11/10/2010

The Vision Numerics Library (VNL)

Tobias Heimann

dkfz.
GERMAN
CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION

What is VNL?

- Part of VXL (Vision-*something*-Libraries): collection of C++ libraries designed for computer vision research and implementation
- Distributed with ITK (Utilities/vxl/core/vnl) and thus available in MITK

> "VNL is intended to provide an environment for numerical programming which combines the ease of use of packages like Mathematica and Matlab with the speed of C and the elegance of C++. It provides a C++ interface to the high-quality Fortran routines made available in the public domain by numerical analysis researchers."

Data classes

- Vector and matrix classes
  - Templated, variable or fixed size
  - Loads of operators: min, max, rms, magnitude, dot / cross products
  - Accessible from corresponding ITK classes by GetVnlX / SetVnlX methods
- Polynomial function classes
  - Evaluate value, derivative, integral
- Quaternions
  - Conversion from and to Euler angles

Advantages against ITK classes:

- Leaner and meaner!
- No differentiation between points and vectors

Algorithms I: Matrix Decompositions

- Eigenvalue decomposition
  - For square matrices
  - Special versions for symmetric and sparse matrices
- Singular value decomposition
- Cholesky decomposition
  - For symmetric, positive-definite matrices
  - Also optimized LDL variant available
- QR decomposition

Solving systems of equations

For regular matrices:

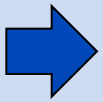
```
vnl_matrix<double> matrix;
Vnl_vector>double> vecRight, vecSolution;
…
vnl_linear system lse( matrix, vecRight );
vnl_lsqr solver( lse );
solver.set_max_iterations( 20 * lse.get_number_of_unknowns() );
solver.minimize( vecSolution );
```

Variant for sparse matrices:

```
vnl_sparse_matrix<double> smatrix;
Vnl_sparse_matrix_linear_system<double> slse( smatrix, vecRight );
```

Algorithms II: Optimizers

- Nelder-Meade downhill simplex
  - For noisy functions without derivatives
- Conjugate Gradient
  - Classic method to solve large linear systems of equations
- LBFGS
  - Best optimization for well-behaved functions with 1st derivatives
- Levenberg Marquardt
  - Nonlinear least squares optimization
- Powell
  - For many-dimensional, expensive functions (no derivative)



All VNL optimizers are accessible by ITK wrappers

Some useful Utilities…

- Fast Fourier transforms (1D+2D)
- Random number generation (normal+box distribution)
- Chi square distribution