Robot Vision Systems

Exercise 2: Adding functionality to sparse matrices in OpenCV

Michael Felsberg

michael.felsberg@liu.se

Preliminaries

- Test all functions for a correct result by casting to dense matrices and comparing the results
- Use a timer to get the timing of your function if you want to compare several implementations
- Use CV_Assert or exceptions for ensuring requirements

Scalar product

- There is no scalar product on 1D sparse matrices (sparse vectors)
- Task: implement an efficient function for two vectors of type float/double
- Optional: complex and or quaternion
- Hint1: use example from documentation (core)
- Hint2: exploit properties of class Scalar

ND-1D product

- There is no matrix-vector product or generalized ND-array-vector product
- Task: implement a product between an ND-array and a vector, where contraction is done in the last dimension of the array
- Same data type as for task1
- Optional: an additional parameter determines the index to be contracted
- Hint: run one single iterator of the array

ND-MD Product

- Two multi-dim arrays A and B can be multiplied in the same way as in the previous task, if the index set of B is a subset of the one of A
- Task: write a function for contracting A with B, given that the index set of B is equal to the last M indices in A
- Optional: give an association vector (similar to mixchannels) for contracting over an arbitrary subset

Specialized functions

- For certain cases, the above defined functions have known names, e.g. Frobenius product.
- Task: define wrappers for matrix-vector product, Frobenius product of matrices, and Frobenius norm of a matrix

Questions

- What makes it difficult to implement an efficient sparse general matrix product?
- How could the previously defined matrix-vector product be used for the general product? Which preprocessing is required?
- Optional: write a function for the general matrix product
- Optional2: write an outer product function (Kronecker product)