AP238 Extension for Powder Bed Fusion

1 Application Objects

1.1 Am_pbf_operation

The Am_pbf_operation application object defines the machine functions and technology to be used in a powder bed fusion operation. The Am_pbf_operation application object inherits from the am_operation as defined in ISO 14649-17 [1]:

```
ENTITY am_pbf_operation
ABSTRACT SUPERTYPE OF (ONE OF (am_oned_pbf_operation, am_twod_pbf_operation,
am_threed_pbf_operation))
SUBTYPE OF (am_operation);
   its_machine_functions: am_machine_functions;
   its_technology:        OPTIONAL am_technology;
END_ENTITY;
```

1.1.1 its_machine_functions

defines the machine functions required for this operation.

1.1.2 its_technology

defines the technology associated with this operation.

1.2 AM_oned_pbf_operation

The AM_oned_operation defines a path for the laser. The operation may include approach and retract strategies for approaching and exiting the scan path. The operation may include a reference to the twod_pbf_operation that was used to generate this oned entity.

```
ENTITY am_oned_pbf_operation
SUBTYPE OF (am_pbf_operation);
ABSTRACT SUPERTYPE OF (ONE OF (am_oned_pbf_contour_operation,
am_twod_pbf_scan_operation))
approach:          OPTIONAL approach_retract_strategy;
retract:           OPTIONAL approach_retract_strategy;
its_twod_operation: OPTIONAL am_twod_pbf_operation
DERIVE
  scan_path: toolpath_list :=
      SELF\Operation.its_toolpath;
```

```
END_ENTITY;
```

Note: The scan path is inherited from the operation entity

1.2.1 approach

the strategy required for approaching the scan path.

1.2.2 approach

the strategy required for exiting the scan path.

1.2.3 its_twod_operaton

the twod_operaton used to generate this oned_operation.

1.3 AM_oned_pbf_contour_operation

The AM_oned_contour_operation defines a contour path for the laser. A contour path is used to define an edge.

```
ENTITY am_oned_pbf_contour_operation
SUBTYPE OF (am_oned_pbf_operation);
DERIVE
  contour_path: toolpath_list :=
      SELF\Operation.its_toolpath;
END_ENTITY;
```

Note: The contour path is inherited from the operation entity

1.4 AM_oned_pbf_scan_operation

The AM_oned_scan_operation defines a scan path for the laser. A scan path is used to fill an area.

```
ENTITY am_oned_pbf_scan_operation
SUBTYPE OF (am_oned_pbf_operation);
DERIVE
  scan_path: toolpath_list :=
      SELF\Operation.its_toolpath;
END_ENTITY;
```

Note: The contour path is inherited from the operation entity

1.5 AM_twod_pbf_operation

The AM_twod_operation defines information for generating AM_oned operations.
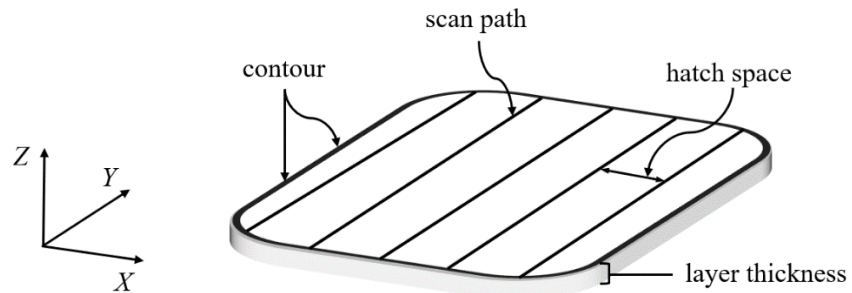

```
ENTITY am_twod_pbf_operation
SUBTYPE OF (am_pbf_operation);
    basis_surface:              OPTIONAL elementary_surface;
    same_sense:                 OPTIONAL BOOLEAN;
    hatch_space:                length_measure;
    its_threed_operation:       OPTIONAL am_threed_pbf_operation;
    its_contour_strategy:       OPTIONAL am_contour_strategy;
    its_hatch_strategy:         OPTIONAL am_hatch_strategy;
END_ENTITY;
```

1.5.1 basis_surface

the surface for the scan paths on this layer.

