Flexible Audio Source Separation Toolbox (FASST)
Version 1.0
User Guide

Alexey Ozerov [1], Emmanuel Vincent [1] and Frédéric Bimbot [2]

[1]INRIA, Centre de Rennes - Bretagne Atlantique
[2] IRISA, CNRS - UMR 6074
Campus de Beaulieu, 35042 Rennes cedex, France
{alexey.ozerov, emmanuel.vincent}@inria.fr, frederic.bimbot@irisa.fr

October 5, 2011

1 Introduction

This user guide describes how to use FASST, an implementation of the general flexible source separation framework presented in [1]. Before reading the user guide you are strongly encouraged to read [1], at least the two first sections.

This guide is organized as follows. Some notations and abbreviations used throughout this document are listed in section 2. Section 3 gives a detailed specification of the *mixture structure* (a Matlab structure), used to define the available prior information. The main functions the user should know about are listed in section 4 and an example of usage is given in section 5.

2 Some abbreviations and notations

2.1 Abbreviations

GMM	Gaussian mixture model
GSMM	Gaussian Scaled Mixture Model
HMM	Hidden Markov Model
NMF	Nonnegative matrix factorization
PSD	Power Spectral Density
QERB	Quadratic Equivalent Rectangular Bandwidth transform
S-HMM	Scaled Hidden Markov Model
STFT	Short-Time Fourier Transform

2.2 Notations

F	Number of frequency bins in the corresponding time-frequency representation
N	Number of time frames in the corresponding time-frequency representation
I	Number of channels (this version is only implemented for $I = 1$ or $I = 2$)
J_{spat}	Number of spatial components (see Section 3)
J_{spec}	Number of spectral components (see Section 3)
R_j	Rank of the covariance matrix of the j-th spatial component
C_j	Number of factors in the j-th spectral component – $C_j = 1$: direct model – $C_j = 2$: factored excitation-filter model
L_j^{ex}	Number of narrowband excitation spectral patterns (see [1]) in the j-th spec. comp.,
K_j^{ex}	Number of characteristic excitation spectral patterns (see [1]) in the j-th spec. comp.,
M_j^{ex}	Number of time-localized excitation patterns (see [1]) in the j-th spec. comp.,
L_j^{ft}	Number of narrowband filter spectral patterns (see [1]) in the j-th spec. comp.,
K_j^{ft}	Number of characteristic filter spectral patterns (see [1]) in the j-th spec. comp.,
M_j^{ft}	Number of time-localized filter patterns (see [1]) in the j-th spec. comp.,
$\mathbf{A}_j$	Mixing parameters ($\in \mathbb{C}^{I\times R_j \times F \times N}$) in the j-th spatial comp. (see [1]),
$\mathbf{W}_j^{\text{ex}}$	Narrowband excitation spectral patterns ($\in \mathbb{R}_+^{F\times L_j^{\text{ex}}}$) in the j-th spec. comp. (see [1]),
$\mathbf{U}_j^{\text{ex}}$	Excitation spectral pattern weights ($\in \mathbb{R}_+^{L_j^{\text{ex}}\times K_j^{\text{ex}}}$) in the j-th spec. comp. (see [1]),
$\mathbf{G}_j^{\text{ex}}$	Excitation time pattern weights ($\in \mathbb{R}_+^{K_j^{\text{ex}}\times M_j^{\text{ex}}}$) in the j-th spec. comp. (see [1]),
$\mathbf{H}_j^{\text{ex}}$	Time-localized excitation patterns ($\in \mathbb{R}_+^{M_j^{\text{ex}}\times N}$) in the j-th spec. comp. (see [1]),
$\mathbf{W}_j^{\text{ft}}$	Narrowband filter spectral patterns ($\in \mathbb{R}_+^{F\times L_j^{\text{ft}}}$) in the j-th spec. comp. (see [1]),
$\mathbf{U}_j^{\text{ft}}$	Filter spectral pattern weights ($\in \mathbb{R}_+^{L_j^{\text{ft}}\times K_j^{\text{ft}}}$) in the j-th spec. comp. (see [1]),
$\mathbf{G}_j^{\text{ft}}$	Filter time pattern weights ($\in \mathbb{R}_+^{K_j^{\text{ft}}\times M_j^{\text{ft}}}$) in the j-th spec. comp. (see [1]),
$\mathbf{H}_j^{\text{ft}}$	Time-localized filter patterns ($\in \mathbb{R}_+^{M_j^{\text{ft}}\times N}$) in the j-th spec. comp. (see [1]),
$\mathbb{R}$	Set of real numbers
$\mathbb{R}_+$	Set of nonnegative real numbers
$\mathbb{C}$	Set of complex numbers

