Definitions of Parameters

Intelligent Robotics Laboratory
University of Birmingham

Version 1.0

Contents

1 Introduction

In this document, parameters of the processes and the functions implemented in Library of Hierarchy of Parts (LibHOP) are explained. Then, their employment in a sample configuration file is given.

2 Parameters of Learning and Inference Processes used in LibHOP

2.1 Inference using LibHOP

2.1.1 Inference at Layer L=1

At the first layer L=1, Gabor features are extracted from the images and the graph structure is initialized using `hop1create`. Its usage is as follows;

```
hop1create config_file file_pattern "parameter1=value1;parameter2=value2;..."
```

Note: Parameter values specified in the command line prevail over those specified in the configuration file `config_file`.

Definitions of the parameters are given below:

- **type (string, default value: struct)**: Filter types.
  - struct: Structure of 5×5 kernel,
  - app: Polar filters.
- **out_dir (string, default value: empty)**: Destination directory (for new files);
  - Empty string: Current directory.
- **src_dir (string, default value: empty)**: Source directory;
  - Empty string: Nothing will be appended to the `file_pattern` parameter.
- **part_lib_name (string, default value: empty)**: If the value is non-empty, then the first layer part library is saved to a file with this value.
- **layer1_threshold (double, default value: 0.1)**: Only nodes with response values that are greater than or equal to the value of **layer1_threshold** are added to the list.
- **scale_sigma (double, default value: 0.0)**: Sigma of the initial Gaussian filter used for smoothing the image;
  - Default value 0.0: The image will not be blurred with a Gaussian filter.
- **scale_factor (double, default value: 0.8)**: Quotient of the sizes of the images considered at two consecutive layers of the LibHOP and must be less than 1.0.

- **init_size (integer, default value: -100)**: Initial size of the image that will be processed at the first layer of the LibHOP, i.e. original image will be resized using this value.
  - If this value is negative, then the image is resized to (-init_size/100)-times its size.
- **scale_limit (integer, default value: 200)**: Minimum size of image at the smallest scale (Note that one scale is always saved).
- **border_size (integer, default value: 0)**: Size of the border around the picture. The recommended value is 100.
- **response_percent (double, default value: 0.8)**: The threshold of a Gabor filter at a point relative to the maximum of the Gabor response computed at that point.
- **normalization_percent (double, default value: 0.0, range: [0, 1])**: Normalizes the top normalization_percent responses to the values defined in the parameter range,
  - If **normalization_percent** = 0.2, then the best 20% of (best) responses are normalized to 1.0.
- **split (int, default value: 1)**: Splits the input image into subimages with size $split \times split$, and the value of the overlap between subimages is **split_overlap**.
- **split_overlap (int, default value: 100)**: Overlap between subimages when the value of the split is greater than 1.
- **separate_colors (boolean, default value: false)**: Seperates colored (RGB) image into R, G, B bands and process each of them (instead of using grayscale as by default). Then, all parts computed at different bands are combined into one result.

2.1.2 Inference at a Layer L>1

Inference algorithms for layers L > 1 are implemented using `hopncreate`. Its usage is as follows;

```
hopncreate config_file file_pattern "parameter1=value1;parameter2=value2;..."
```

Note: Parameter values specified in the command line prevail over those specified in the configuration file `config_file`.

The definitions of the processes called during inference and their parameters are given below:

- **part_lib_name (string, mandatory)**: Name of the file in which the part library will be stored.
- **result_extension (string, default value: .lyx)**: The extension of the name of the file that contains the graph and part structures computed at the layers $L = 1, 2, \ldots, x$. For instance, .ly2 is used as an extension for a file that contains the structures computed at the layers 1 and 2.