1.5.2 same_sense

false if the top is on the opposite side of the surface.

NOTE: the surface may be used elsewhere for other purposes

1.5.3 hatch_space

the distance between two consecutive scan paths in the in-fill.

1.5.4 pre_contour

the number of contours to execute before scanning the in-fill.

1.5.5 post_contour

the number of contours to execute after scanning the in-fill.

1.5.6 its_threed_operaton

the threed_pbf_operaton used to generate this twod_pbf_operation.

1.5.7 its_hatch_strategy

defines the strategy used to generate hatch paths for this operation.

1.5.8 its_contour_strategy

defines the strategy used to generate contour paths for this operation.

1.6 AM_threed_pbf_operation

The AM_threed_operation application object describes process parameters for generating AM_twod_operation application objects as a list of layers that together make a solid.
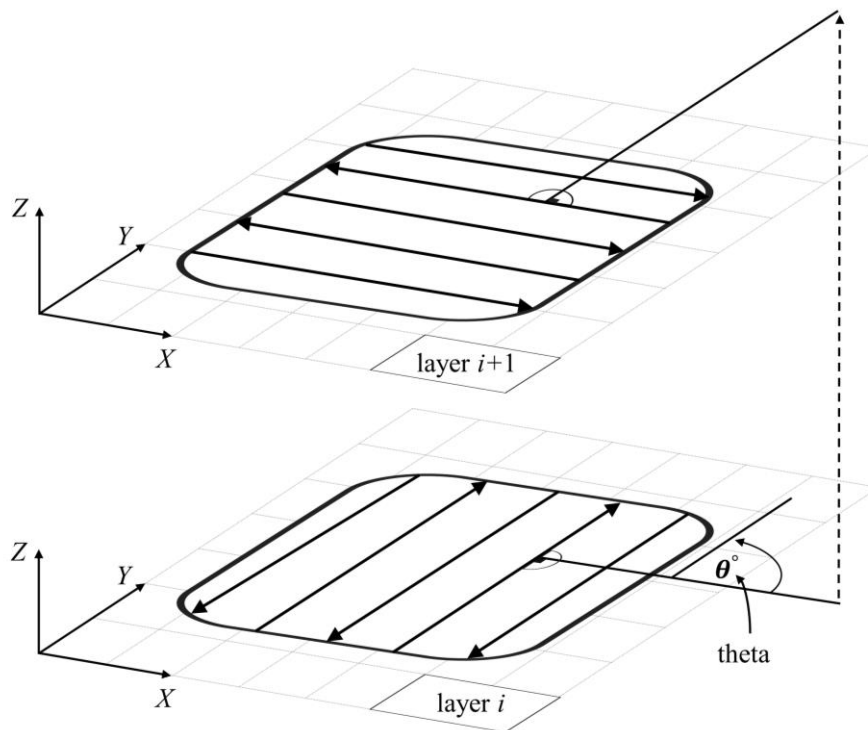

```
ENTITY am_threed_pbf_operation
SUBTYPE OF (am_pbf_operation);
```

```
    theta_interlayer_rotation:          angle_data_element;
    theta_initial_layer_rotation:       OPTIONAL angle_data_element;
    restriction_angle:                  OPTIONAL angle_data_element;
    layer_thickness:                    length_measure;
    its_hatch_strategy:                 OPTIONAL am_hatch_strategy;
    its_contour_strategy:               OPTIONAL am_contour_strategy;
END_ENTITY;
```

1.6.1 theta_interlayer_rotation

the rotation angle (θ) for the scan strategy of the current layer with respect to the scan strategy of the previous layer. The rotational direction of the scan strategy of a layer can be specified with a positive angle value indicating a counterclockwise rotation, or a negative angle value indicating a clockwise rotation.

1.6.2 theta_initial_layer_rotation

the rotation angle (θ) for the scan strategy of the first layer with respect to the coordinates of the build plate. The rotational direction of the scan strategy of the initial layer can be specified with a positive angle value indicating a counterclockwise rotation, or a negative angle value indicating a clockwise rotation.