3 Mixture structure

The mixture structure is a Matlab structure that is used to incorporate prior information into the framework. The structure has a hierarchical organization that can be seen from the example in figure 1. Global parameters (e.g., signal representation) are defined on the first level of the hierarchy. The second level consists of J_{spat} spatial components and J_{spec} spectral components. Each source is typically modeled by one spectral component, although some sources (e.g., drums) might be modeled by several spectral components (e.g., bass drum, snare, etc.). Furthermore, each spectral component must be associated with one spatial component, and each spatial component must have at least one spectral component associated to it. [1] Compared to the description of the framework in [1], this implementation is more general in the sense that the number of spectral components is

[1]This extension makes it possible to model the fact that several sources have the same direction, which is very often the case for professionally produced music recordings. It is implemented by simply adding the power spectrograms of the spectral components corresponding to the same spatial component.

not necessarily equal to that of spatial components, and more precisely $J_{\mathrm{spec}} \geq J_{\mathrm{spat}}$. The third level of the hierarchy consists in factorizing each spectral component into one or more *factors* representing for instance excitation and filter structures (see [1]) [2]. Finally, on the fourth level of the hierarchy, each factor is represented as the product of three or four matrices (see Table 5), which are not represented in figure 1. For instance, the factor representing excitation structure is either represented as the product of four matrices $\mathbf{W}_j^{\mathrm{ex}}\mathbf{U}_j^{\mathrm{ex}}\mathbf{G}_j^{\mathrm{ex}}\mathbf{H}_j^{\mathrm{ex}}$ representing, respectively, narrowband spectral patterns, spectral pattern weights, time pattern weights and time-localized patterns (see [1]) or as the product of threes matrices $\mathbf{W}_j^{\mathrm{ex}}\mathbf{U}_j^{\mathrm{ex}}\mathbf{G}_j^{\mathrm{ex}}$ when $\mathbf{H}_j^{\mathrm{ex}}$ is marked by the empty matrix `[]` [3]. Almost all the fields of the mixture structure must be filled as specified in Tables 2, 3, 4 and 5, except those marked by the empty matrix `[]`.