- **out_dir (string, default value: empty)**: Destination directory (for new files);
  - Empty string: Current directory.
- **src_dir (string, default value: empty)**: Source directory; empty string means that nothing is appended to the **file_pattern** program parameter.
- **reconstruction_type (integer, default value: 1)**: Type of inference algorithm. Current types are:
  - 1: default algorithm,
  - 9: Should be used for object layer only.
- **layer_index (integer, mandatory)**: Layer that will be inferred. Note that it uses only the nodes which are not yet covered.
  - `layer_index (integer, mandatory)`= 2 is used to construct parts and compositions at the layer L=2 (which corresponds to the first layer in C implementation).
- **layer_contraction (integer, default value: 2.0)**: Contraction value which is employed after a part is inserted as a node to the graph structure.
  - default value: 2.0 divides coordinates of newly inserted nodes by a factor of 2.0.
- **candidate_r_threshold (double, default value: 0.2)**: Only subparts whose r-responses are greater than or equal to **candidate_r_threshold** are considered as central parts in the inference of a part.
- **candidate_g_threshold (double, default value: 0.2)**: Only subparts whose g-responses are greater than or equal to **candidate_g_threshold** are considered as central parts in the inference of a part.
- **candidate_r_threshold_percent, candidate_g_threshold_percent (double, default value: NaN)**: If the value is not NaN (not-a-number, i.e. undefined), then
  - **candidate_r_threshold** is set to the value of **candidate_r_threshold_vf** $*$ **the best r-response** , or
  - **candidate_g_threshold** is set to the value of **candidate_g_threshold_vf** $*$ **the best g-response**.
- **convolution_threshold (double, default value: 0.1)**: Only subparts which are convolved with the map and have the magnitudes that are greater than or equal to the value of **convolution_threshold** are used.
- **convolution_link_threshold (double, default value: 0.0)**: Links to subpart are added only if its response value is greater than or equal to the value of **convolution_link_threshold** * **best response**.

- **r_response_threshold (double, default value: 0.3)**: Threshold for r_response (it is used if an r-threshold is not learned using the training data for a particular part).

- **g_response_threshold (double, default value: 0.0)**: Threshold for g_response (it is used if an g-threshold is not learned using the training data for a particular part).

- **r_response_pow (double, default value: 1.0)**: Power correction to the parts.

- **map_product_response_power (double, default value: 1.0)**: The value of the power factor in the exponent when making product with a map. This parameter is used while learning the parameters of distributions of g-responses.

- **realization_ratio_threshold (double, default value: 0.5)**: Threshold value which is used for the construction of subpart realizations. The number of subpart realizations is determined by computing the ratio of the number of realized subparts without center to the total number of subparts without center. The part realizations are constructed until this ratio is less than the parameter value. If the library contains learned thresholds (RR_THRESH) for a particular type, then the learned threshold is used.

- **min_factor (double, default value: 1.0)**: The minimum factor value used for *stretching* the object model. In the inference process the object positions of parts relative to the center are multiplied by factors between **min_factor** and **max_factor** and the best result is used. It is implemented if **reconstruction_type** = 9 is used.

- **max_factor (double, default value: 1.0)**: See **min_factor** above.

- **type_threshold (double, default value: 1.0)**: Threshold used for computing part similarity. This works only if **reconstruction_type** = 1 and the library contains similar edges. If there are no similar edges, then each part is only similar to itself with a similarity value 1.0.

- **ignore_texture (bool, default value: false)**, **texture_radius (integer, default value: 4)** and **texture_parts (integer, default value: 4)**: If **ignore_textue** = true and there are more than **texture_parts** number of different parts in the neighborhood of some part p with radius **texture_radius'**, then p can not be used in the inference process. It is used if **reconstruction_type** = 1.

2.2 Learning and Optimization using LibHOP

Learning algorithms are implemented using `hoplearning`. Its usage is as follows

```
hoplearning config_file file_pattern "parameter1=value1;parameter2=value2;..."
```

Note: Parameter values specified in the command line prevail over those specified in the configuration file `config_file`.

2.2.1 Step 1: Map Learning

Parameters that are used in Map Learning are given below:

- **General parameters**:
  1. **action = update**: Invokes learning of maps.
- **Parameters for input/output**:
  1. **src_dir** (string, default value: empty): Source directory for *.lyX files.
  2. **nbhoods_file (string, mandatory)**: Result of the map update process.
  3. **save_nbhoods (bool, default value: false)**: To determine saving maps or not.
  4. **image_name (string, default value: c:_%d%d.bmp)**: printf string pattern for the names of map image files. The first %d is filled with the index of center part, and the second %d with the index of "other" part
  5. **matrix_name (string, default value: empty)**: printf string pattern for the names of "matrix" files (in Mathematica format).
  6. **part_lib_name (string, mandatory)**: Name of the files that contain part libraries.
- **Parameters for thresholds**:
  1. **region_intersection_threshold (double, default value: 0.3)**: Intersection between the part and the central element must be less than or equal to the value of **region_intersection_threshold**.

