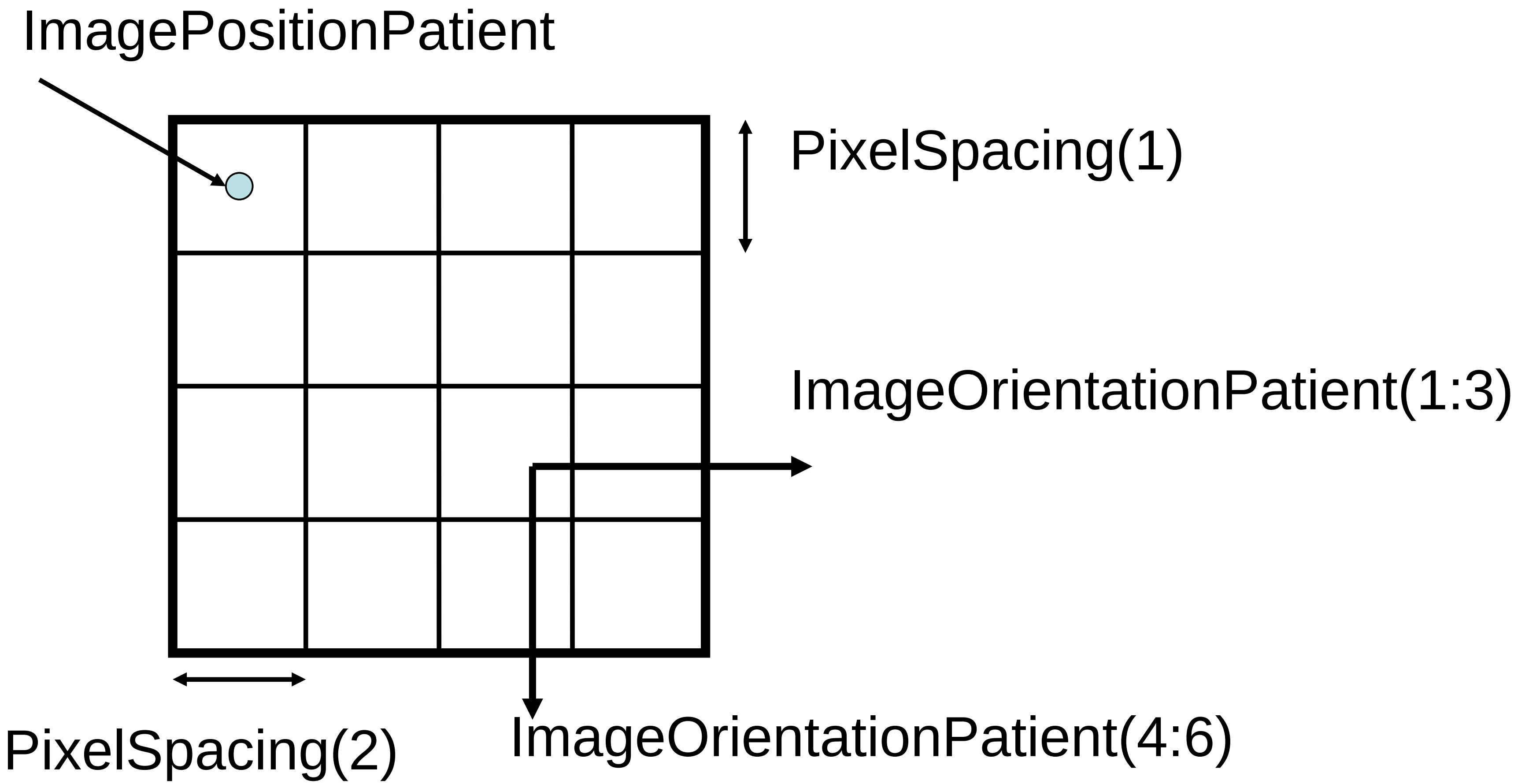
[medical.nema.org]

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

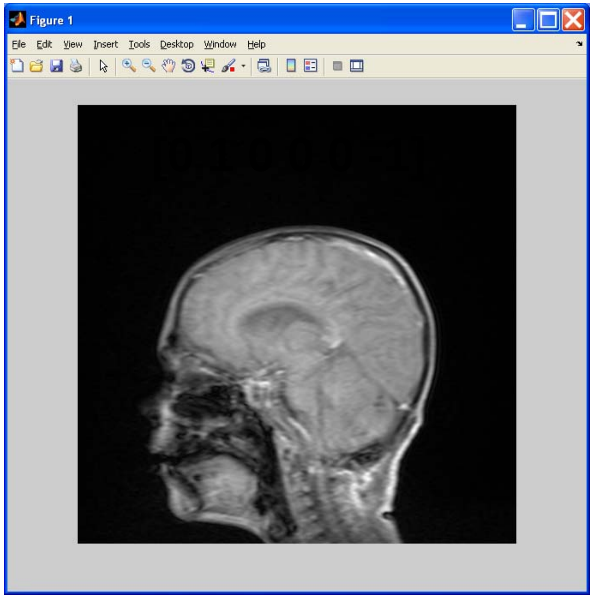


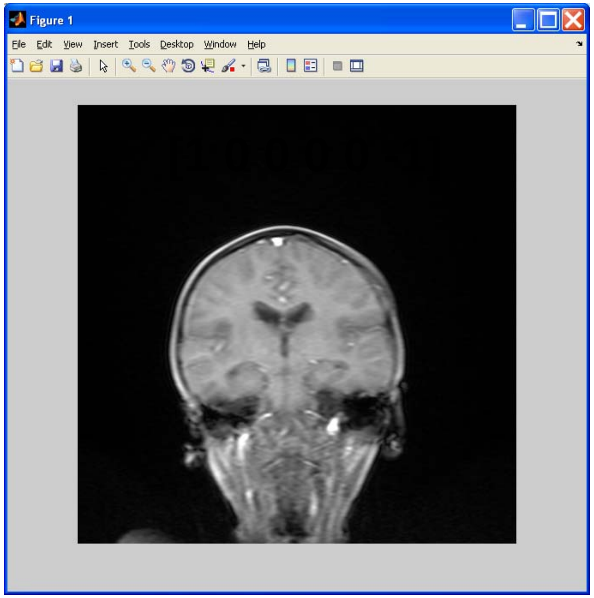
PixelSpacing and ImagePositionPatient are in mm
ImageOrientationPatient are two unit vectors (direction cosines)

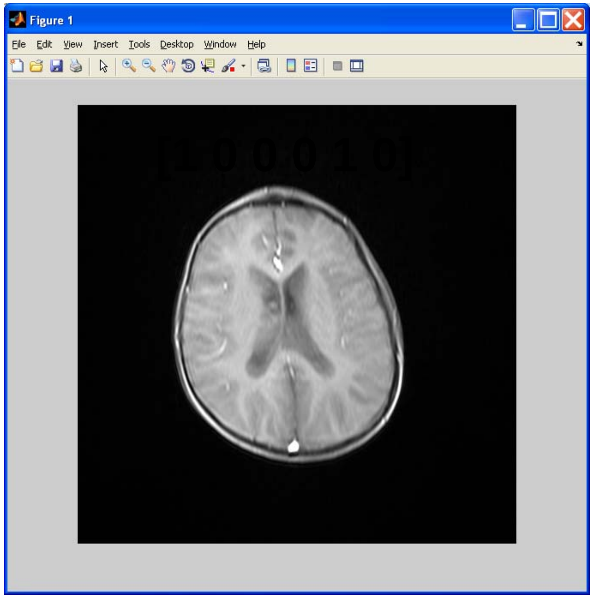
Height and Width give number of rows and columns.

Quiz.

