

DUOADD[§]: A SOFTWARE TO DETECT AND EXPORT DAMAGES OF 3D SCANNED OBJECTS

MATTEO PERINI^{*}, PAOLO BOSETTI[†]

University of Trento
Department of Industrial Engineering

^{*} e-mail: matteo.perini@unitn.it

[†] e-mail: paolo.bosetti@unitn.it

web page: <https://www.promfacility.eu/>

Key words: Direct Laser Deposition, Object Repair, Octree, Hybrid Manufacturing

Abstract.

The recent appearance of hybrid CNC machines – which can both add and remove material to an object – has facilitated a new type of approach to the problem of repairing damaged components. Up to now, repair operations have been carried out manually and for this reason they are errors prone, costly and time consuming. The use of precise and repeatable CNC machines in this field is therefore very attractive for the sake of reliability and repeatability. One of the biggest obstacles on the automation of the repairing process is that the CAM software requires a solid CAD model to create the toolpaths needed to perform additive operations. The present work proposes the use of octrees to detect the damaged spot starting from the 3D scan of the damaged object. A software named DUOADD has been developed to convert these informations into a CAD model suitable to be used by the CAM software. The new workflow designed to perform a complete repair operation is here described. DUOADD allows to approach the repairing problem from a new point of view opening the doors to save both time and money. The successful application of the entire process to repair a damaged die injection mold is also here reported.

[§] DUOADD = DUOADD Uses Octrees As Damage Detector

1 INTRODUCTION

Direct laser deposition (DLD)[1, 2, 3] is an additive manufacturing (AM) technique[4, 5] that allows metal material to be deposited using a process similar to FDM (Fusion Deposition Modeling) 3D printing. The main difference between the two is that the raw material is powder rather than wire. The metal powder is sprayed by a nozzle with a flow of inert gas and is melted by a laser beam. The powder ends up in a pool of molten material and becomes part of the object, adding thickness to the working surface. By adding material layer by layer, it is possible to create three-dimensional objects.

One of the very interesting aspects of this technology is that the printhead can be mounted on various machines such as industrial robots or 5-axis CNC machines. This allows the deposition of material even on free form surfaces and/or to create very complex objects. In the present

work the Authors used a hybrid manufacturing machine: the DMG MORI Lasertec 65 3D. Hybrid manufacturing in this case means that the 5-axis CNC machine has the ability both to add material and to remove it through a milling process.

The ability to add new material to an existing object and remove any excess material makes this 3D printing technique particularly suitable for repairing damaged objects[6, 7, 8]. Nowadays it often happens that critical components are replaced in their entirety after suffering minor damages that affect the part functionality. This results in a huge waste of material, energy and money. Being able to refurbish these objects becomes very important both from an economic point of view and from that of environmental sustainability[9]. Extending the life of objects becomes crucial if, as the European Commission has recently pointed out¹, companies are invited to develop production techniques that are increasingly geared towards reduced environmental impact and the circular economy. Until now, the repair of mechanical components is carried out manually by highly skilled workers through welding and finishing operations. This makes repair operations very expensive and prone to errors due to the non-repeatability of the process.

The idea of using the precise and repeatable DLD technology to repair and finish components by milling is therefore very attractive[10, 11], with the added advantage of using only one machines during the entire process. One of the main problems to face in using this technology for repair damaged object is the difficulty in detecting the volume where the material needs to be deposited. In fact, CAM software use three-dimensional solid models, not meshes, to create the toolpaths that guide the CNC machine through additive operations. In this work the Authors present a software and a procedure that allow the identification of the damage, its digitalization and finally the creation of a digital model in a format that can be used by the CAM.

The process consists of 5 main steps:

1. 3D scanning of the damaged component;
2. alignment of the scan with the original solid model;
3. discretization via octree[12, 13] of both models;
4. boolean comparison between the two discretized objects
5. conversion of the damage representation into a solid model.

This paper will describe in detail these steps with particular emphasis on the software part. Finally, the results obtained on a case study showing the effectiveness of the process developed in the repair of real objects will be reported and discussed.

2 3D SCAN AND ALIGNMENT

The process of identifying a damage on a component requires a couple of three-dimensional models, the first obtained by 3D scanning of the damaged object, and the second representing the theoretical solid model of the original geometry. To obtain the model of the damage,

¹http://europa.eu/rapid/press-release_IP-15-6203_en.htm

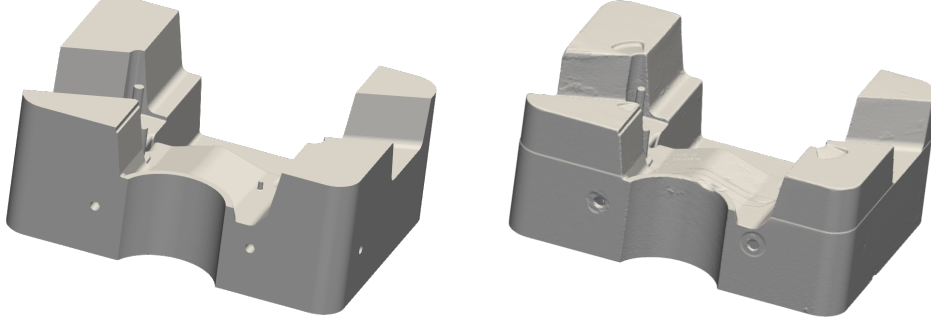


Figure 1: Left: The original solid model. Right: The mesh of the 3D scan.

these two models must be compared by means of a Boolean operation, see Figure 1. When boolean operations are performed by working with surfaces represented as triangulated meshes – typically obtained by 3D scans – it is very easy to incur in broken geometries or a multitude of micro-volumes that have no practical value[14]. For this reason, after several attempts to directly compare two mesh models, it was decided to change approach and discretize both geometries using octrees. This technique will be explained in detail in section 3.1.