2.2.2 Step 2: Part Learning

Parameters that are used in Part Learning are given below:

- **General parameters**
  1. **action = learn_parts**: Invokes learning of parts.
- **Parameters for input/output**:
  1. **src_dir (string, default value: empty)**:Source directory for *.lyX files.
  2. **nbhoods_file (string, mandatory)**: Result of the part learning process.
  3. **save_maxima (bool, default value: false)**: To decide whether to save "maximal" values of parts or not.
  4. **image_name (string, default value: c:_%d%d.bmp)**: printf string pattern for the names of maxima image files. The first %d is filled with the index of center part, and the second %d with the index of "other" part.

5. **library_image_file (string, default value: empty)**: Name of the image file with all parts computed at the layer L. If string is empty (default value) nothing is saved.

   Note that if a large number of parts is produced (and hence the image is large), then the export may take some time.

6. **show_labels (bool, default value: false)**: Show labels of parts (their index in the library) when saving parts, or not (see
7. **library_image_file**).
8. **save_library (string, default value: empty)**: File name of the output library. It is not saved if the value is empty string.

- **Other parameters**:

  1. **nb_size (odd integer, default value: 17)**: Dimension of the neighborhood square used in the update process.
  2. **min_part_length (integer, default value: 2)**: Minimal size of the part (including center).
  3. **max_part_length (integer, default value: 3)**: Maximal size of the part (including center).
  4. **layer_contraction (double, mandatory)**: Contraction of the previous layer;
     - It is taken as 1.0 when learning 2nd layer, as 2.0 when learning 3rd layer, etc.
  5. **library_indices (vector of integers, default value: empty)**: Indices of parts which should be added to the library;
     - Empty vector adds first **part_max_number** parts.
  6. **learn_g_distributions (bool, default value: true)**: Learning parameters of normal distribution of g-responses.
     - Important: Inference parameters should be used in namespace **g_distribution** (i.e. `g_distribution.g_response_threshold`, etc.)
  7. **seq_threshold (double, default value: 0.3)**: Threshold used for convolution while selecting subparts.
  8. **seq_center_threshold (double, default value: 0.3)**: Threshold used for computing the part centers for the selection of subparts.

- **Thresholds used for finding maxima of the map**

  1. **max_threshold_percent (integer, default value: 0.5)**: Only parts whose response values are greater than **max_threshold_percent**$*$ value of **max_max** are considered in the computations.
  2. **max_max (integer, default value: -1)**: Maximal value of the histogram peak. Default value -1 means no limit.

3. **min_update_count_percent (double, default value: 0.1)**: Only those values computed in the map updates which are greater than or equal to **min_update_count_percent**$*$ maximum value computed in the map (or of all maps, see **individual_max** parameter) are used.
4. **individual_max** (boolean, default value: false): Specifies which option will be used for the computation of the maximum in the description of **min_update_count_percent**;
   - If the value is true, then the maximum value computed in the map is used,
   - If the value is false, then the maximum value computed over all of the maps is used.
   - The value should be true if statistics is poor at the higher layers and/or during optimization.

2.2.3 Optimization Steps

- **Local Optimization step: Optimization in a receptive field**
  Inference parameters and the optimization specific thresholds should be in namespace l_optimization (ie. l_optimization.g_response_threshold, etc.)
  1. **local_optimization_step (bool, default value: false)**: Perform local optimization step, additional thresholds are given below.
  2. **sample_size (int, default value: -1)**: The number of neighborhoods that are checked, -1 means all.
  3. **update_threshold (double, default value 0.7)**: If, in a 3 x 3 neighborhood of the n^{th} part, a part is found from the current part set and with g-response which is greater than or equal to **update_threshold**, then this neighborhood is not considered, else if g_response of the n^{th} part is greater than or equal to **update_threshold**, then the statistics of the n^{th} part is updated.
  4. **statistics_threshold (double, default value: 0.01)**: Optimization is finished when statistics of the best part falls under statistics_threshold.
  5. **bite_size (int, default value: 10)**: The number of parts that are taken in a greedy step.
- **Global Optimization step: Optimization across the whole image**

  Inference parameters and this optimization specific thresholds should be in namespace g_optimization (i.e. g_optimization.g_response_threshold, etc.)
  1. **global_optimization_step (bool, default value: false)**: Perform global optimization step, additional thresholds are given below.
  2. **cover_threshold (double, default value: 0.8)**: Image is skipped if it is covered with a set of parts whose size is greater than or equal to cover_threshold.
  3. **intersection_threshold (double, default value: 0.2)**: If the p^{th} part intersects with the currently covered set in more than **intersection_threshold** $*$ "`size of p`" pixels, do not update statistics for that part.

