23. Juni 2014

Platform project restructuring of basic MITK types

Sebastian Wirkert

Medizinische und Biologische Informatik
Deutsches Krebsforschungszentrum (DKFZ), Heidelberg

dkfz. DEUTSCHES KREBSFORSCHUNGSZENTRUM IN DER HELMHOLTZ-GEMEINSCHAFT

Platform project overview

- Switched mitk::ScalarType from float to double
  - To be more in line with itk, which mostly typedefs to double
- Enabled easier type conversion from and to basic mitk types as Vector, Point and Matrix
- Restructured monolithic mitkVector.h header

Typ conversions (1)

- Goal: easier conversion of mitk basic datatypes:
  - mitk::Point
  - mitk::Vector
  - mitk::Matrix
- E.g. to vnl_vector, vnl_vector_fixed, cv::Point

Typ conversions (2)

- mitk::Point, mitk::Vector and mitk::Matrix now proper classes
  - Derived from itk::Point/Vector/Matrix
- Mitk::Vector has implicit conversion functions for vnl_vector, vnl_vector fixed and itk base classes:

```
void Mitk2Itk(void)
{
  Vector3D vector3D = valuesToCopy;
  itk::Vector<ScalarType, 3> itkVector = originalValues;

  itkVector = vector3D;

  TestForEquality(itkVector, vector3D, "itk::Vector", "mitk::Vector3D");
}


void Vnlfixed2Mitk(void)
{
  mitk::Vector3D vector3D = originalValues;
  vnl_vector_fixed<ScalarType, 3> vnlVectorFixed(valuesToCopy);

  vector3D = vnlVectorFixed;

  TestForEquality(vector3D, vnlVectorFixed, "mitk::Vector3D", "vnl_vector_fixed<ScalarType>");
}


void Mitk2Vnlfixed(void)
{
  vnl_vector_fixed<ScalarType, 3> vnlVectorFixed(originalValues);
  mitk::Vector3D vector3D = valuesToCopy;

  vnlVectorFixed = vector3D;

  TestForEquality(vnlVectorFixed, vector3D, "vnl_vector_fixed<ScalarType>", "mitk::Vector3D");
}
```

Typ conversions (3)

- Impicit conversion not possible for all classes (e.g., cv::Point, POD)
  - Use ToArray and FillVector/FillPoint/FillMatrix methods for classes/types implementing the []-operator
  - Doesn‘t care about copying from double to float

```
void ToArray_DifferentType(void)
{
  float podArray[3];
  for (int var = 0; var < 3; ++var) {
    podArray[var] = originalValues[var];
  }
  mitk::Vector3D vector3D = valuesToCopy;

  vector3D.ToArray(podArray);

  TestForEquality(podArray, vector3D, "float POD", "mitk::Vector3D", epsDouble2Float);
}


void Fill_DifferentType(void)
{
  mitk::Vector3D vector3D = originalValues;
  float podArray[3];
  for (int var = 0; var < 3; ++var) {
    podArray[var] = valuesToCopy[var];
  }

  vector3D.FillVector(podArray);

  TestForEquality(vector3D, podArray, "mitk::Vector3D", "float POD", epsDouble2Float);
}
```

Typ conversions (4)

- Vector to Point
  - Vector is treated as vector from origin

```
void Vector2Point()
{
  itk::Point<double, 3> point3D   = valuesToCopy;
  itk::Vector<double, 3> vector3D = originalValues;

  point3D = vector3D;

  TestForEquality(point3D, vector3D, "mitk::Point", "mitk::Vector");
}
```

Typ conversions (5)

- Point to Vector
  - Use GetVectorFromOrigin method

```
void Point2Vector()
{
  mitk::Point3D point3D   = originalValues;
  mitk::Vector3D vector3D = valuesToCopy;

  vector3D = point3D.GetVectorFromOrigin();

  TestForEquality(point3D, vector3D, "mitk::Point3D", "mitk::Vector3D");
}
```

mitkVector.h restructuring (1)

- before: everything related to basic datatypes located in mitkVector.h
- now
  - In mitkNumericTypes.h
  - Capsuled in subheaders:

```
#ifndef MITKNUMERICTYPES_H_
#define MITKNUMERICTYPES_H_


#include "mitkNumericConstants.h"
#include "mitkQuaternion.h"
#include "mitkAffineTransform3D.h"
#include "mitkPoint.h"
#include "mitkVector.h"
#include "mitkMatrix.h"
#include "mitkEqual.h"


// this include hold the old deprecated ways to convert from itk 2 vtk and the likes.
// calls to these functions shall be removed in future bugsquashings so that this include can be removed.
#include "mitkVectorDeprecated.h"



#endif /* MITKNUMERICTYPES_H_  */
```

mitkVector.h restructuring (2)

- Extract from documentation:

- -# mitkNumericConstants.h : contains basic constants like mitk::ScalarType or mitk::eps
- -# mitkArray.h : copy itk::FixedArrays (like itk::Point and itk::Vector) from and to types which implement the [] operator (array-types), like e.g. Plain Old Datatypes (POD) or the opencv vector
- -# mitkPoint.h : the mitk::Point class. This is basically the itk::Point with the added ToArray and Fill members to conveniently copy from and to array-types. In MITK, a point is considered a fixed geometric location and thus cannot be summed or multiplied.
- -# mitkVector.h : the mitk::Vector class. This is an itk::Vector, but with the possiblity to implicitly convert to vnl_vector and vnl_vector_fixed. In MITK, vectors denote directions and can be summed or multiplied with scalars.
- -# mitkMatrix.h : the mitk::Matrix class. This is an itk::Matrix with the added ToArray and Fill members to conveniently copy from and to array-types.
- -# mitkQuaternion.h : a typedef to vnl_quaternion.
- -# mitkAffineTransform3D.h : a typedef to itk::AffineGeometryFrame<ScalarType, 3>
- -# mitkNumericTypes.h : this file includes all of the above as a convenience header

Find more

- More examples for type conversions:
  - mitkArrayTypeConversionTest.cpp
  - mitkPointTypeConversionTest.cpp
  - mitkVectorTypeConversionTest.cpp
  - mitkMatrixTypeConversionTest.cpp
- More information:
  - Concepts Page*: „BasicDataTypesPage Numeric MITK data types and their usage”*
  - (BasicDataTypes.dox)