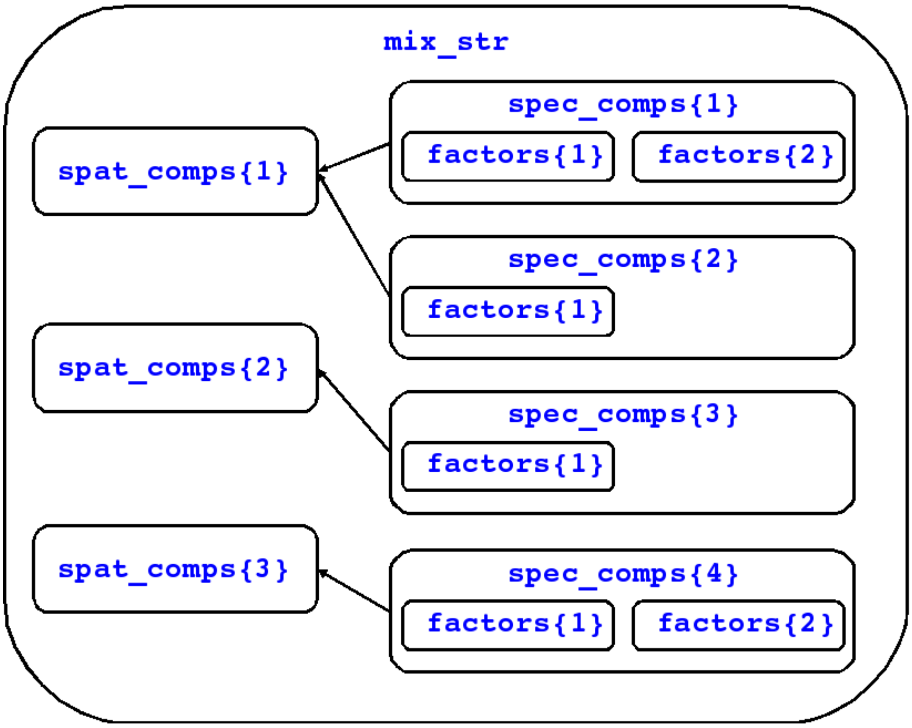

Figure 1: Visualization of a mixture structure example.

4 Main functions

The user should know about three main functions `comp_transf_Cx`, `estim_param_a_post_model` and `separate_spec_comps`, allowing, respectively, to compute the input time-frequency transform, estimate the model parameters and separate the spectral components. The headers of these functions are listed in Tables 6, 7 and 8.

[2]Note that in [1] the usage of two factors (excitation and filter) is described. The implementation presented here is more flexible, since one can use any number of factors C_j, and it reduces to [1] when $C_j = 2$. This is done for convenience of usage. For example if one needs to implement an excitation model only or a filter model only (*direct model*), one simply needs to choose $C_j = 1$ without bothering to specify and to process an additional dummy factor.

[3]In [1] only the case of four matrices is considered, and the case of three matrices $\mathbf{W}_j^{\mathrm{ex}}\mathbf{U}_j^{\mathrm{ex}}\mathbf{G}_j^{\mathrm{ex}}$ is just equivalent to fixing $\mathbf{H}_j^{\mathrm{ex}}$ to the $N \times N$ identity matrix. Since N may be quite big, we fix $\mathbf{H}_j^{\mathrm{ex}}$ to `[]` by convention in the latter case in order to avoid storing a big identity matrix in memory.

5 Examples of usage

The user should also know how to fill and browse the mixture structure and how to use the above-mentioned three functions. An example of mixture structure filling and browsing is given in Tables 9 and 10. An example script for the separation of an instantaneous mixture of music signals is given in Table 11.

Function `EXAMPLE_prof_rec_sep_drums_bass_melody.m` contains a more sophisticated example allowing the separation of the following four sources:

- drums,
- bass,
- melody (singing voice or leading melodic instrument),
- remaining sounds,

from a stereo music recording. Due to memory limits in Matlab this function cannot process sound excerpts longer than 30 seconds. For full length music recording the function
`EXAMPLE_prof_rec_sep_drums_bass_melody_FULL.m`
should be used. This function simply cuts the full recording into small parts, and applies
`EXAMPLE_prof_rec_sep_drums_bass_melody.m` to each of them.

References

[1] A. Ozerov, E. Vincent, and F. Bimbot, "A general flexible framework for the handling of prior information in audio source separation," *IEEE Transactions on Audio, Speech and Signal Processing*, to appear. [Online]. Available: http://hal.inria.fr/hal-00626962/

[2] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 3, pp. 550–563, March 2010.

Field	Description	Value
Cx	$F \times N \times I \times I$ complex-valued tensor of local mixture covariances	$\in \mathbb{C}^{F \times N \times I \times I}$
transf	Input time-frequency transform	'stft' for STFT
		'qerb' for QERB
fs	Sampling frequency in Hz	$\in \{16000, 44100, \ldots\}$
wlen	Analysis window length (used to compute STFT or QERB) in samples	$\in \{512, 1024, \ldots\}$
Noise_PSD	$F \times 1$ real-valued nonnegative vector of additive noise PSD, e.g., for annealing	$\in \mathbb{R}^{1 \times F}$ or []
spat_comps	$1 \times J_{\text{spat}}$ cell array of spatial component structures	see Table 3
spec_comps	$1 \times J_{\text{spec}}$ cell array of spectral component structures	see Table 4

Figure 2: Specification of the mixture structure (mix_str).

