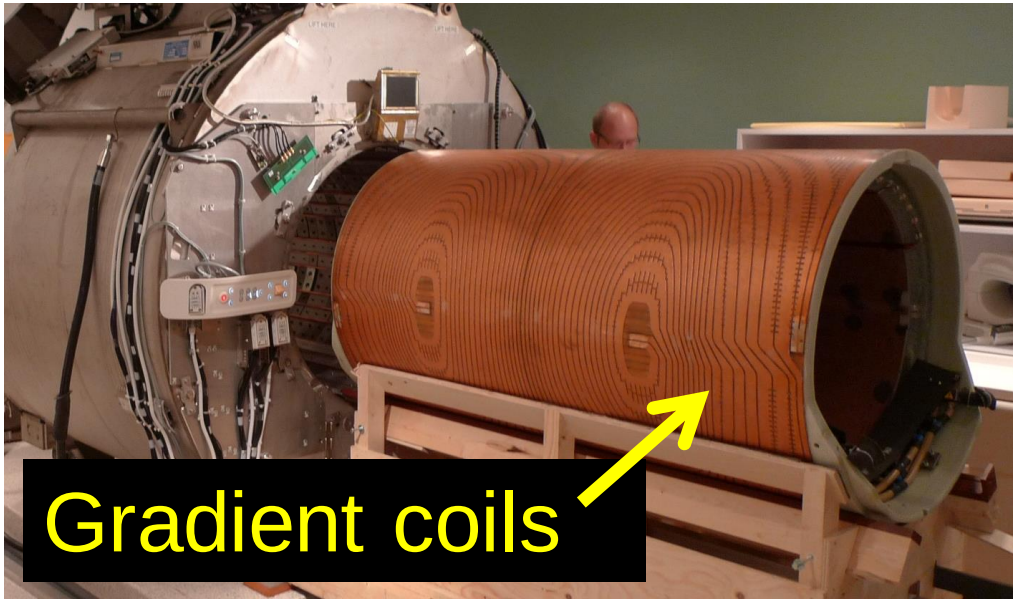
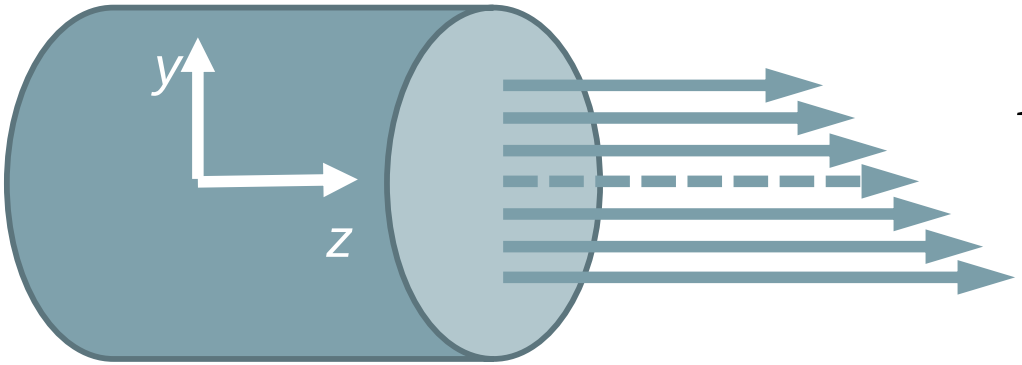
Geometry: PET-MR and DICOM

David Atkinson

University College London





$$B_z(y) = B_0 + G_y.y$$

$$\omega(r) = \gamma\, B_z(r)$$

Spatial Encoding

- Strong (3T) uniform field along bore (z).
- Spin frequency proportional to magnetic field.
  - 128MHz at 3T for protons in water
  - Slightly different for protons in fat
- Gradients change B_z as a function of space
  - Encodes spatial info in received frequency and phase.
  - x,y,z gradients can be applied simultaneously to give resultant in any direction desired – angulated slices.
  - Field from gradients at isocentre always zero.

Field Imperfections - Distortions

- Gradients not a perfectly linear change of field.
  - Consequence of technology and Maxwell's equations.
  - Can be calibrated.

