

Image-source method: Matlab code implementation

Summary and disclaimer

The Matlab files on this page provide an implementation of the image-source method (ISM) described in [1] for the purpose of simulating reverberant audio data in small-room acoustics. Please refer to [1] for more detailed information on this ISM implementation. General information can also be found in the [Background](#) section of this web site. The next section below provides a typical example of how to use the different Matlab functions provided on this page. Note that these files can also be downloaded from the [MathWorks](#) website.

As per usual, simply download and use these files as you like, but remember that all material available on this page is subject to the [GNU general public license](#) and comes with NO WARRANTY whatsoever! Please don't hesitate to [contact me](#) if you find any bug or if you have interesting suggestions to make.

And of course, an acknowledgement to [1] will also be deeply appreciated if you happen to publish results obtained with the help of the code provided on this page. Thanks! ;)

Image-source code

The Matlab functions provided in the table below make it very easy and straightforward to generate samples of reverberant audio data for a source moving across a given environment, using ISM simulations. You can refer to each function's Matlab help section for detailed usage information, but in short, the basic process is as follows.

Typical Matlab usage:

1. Create a setup file for the simulation, e.g.: `my_ISM_setup.m` (similar to the example [ISM_setup.m](#) in the table below)
2. Create a 1D vector of source signal data, e.g.: `SrcSignalVec = wavread('speech.wav')`
3. Simply execute the following two Matlab commands:

```
>> ISM_RIR_bank(my_ISM_setup, 'ISM_RIRs.mat');  
>> AuData = ISM_AudioData('ISM_RIRs.mat', SrcSignalVec);
```

The first command `ISM_RIR_bank(...)` will create and save a bank of RIRs into the file `ISM_RIRs.mat`. This process can take quite a while depending on the number of RIRs to compute, size of the room, reverberation time, etc. The second command `ISM_AudioData(...)` takes the `.mat` file and the source signal vector as inputs, and computes the sensor data `AuData` by convolution (overlap-add). This function also offers the possibility to write the resulting audio into a `.wav` or `.mat` file. The resulting audio data is a matrix where each column contains the signal generated for the corresponding microphone. Of course, the same process can also be used if the source remains stationary and/or in environments containing only one sensor: simply define the simulation parameters accordingly in the setup file `my_ISM_setup.m`!

The following samples of audio data were generated using the above commands for a source moving along a [circle trajectory](#) around a couple of microphones, in a 4m x 5m x 2.7m room: [.wav sample](#) (570kB) with $T_{60} \approx 0.3s$, and [.wav sample](#) (583kB) with $T_{60} \approx 0.6s$ (both files had white noise added with an SNR level of about 35dB). These are stereo signals, with each channel containing the signal recorded by one microphone, so try listening to them with headphones.

If desired, the function `ISM_RT_check.m` can also be used in order to obtain precise information

about the exact reverberation level in the considered environment. However, because the simulation files below make use of the reverberation time (RT) prediction method described in [1] and [2], the room's resulting RT will in general match quite closely the desired value of reverberation time defined in the setup file `my_ISM_setup.m`. An assessment of the accuracy of this RT prediction method can be found in [1,2], and is also demonstrated [here](#) for a range of simulation environments (room sizes, absorption coefficient ratios, etc.).

