## An Efficient Solution to Structured Optimization Problems using Recursive Matrices

Darius Rückert<sup>1</sup> and Marc Stamminger<sup>1</sup>

<sup>1</sup> University of Erlangen-Nuremberg, Germany



High-Performance Graphics 2019, Strasbourg



### Motivation

Branch: master - ceres-solver / inter	nal / ceres /	Create new file	Upload files	Find file	Histor
alexsmac Default to any other sparse librar	ies over Accelerate		Latest commi	t c4dbc92 2	days ag
 benchmarks	Update the sizes in small_blas_gem?_benchmark.cc				last yea
generated	Add the 3,3,3 template specialization.		4 months ag		
generated_bundle_adjustment_tests	Add Apple's Accelerate framework as a sparse linear algeb	ora library.	a library. last yea		
gmock	Update googletest/googlemock to db9b85e2.			4 moi	nths ag
gtest	Update googletest/googlemock to db9b85e2.			4 mor	nths ag
miniglog/glog	Make miniglog threadsafe on non-windows system by usin	g		4 ye	ears ag
CMakeLists.txt	Speed up InvertPSDMatrix			28 0	lays ag
accelerate_sparse.cc	Recycle numeric factorisation when using Accelerate Spar	se.			last yea
accelerate_sparse.h	Recycle numeric factorisation when using Accelerate Spar	se.			last yea
algorithm_test.cc	Switch to FixedArray implementation from abseil.			3 moi	nths ag
array_utils.cc	Deprecate macros.h and fpclassify.h				last yea
array_utils.h	Fix typos.			9 moi	nths aç

> 50 % of the code is for

- Sparse Block Matrices
- Linear Solvers

Can we get rid of that?

#### https://github.com/ceres-solver/ceres-solver

### Overview



#### https://eigen.tuxfamily.org

### What is a Recursive Matrix?

### Definition

1

A recursive matrix is a rectangular array of numbers or recursive matrices.

$$\begin{bmatrix} -1 & 1 \\ 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{bmatrix}$$

## **Recursive Matrix Operations**

- Multiplication
- Addition
- Transposition



## **Recursive Matrices in Eigen**

Scalar Matrix

Matrix<float,3,3>

Dense Block Matrix

Matrix<Matrix<float,2,2>2,2>

Sparse Block Matrix

SparseMatrix<Matrix<float,2,2>>

Recursive Operations in Eigen

> Not working without our extension (see Paper)

$$\begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}$$
$$\begin{bmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \end{bmatrix}$$

 $\left| \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right|$ 

 $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ 

 $\begin{bmatrix} \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ 

 $\begin{pmatrix} 5\\ 0 \end{pmatrix}$  $\begin{pmatrix} 0\\ 5 \end{pmatrix}$   $\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix}$ 



 $\begin{pmatrix} 4 & 1 \\ 2 & 3 \end{pmatrix}$ 





High-Performance Graphics



### Mixed Matrix

### Definition

A mixed matrix is a rectangular array, where each element can be of a different type.

$$\begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}$$
 42  
31 + 5i "Hello"

MixedMatrix22<
 Matrix<float,3,3>,
 float,
 complex<float>,
 string
 > A;

### **Recursive Bundle Adjustment**



MixedMatrix22<

DiagonalMatrix<Matrix<float,3,3>>,
SparseMatrix<Matrix<float,3,6>>,
SparseMatrix<Matrix<float,6,3>>,
DiagonalMatrix<Matrix<float,6,6>>
> H;

How do we solve

 $H\Delta x = b$  ?

### **Recursive Linear Solvers**

"Normal" recursive matrices

- Recursive CG, Recursive LDLT,...
- Straight forward (see paper)

Mixed recursive matrices

- General solvers (CG)
- Partially specialized solvers

## **Partially Specialized Solvers**



#### Implementation

- 1. Invert Diagonal Matrix
- 2. Compute Schur Complement
- 3. Solve Reduced System (Recursive Call!)
- 4. Compute Solution for Initial System

Mixed 2x2 Matrix

### Better than general solver

### **Template Matching**



### Results

### **Structured Optimization**

High-Performance Graphics

**ARAP** - Speedup

Bundle Adjustment - Speedup



High-Performance Graphics

### Results

## Sparse Block Matrix Multiplication

### Sparse Block Matrix-Vector

### Sparse Block Matrix-Matrix



Clang 8.0 - 1 Thread on i7-7700K, SSE+AVX

# Application

- Camera Tracking (SLAM)
  - Local/Global BA
  - Pose Refinement
  - Pose Graph Optimization



### > 1.17 ms/frame (~850 FPS)\*

\*4 Threads on i7-8850H

## Any Questions?





https://github.com/darglein/EigenRecursive
 darius.rueckert@fau.de

High-Performance Graphics