Field	Description	Value
time_dep	Stationarity of mixing	'indep' for time-invariant mixing
		'dep' for time-varying mixing
mix_type	Mixing type	'inst' for instantaneous (freq.-indep.)
		'conv' for convolutive (freq.-dep.)
frdm_prior	Degree of adaptability	'free' for adaptive
		'fixed' for fixed
params	Tensor of mixing parameters	$\in \mathbb{R}^{I \times R_j}$ for mix_type = 'inst'
	(corresponding to $\mathbf{A}_j$ from [1])	$\in \mathbb{C}^{I \times R_j \times F}$ for mix_type = 'conv'

Figure 3: Specification of the spatial component structure (spat_comps{j}, $j = 1, \ldots, J_{\text{spat}}$).

Field	Description	Value
spat_comp_ind	Index of the corresponding spatial component	$\in \{1, \ldots, J_{\text{spat}}\}$
factors	$1 \times L_j$ cell array of factor structures	

Figure 4: Specification of the spectral component structure (spec_comps{j}, $j = 1, \ldots, J_{\text{spec}}$).

Field	Description	Value
FB_frdm_prior	Degree of adaptability	'free' for adaptive
	for narrowband spectral patterns	'fixed' for fixed
FW_frdm_prior	Degree of adaptability	'free' for adaptive
	for spectral pattern weights	'fixed' for fixed
TW_frdm_prior	Degree of adaptability	'free' for adaptive
	for time pattern weights	'fixed' for fixed
TB_frdm_prior	Degree of adaptability	'free' for adaptive
	for time-localized patterns	'fixed' for fixed
FB	Narrowband spectral patterns (Frequency Blobs) (corresponding to $\mathbf{W}_j^{\mathrm{ex}}$ or $\mathbf{W}_j^{\mathrm{ft}}$)	$\in \mathbb{R}_+^{F \times L_j^{\mathrm{ex}}}$ or $\in \mathbb{R}_+^{F \times L_j^{\mathrm{ft}}}$
FW	Spectral pattern weights (Frequency Weights) (corresponding to $\mathbf{U}_j^{\mathrm{ex}}$ or $\mathbf{U}_j^{\mathrm{ft}}$)	$\in \mathbb{R}_+^{L_j^{\mathrm{ex}} \times K_j^{\mathrm{ex}}}$ or $\in \mathbb{R}_+^{L_j^{\mathrm{ft}} \times K_j^{\mathrm{ft}}}$
TW	Time pattern weights (Time Weights) (corresponding to $\mathbf{G}_j^{\mathrm{ex}}$ or $\mathbf{G}_j^{\mathrm{ft}}$)	$\in \mathbb{R}_+^{K_j^{\mathrm{ex}} \times M_j^{\mathrm{ex}}}$ or $\in \mathbb{R}_+^{K_j^{\mathrm{ft}} \times M_j^{\mathrm{ft}}}$
TB	Time-localized patterns (Time Blobs) (corresponding to $\mathbf{H}_j^{\mathrm{ex}}$ or $\mathbf{H}_j^{\mathrm{ft}}$)	$\in \mathbb{R}_+^{M_j^{\mathrm{ex}} \times N}$, $\in \mathbb{R}_+^{M_j^{\mathrm{ft}} \times N}$ or []
TW_constr	Constraint on the time pattern weights	'NMF' no constraint
	(note that nontrivial constraints, i.e.,	'GMM' for GMM
	different from 'NMF' are not	'HMM' for HMM
	compatible with nonempty time patterns TB)	'GSMM' for GSMM
		'SHMM' for S-HMM
TW_DP_params	Discrete probability (DP) parameters for the time pattern weights (needed only when TW_constr $\neq$ 'NMF')	$1 \times K_j^{\mathrm{ex}}$ ($1 \times K_j^{\mathrm{ft}}$) vector of Gaussian weights for GMM or GSMM
		$K_j^{\mathrm{ex}} \times K_j^{\mathrm{ex}}$ ($K_j^{\mathrm{ft}} \times K_j^{\mathrm{ft}}$) matrix of transition probabilities for HMM or S-HMM
TW_DP_frdm_prior	Degree of adaptability for DP parameters	'free' for adaptive
	(needed only when TW_constr $\neq$ 'NMF')	'fixed' for fixed
TW_all	Matrix of all time weights (corresponding to $\tilde{\mathbf{G}}_j^{\mathrm{ex}}$ or $\tilde{\mathbf{G}}_j^{\mathrm{ft}}$ from [1]) (needed only when TW_constr $\neq$ 'NMF')	Nonnegative real-valued matrix of the same size as TW