Download all the files below in a .zip file: ISM.zip	
ISM_setup.m	Edit this file to define the various environmental parameters of the desired image method simulation. These include parameters such as the sampling frequency, room dimensions, desired reverberation time (T_{60} or T_{20}), positions of the acoustic sensor (or an array of them), location of a sound source (or a series of trajectory points for a sound source moving across the environment), etc.
ISM_RoomResp.m	This function computes the RIR between a source and a receiver in a given simulation setup, using the image-source method described in [1]. This implementation generates fractional delays for each image source. This Matlab function was specifically optimised for execution speed by only considering the image sources relevant to the final transfer function, so the computation time will be as minimal as possible. An example RIR obtained with this implementation can be found on this page (Fig. 3).
ISM_RIR_bank.m	Computes a bank of RIRs corresponding to a simulation setup defined in a file such as <code>ISM_setup.m</code> (see above). One impulse response is computed for every possible combination of the sensors and source trajectory points.
ISM_AudioData.m	Generates samples of audio data (one sample per sensor channel) using the bank of impulse responses pre-computed with <code>ISM_RIR_bank.m</code> . This function takes the source signal as input argument, and allows to set the source direction, disable/enable file saving, and disable/enable additive noise (see Matlab help for details).
ISM_RIR_DecayTime.m	For a given environmental setup, this function provides an estimate (prediction) of the time required by a RIR's energy to decay by a given factor (in dB). This function is used in <code>ISM_RoomResp.m</code> and is based on the energy decay approximation method of [1].
ISM_AbsCoeff.m	Also based on the research proposed in [1], this Matlab function can be used to predict the absorption coefficient values required with a specific environmental setup in order to achieve a desired level of reverberation. This function can be used in order to generate an input argument to <code>ISM_RoomResp.m</code> , and is required by the functions <code>ISM_RIR_bank.m</code> .
ISM_RIRpow_approx.m	This function computes the predicted power values for ISM-simulated RIRs for a given environmental setup, estimated by means of the EDC approximation method in [1]. It is used by the functions <code>ISM_RIR_DecayTime.m</code> and <code>ISM_AbsCoeff.m</code> .
ISM_RT_check.m	This function can be used to perform a statistical analysis of the reverberation characteristics of a simulated environment, defined in a file such as <code>ISM_setup.m</code> (see above). This analysis is carried out by generating a number of random impulse responses in the desired environment (potentially very time-consuming!), and computing the reverberation time from the resulting RIRs. This function is purely informative and is not required in the overall RIR simulation process implemented in the previous files. It is typically used to gain some insight into the "true" reverberation time resulting in the considered room, which should match the desired reverberation time defined in

Miscellaneous files

Below is a list of various custom functions that may be required by the Matlab files described above. See the files' help section for specific usage information. Also, please [let me know](#) if a function is missing from the list below.

Download all the files below in a .zip file: ISM_misc.zip	
PrintLoopPCw.m	Prints the execution percentage on screen in a FOR or WHILE loop.
SetUserVars.m	Checks and initialises some input variables from a set of user-definable parameters. To be used for functions that accept a series of ARGNAME—ARGVAL pairs to set user-definable parameters.
reusefig.m	Creates or re-uses a figure with specific tag, regardless of the currently active window. When defined in a Matlab function, this command re-uses the same figure between consecutive runs and hence avoids plotting in an unwanted window or creating a new figure every time the function is executed.
freq_conv.m	Performs the convolution of two vectors in the frequency domain. This provides significant savings in computation time compared to the time-domain convolution, as implemented in Matlab's <code>conv</code> function.

List of changes

Check the table below to see which of the files above have been updated and to make sure that you have the latest version available.

Date	Summary of changes
10 Mar. 2012	Minor updates to <code>ISM_RoomResp.m</code> and <code>ISM_RIR_DecayTime.m</code> so as to better handle anechoic environments (thanks to comments from Maryam Naghibolhosseini). Anechoic responses can now be achieved by setting either <code>RT_VAL=0</code> or <code>BETA=[0 0 0 0 0 0]</code> (or both).
28 Nov. 2009	Files updated! The function <code>ISM_RIR_DecayTime.m</code> was updated to allow for a greater accuracy in the predicted decay time results. Several other function were modified as a result of this update, including <code>ISM_RoomResp.m</code> , <code>ISM_RT_check.m</code> , <code>ISM_AbsCoeff.m</code> and <code>ISM_RIR_bank.m</code> . The function <code>ISM_RIRpow_approx.m</code> was also added to the list of required Matlab functions.
08 Aug. 2008	Small bug fixed in <code>ISM_AudioData.m</code> (incorrect number of frames with 'TrajDir' options 'SES' and 'ESE'). Thanks to Benedikt L��sch for spotting this!

References

- [1] E. Lehmann and A. Johansson, *Prediction of energy decay in room impulse responses simulated with an image-source model*, Journal of the Acoustical Society of America, vol. 124(1), pp. 269-277, July 2008.
- [2] E. Lehmann, A. Johansson, and S. Nordholm, *Reverberation-time prediction method for room impulse responses simulated with the image-source model*, Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'07), pp. 159-162, New Paltz, NY, USA, October 2007.