4. **bite_size (int, default value: 10)**:The number of parts that are taken in a greedy step.

– **EM-step (Expectation-maximization step)**

1. **EM_step (bool, default value: false)**: Perform EM-step or not.

2.2.4 Learning of objects (object layer)

∗ **General parameters**:

1. **action = learn_objects**: Invokes learning of objects.
2. **src_dir (string, default value: empty)**: Source file for *.lyX files.
3. **library (string, mandatory)**: Name of the file in which the library is saved.
4. **out_library (string, mandatory)**: The name of the file in which the results will be saved.

∗ **Model creation processes**

1. **contraction (double, default value: 1.0)**: Should be 1.0.
2. **max_objects (integer, default value: 4)**: Number of object models produced and validated.
3. **max_cluster_n (integer, default value: 6)**: Maximum number of subparts in each model.
4. **min_cluster_n (integer, default value: 4)**: Minimum number of subparts in each model.
5. **layer (integer, mandatory)**: Layer index of the "last layer" (object models will be added to layer layer + 1).
6. **cluster_member_threshold (double, default value: 0.5)**: "Similar parts" threshold, i.e: If a part is chosen to be in the model, then all parts at the same position and covering at least (**cluster_member_threshold**∗100) percent of this part are considered equivalent (similar) to this part.
7. **cover_threshold (double, default value: 0.75)**: Threshold for the computation of 'model cover projection size'/'layer 0 size'. Union of projections to the first layer (layer of index 0) of all parts in the model / by all of the parts on the 1st layer must be greater than **cover_threshold**.
8. **intersection_threshold (double, default value: 0.2)**: The number of intersections between any two parts in the model (i.e. projections to layer 0) must be greater than or equal to **intersection_threshold**.
9. **max_depth (int, default value: 0)**: Determines the number of layers that are used to construct the object model below the last layer.
   · 0: Only parts from layer L are used;
   · 1: Parts from layers L and L-1 are used.

   Note that parameters '**max_object**', '**max_cluster_n**', '**min_cluster_n**', '**cluster_member_threshold**', '**cover_threshold**', '**intersection_threshold**' can be vectors such that the i^{th} variable of the vector represents the

parameters that are used used at depth $L = i - 1$. If the vector is not long enough, the last variable of the vector is used.

2.2.5 Processes and Functions used for Learning Thresholds

The parameter and its value which is used for learning the thresholds is **action = learn_thresholds**.

1. **positive_src_dir (string, default value: empty)**: Directory which contains positive examples.
2. **negative_src_dir (string, default value: empty)**: Directory which contains negative examples.
3. **library (string, mandatory)**: Input library name.
4. **out_library (string, mandatory)**: Library with learned thresholds.
5. **category_layer (integer, mandatory)**: Index of the category layer (one above the object layer).
6. **use_groundthruth (bool, default value: true)**: Use the groundtruth information when processing positive files.
7. **false_is_negative (bool, default value: true)**: Consider detections outside the groundtruth as negatives.
8. **tf_ratio (double, default value: 0.75)**: The ratio of the number of positive samples to the number of all samples must be greater than or equal to this threshold.

3 Configuration File Organization

In this section, some parameters that are used the configuration files are defined.

3.1 Inference

Contents of a sample configuration file ly1-6_learning.cfg) are given below.

Layer 1

```
#### Inference for layer one ####

inference.ly1.type                  = struct
inference.ly1.layer1_threshold      = 0.1
inference.ly1.power_correction      = 0.6
inference.ly1.init_size             = -50
inference.ly1.scale_limit           = 1000
inference.ly1.border_size           = 100
inference.ly1.scale_factor          = 0.707
```

```
inference.ly1.response_percent       = 0.6
inference.ly1.result_extension       = .ly1

####
```

`type = struct` Filtering with Gabor filters, other options exist but have not been tested.