Figure 5: Specification of the spectral component factor structure (factors{l}, $l = 1, \ldots, L_j$).

```
function Cx = comp_transf_Cx(x, transf, win_len, fs, qerb_nbin)

%
% Cx = comp_transf_Cx(x, transf, win_len, fs, qerb_nbin);
%
% compute spatial covariance matrices for the corresponding transform
%
%
% input
% -----
%
% x                   : [I x nsampl] matrix containing I time-domain mixture signals
%                       with nsampl samples
% transf              : transform
%                         'stft'
%                         'qerb'
% win_len             : window length
% fs                  : (opt) sampling frequency (Hz)
% qerb_nbin           : (opt) number of bins for qerb transform
%
% output
% ------
%
% Cx                  : [F x N x I x I] matrix containing the spatial covariance
%                       matrices of the input signal in all time-frequency bins
%
```

Figure 6: `comp_transf_Cx` : FASST function for the computation of the input time-frequency transform.

```
function [mix_str, log_like_arr] = estim_param_a_post_model(mix_str_inp, ...
    iter_num, sim_ann_opt, Ann_PSD_beg, Ann_PSD_end)

%
% [mix_str, log_like_arr] = estim_param_a_post_model(mix_str_inp, ...
%     iter_num, sim_ann_opt, Ann_PSD_beg, Ann_PSD_end);
%
% estimate a posteriori mixture model parameters
%
%
% input
% -----
%
% mix_str_inp         : input mixture structure
% iter_num            : (opt) number of EM iterations (def = 100)
% sim_ann_opt         : (opt) simulated annealing option (def = 'ann')
%                          'no_ann'      : no annealing (zero noise)
%                          'ann'         : annealing
%                          'ann_ns_inj'  : annealing with noise injection
%                          'upd_ns_prm'  : update noise parameters
%                                          (Noise_PSD is updated through EM)
% Ann_PSD_beg         : (opt) [F x 1] beginning vector of annealing noise PSD
%                             (def = X_power / 100)
% Ann_PSD_end         : (opt) [F x 1] end vector of annealing noise PSD
%                             (def = X_power / 10000)
%
%
% output
% ------
%
% mix_str             : estimated output mixture structure
% log_like_arr        : array of log-likelihoods
%
```

Figure 7: `estim_param_a_post_model` : FASST function for the estimation of the model parameters.

```
function ie = separate_spec_comps(x, mix_str, sep_cmp_inds)

%
% ie = separate_spec_comps(x, mix_str, sep_cmp_inds);
%
% separate spectral components
%
%
% input
% -----
%
% x                 : [nchan x nsampl] mixture signal
% mix_str           : input mix structure
% sep_cmp_inds      : (opt) array of indices for components to separate
%                     (def = {1, 2, ..., K_spec})
%
%
% output
% ------
%
% ie                : [K_sep x nsampl x nchan] estimated spectral components images,
%                     where K_sep = length(sep_cmp_inds) is the number of
%                     components to separate
%
```