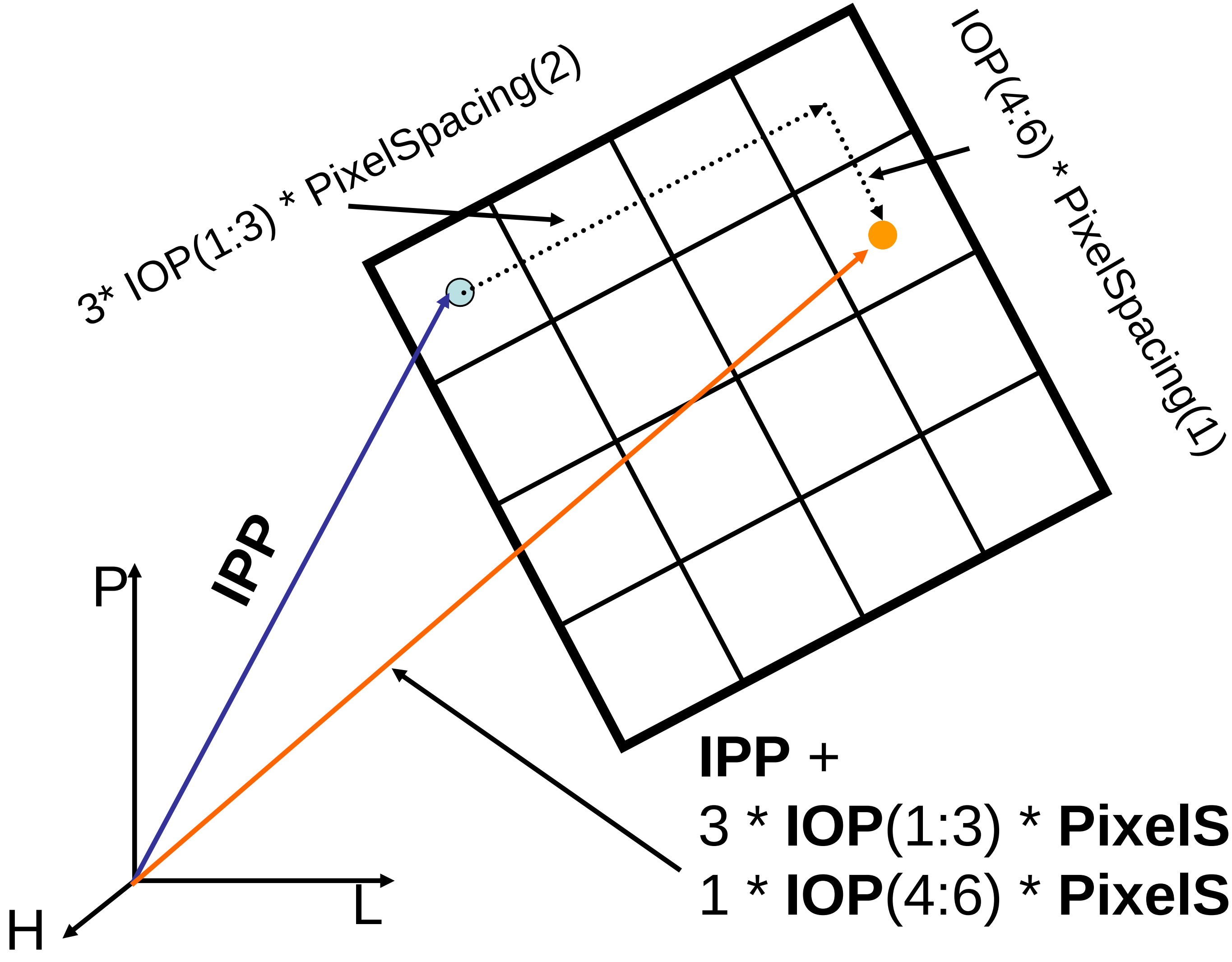
- What is the ImageOrientationPatient vector for these images?





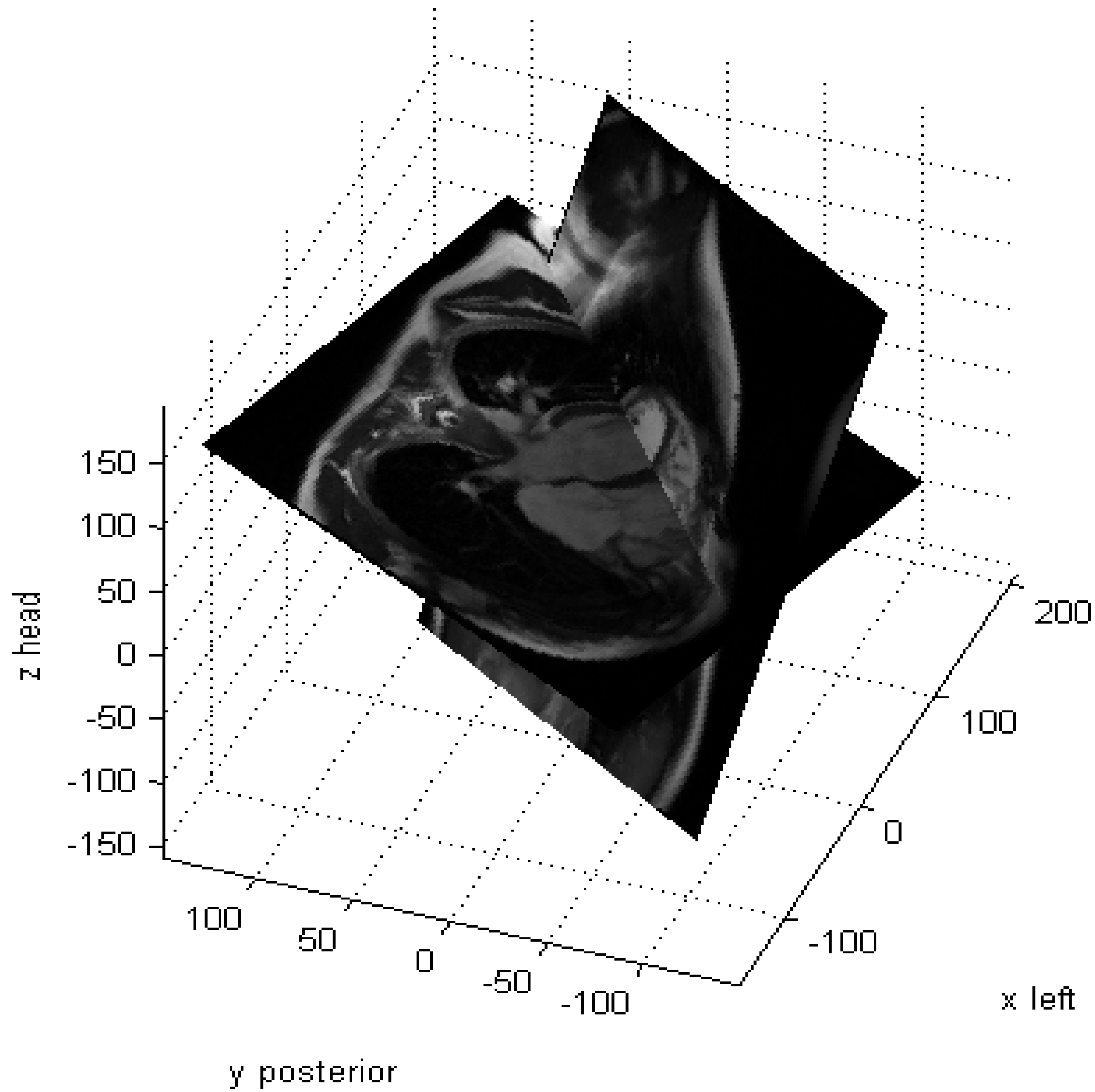


Finding an Image Pixel Coordinate in LPH



IPP +
3 * **IOP**(1:3) * **PixelSpacing**(2) +
1 * **IOP**(4:6) * **PixelSpacing**(1)

IPP = ImagePositionPatient
IOP = ImageOrientationPatient


150
100
50
0
-50
-100
-150
z head
100
50
0
-50
-100
y posterior
200
100
0
-100
x left

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.

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

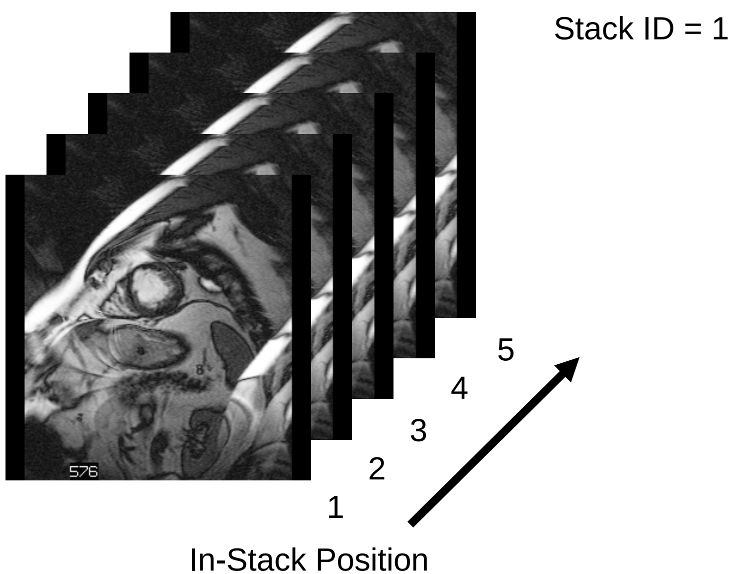
Organization of Data

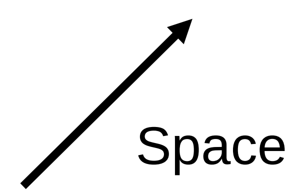
- Goal is to reduce the work that the receiving application has to do to “figure out”
  - How the data is organized
  - Why it is organized that way
- Without preventing use of the data in unanticipated ways
  - E.g. 3D on a dataset not intended as a volume
- Two levels
  - The detailed shared & per-frame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