1.6.3 Restriction angle

Scan paths are not to be planed within this angle of the direction of the gas flow.

NOTE: Layers, islands or stripes that will make paths within this angle should be skipped.

1.6.4 layer_thickness

the thickness of each layer.

Note: the layer_thickness is used to generate the layers and may differ from the actual thickness as stored by the coordinates of each layer.

1.6.5 its_hatch_strategy

defines the strategy used to generate hatch paths for this operation.

NOTE: If there is a conflict between the strategy of an am_twod_pbf_operation and the strategy of its associated am_threed_pbf_operation then the strategy of the am_twod operation shall take precedence.

1.6.6 its_contour_strategy

defines the strategy used to generate contour paths for this operation.

1.7 AM_contour_strategy

The AM_scan_strategy application object defines a strategy for generating scan paths onto a layer.

```
ENTITY am_contour_strategy;
    region_type:           strategy_category;
    layer_count:           OPTIONAL count_measure;
    pre_countour_count:    count_measure;
    pre_contour_offset:    OPTIONAL length_measure;
    post_contour_count:    count_measure;
    post_contour_offset:   OPTIONAL length_measure;
    compensate:            boolean;
    edge_overlap:          length_measure;
END_ENTITY;

TYPE strategy_category = ENUMERATION OF (core, upskin, downskin, future_hole,
support); END_TYPE;
```

1.7.1 region_type

the category of region where this strategy applies.

- core if this strategy applies to core regions of the part
- upskin if this strategy applies to regions near a top face of the part
- downskin if this strategy applies to regions near a bottom face of the part
- future_hole if this strategy applies to a feature that will be removed by machining
- support if this strategy applies to support structures

NOTE: The application shall determine where the strategy shall apply.

1.7.2 layer_count

if the strategy is being applied to a region that is close to a surface, then the number of layers where this strategy should apply.

1.7.3 pre_contour_count

the number of times to run the contour before hatching the in-fill.

1.7.4 pre_contour_offset

distance between consecutive pre_contours

1.7.5 post_contour_count

the number of times to run the contour after hatching the in-fill.

1.7.6 post_contour_offset

distance between consecutive post_contours

1.7.7 compensate

move the path of the contour so that the edge of the beam coincides with the edge of the part.

1.7.8 edge_overlap

move the path beyond the edge to ensure corners are completely filled.

NOTE: Moving the path beyond the edge ensures that all required material is scanned but at the cost of scanning additional material beyond the edge. If the additional material means the part tolerance will not be met, then the excess material can be removed by machining.
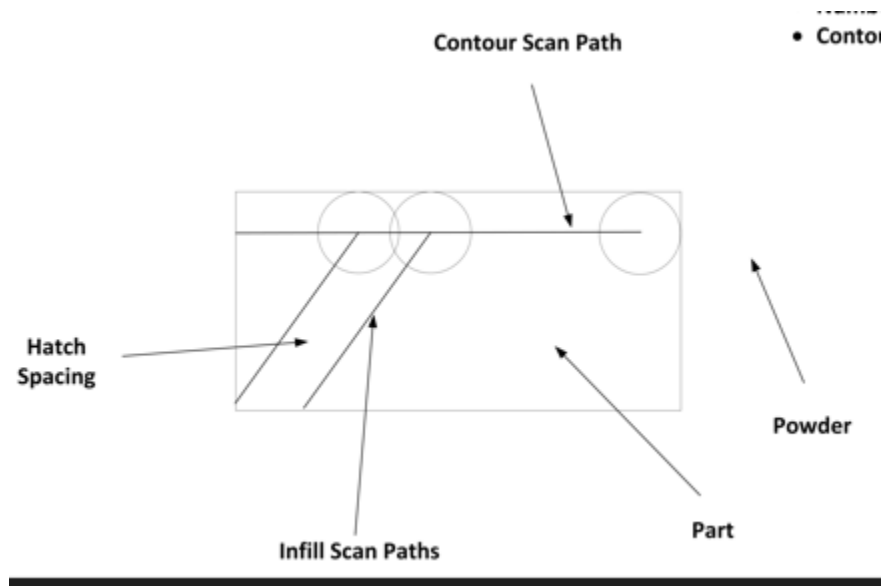
A compensated contour scan path that does not fuse material in the top right corner