`layer1_threshold` Threshold for Gabor filter responses (percent of the max response).

`power_correction` Power correction of computed responses (0.5 is square root).

`init_size` If positive then the entire image is resized s.t. the maximal dimension is set to `init_size`. If negative, then the image is resized by factor $-$`init_size`/100.

`scale_limit` File is rescaled using `scale_factor` until the maximal dimension is greater than `scale_limit`. At least one scale is processed even this value is large (Here the value is large to force the algorithm to produce only one scale).

`border_size` Border around image obtained *after* filtering. Should be 100.

`response_percent = 0.6` At the same location only orientations with at least 60% of the strongest orientation response are kept.

`result_extension` Extension of output files.

Layer 2 (and above)

```
#### Config values for layer 2 inference  ####

inference.ly2.layer_contraction     = 2.0
inference.ly2.g_response_threshold  = 0.8
inference.ly2.r_response_threshold  = 0.3
inference.ly2.candidate_r_threshold = 0.25
inference.ly2.candidate_g_threshold = 0.4
inference.ly2.g_response_pow        = 0.6
inference.ly2.candidate_g_threshold_percent = 0.35
inference.ly2.realization_ratio_threshold   = 0.9
inference.ly2.ignore_g_distribution = true
inference.ly2.add_edge_names        = true
inference.ly2.convolution_link_threshold = 0.95
inference.ly2.result_extension      = .ly2

####
```

`layer_contraction` Contraction of coordinates, i.e., "reduction in spatial resolution".

`g_response_threshold` Threshold for *score* value defined above.

`r_response_threshold` Threshold for "contrast score" which is an average of all layer 1 filter responses in the support of the detection (hidden state).

`candidate_g_threshold` Threshold for score of the central (reference) part.

`candidate_r_threshold` Threshold for contrast score for the central part.

`g_response_pow` Power factor used for g_response (score), similar as `power_correction` on layer 1.

`candidate_g_threshold_percent` Threshold for score of the central part, set as percent of the best score on layer rather than set explicitly by `candidate_g_threshold`. Note that this overrides `candidate_g_threshold`.

`realization_ratio_threshold` Threshold on the ratio between the number of parts "realized" on the first layer and the total number of parts realized on the first layer of the composition.

`ignore_g_distribution` Should always be `true` (legacy of experimenting with different scoring functions).

`add_edge_names` Should be true in learning phase, later it can be set to false to speed up the inference process.

`convolution_link_threshold` Should be set to a large value less than 1 for performance reasons.

`result_extension` Extension of output files.

3.2 Learning

3.2.1 Parameters used in the Optimization Step

```
#### Optimization step settings (selection of compositions) ####

learning.optimization.overall_steps         = 1
learning.optimization.save_test_set         = true

learning.optimization.optimization_type     = default
learning.optimization.min_layer             = 2
learning.optimization.max_layer             = 4
learning.optimization.start_layer           = 2
learning.optimization.end_layer             = 4

learning.optimization.ly2.loops = 2
learning.optimization.ly2.part_max_number                = 10,20
learning.optimization.ly2.merge_distance_threshold       = 2000.
learning.optimization.ly2.merge_sc_threshold             = 0.01
learning.optimization.ly2.cluster_size                   = 7
learning.optimization.ly2.optimize.covered_threshold     = 0.8,0.9
learning.optimization.ly2.optimize.intersection_threshold = 0.1,0.3
learning.optimization.ly2.optimize.optimization          = 1

####
```

`overall_steps` Should be set to 1.

`save_test_set` True if we want the train files to be processed with the obtained library and saved (names are saved in the form res*.lyX).

`optimization_type` Should be `default`.

`start_layer, end_layer` Range of new layers to be learned.

`min_layer, max_layer` Range of configuration file entries for learning (in most cases this range is the same as `start_layer` – `end_layer`).

`loops` Number of iterations when adding part to a new layer. Should be at least 1.

`part_max_number` Number of parts taken at each iteration step. It is a vector of numbers. If the size of the vector is shorter than `loops`, then the last value in the vector is taken.

`merge_distance_threshold` Set it to a large value greater than 1000). It is used while with merging of similar compositions.