Dimensions

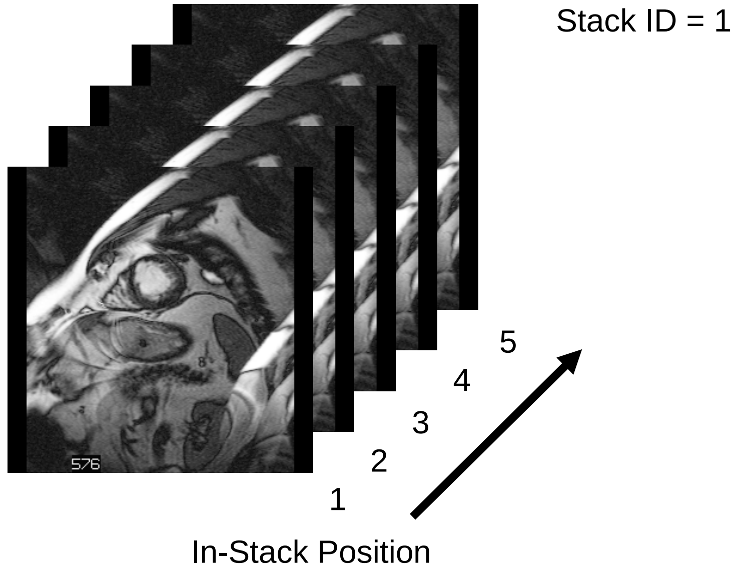
Start with a dimension of space.

A set of contiguous slices through the heart.



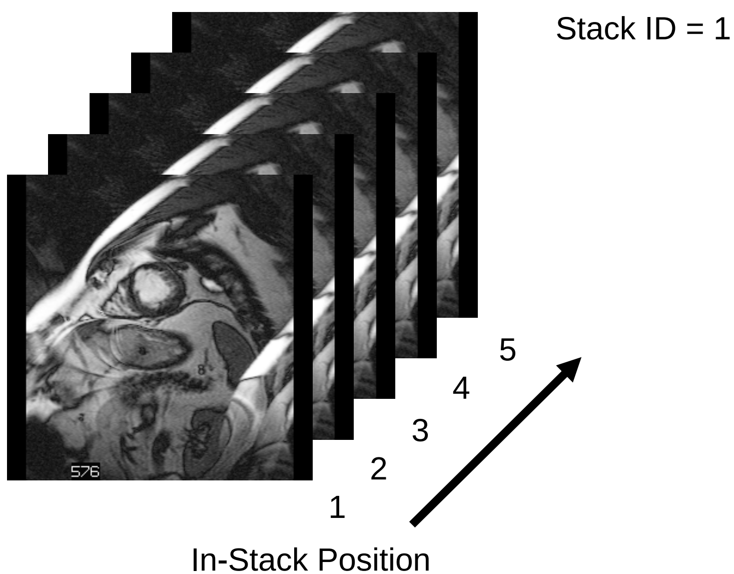


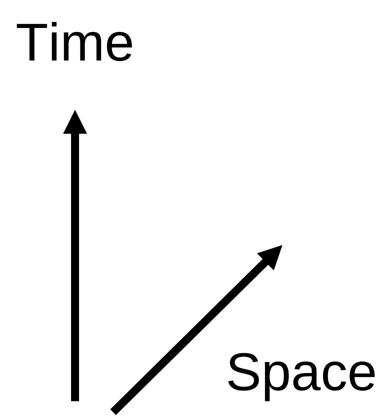
Trigger Delay Time	Temporal Position Index
48 ms	2
0 ms	1

