Transformations in Aims and Motion class change

(this doc is also in the C++ API documentation of the Transformation3d class)

WARNING: Affine 3D transformations have changed in AIMS 4.0: the old Motion class of Aims 3.x, has been renamed and changed.

A new transformation classes tree has been setup to allow non-linear transformations: see Transformation and Transformation3d classes. AffineTransformation3d now inherits Transformation3d.

A typedef is still provided for backward compatibility, in aims/resampling/motion.h (the former location of the Motion class) and should minimize inconvenience when compiling old code which used Motion. However a few API differences may cause compilation problems:

- forward declatations for the Motion class will not work any longer, since Motion is not a class anymore but a typedef. It is still possible to replace:

  ```
  class Motion;
  ```

  by:

  ```
  namespace aims { class AffineTransformation3d; }
  typedef aims::AffineTransformation3d Motion;
  ```

  but of course it is better to use directly the AffineTransformation3d class under its real name.

- the Motion::transform_normal() method is now transformUnitNormal()

- there are now several overloaded transform() methods taking double, or float numbers, or Point3df or Point3dd arguments. As there were formerly only float and Point3df arguments, ambiguities may raise from calling them with mixed double/float arguments

- the base aims::Transformation class introduces a name ambiguity with anatomist anatomist::Transformation class, so it now requires to handle namespaces carefully.

- the Motion DataTypeCode has also changed to AffineTransformation3d.