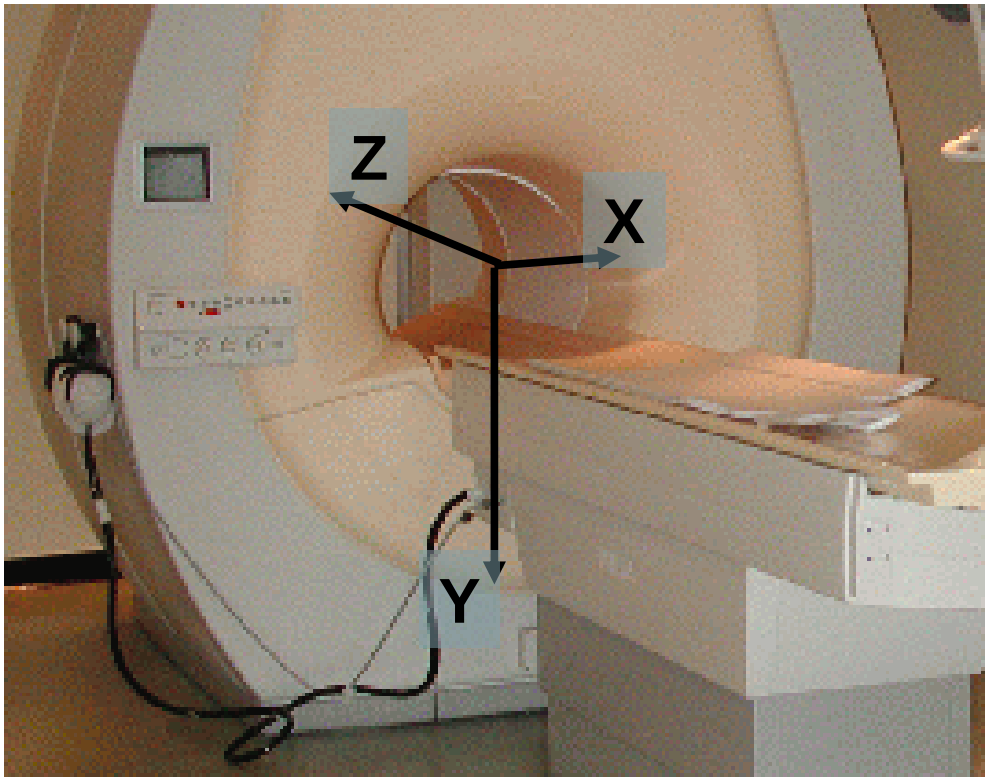
- Susceptibility differences.
  - Air-tissue interfaces
  - Local fields, patient and sequence-specific distortions.

- Chemical shifts.
  - e.g. water and fat

Coordinate Systems

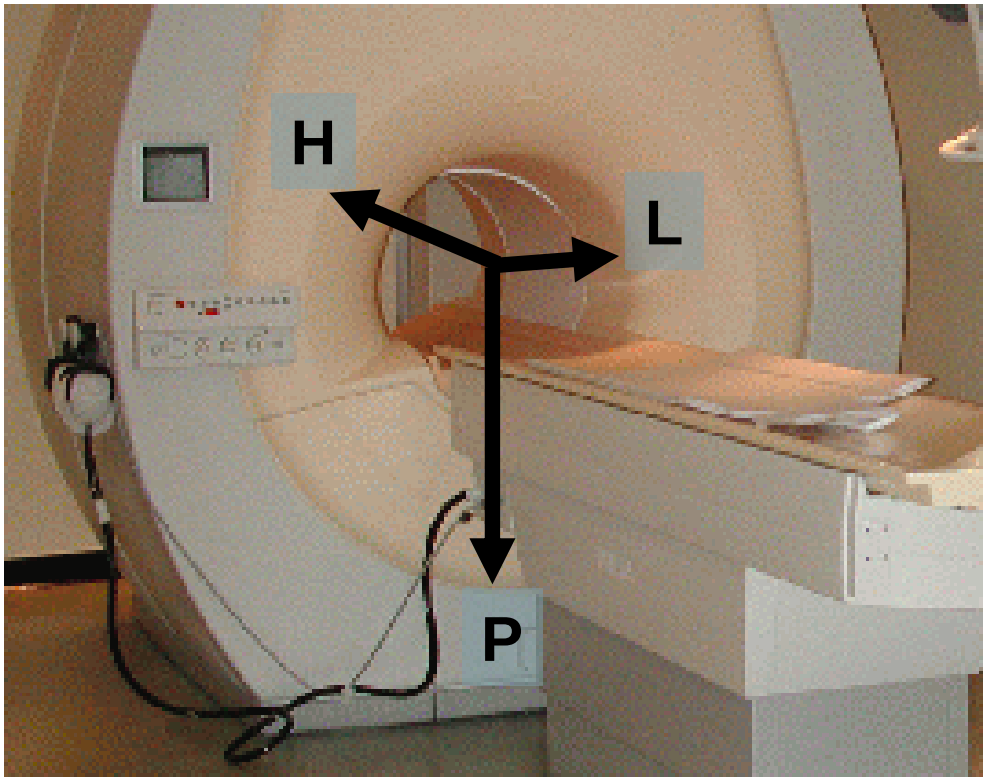
- Specified by an origin and orthogonal axes.
- Scanner, Patient, Image and Gradient systems.
- Vector cross product applies to right-handed systems
  - Changing sign of 1 axis switches between right and left handed.

