June 2022

A METHOD TO CONVEY AND ENFORCE PART REQUIREMENTS FOR AUTOMATIC NESTING OF 3D PARTS INTO A BUILD

HP INC

A METHOD TO CONVEY AND ENFORCE PART REQUIREMENTS FOR AUTOMATIC NESTING OF 3D PARTS INTO A BUILD

Introduction

In batch additive manufacturing processes such as MJF and MetalJet, parts are produced in batches as opposed to printed individually. A pre-requisite is that parts are grouped and oriented into what is called a build. This process is sometimes called 'nesting' or 'packing' and is performed by professionals sometimes referred to as DFM Engineer or simply DFM (Design for Manufacturing). As a costly process, solutions have been developed to automate it. Specific build preparation guidelines are developed to specify things like minimal distance between parts, between part and the walls of the build volume, part orientation (flat vs tilted) and placement or not of control parts.
While some automatic nesting solutions are provided as part of an attended workflow, in scenarios where automatic nesting should be done completely unattended, a filtering step is needed to prevent non-complying parts to be placed in a build.

Problem solved

The nesting of parts has been identified as a critical factor for final part quality, and specific nesting guidelines should be defined as part of process development to ensure final parts meet critical quality attributes, as it is specific to the part general geometry, material, etc. These guidelines are not only followed but usually enforced by the DFM engineer, who might identify incoming parts somehow prevents following the guidelines, and through manual intervention either fix or reject the part. An example would be a part that arrives vertically oriented, while a guideline might state that 1) parts should not be rotated and 2) part should be oriented flat across the XY axis.
A vertically oriented part like described above would most likely be the result of a simple mistake upstream, and while it would be easily identified by a person doing the nesting, represents a problem for the automated nesting processes existing today, as they focus on the specific and very challenging task of finding optimal part placement within a build space according to pre-defined rules.
While some automatic nesting solutions are provided as part of an attended workflow (where a DFM engineer is still involved), in scenarios where automatic nesting should be done completely unattended, a filtering step is needed to prevent parts that cannot comply with nesting requirements to be placed in a build. The consequences of not filtering parts very likely will compromise the whole build, resulting in wasted time and raw material.

Description

Different automatic nesting solutions may take different input parameters. These are defined based on specific build preparation guidelines. Combined, the specific values for the nesting solution can be seen as a 'nesting recipe' of a particular process. This nesting recipe is defined before-hand and potentially stored electronically to be used in production/run time.

To achieve a higher level of automation, our proposed solution is structured in two steps:

1. A method used define part requirements for specific nesting recipes prior to run time, including available attributes and clauses that can be used to express the requirements.

2. A method and system used to check whether specific incoming parts meet the intended part requirements for a given nesting recipe and filter out those parts that cannot meet the part requirements prior to initiate the automatic nesting process

Step1: Definition of part requirements

Part requirements are considered entities defined by DMFs and encoded into an electronic medium. They are defined using a set of known operators plus part attributes that can be combined to express part requirements. Operators can be used to test for inequalities (i.e., less than, greater or equal to) and also logical conditions. Part attributes are things like width, height, depth, largest_xy (largest dimension in the XY plane - width and depth), smallest_xy (smallest dimension in the XY plane - width and depth). In conjunction, an examples of a part requirements could be something like:

1) Width <= 380 and depth <= 284 and height <= 380

This example states that a part should fit into the build platform as is, without any rotation in any of the axes. Useful if during the automatic nesting process, parts are expected to be oriented as they arrive. The usable area of the build platform of the MJF 5200 is 380mm x 284mm x 380mm.

2) largest_xy <= 380 and smallest_xy <= 284 and height <= 380

This example is similar then the previous one, but would be useful when part rotations are only useful in the XY plane but not in Z. This is a very typical real-world scenario, as orientation in Z should be carefully determined to avoid part quality defects, but XY could vary to optimize the number of the parts in a build.

3) height < {largest_xy}

This example states a part should have its height smaller than the value of the largest dimension in the XY plane.

Part requirements should be associated to the definition of the nesting recipe. See below an example of how such relationship can be implemented:

```
{"process_1": {
    "nesting_recipe_1": {
        "other_parameters": "some_value",
        "part_requirements": "largest_xy <= 370 and smallest_xy <= 274 and z < {largest_xy}"
 }}}
```

Step 2: Check part requirements & filter out non-compliant parts

Typically, automated nesting is achieved by receiving a list of parts that should be nested according to a given recipe. Step 2 should be done prior to the actual automatic nesting process, inspecting each incoming part to confirm if it complies with the requirements specified for the target recipe.
The part requirement statement can be seen as a rule that should be matched, resulting in true/false value for each part that is checked.
The result of this step should be a new list of parts, with indication on whether the intended nesting recipe can be achieved. The resulting list can be passed to the nesting module.

Advantages

- Allow further automation of the nesting process, reducing labor cost
- Reduce scrap and quality issues by filtering out non-complaint parts

Disclosed by Andre Rabelo, HP Inc.