Guidelines of Using SLD Functions

LEE, Kwan Kit

October 17, 2021

Contents

1 Introduction

This document gives the guidelines of using the MATLAB function *measurement.m* to 1).simulate the photon reception process; 2).perform adaptation measurement to estimate the brightness and location of the point sources. Fig. 1 is the flow chart of the algorithm

The details of each part (the yellow arrows in Fig. 1) will be discussed in the following sections. For a single scene, one can generate many trials to obtain results for error analysis.

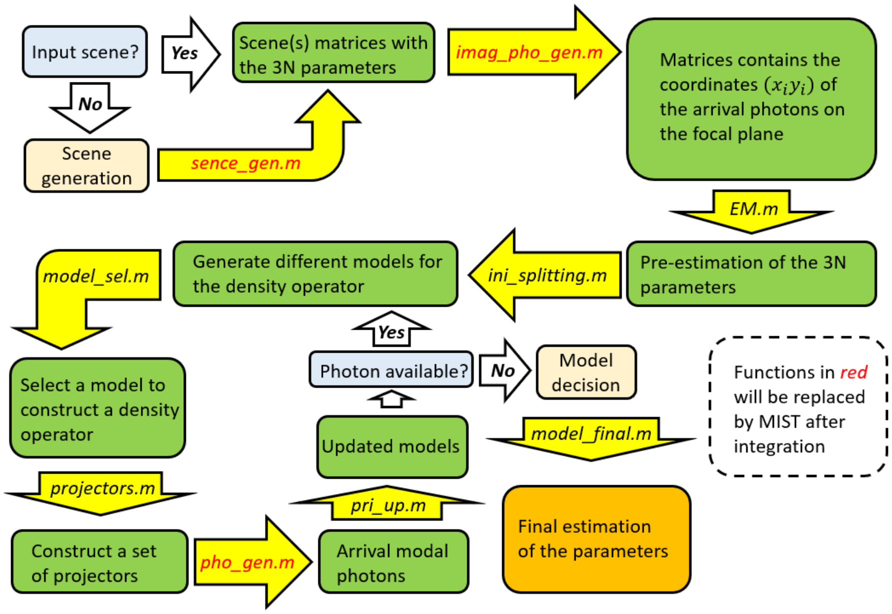

Figure 1: The flow of the algorithm.

2 Single Photon Adaptation

2.1 *measurement.m*

2.1.1 Description

This is the main part of the programme. Once it is called, it goes through the whole process in Fig. 1. The following are the input and output of *measurement.m*. The average number of arrival photons are defined by the user and iteration stops when the photons run out.

2.1.2 Input

Optional parameters:

- *seed*
  An integer used as the seed of random generator. The default value is 0.
- *n_exp*
  An integer specifying the number of trials for a single scene. The default value is 1.
- *n_imag_mu*
  An integer specifying the average number of photon arrived in the pre-estimation (direct detection) precess. The default value is 500.

- *n_pho_SLD*
  An integer specifying the expected number of photon arrived in the modal projection precess. The default value is 9500.

- *scene*
  A $N \times 3$ matrix specifying the parameters of the ground truth. N has to be smaller than N_{max}. If it is empty, a scene will be generated with N close to N_{max}. The default value is [ ] and $N_{max} = 10$.

- *use_SLD*
  A value determines where the modal projection measurement is used. If logical(use_SLD) = 1, the modal measurement will be turned on; otherwise only direct detection will be performed. The default value is 1.

2.1.3 Output

The default output is a file named "*Estimation_xxx.m*" where *xxx* is the seed number. In the file, one can find:

- *scene*
  Same as the scene in the input.

- *cand*
  A cell contains all a prediction for all candidates in every trials. Only exists if the modal measurement is turned on.

- *SLD_est*
  A cell contains the chosen candidate in every trials (i.e. the final result). Only exists if the modal measurement is turned on.

- *tp*
  A matrix contains the initial guess of he parameters. Only exists if the modal measurement is turned on.

- *imag_est*
  A matrix contains the estimation given by direct detection followed by EM algorithm. Only exists if the modal measurement is turned off.

2.2 *scene_gen.m*

If the input of 'scene' in measurement is null, a scene with number of point sources less than N_{max} is generated. The point sources are specified by their relative brightness and the coordinates on the image plane. In other words, this scene is the ground truth of the parameters. This part will be replaced after integration with MIST.

2.3 *imag_pho_gen.m*

By knowing the scene parameters and the point spread function, the intensity profile on the focal plane can be constructed under diffraction limit. The function *imag_pho_gen.m* will draw n_{imag} photons according to the intensity profile. At the moment we assume that the field of view captures (almost) all the intensity. The arrival photons are specified by the coordinates on the focal plane. This part will be replaced after integration with MIST.

3 *EM.m*

With the photon samples given by *imag_pho_gen.m*, a pre-estimation of the parameters is found by the expectation maximization (EM) algorithm.

3.1 *ini_splitting.m*

Using the pre-estimation from the EM algorithm, density operator candidates are constructed by *ini_splitting.m*. It also gives the prior distribution of the parameters.

3.2 *model_sel.m*

A density operator is selected from the candidates given by *ini_splitting.m*. The criterion using now is to select the candidate with highest likelihood.

3.3 *pho_gen.m*

A measurement is performed using the modal projection selected. *ini_splitting.m*. The criterion using now is to select the candidate with highest likelihood. This part will be replaced after integration with MIST.

3.4 *pri_up.m*

The parameters for the posterior is calculated and it is used as the prior for the next iteration.

3.5 *model_final.m*

The points which are close to each other are merged and the final candidate is selected. The criterion using now is to select the candidate with highest likelihood.

3.6 Example

The scene shown in the second milestone report is saved as "*scene.mat*". To get the estimation of the parameters one can run the script "*script_run.m*". To plot the intensity of the scene and visualize the estimations, one can run the script "*script_plot.m*".

4 Multi-Photon Adaptation

For multi-photon adaptation, call *measurement_multipho.m*. The only required input is the *scene* in the format stated about.

4.0.1 Input

There are some now optional parameters:

- *n_max*
  The maximum number of the sources. It was a global parameter and now it is an optional parameter.

- *n_HG_modes*
  The label of the HG modes which determine the dimension space. For example, if *n_HG_modes* = 1, only (0,0) mode is considered; *n_HG_modes* = 2, (0,0), (1,0), (0,1) modes are considered; *n_HG_modes* = 3, (0,0), (1,0), (0,1), (2,0), (1,1), (0,2) modes are considered, so forth and so on.

- *n_pho_group*
  An integer specifying the average number of photon collected in each iteration cycle.

- *n_pri*
  A boolean parameter determine whether the number of sources is known. If *n_pri* = 1, *n_max* has to be the exact number of sources.

- *n_sam*
  The number of samples for numerical integration.

- *bri_known*
  A boolean parameter determine whether the even source prior is known.

4.0.2 Output

The default output is a file named "*SLD_Est_xxx.m*" and "*Imag_Est_xxx.m*", for *use_SLD* = 1 and 0 respectively, where *xxx* is the seed number. The outputs are similar to those in the single photon adaptation.

4.1 Example

Please run *script_multiphoton_run* as an example. To visualize the result, one may use the function *plot_now*(*scene*, *estimation*).