Scanner Coordinate System



- Schematic only. No published definitions

DICOM Patient Coordinate System
Left-Posterior-Head



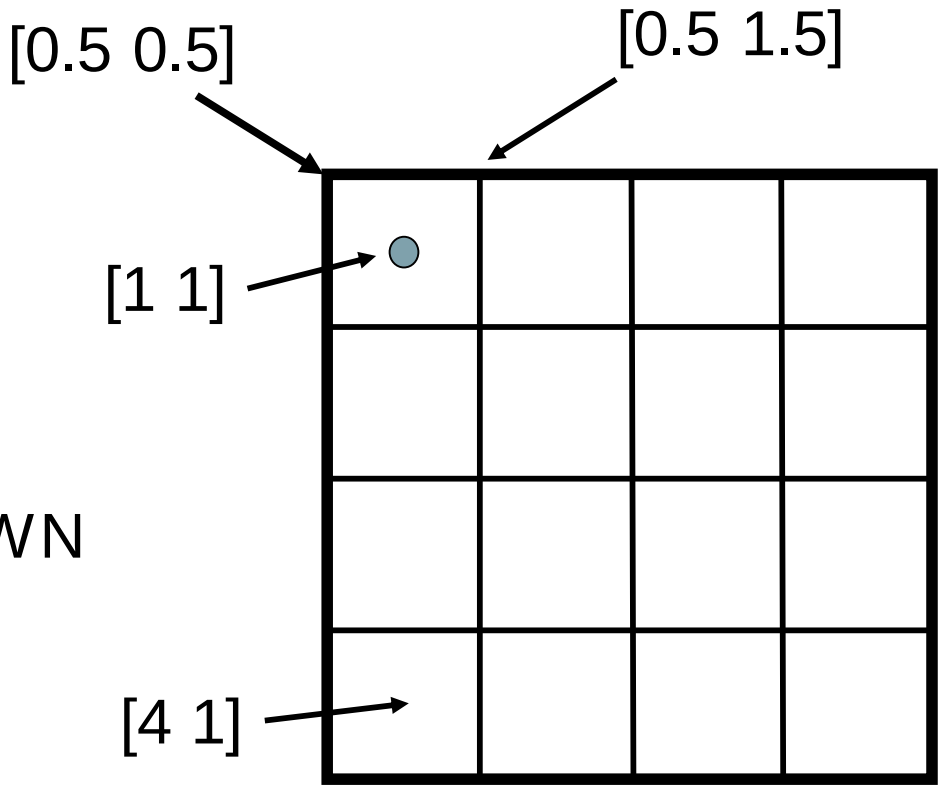
Schematic for supine (face-up), head-first patient.

It is ESSENTIAL the radiographer enters the correct patient orientation.

DICOM

- Specifies geometry.
- All manufacturers claim to support.
- Allows surgery/radiotherapy planning, image fusion.
- Relative to scanner, origin shifts with the bed.
- Origin is consistent across all scans with the same FrameOfReferenceUID.
- Origin has no physical relation to isocentre.
- Clearly defined relation to Image coordinates.