`merge_sc_threshold` Threshold for distance between compositions when merging them into "OR-compositions". Set to small value to prevent this process.

`cluster_size` Threshold for the size of OR-composition (for the number of parts merged together). With this threshold we prevent keeping a large number of similar parts in the final vocabulary layer.

`optimize.*` These are settings for "optimization" step.

`optimize.optimization` The value is set to 1 to perform optimization, and 0 to skip this step (in the latter case, `part_max_number` parts are taken in each "loop").

`optimize.covered_threshold` Percent of the first layer to be covered with this layer relative to the cover of the first layer of the previous layer.

`optimize.intersection_threshold` In the optimization step, update part statistics if the size of the intersection of this part with already covered part of image is greater than or equal to `intersection_threshold` · support of the part.

`optimize.bite_size` Specifies the number of parts that are processed in one greedy step.

3.2.2 Parameters used in the Computation of Statistics and Compositions

```
#### Learning (statistics and composition creation) ####

learning.optimization.ly2.source_layer_index = 1

# map update
learning.optimization.ly2.learn.nb_size = 17
learning.optimization.ly2.learn.center_val_threshold_rel = 0.5
learning.optimization.ly2.learn.nb_val_threshold_rel = 0.6
```

```
# finding maxima
learning.optimization.ly2.learn.max_max = 4              # def val = 4
learning.optimization.ly2.learn.max_val_threshold = 0.01 # def val = 0.01
learning.optimization.ly2.learn.individual_max = false   # def val = false
learning.optimization.ly2.learn.max_sigma = 0.0          # def val = 0.0
learning.optimization.ly2.learn.max_nbhood_mask = 5      # def val = 5
learning.optimization.ly2.learn.max_radius = 2           # def val = 2

# composition creation
learning.optimization.ly2.learn.max_candidates = 2    # def val = 5
learning.optimization.ly2.learn.min_part_length = 3   # def val = 2
learning.optimization.ly2.learn.max_part_length = 3
learning.optimization.ly2.learn.max_seq_size = 3      # def val = 3
learning.optimization.ly2.learn.seq_min_intersection_percent = 0.01  # 0.0
learning.optimization.ly2.learn.seq_max_intersection_percent = 0.3   # 0.5

####
```

`source_layer_index` Should be set to L (e.g. 1) for learning the parts at the layer L+1 (e.g. 2), etc.

`center_val_threshold_rel` Only consider nodes with score greater than $(1-\sim)$-quantile of all nodes on "source" layer. This overrides `center_val_threshold`.

`center_val_threshold` Only consider nodes with score greater than $\sim$.

`nb_size` Receptive field size (should be positive odd integer).

`nb_val_threshold_rel` Only nodes n with

$$(2-\sim)\cdot(\text{score of center}) > \text{score}(n) > \sim\cdot(\text{score of center})$$

update histogram value.

`max_max` Maximal number of histogram maxima (largest $\sim$ maxima are taken).

`max_val_threshold` Only positions with value greater than (max histogram value)$\cdot\sim$ are considered for local maxima; maximum can be global maximum or "map-local" (relative to all histograms or to a single histogram).

`individual_max` Take global od map-local maximum (see `max_val_threshold`).

`max_sigma` Sigma of "artificial" spatial distribution (position of subpart). 0 means that we take distribution from the histogram. If statistics is insufficient (on higher layers) then this value should be set to a value greater than 0.

`max_nbhood_mask` Spatial distribution size (position of subpart).

`max_radius` Minimal distance between the peak and center of the histogram (central part).

`max_candidates` Number of (best) candidates examined at each (greedy) step when adding new subpart to the already constructed composition. Default value is 5 but can be smaller for performance reasons.

`min_part_length` Minimal number of subparts.

`max_part_length` Maximal number of subparts.

`min_seq_size` Same as `min_part_length`. Used for compatibility with older cfg. files.

`max_seq_size` Same as `max_part_length`. Used for compatibility with older cfg. files.

`seq_min_intersection_percent` Lower bound for quotient between size of intersection of the node support with center support and size of center support (only compositions with quotient larger than this value can be taken as subparts). Set it to small value if we want compositions to be "connected". 0 allows disconnected compositions.

`seq_max_intersection_percent` Upper bound for quotient between size of intersection of the node support with center support and size of center support (only compositions with quotient smaller than this value can be taken as subparts).