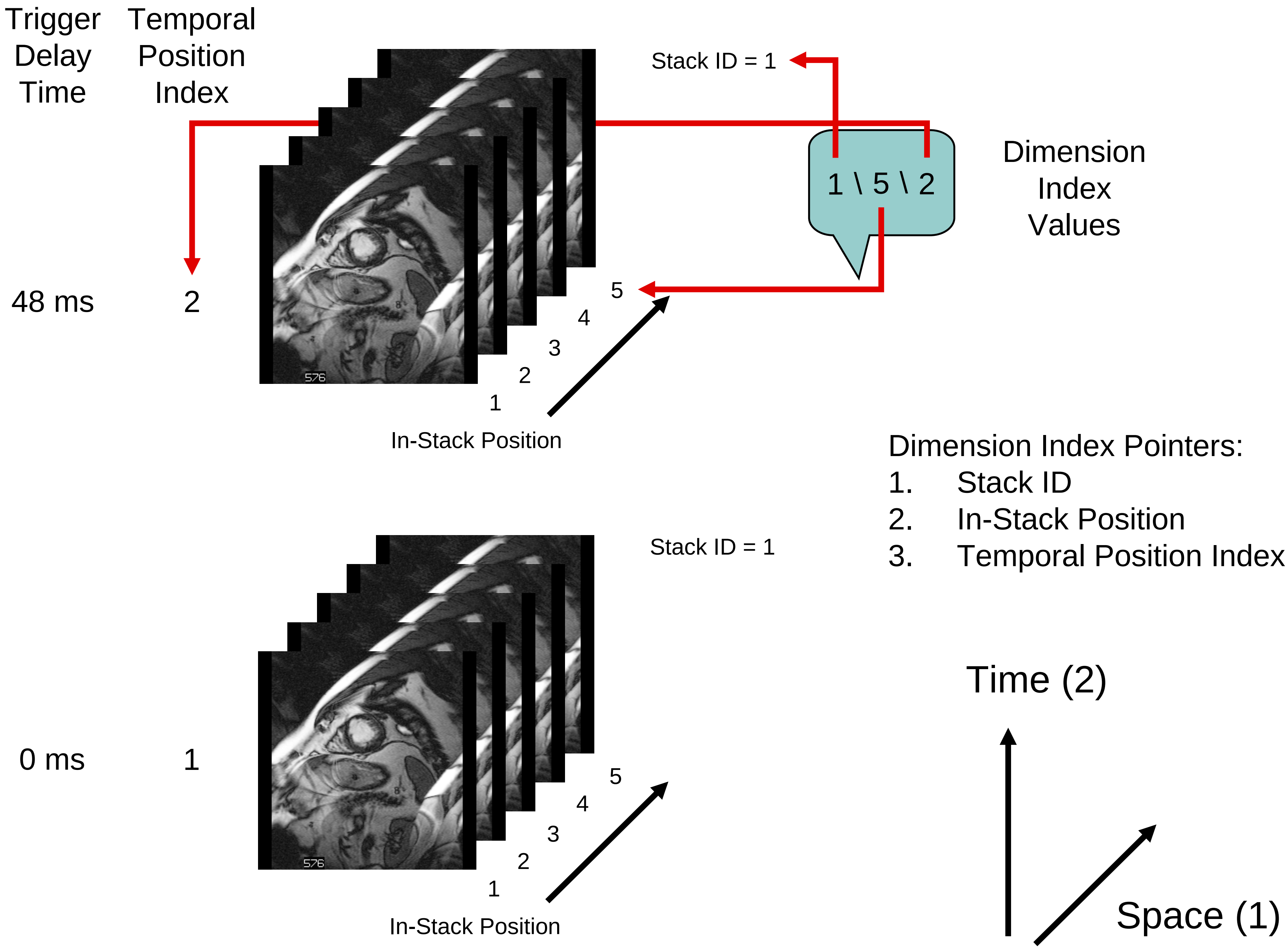


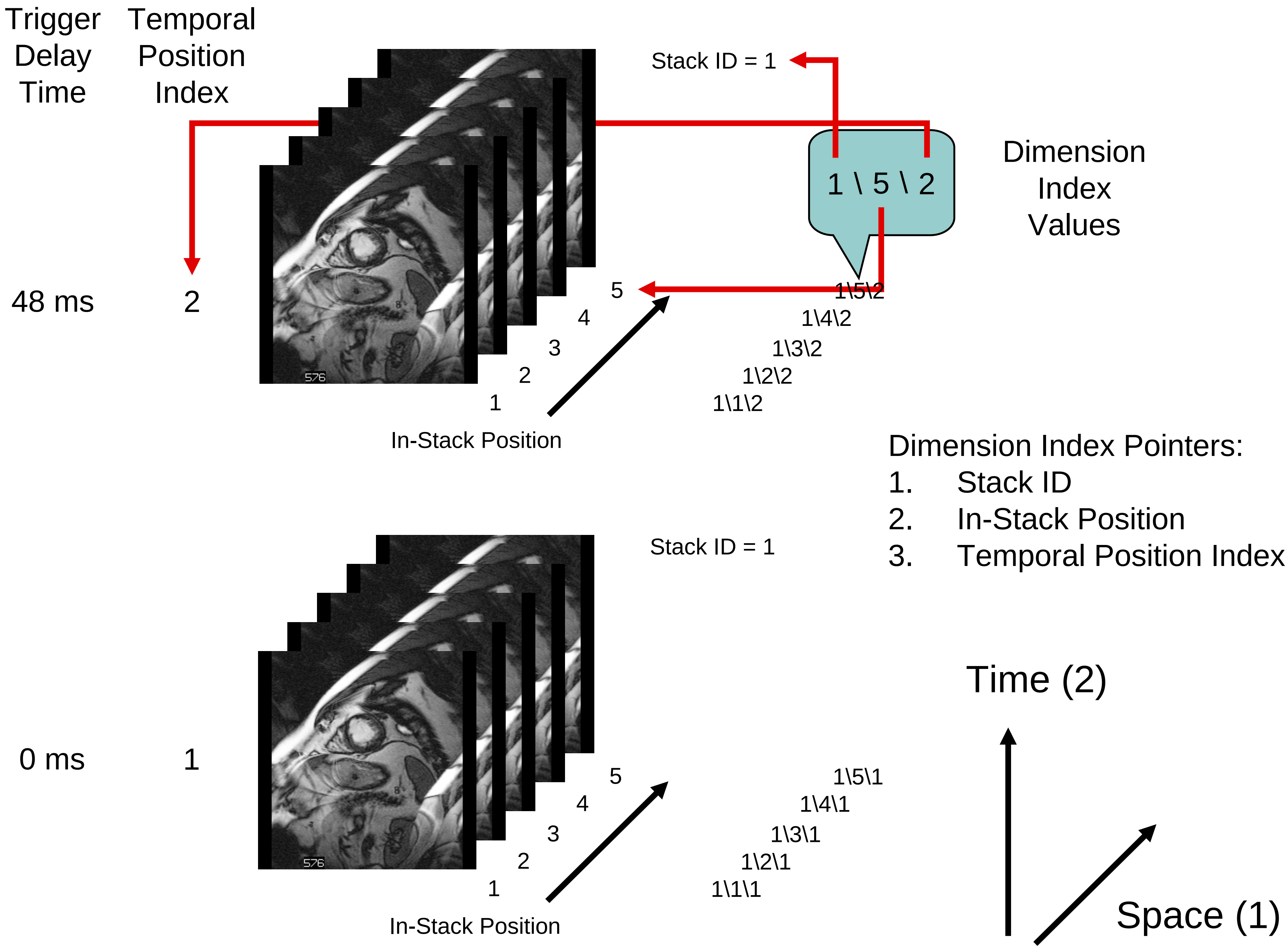
Add dimension of time (delay time from R-wave).

Sets of contiguous slices throughout cardiac cycle.

