Update on Matrix Algebra

Beomki Yeo
Math/Backends of Algebra-plugins

<table>
<thead>
<tr>
<th>Math</th>
<th>Backends</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPU</td>
<td>CUDA</td>
</tr>
<tr>
<td>cmath (home-made)</td>
<td>green</td>
<td>green</td>
</tr>
<tr>
<td>Eigen</td>
<td>green</td>
<td>yellow</td>
</tr>
<tr>
<td>SMatrix</td>
<td>green</td>
<td>grey</td>
</tr>
<tr>
<td>VC</td>
<td>green</td>
<td>grey</td>
</tr>
</tbody>
</table>

Backend is supported

Supported but not tested

Backend is NOT supported

Developed algebras before v0.8.0:
- vector algebra
- local $\leftrightarrow$ global transform
- matrix generation and element access
  - matrix algebra had been missing
Required Matrix Operations

- Matrix Creation
  - Zero
  - Identity

- Addition and Subtraction

- Multiplication
  - Normal multiplication
  - Blocked multiplication

- Transpose

- Inverse
  - 2x2 (KF updater)
  - 4x4 (local ↔ global transform)
  - 6x6 (KF smoother)
  - Maybe more?
Writing the matrix algebra for Eigen and SMatrix is just porting the existing functions.
cmath Implementation

- For cmath, we can try something better
  - User can decide which specific algorithm will be used for which matrix dimensions in compile time
  - Various matrix_actor can be defined to test different aggregations

```cpp
// Define inverse algorithm
// Base algorithm is cofactor method
// For 2x2 and 4x4 matrix, hard coded method is used
// For 3x3 and 5x5 matrix, LU decomposition is used
using inverse_actor = matrix::inverse::actor<cofactor>,
  hard_coded<2,4>, LU_decomposition<3,5>>

// Define matrix actor
using matrix_actor = matrix::actor<scalar, inverse_actor>

matrix<2,2> m22;
matrix<3,3> m33;
matrix<7,7> m77;

m22_inv = matrix_actor().inverse(m22) // hard-coded
m33_inv = matrix_actor().inverse(m33) // LU decomposition
m77_inv = matrix_actor().inverse(m77) // cofactor
```
Customizable algorithms in cmath matrix

- Determinant
  - cofactor
  - hard-coded for 2x2 and 4x4

- Inverse
  - cofactor
  - hard-coded for 2x2 and 4x4
Outlooks

- Similar concept of algorithm aggregation can be applied to Eigen and SMatrix implementation

- cmath needs 6x6 hard coded inversion or LU decomposition
  - cofactor is slow!

- cmath matrix multiplication relies on * operator which does standard multiplication
  - Needs to add customizable actor for multiplication as done for inverse and determinant