

Compressive Sensing in Video Reconstruction

Compilation

For **serial** compilation: Make sure to comment out the line `#define USE_MPI` flag at the top of `main.cpp`. Then compile using

```
$ g++ main.cpp -llapack -lblas -std=c++11 -O3 -o main
```

or

```
$ g++ main.cpp -llapack -lblas -std=c++0x -O3 -o main
```

For **parallel** compilation: Make sure to uncomment the `#define USE_MPI` flag at the top of `main.cpp`. Then compile using

```
$ mpic++ main.cpp -llapack -lblas -std=c++11 -O3 -o main
```

or

```
$ mpic++ main.cpp -llapack -lblas -std=c++0x -O3 -o main
```

Usage

To print the help message:

```
$ ./main -h
```

or

```
$ ./main --help
```

To use in **serial** mode:

```
$ ./main [configfile]
```

To use in **parallel** mode:

```
$ mpirun -np <n> ./main [configfile]
```

where `n` is the number of processes. For more options on MPI, check the [open-mpi faq](#).

If no 'configfile' is specified, the program will look for settings file `.rvmsettings.cfg` in the current directory and attempt to use it.

Configuration File

The format of the configuration file is:

```
setting1 = value1
setting2 = value2
...
```

Leading and trailing whitespace is mostly ignored. Each line can hold at most one setting/value pair. Setting/value statements must contain `=`. The order of the statements is irrelevant. To use the default value for a particular setting, remove (or comment it out using `#`) that setting from the configuration file.

The names of the output files will be saved in a `.fls` file where the stem of the filename is the same as that of the settingsfile (so the default is `.rvmsettings.fls`). This is to allow interfacing with other programs for later analysis.

Parameter Settings

`inputFile`: Input file name. Txt file containing input signal pixel values. Frames are separated by empty lines. Each frame must have a consistent number of rows and columns, respectively (required).

`outputDirectory`: Name of output directory (default: `./`).

`outputName`: A label for the names of the output files. Everything up to final '/' character will be ignored. If no `outputName` is provided, one will be constructed using the signal settings.

`blockHeight`: Height of each signal block. (default: `2`)

`blockWidth`: Width of each signal block. (default: `2`)

`blockFrames`: Depth of each signal block. (default: `2`)

`simulateCorruption`: If `0`, the sensing matrix is assumed to be a mask and a mask file must be provided. If `1`, we simulate the sensing mechanism within the program (default: `1`).

`maskFile`: File name of signal mask. Required if `simulateCorruption = 1`. Ignored if `simulateCorruption = 0`.

`percentage`: Percentage of Compressive Sensing measurements relative to the full signal length. Ignored if `simulateCorruption = 0`. (default: `50`).

`sensorMode`: Type of sensing matrix. Possible values: `mask`, `gaussian` or `bernoulli`. Ignored if `simulateCorruption = 0` (default: `mask`).

`maskMode`: Decimation pattern of simulated mask. Possible values: `uniform`, `timeRays`, `verticalFlicker`, `horizontalFlicker`, `missingFrames`, `verticalLines` or `horizontalLines`. Ignored if `simulateCorruption = 0` or if `sensorMode` does not equal `mask`. (default: `uniform`).

`maskFill`: If `sensorMode = mask`, we fill up the recovered signals with original measurements where possible.

`basisMode`: Type of basis function used to represent the signal. Possible values: `haar`, `dct` (default: `dct`)

`basisStartScale`: Minimum scale of wavelet basis functions for the cascade of RVMs. Ignored if `basisMode = dct` (default: `1`).

`basisEndScale`: Maximum scale of wavelet basis functions for the cascade of RVMs. Ignored if `basisMode = dct` (default: `1`).

`stdDev`: Standard deviation of the noise in the RVM (default: `1.0`)

`deltaML_threshold`: Convergence threshold for the change in marginal likelihood in the RVM (default: `1.0`)

`rngSeed`: Seed for random number generator for reproducibility. If no seed is specified, the current date and time will be used.

`computePSNR` : If 1, the Mean Square Error and Peak Signal-to-Noise Ratio for each reconstruction stage will be computed and displayed on standard out. Note that this computation will cap Signal entries in the range [0,255]. Actual saved output, however, is not affected by this capping. (default: 1)

`printToLogFile` : If 1, we redirect stdout and stderr to a log file (default: 1)

`logFile` : File name for a log file. Ignored if 'printToLogFile = 0'. The default name is the same as that of the settings file but with the `.log` extension.

`convertToMedia` : If 1, the output files will be passed Matlab and converted to media files. Images will be saved in `.png` format. Videos will be saved in lossless `.avi`. Requires MATLAB to be installed (and be in the Path). (default: 1)

`frameRate` : Frame rate for output videos. Ignored if input signal is an image or if `convertToMedia=0` . (default: 25);

Output

The algorithm will save the output in `.txt` files. There are several output files that all utilize the name specified in `outputName` as well as an additional label to differentiate them.

The format of the output file name is "`<outputName>_<label>.txt`".

The following labels will be produced:

`MASKED` : Contains the masked signal. This will only be produced if `sensorMode=mask` .

`MEASUREMENTS` ; A text file containing all the Compressive Sensing Measurements. The file will be filled with zeros until it has the same dimensions as the input signal.

`RECOVERED_<p>_OF_<q>` : Contains the output of cascade with wavelet scale `p`, where the total number of cascades is `q`.