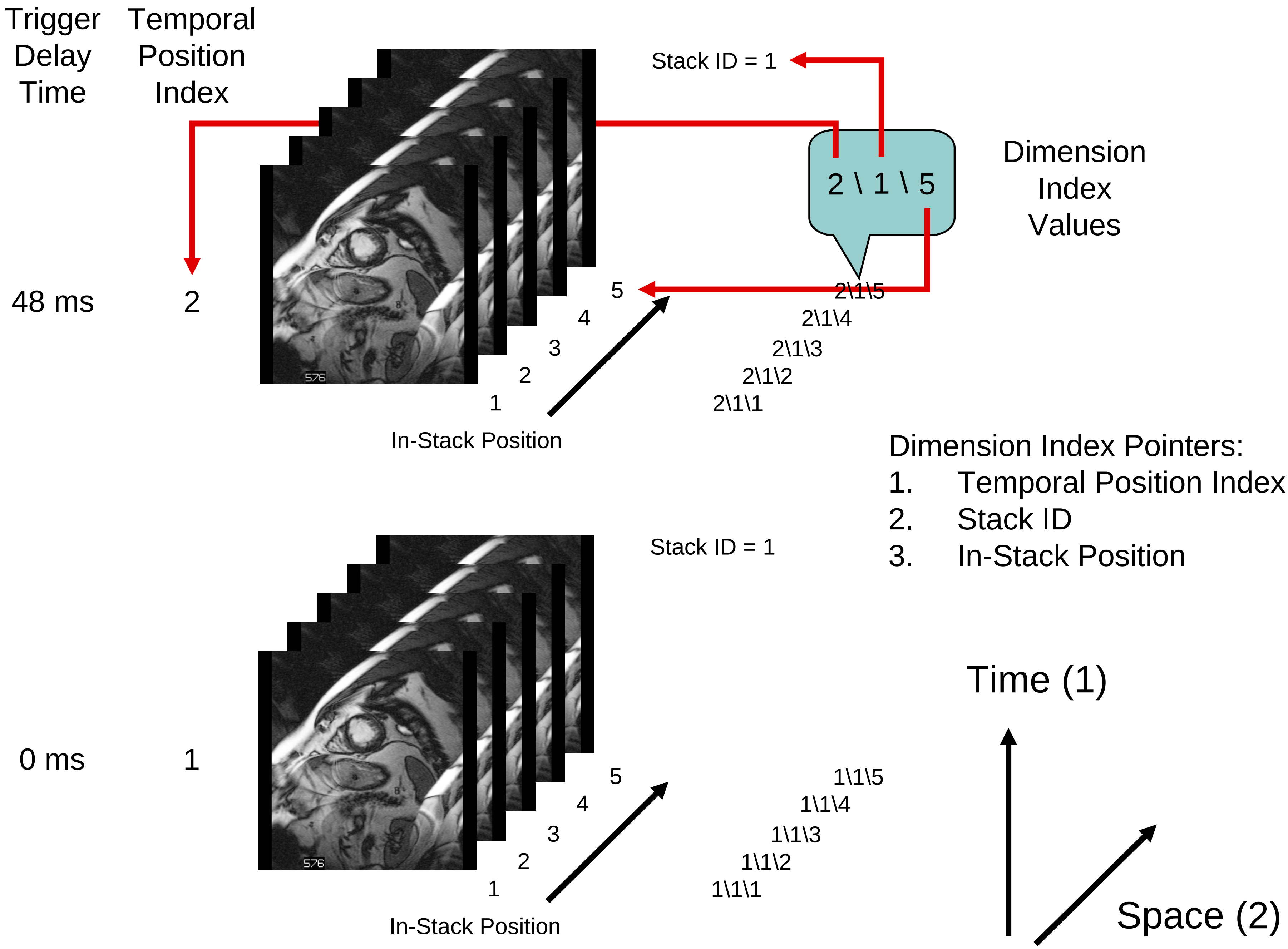




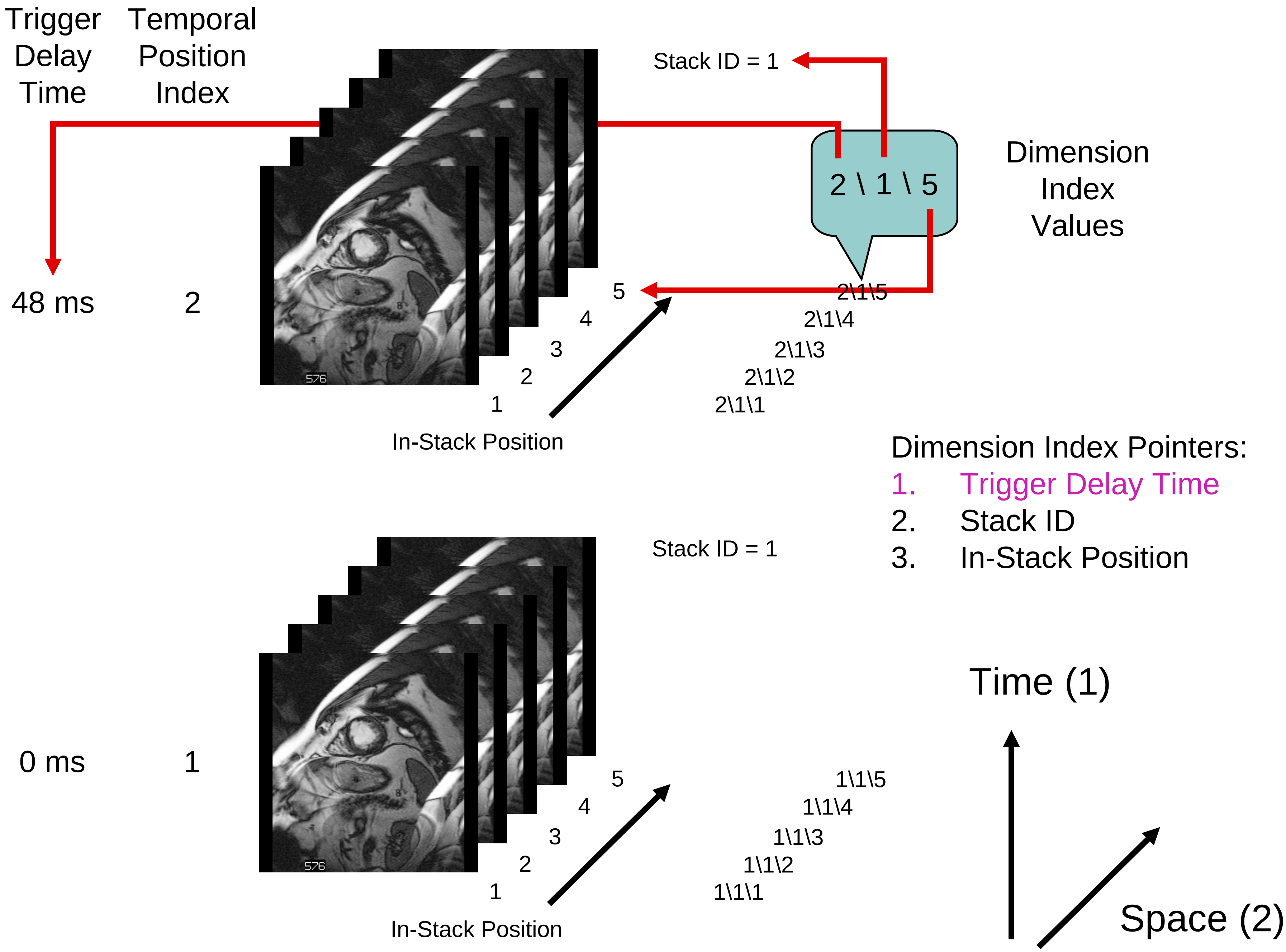

Trigger Delay Time
Temporal Position Index
Stack ID = 1
1 \ 5 \ 2
Dimension Index Values
48 ms
2
5
4
3
2
1
In-Stack Position
Dimension Index Pointers:
1. Stack ID
2. In-Stack Position
3. Temporal Position Index
Stack ID = 1
0 ms
1
5
4
3
2
1
In-Stack Position
Time (2)
Space (1)



Trigger Delay Time
Temporal Position Index
Stack ID = 1
1 \ 5 \ 2
Dimension Index Values
48 ms
2
5
4
3
2
1
1\5\2
1\4\2
1\3\2
1\2\2
1\1\2
In-Stack Position
Dimension Index Pointers:
1. Stack ID
2. In-Stack Position
3. Temporal Position Index
Stack ID = 1
0 ms
1
5
4
3
2
1
1\5\1
1\4\1
1\3\1
1\2\1
1\1\1
In-Stack Position
Time (2)
Space (1)



Trigger Delay Time
Temporal Position Index
Stack ID = 1
2 \ 1 \ 5
Dimension Index Values
48 ms
2
5
4
3
2
1
2\1\5
2\1\4
2\1\3
2\1\2
2\1\1
In-Stack Position
Dimension Index Pointers:
1. Temporal Position Index
2. Stack ID
3. In-Stack Position
Stack ID = 1
0 ms
1
5
4
3
2
1
1\1\5
1\1\4
1\1\3
1\1\2
1\1\1
In-Stack Position
Time (1)
Space (2)


Trigger Delay Time
Temporal Position Index
48 ms
2
Stack ID = 1
2 \ 1 \ 5
Dimension Index Values
5
4
3
2
1
2\1\5
2\1\4
2\1\3
2\1\2
2\1\1
In-Stack Position
Dimension Index Pointers:
1. Trigger Delay Time
2. Stack ID
3. In-Stack Position
0 ms
1
Stack ID = 1
5
4
3
2
1
1\1\5
1\1\4
1\1\3
1\1\2
1\1\1
In-Stack Position
Time (1)
Space (2)