1.8 AM_hatch_strategy

The AM_hatch_strategy application object defines a strategy for generating hatch paths onto a layer.

```
Entity am_hatch_strategy
ABSTRACT SUPERTYPE OF (ONE OF (am_stripe_strategy, am_chess_strategy))
   region_type:                       strategy_category;
   layer_count:                       OPTIONAL count_measure;
   maximum_time_between_adjacent:     OPTIONAL time_measure;
   minimum_time_between_adjacent:     OPTIONAL time_measure;
END_ENTITY;
```

1.8.1 region_type

the category of region where this strategy applies.

1.8.2 layer_count

if the strategy is being applied to a region that is close to a surface, then the number of layers where this strategy should apply.

1.8.3 maximum_time_between_adjacent

The maximum time gap allowed between hatching of adjacent islands or stripes.

Note: After this time the material will be too cold.
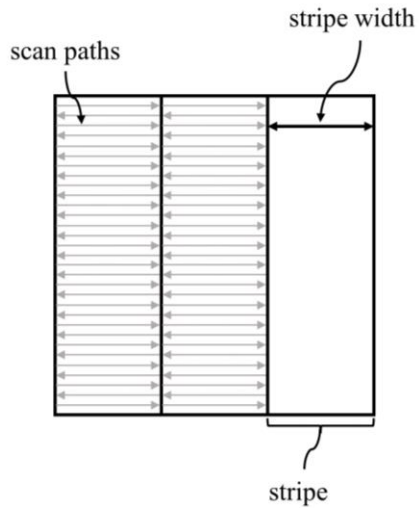
1.8.4 minimum_time_between_adjacent

The minimum time gap allowed between hatching of adjacent islands or stripes.

Note: Before this time the material will be too hot.

1.9 AM_stripe_strategy

The AM_stripe_strategy application object divides each layer into stripes of a given width.

stripe width

scan paths

stripe

```
ENTITY am_stripe_strategy
SUBTYPE OF (am_scan_strategy);
    stripe_width:                           length_measure;
    contour_overlap:                        length_measure;
    stripe_overlap:                         length_measure;
END_ENTITY;
```

1.9.1 stripe_width
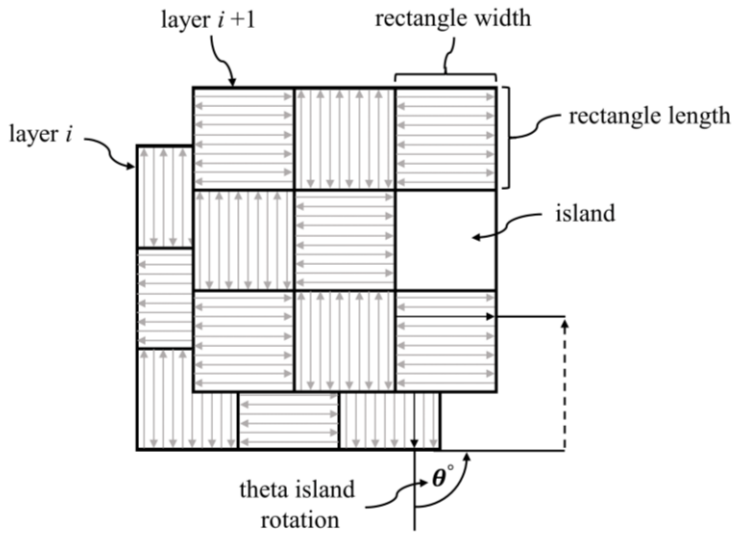
the width of each stripe.

1.9.2 contour_overlap

overlap of a stripe at an edge into the final contour to ensure seamless melting.

1.9.3 stripe_overlap

overlap between stripes to ensure seamless melting.