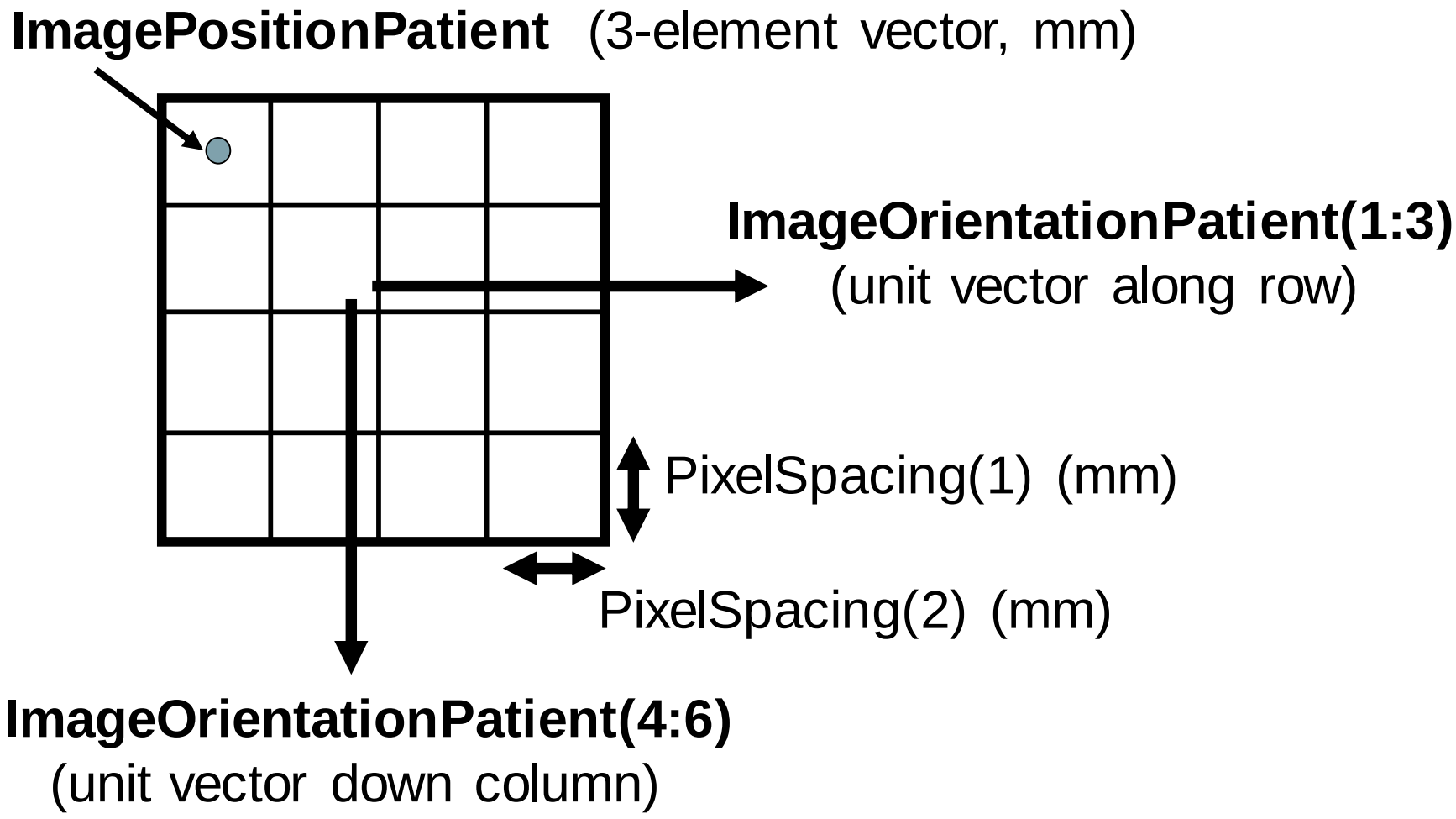
Image Coordinate System (MATLAB Default)



Row numbers increase going DOWN

Image coordinates in [row column] order
Pixel centres are integer coordinates.

DICOM Parameters Linking Patient (3D) to Image Plane (2D)