Figure 8: separate_spec_comps : FASST function for the separation of the spectral component signals.

```
function mix_str = init_mix_struct_Mult_NMF_inst(Cx, J, K, transf, fs, wlen)

%
% mix_str = init_mix_struct_Mult_NMF_inst(Cx, J, K, transf, fs, wlen);
%
% An example of mixture structure initialization, corresponding to
% multichannel NMF model (instantaneous case)
% Most of parameters are initialized randomly
%
% input
% -----
%
% Cx              : [F x N x I x I] matrix containing the spatial covariance
%                   matrices of the input signal in all time-frequency bins
%                   or [F x N] single channel variance matrix
% J               : number of components (here J_spat = J_spec)
% K               : number of NMF components per source
% transf          : transform ('stft' or 'qerb')
% fs              : sampling frequency in Hz
% wlen            : length of the time integration window (must be a power of 2)
%
% output
% ------
%
% mix_str         : initialized mixture structure
%

rank = 1;

[F, N, I, I] = size(Cx);

mix_str.Cx         = Cx;
mix_str.transf     = transf;
mix_str.fs         = fs;
mix_str.wlen       = wlen;
mix_str.spat_comps = cell(1,J);
mix_str.spec_comps = cell(1,J);

for j = 1:J
    % initialize spatial component
    mix_str.spat_comps{j}.time_dep   = 'indep';
    mix_str.spat_comps{j}.mix_type   = 'inst';
    mix_str.spat_comps{j}.frdm_prior = 'free';
    mix_str.spat_comps{j}.params     = randn(I, rank);

    % initialize single factor spectral component
    mix_str.spec_comps{j}.spat_comp_ind = j;
    mix_str.spec_comps{j}.factors       = cell(1, 1);

    factor1.FB            = 0.75 * abs(randn(F, K)) + 0.25 * ones(F, K);
    factor1.FW            = diag(ones(1, K));
    factor1.TW            = 0.75 * abs(randn(K, N)) + 0.25 * ones(K, N);
    factor1.TB            = [];
    factor1.FB_frdm_prior = 'free';
    factor1.FW_frdm_prior = 'fixed';
    factor1.TW_frdm_prior = 'free';
    factor1.TB_frdm_prior = [];
    factor1.TW_constr     = 'NMF';

    mix_str.spec_comps{j}.factors{1} = factor1;
end;
```

Figure 9: Example of filling of the mixture structure corresponding to the multichannel NMF method [2] (instantaneous case).

```
>> mix_str

mix_str =

            Cx: [4-D double]
        transf: 'stft'
            fs: 16000
          wlen: 1024
    spat_comps: {[1x1 struct]  [1x1 struct]  [1x1 struct]}
    spec_comps: {[1x1 struct]  [1x1 struct]  [1x1 struct]}
     Noise_PSD: [513x1 double]

>> mix_str.spat_comps{2}

ans =

      time_dep: 'indep'
      mix_type: 'inst'
    frdm_prior: 'free'
        params: [2x1 double]

>> mix_str.spec_comps{3}

ans =

    spat_comp_ind: 3
          factors: {[1x1 struct]}

>> mix_str.spec_comps{3}.factors{1}

ans =

               FB: [513x4 double]
               FW: [4x4 double]
               TW: [4x98 double]
               TB: []
      FB_frdm_prior: 'free'
      FW_frdm_prior: 'fixed'
      TW_frdm_prior: 'free'
      TB_frdm_prior: []
        TW_constr: 'NMF'
```

Figure 10: Browsing in Matlab of the example mixture structure in Table 9.

```
function EXAMPLE_ssep_Mult_NMF_inst()

data_dir    = 'example_data/';
result_dir  = 'example_data/';
file_prefix = 'Shannon_Hurley__Sunrise__inst_';

transf     = 'stft';
wlen       = 1024;
nsrc       = 3;      % number of sources
NMF_ncomp  = 4;      % number of NMF components
iter_num   = 200;

% load mixture
fprintf('Input time-frequency representation\n');
[x, fs, nbins]=wavread([data_dir file_prefix '_mix.wav']);
x = x.';
mix_nsamp = size(x,2);

% compute time-frequency representation
Cx = comp_transf_Cx(x, transf, wlen, fs);

% fill in mixture structure
mix_str = init_mix_struct_Mult_NMF_inst(Cx, nsrc, NMF_ncomp, transf, fs, wlen);

% reinitialize mixing parameters
A = [sin(pi/8), sin(pi/4), sin(3*pi/8); cos(pi/8), cos(pi/4), cos(3*pi/8)];
for j = 1:nsrc
    mix_str.spat_comps{j}.params = A(:,j);
end;

% run parameters estimation (with simulated annealing)
mix_str = estim_param_a_post_model(mix_str, iter_num, 'ann');

% source separation
ie_EM = separate_spat_comps(x, mix_str);

% Computation of the spatial source images
fprintf('Computation of the spatial source images\n');
for j=1:nsrc,
    wavwrite(reshape(ie_EM(j,:,:),mix_nsamp,2),fs,nbins, ...
        [result_dir file_prefix '_sim_' int2str(j) '.wav']);
end
```

Figure 11: Example of usage involving all three main functions (runs the multichannel NMF method [2] in the instantaneous case).