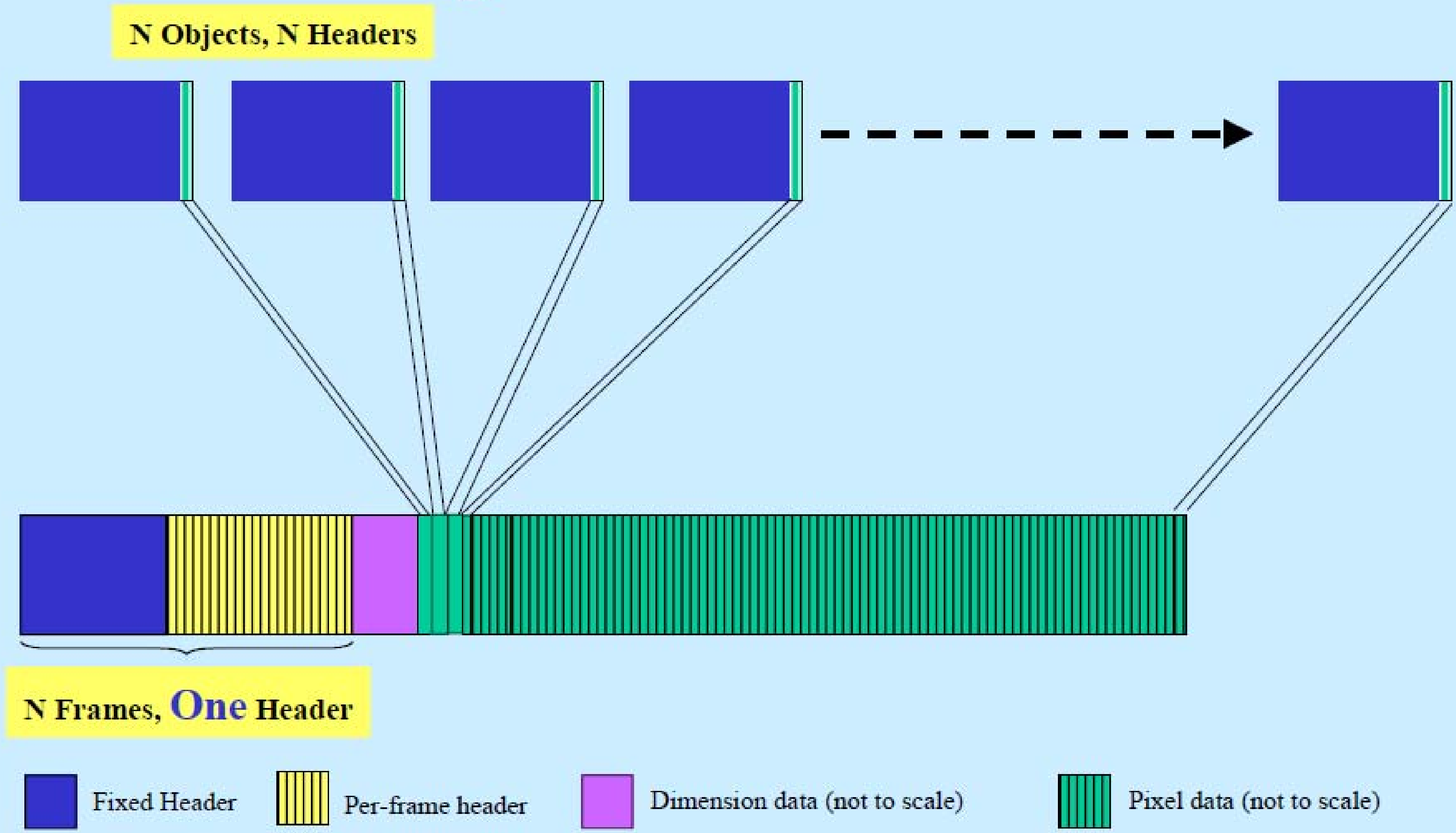
Dimension features

- 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

Dimension applications

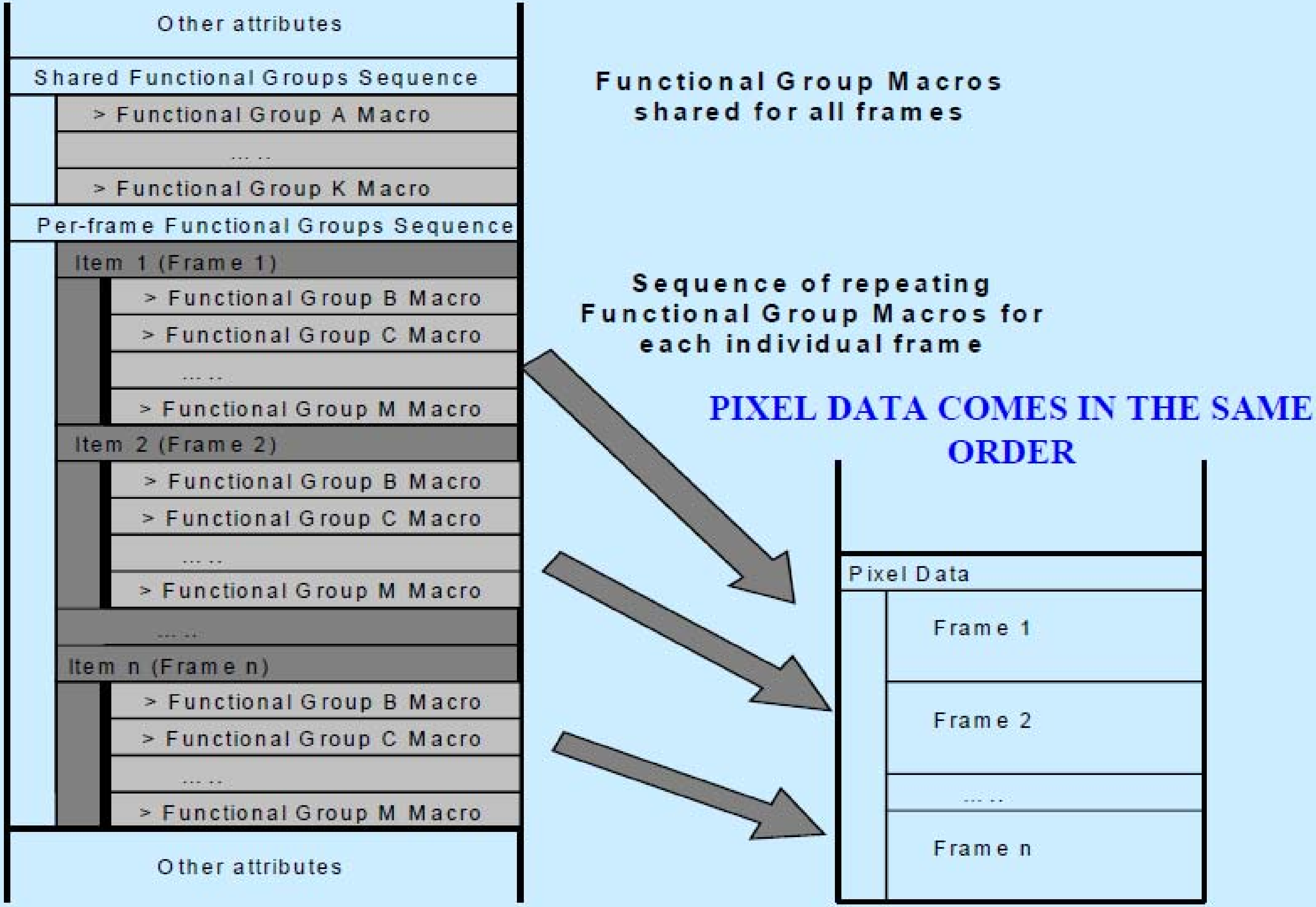
- Selection of sort order for simple viewing
- Partitioning of frames for hanging
- Selection of frames that constitute a
  - volume in space
  - temporal sequence
  - contrast administration phase
  - physiological parameter, e.g. diffusion b value

From Single-frame to MultiFrame



[SPIE 2003]

The multiframe header



e Functional Group Macros A, B, C, etc. are examples to illustrate the Multi-frame Functional Groups. The
ual Functional Group Sequences are defined elsewhere.

[SPIE 2003]

Geometry Fields in Multi-frame DICOM

ImageOrientationPatient

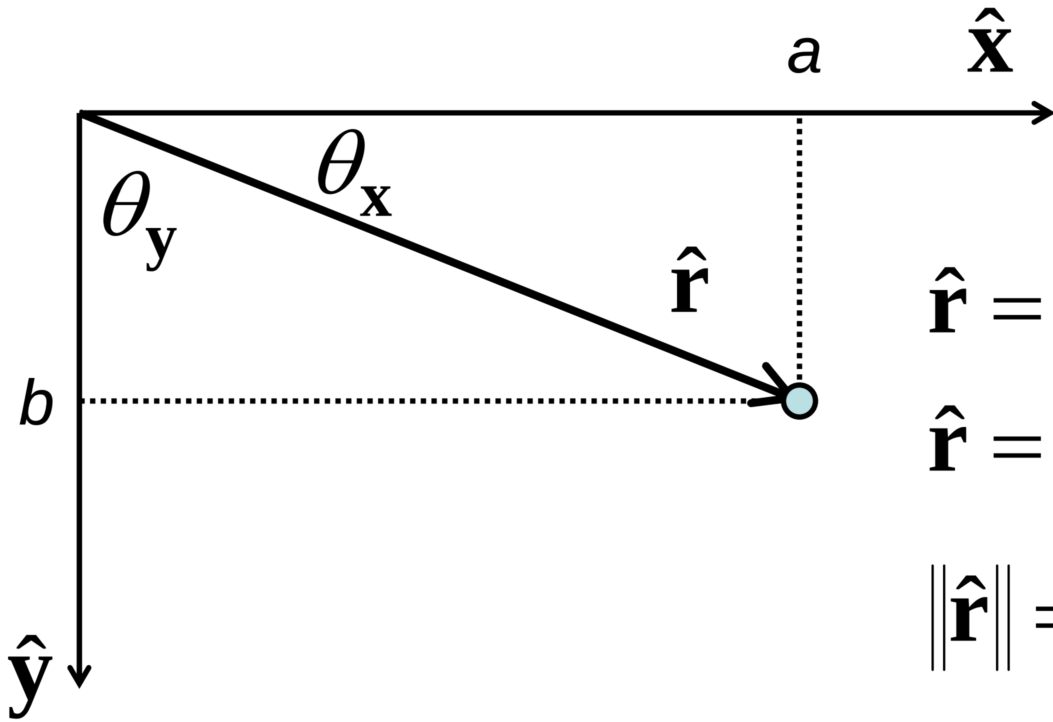
SharedFunctionalGroupsSequence.Item_1.PlaneOrientationSequence.Item_1.ImageOrientationPatient

ImagePositionPatient

PerFrameFunctionalGroupsSequence.Item_168.PlanePositionSequence.Item_1.ImagePositionPatient

(ignore specific Item numbers here)

Direction Cosines



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

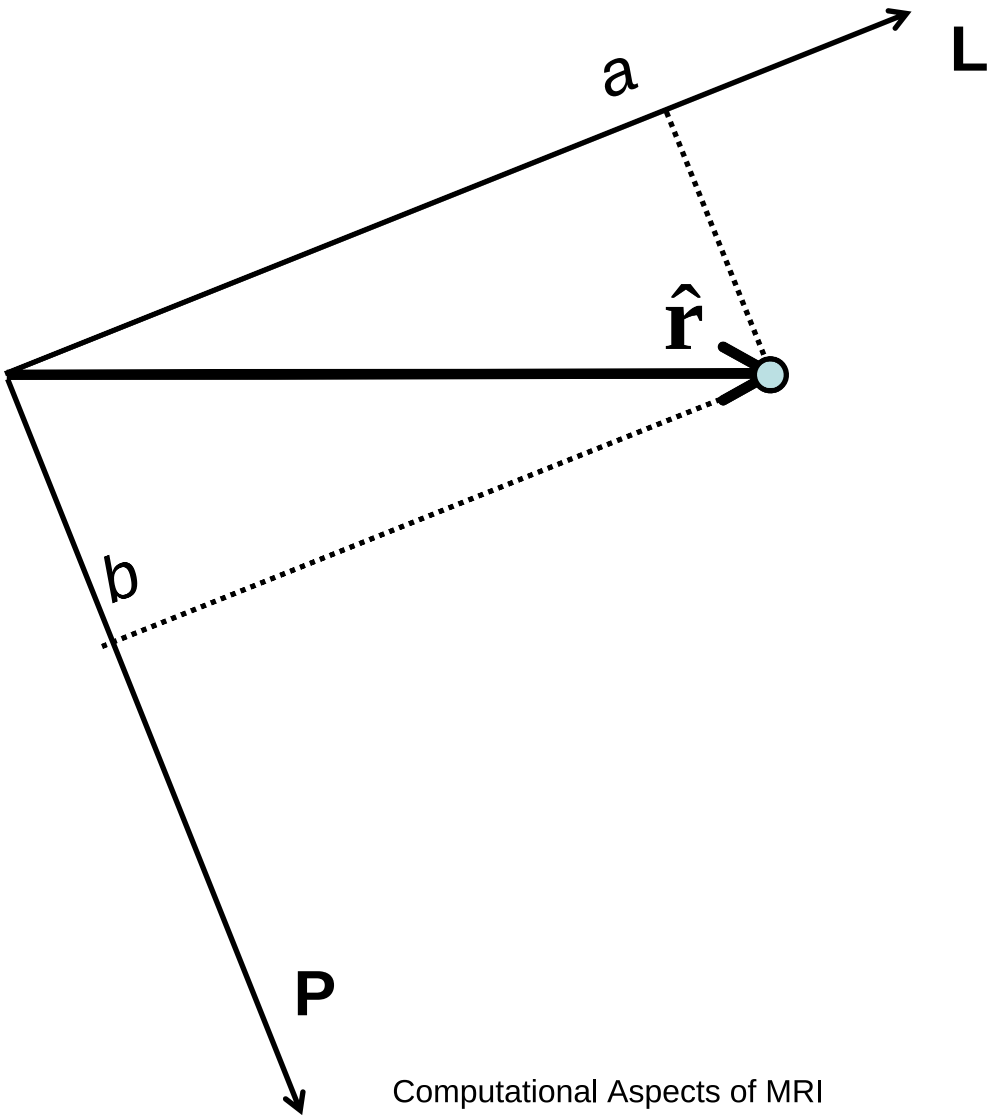
$$\hat{\mathbf{r}} = a\hat{\mathbf{x}} + b\hat{\mathbf{y}}$$