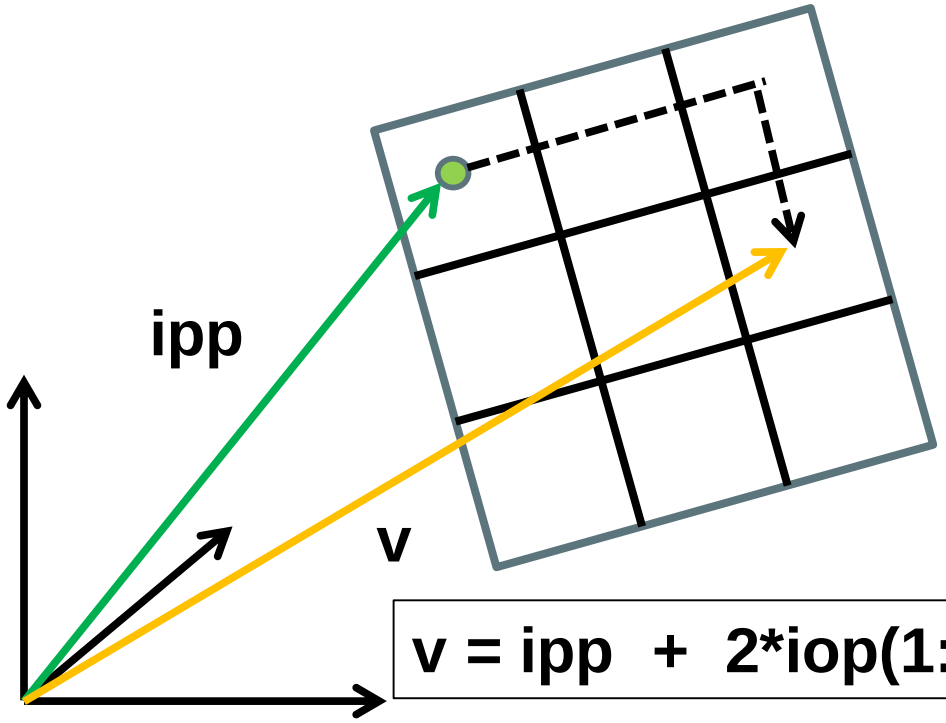
DicomFinishGadget.h

```
// Image Position (Patient)
float corner[3];

corner[0] = m1->getObjectPtr()->position[0] -
    (m1->getObjectPtr()->field_of_view[0] / 2.0f) * m1->getObjectPtr()->read_dir[0] -
    (m1->getObjectPtr()->field_of_view[1] / 2.0f) * m1->getObjectPtr()->phase_dir[0];
corner[1] = m1->getObjectPtr()->position[1] -
    (m1->getObjectPtr()->field_of_view[0] / 2.0f) * m1->getObjectPtr()->read_dir[1] -
    (m1->getObjectPtr()->field_of_view[1] / 2.0f) * m1->getObjectPtr()->phase_dir[1];
corner[2] = m1->getObjectPtr()->position[2] -
    (m1->getObjectPtr()->field_of_view[0] / 2.0f) * m1->getObjectPtr()->read_dir[2] -
    (m1->getObjectPtr()->field_of_view[1] / 2.0f) * m1->getObjectPtr()->phase_dir[2];

key.set(0x0020, 0x0032);
ACE_OS::snprintf(buf, BUFSIZE, "%.4f\\%.4f\\%.4f", corner[0], corner[1], corner[2]);
WRITE_DCM_STRING(key, buf);

// Image Orientation
// read_dir, phase_dir, and slice_dir were calculated in
// a DICOM/patient coordinate system, so just plug them in
key.set(0x0020, 0x0037);
ACE_OS::snprintf(buf, BUFSIZE, "%.4f\\%.4f\\%.4f\\%.4f\\%.4f\\%.4f",
    m1->getObjectPtr()->read_dir[0], m1->getObjectPtr()->read_dir[1], m1->getObjectPtr()->read_dir[2],
    m1->getObjectPtr()->phase_dir[0], m1->getObjectPtr()->phase_dir[1], m1->getObjectPtr()->phase_dir[2]);
WRITE_DCM_STRING(key, buf);
```



$$v = ipp + 2*iop(1:3)*PS(2) + iop(4:6)*PS(1)$$

ipp = ImagePositionPatient (mm)

iop = ImageOrientationPatient (unit vectors)

