REPORTING A PRINTABLE BOX WITH RESTRICTED ZONES

HP INC
Reporting a Printable Box with Restricted Zones

In this paper we present a method to report a Printable Box with restricted zones from a 3D Printer, so that the pre-print SW should not add content to the restricted zones despite they are defined inside the Printable Box of the printer.

Some 3D printers define a maximum build bed box, and a printable bed box (see Figure 1). Build bed Max Box dimension determines the limits where printer’s mechanic can (physically) place agent fluids used during object generation. On the other hand, the Printable Box determines the locations where parts can be placed in order to get optimal Part Quality. These two boxes are reported to the software, so that it can properly configure a 3MF by placing all the content to be printed inside the printable box. Once the user designs the build configuration, a 3MF file is generated and sent to the printer.

The Printable Box is reported via the printer’s identification web service as UsablePlatform parameter. Our proposal is to include under this element, a list of restricted zones, namely RestrictedZones. This list could include an arbitrary list of restricted zones. For the sake of simplicity, in the example of Figure 2 we have included 2 restricted zones, but our approach allows including as many restricted zones as required. It is worth to mention that Figure 2 is just a specific example of the approach, but any other implementation for reporting restricted zones would also be valid.

When the pre-print SW reads these values, it should restrict the user to place content in the restricted areas. If the user places them manually, it should not the user to submit the job if there is content violating the restricted zones, or at least inform him about such situation. In other words,
content inside a restricted zone should be handled in the same way as if it were outside the Printable Box. Also, if the users select to auto-pack the content, the auto-packing algorithm from the pre-print SW application should respect the restricted zones. In the case of the printer FW, despite the pre-print SW should respect the restricted zones, FW cannot assume that the 3MF has not content violating them, so it still would require performing some validations and informing the printer operator in case any violation is detected.

Figure 2. Proposal of MJF Printer Identification web service reporting a printable bed box (UsablePlatform) including 2 restricted zones

When the printer receives the job content (see Figure 3), first it analyzes the content to determine which parser must handle the job. In this case, it would detect that the file is a 3MF and send it to the
3MF parser. The 3MF parser will extract all the content and store it to disk in some intermediate format. At this point the hierarchy of parts and assemblies are created in memory and added to the job. All the parts reference a model which usually is a representation by means of a triangular mesh (or in some cases a stack of polygonal slices). Every model is stored in an intermediate file. Also, the parts have assigned an affine 3D transformation matrix which positions the model content into the bed. After this point the regular processing of every part should start. However, in our proposal we perform a per-part analysis to identify if any of the parts violates the restricted zones. In the case that a part violates the restricted zone, it is marked accordingly.

![Figure 3. Simplified block diagram for job submission with the proposal for validating that models do not fall inside the restricted zones](image)

The process of checking that a model does not fall inside a restricted zone is done for every part belonging to the job. For every part, the intersection of the mesh representing the part (after applying the transformation matrix) is computed against every one of the restricted zones. For every restricted zone, a virtual triangular mesh is generated representing the box it models. Then, the problem is just to compute the intersection between these two meshes. This is a well-known problem and there exist many efficient algorithms for computing intersection between triangular meshes. In our case, we used a class [2] provided by the VTK library [3], but any other algorithm would also fit in our solution. If there are no intersection at all, then the part is processed successfully. Otherwise, the part will be marked as intersecting with restricted zones. In this second case we can avoid the processing of the part as in no case we would allow printing its content inside a restricted zone.

After all the parts are processed, the job is considered as Processed and ready to be printed. When the job is sent to print, then the result of the analysis can be queried from some check before printing. There can be different policies on how to proceed when some violation is detected in the previous check (and the behavior could be configurable):

- Reject the job if some part violates a restricted zone.
- Allow printing but removing the part from the user job which is inside a restricted zone. If that is the case, the parts violating restricted zones will be directly marked as COMPLETED with a specific completion status CANCELLED and reason MISSED_TO_PRINT (all these status values are already defined in current web service API), which would allow to later identify them when analyzing the job data. Marking parts as COMPLETED we assure that their content is not added to the bed-layer images when composing them, so we avoid the overlapping with the content that the printer FW will automatically include to the job.
- Allow printing but the printer does not print any content in the restricted zone which contain some user part.

The proposed method allows reserving zones within the Printable Box so that the printer can embed internally generated content to the user job. Despite the method is not tight to a specific application, we can think in some interesting applications which we enable with our method:

- Printing parts for traceability: In some production environments it is required that specific part/s are printed for traceability. Mainly, these parts include some unique identifier of the printed job (i.e. lot ID), and are stored for some amount of time. Then, if some issue is detected with any part of a given lot, the customer could go back to the traceability part/s for this lot and analyze them to identify any specific issue with the 3D printing process and decide if it should recall all the parts printed within that lot.

- Printing parts for improving part quality: We can think in a method that analyzes the content from the user job and, for instance, based on the layer densities of the user jobs, the printer decides to add certain patches in the reserved zones to compensate re-radiance amplification effects.

*Disclosure by Sergio Gonzalez, Pablo Dominguez Pastor and Jordi Roca, HP Inc.*