$$\|\hat{\mathbf{r}}\| = \sqrt{a^2 + b^2} = 1$$

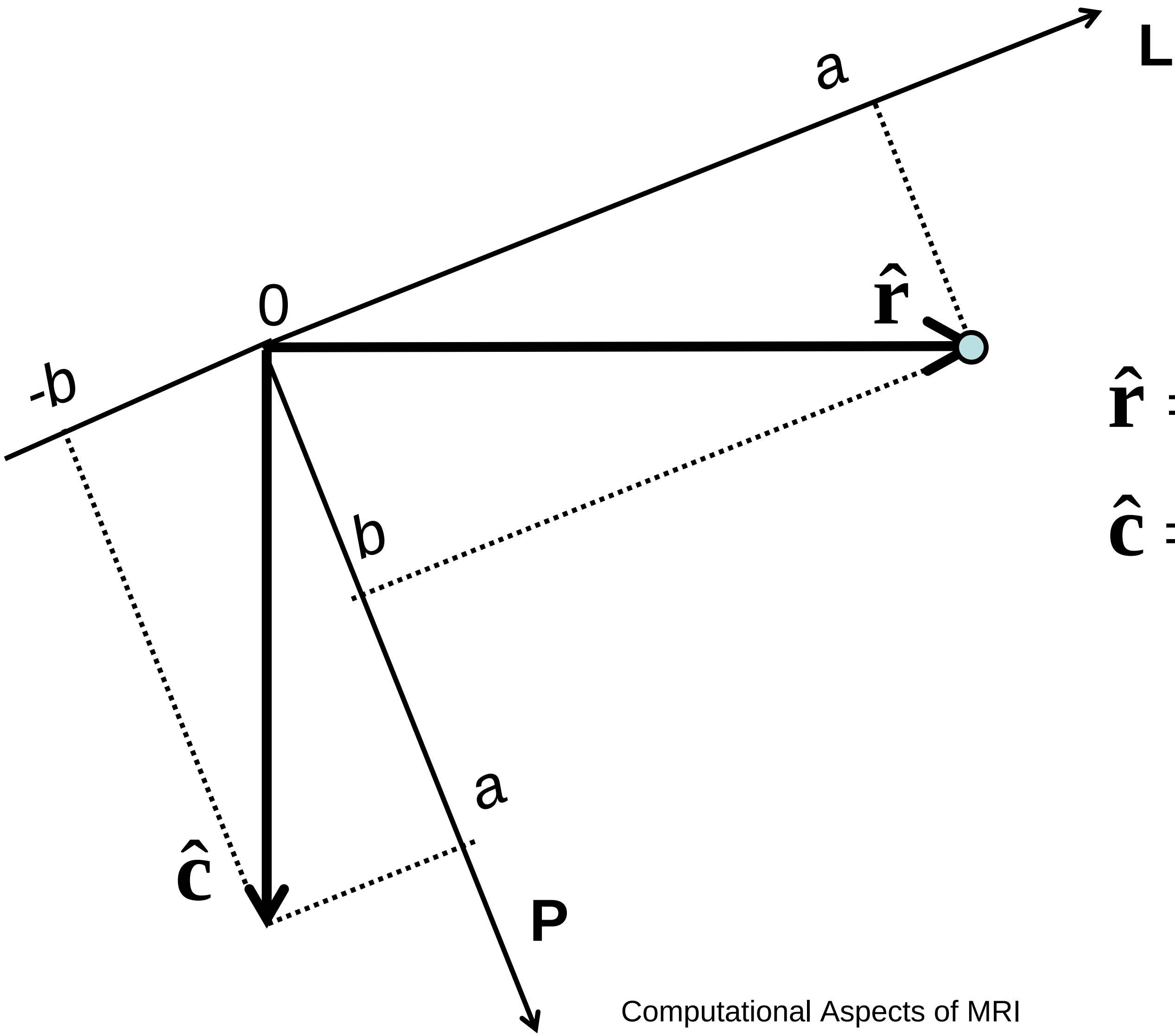
$$\hat{\mathbf{r}} = \cos(\theta_{\mathbf{x}})\hat{\mathbf{x}} + \cos(\theta_{\mathbf{y}})\hat{\mathbf{y}}$$

The components of a unit vector are the cosines of the angles the vector makes with the basis directions

Direction Cosines



Direction Cosines and DICOM



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

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

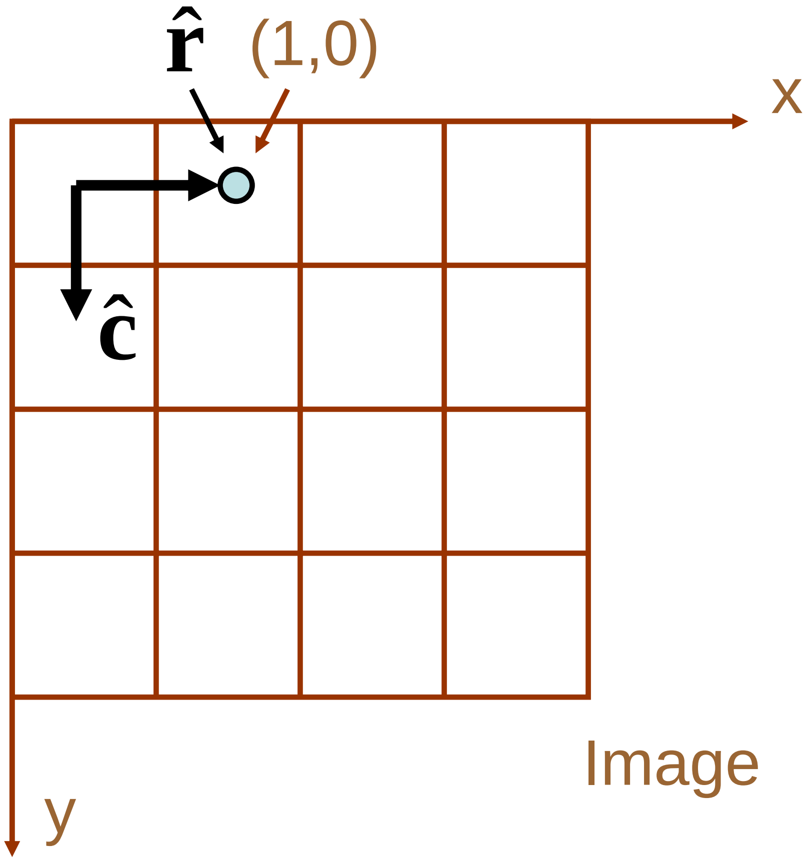
Image to Patient Transform: direction cosines as matrix columns

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

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

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

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



$$\begin{bmatrix} a & -b \\ b & a \end{bmatrix} = [\hat{\mathbf{r}}: \quad \hat{\mathbf{c}}:]$$

