ScattLab Architecture Document

March 27, 2013

1 Filters

1.1 Filter Structure

Filters are defined by a signal size `[N,M]`, a filter type (Morlet, Gabor, spline), and wavelet-specific parameters. The boundary conditions (symmetric or periodic) are also specified when defining a filter. For one-dimensional signals, $M = 1$.

Filter parameters are specified in a parameters structure, `fparam`, containing the following fields:

- `fparam.filter_type`: The wavelet type, such as 'morlet_1d', 'gabor_2d', 'spline_1d', for example.
- `fparam.boundary`: The boundary conditions used in convolutions. Could be either 'symm' or 'per' for symmetric or periodic boundary conditions, respectively.

In addition, the `fparam` structure would contain parameters specific to the wavelet type chosen (see below).

Once filter parameters are entered, the `filter_bank` function is called to generate the filter bank:

```
filters = filter_bank([N M],fparam);
```

This function will then call the appropriate filter bank function (`filter_bank_morlet_1d`, `filter_bank_gabor_2d`, etc.) depending on the value of `fparam.filter_type`.

The returned structure, `filters`, contains the filters ψ and ϕ that form the filter bank, as well as meta information. Specifically, the fields are:

- `filters.psi`: A set of wavelet filters ψ_λ (for definition, see below)
- `filters.phi`: A set of lowpass filter(s) ϕ (for definition, see below)
- `filters.meta`: The meta information on the filter bank. Specifically, the parameters given in `fparam` and the signal size `[N,M]`. For example, `filter.meta.filter_type` gives the type of filters in `filters.psi` and `filters.phi`.

Each filter set (be it `filters.phi` or `filters.psi`), is a structure `fset` containing the following:

- `fset.filter`: A cell array of the actual filter coefficients. These coefficients are implementation-dependent and can encode the filter spatially, in the Fourier domain, at different resolutions, etc.
- `fset.meta`: Contains meta information on the filters. Specifically, it has two fields: `fset.meta.k`, which is the scale indices, and `fset.meta.theta`, which is the angle indices (for two-dimensional filters). Both `fset.meta.k` and `fset.meta.theta` are of the same length as `fset.filter`.

The scale and angle indices are non-negative integers. The scale index rises with increasing scale, while the angle index rises with increasing angle (counterclockwise).

1.2 Morlet/Gabor filter bank

In addition to the parameters listed above, the Morlet/Gabor filter bank has the following options:

- `fparam.V`: The number of wavelets per octave.
- `fparam.nb_scales`: The number of wavelet scales.
- `fparam.sigma_psi`: The standard deviation of the mother wavelet in space.
- `fparam.sigma_phi`: The standard deviation of the scaling function in space.
- `fparam.slant`: The slant of the mother wavelet ellipse in frequency.
- `fparam.nb_theta`: The number of wavelet angles.

The maximal wavelet bandwidth (in space) is determined by $2^{nb\_scales/V}$. If `sigma_psi` is smaller than a certain threshold, a number of constant-bandwidth filters are added, linearly spaced, to cover the low frequencies.

2 Wavelet Modulus and Scattering Transforms

2.1 Wavelet Modulus Transform

The wavelet modulus transform takes a layer of coefficients and calculates the next. This layer has the fields:

- `layer.signal`: A cell array of signals.
- `layer.meta`: The meta information on the signals, such as their path, their resolutions, etc.

The signals are one-dimensional or two-dimensional arrays while `meta` contains the fields `meta.k` and `meta.theta`, which are two-dimensional arrays. The first dimension has length corresponding `layer.signal` while the second dimension as length corresponding to the order of the coefficients. The path of the lth coefficient is thus encoded in `meta.k(l,:)` and `meta.theta(l,:)`, respectively.

A wavelet modulus transform (of which there are multiple) is a function that takes a layer, a filter bank (see previous section), an options structure, and returns the next layer as well as the smoothed output of this layer. Specifically, for a wavelet modulus transform `wavemod`, we have:

```
[U{m+1},S{m}] = ...
    wavemod(U{m},options,filters);
```

Here, `U` and `S` are cell arrays of layers, as described above. This wavelet modulus can be a one-dimensional wavelet modulus transform, a two-dimensional wavelet modulus transform, a joint wavelet modulus transform, etc. It only has to satisfy the above input/output conditions.

2.2 Scattering Transform

By stacking multiple wavelet modulus transforms together, we obtain the scattering transform. Specifically, the `scatt` function, takes a signal, an options structure, a cell array of wavelet modulus transforms (with filters fixed), and returns the scattering coefficients (or wavelet modulus coefficients, if desired). The scattering coefficients are output as a cell array of layers `S`.

The scattering transform could be used like the following

```
fparam1.filter_type = 'gabor';
fparam1.V = 8;
fparam1.sigma_psi = 8;
fparam1.sigma_phi = 8;
fparam1.nb_scales = 80;

fparam2.filter_type = 'morlet';
fparam2.V = 1;
fparam2.sigma_psi = 1;
fparam2.sigma_phi = 1;
fparam2.nb_scales = 13;

filters1 = filter_bank(N,fparam1);
fitlers2 = filter_bank(N,fparam2);

wavemod{1} = ...
    @(x,opt)(wavemod_1d(x,opt,filters1));
wavemod{2} = ...
    @(x,opt)(wavemod_1d(x,opt,filters2));

S = scatt(x,options,wavemod);
```