February 2021

PACKING ALGORITHM WHICH MINIMIZES 3D BUILD COST BASED ON PRE-TRAINED MACHINE LEARNING MODEL

HP INC

Packing algorithm which minimizes 3D build cost based on a pre-trained machine learning model

Abstract

One of the main decision points for selecting a 3DP technology for production applications is the cost per produced part. In order to enable this choice-point it is required to provide a relatively accurate estimation of the cost per part. We have developed a way to calculate the cost automatically, transparently to the user and more accurately than most of the currently available solutions for the different 3DP technologies, by using a trained machine learning model in the nesting algorithm executed in the Pre-print SW application. Then, the proposed method is intended to be combined with any other packing algorithm which may have any optimization target. The packing algorithm generates job nesting configurations and our proposed method analyzes if the solution found meets the cost requirement previously defined by the user. If the solution meets the cost requirement it is accepted. In any other case the solution is rejected, and the packing algorithm is re-executed to find a different solution.

Problem solved

Estimating the cost per part is key for enabling new applications for a 3DP technology. Having a way to provide an accurate estimation of the cost per part before it is even sent to the printer is required to provide the proper mechanism to compare the performance of the technology with others. Not having an accurate cost estimation makes the perception of the technology as not predictable as the actual cost of the printed part will not match (and not even close) to the original estimation, being this a relevant drawback for the technology. For that reason, it is required a mechanism to provide accurate cost estimations. In addition to this, it is always desirable to minimize the cost per part to improve the performance of a 3DP technology. Consequently, we have developed a way to calculate the cost automatically, transparently to the user and more accurately than the actual solution, which also allows to minimize the cost per part by optimizing the job 3D nesting.

Prior solutions

There exist tools or solutions that allow to calculate the approximate cost per build. But they have two main problems:

- They were not implemented to be included as a part of a packing algorithm and run in real-time in the pre-print SW application.
- Its accuracy usually changes depending on the model’s geometry, position in the build, material and print profile configurations, among others.

Description

The proposed solution consists on an additional step after the packing computational optimization algorithm. The nesting optimization algorithm calculates the most suitable solution according to a certain target (thermal uniformity, layer density, height minimization, …). Our solution adds a step that discards or not the packing algorithm solution according to a given cost requirement. This cost requirement could be configured in two operational modes:

- First, by a target cost set by the user. So that the algorithm is executed till a solution which cost is below the target cost is obtained.
- Or by a limit of number of executions (or time) set by the user, so that the algorithm is executed the number of configured times and the solution providing the less cost is chosen.

Our solution includes the following steps (see Figure 1):

1. Select the parts to be packed and the target optimization algorithm (e.g. thermal uniformity, layer density, …)
2. Obtain a valid packing solution according to the optimization target algorithm.
3. Computation of the build estimated cost by introducing its features to our model trained offline with machine learning.
4. If the estimated build cost is lower than the best solution so far, then the new packing is selected as solution.
5. If the cost of the best solution found so far is higher of the target cost, or the number of iterations is still not met, then return to 2 and start again the optimization algorithm with a different seed for the base optimization algorithm. Otherwise, the loop is finished, and the best solution found so far is returned as solution.

For step 3, a model that calculates the estimated cost is trained offline using machine learning techniques (SVM, CNN, GAN…), and it will be capable of obtaining the build estimated cost by looking at the build features. So, first build features are extracted, both at job level like job height, packing density or minimum part distance, and at part level the geometry descriptors like volume, surface, number of triangles or density. These build features are used to evaluate the pre-trained cost model, and an accurate cost estimation is obtained. It is worth to mention that our method is not limited to a concrete set of build features, so it could scale to new types of build features without loss of generality.

To obtain the ground truth it is necessary to send first several builds to the corresponding 3D printer model and obtain the actual values of the agents, powder, wiper and printing/annealing/cooling times, from where it is possible to calculate the cost. Then, these costs are mapped to the corresponding build features from the source jobs being printed, and the machine learning is training to generate the model which will be evaluated by our method.

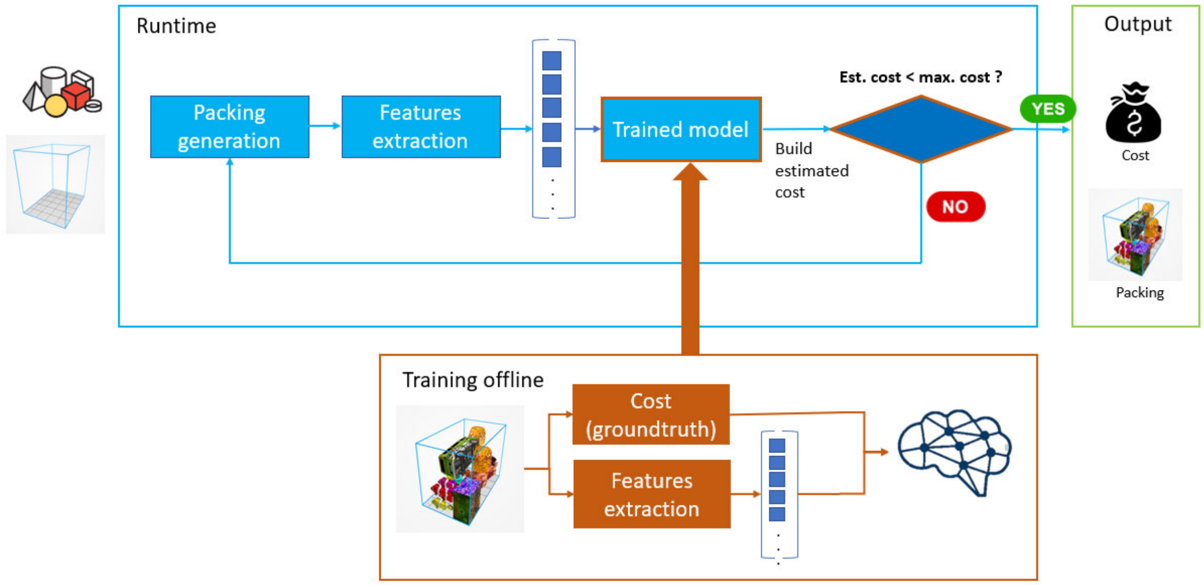

Figure 1. Diagram of the steps in proposed method

Advantages

The proposed solution has several advantages which we enumerate below:

- It provides to the user a way to easily see cost changes depending on the configuration (for different platform, material and print profile configurations…)
- The cost calculation is transparent to the user.
- The nesting algorithm can be combined with any other packing algorithm with any other optimization goal (e.g. thermal uniformity, …).
- It is executed in real time and it is transparent to the user since it runs during the nesting execution.
- It does not require to submit the job to the printer, as the analysis during the job nesting is done in the pre-print SW application.

Disclosed by: Cristina González, Quim Muntal, Sergio González, Jordi Roca, Marc Borràs, HP Inc.