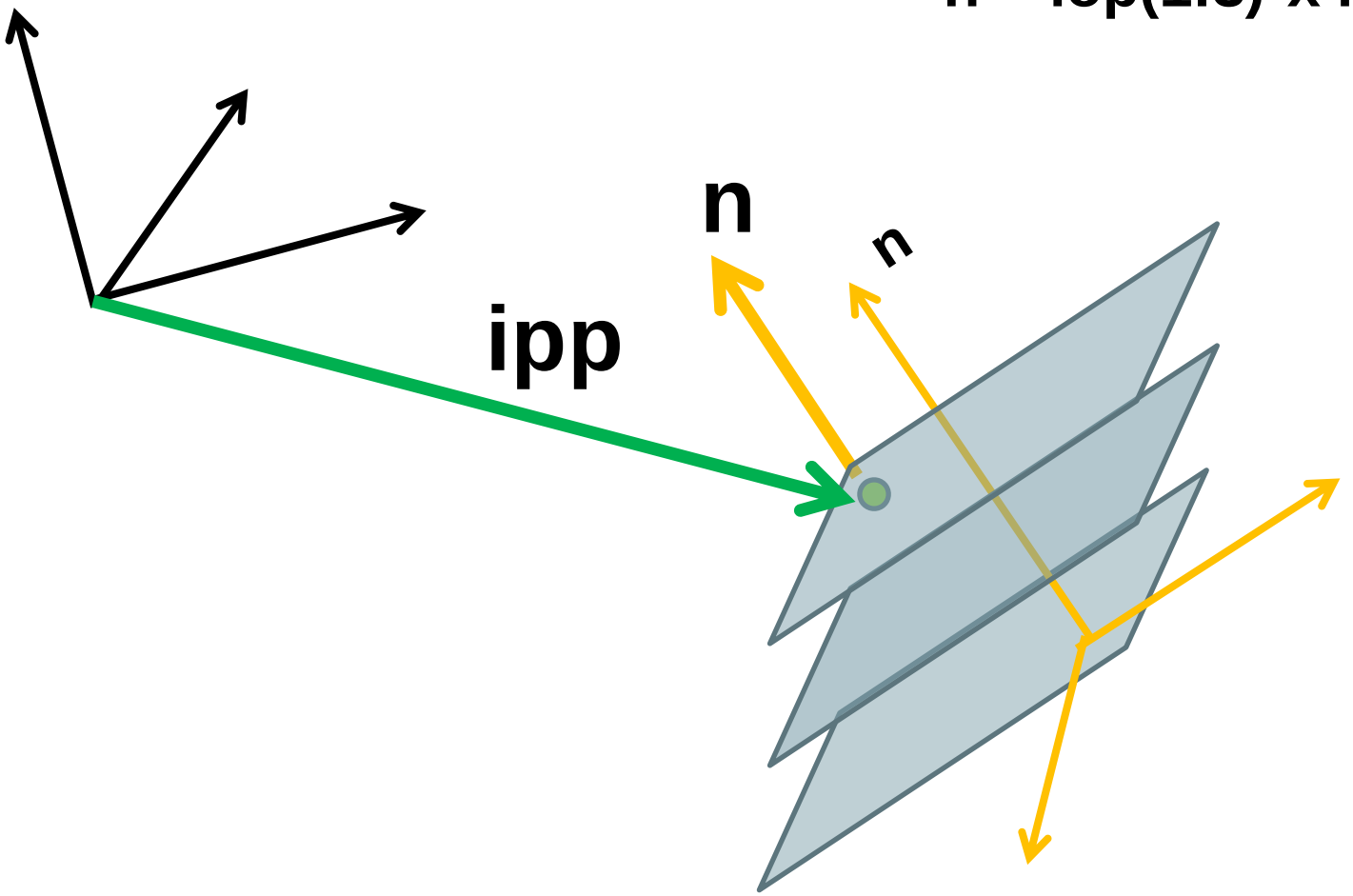
PS = PixelSpacing (mm)

Through Slice Direction

- Direction: normal to image plane = vector cross product of the row and column **ipp** vectors.
- Do not rely on manufacturer "slice number" or anything with a similar name for ordering.
- Scalar product of **iop** with slice normal can be used for ordering.
  - **iop, ipp** and PixelSpacing are all you need.
- Slice "thickness" may be unclear (acquired or reconstructed?). Slices can have gaps, gaps can be negative.

Sort Slices on Scalar Product ipp.n

n = iop(1:3) x iop(4:6)



Gradient Coordinate System

- Origin at isocentre.
- Axes:
  - Measurement (or ‘readout’ or ‘frequency’)
  - Phase
  - Slice
- Axes rotate with slice angulation wrt scanner.
- Change of gradient sign (fat shift direction) changes right/left handedness.
- **Space of the Raw Data**.

CCP

- Primary challenge is to align PET raw reconstruction with MR raw reconstruction.
- DICOM is a valuable debugging tool.
- Recommend we adopt DICOM conventions whenever there is uncertainty.

Where is my MR reconstruction origin?

- Replace acquired k-space with 1’s.
- Reconstruct (may include cropping and shifts).
- Single bright pixel should be at the reconstruction origin.

Where is the isocentre in my MR reconstruction?

- Probably enough information in the raw data header to work this out.
- Possible experiment:
  - Image structured phantom with repeated, different angulations.
  - Rotate slice by single angles only and no offsets.
  - Register using known angles and unknown rotation centre.
  - Rotation centre will be isocentre.

Can I account for susceptibility shifts in recon?

- In principle “Yes”, but requires a field map.

- Post-processing warpings also possible.

and gradient non-linearity distortions?

- Order of mm away from isocentre.
- Calibration requires a large (heavy) phantom.
- Best to get data from manufacturer…?