Direction Cosines in 3D with homogeneous coordinates

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

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

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

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

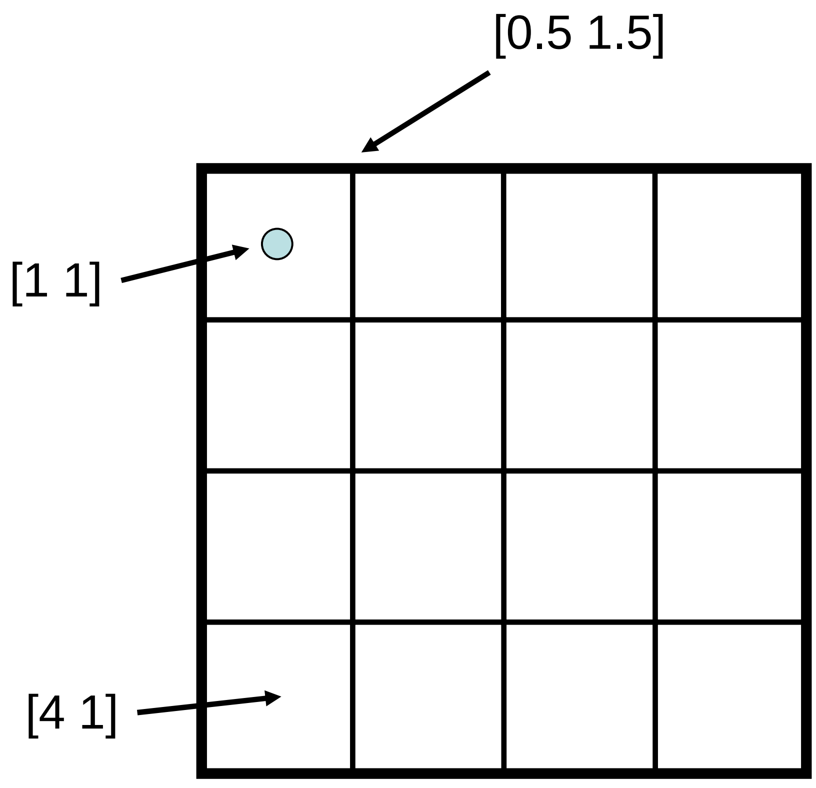
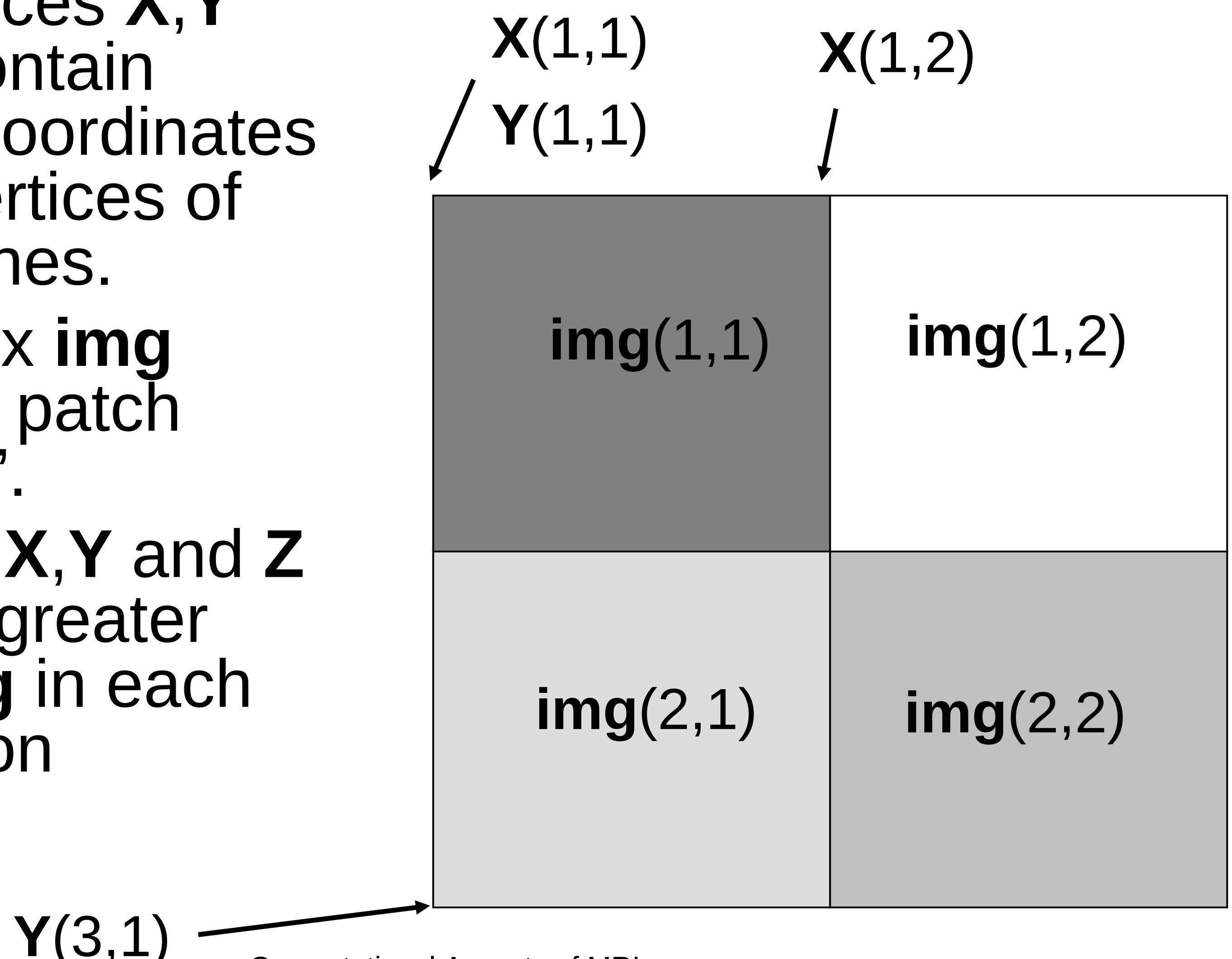


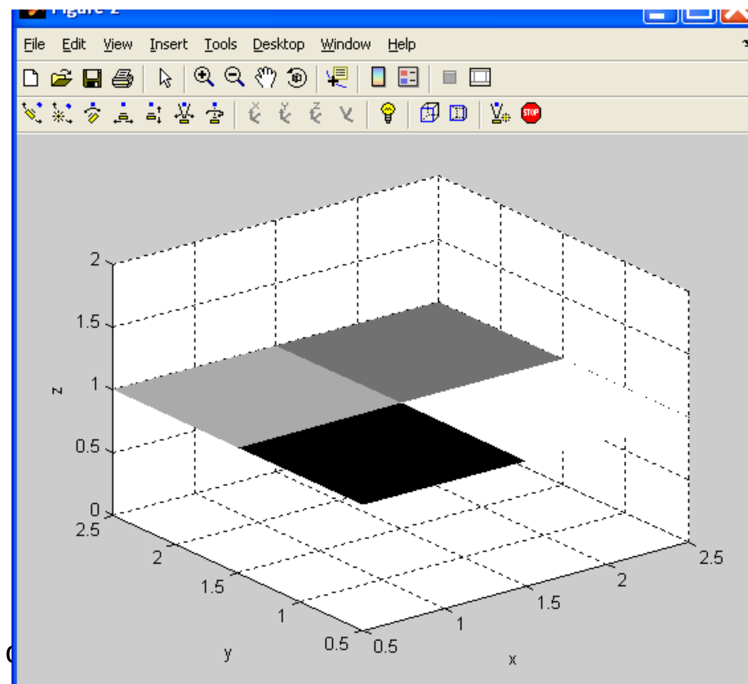
Image coordinates in [row column] order

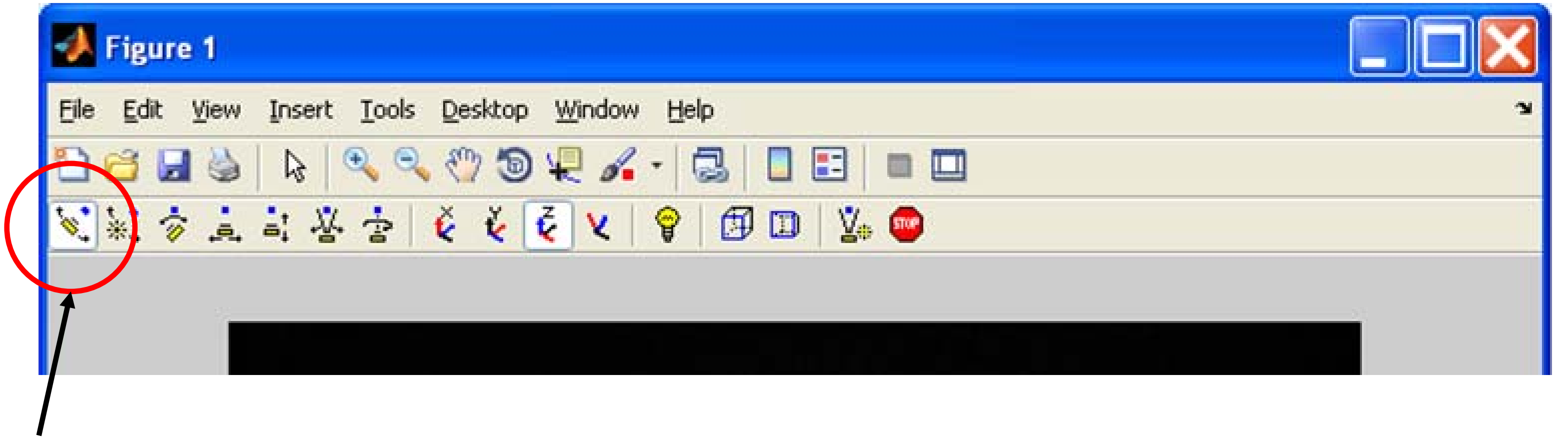
Displaying a 2D image in 3D using **surf(X,Y,Z,img)**

- 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)