1.10 AM_chess_strategy

The AM_chess_strategy application object divides each layer into rectangular patches, also called islands. A length and width are specified for each island. The theta_island_rotation describes an optional rotation from the previous island.

```
ENTITY am_chess_strategy
SUBTYPE OF (am_scan_strategy);
    rectangle_length:                      length_measure;
    rectangle_width:                       length_measure;
    theta_island_rotation:                 OPTIONAL angle_data_element;
    contour_overlap:                       length_measure;
    island_overlap:                        length_measure;
END_ENTITY;
```

1.10.1 rectangle_length

the length of an island.

1.10.2 rectangle_width

the width of an island.

1.10.3 theta_island_rotation

a rotation angle (θ) to change the orientation of the scan paths in each new island.

1.10.4 contour_overlap

overlap of an island at an edge into the final contour to ensure seamless melting.

1.10.5 island_overlap

Overlap between islands to ensure seamless melting.

1.11 AM_technology

The AM_technology application object provides categorization of AM processes based on the characteristics that determine each additive process as defined in ISO/ASTM 52900 [2]. The AM technology entity is a subtype of the Technology entity that is defined in ISO 14649-10.

```
ENTITY am_technology
SUBTYPE OF (technology);
END_ENTITY
```

1.12 Powder_bed_fusion_technology

Powder bed fusion application object is a category of AM technologies that uses a thermal energy beam to selectively fuse powder material into geometric patterns. The powder_bed_fusion defines the power, spot diameter, and scan speed of the thermal energy beam.

```
ENTITY powder_bed_fusion_technology
SUBTYPE OF (am_technology);
   beam_diameter:          length_measure;
   beam_path_mode:         OPTIONAL beam_path_mode_type;
   beam_power:             power_data_element;
   beam_power_mode:        OPTIONAL beam_power_mode_type;
   scan_speed:             velocity_data_element;
END_ENTITY;

TYPE beam_path_mode_type = ENUMERATION OF (exact_stop, constant_build_speed,
continuous);
END_TYPE;

TYPE beam_power_mode_type = ENUMERATION OF (constant_power, constant_power_density);
END_TYPE;
```

1.12.1 beam_diameter

the spot diameter of the beam.

Note: d4 sigma measure of gaussian distribution

1.12.2 beam_path_mode

the type of transition between scan paths.

- exact_stop: stop the beam movement at the exact end of each move;

- constant_build_speed: keep the motion at a constant speed while the beam is switched on;
- continuous: matches the final speed with the starting speed of the following move.

1.12.3 beam_power

the energy output of the beam.

1.12.4 beam_power_mode

the type of power management.

- constant_power: keeps the beam power constant during each move;
- constant_power_density: hold the power to speed ratio at a predefined constant during each move.

The formula for deriving the constant power density is given in equation (1) as follows,

$$L = \left(\frac{V}{V_0}\right) \cdot C \cdot L_0 + (1 - C) \cdot L_0 \qquad , \qquad (1)$$

where V is the instantaneous speed in millimeters per second (mm/s), V_0 is the nominal speed in mm/s, C is a unitless weighting factor between 0 and 1, L_0 is the nominal laser power in Watts, and L is the applied laser power in Watts.

Note: The constant power density is controlled on the machine so the value of (C) does not need to be programmed.

1.12.5 scan_speed

the rate at which the beam moves over the scan path.

1.13 AM_machine_functions

The AM_machine functions application object provides categorization of AM processes based on the characteristics that determine each additive process as defined in ISO/ASTM 52900 [2]. The AM machine functions entity is a subtype of the Machine functions entity that is defined in ISO 14649-10.

```
ENTITY am_machine_functions
SUBTYPE OF (machine_functions);
END_ENTITY;
```

1.14 Am_coating_function

The am_coating function application object defines a coating operation that adds a new laer of material onto the part.

- coating_time;

```
ENTITY am_coating_function
SUBTYPE OF (am_machine_functions);
    coating_time:         OPTIONAL time_measure;
END_ENTITY;
```

1.14.1 coating_time

the assumed time for coating a new layer.

NOTE: If the actual coating time differs then there may be more or less cooling between layers.