There are various techniques that allow to digitize the geometry of the damaged component[15, 16] but for this work it was decided to use a structured light 3D scanner: the Metrascan 3D CMM by Creaform. This scanner has a resolution of 0.03 mm and an overall accuracy of 0.12 mm. This precision is more than sufficient for repair operations as it is about an order of magnitude smaller than the voxel resolution of the DLD machine.

One of the most advantages of this type of 3D scan is that the entire geometry of the object can be reconstructed at once. In fact, the Metrascan is equipped with a scanning head that can be manually moved and therefore allows to reconstruct the object from many points of view during the same scanning session. The scanner software processes all the information received from the cameras and returns a triangulated mesh. It also allows an adaptive sampling that increases the number of triangles in detail-rich areas leaving less important areas – e.g. flat areas – with fewer triangles. In this way it is possible to obtain a mesh with a good resolution while keeping the number of triangles limited, with the consequence that it will decrease the computational calculation of the following octree creation.

The second model – that represents the nominal geometry – is transformed from a solid CAD model into a mesh by software. This allows one to have absolute control on the maximum deviation error of the mesh from the theoretical model.

The resulting meshes should be checked to ensure that they are watertight and coherently

oriented.

The next step consists in the alignment of the two models and their positioning in space relative to a known coordinate system. While alignment is essential for the comparing operation of the two models, positioning the models is optional but greatly facilitates dealing with the models inside the CAM software.

The procedure that gave the best results places the theoretical model in a convenient position for the identification of a local coordinate system. Subsequently, the 3D model obtained by three-dimensional scanning of the damaged object is aligned with the theoretical model using one of the following software: Creaform VX Elements², Cloud Compare³ or Meshlab [17]⁴.

All the software mentioned use least square algorithms for minimizing alignment errors and gave good results. These software have differences that could sometimes make the alignment process easier. For example, VX Elements allows one to perform alignment by giving more importance to some areas of the mesh than others – i.e. the undamaged areas. Consequently, the choice of the alignment software often depends on the availability of software and on the skill/knowledge of the operator.

The alignment operation results in two meshes that have the same position and orientation in space, these can be used for the next operations.

3 DUOADD

At the time being, there are only few CAM software that can handle DLD processes in 5-axis CNC machines and none of them can use meshes to define where the material has to be added. It is therefore necessary to have a solid model of the volume to be created/added. The purpose of DUOADD is precisely to convert the information from the 3D scan of the damaged component into a model that can be easily used within the CAM. DMG MORI – the company making the Lasertec 65 3D – uses Siemens NX CAD/CAM to prepare part programs for hybrid processes. The process that DUOADD uses to arrive at the STEP [18] file representing the solid damage model is divided into several steps:

1. octree-based discretization of both meshes, the one resulted by the 3D scan and the one obtained from the original solid model;
2. complete the data structure filling the void space inside the shell of the octrees;
3. boolean comparison between the octrees by NOR operation;
4. filter the obtained results by removing isolated nodes;
5. conversion of the octree structure into a solid STEP/IGES model;
6. export of results.

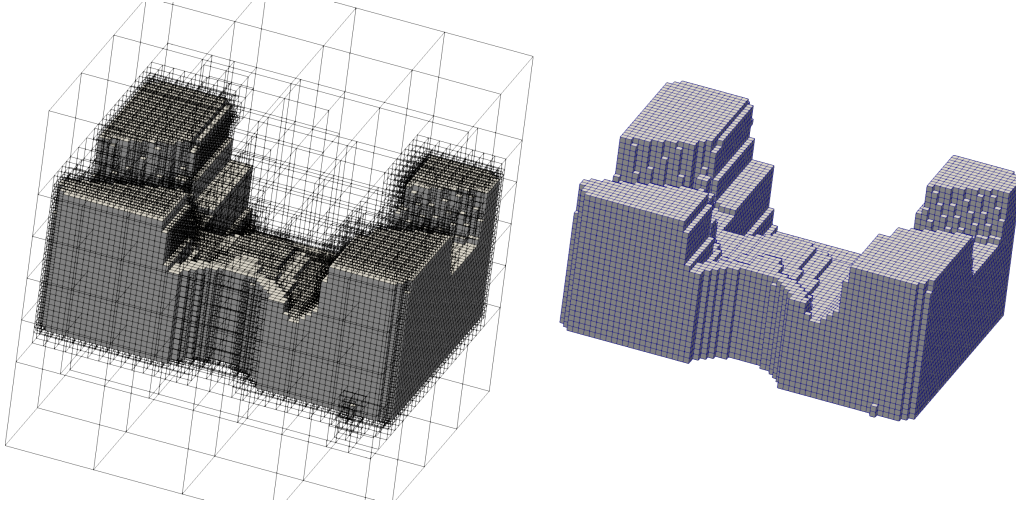


Figure 2: Left: The structure of the octree. Right: The leaves representing the mesh.

3.1 Octree-based discretization

This section describes in detail the process of discretization by octree of a mesh. The octree-based data structure was chosen by the Authors because of its robustness and effectiveness. An octree is a recursive tree-based structure that develops from an initial volume – called root – in the shape of a cube, built around the mesh you want to discretize. Octrees have only a single root which is divided into eight octants – called children – all identical to each other. During the recursive discretization process each child that contains or intersects at least one triangle of the mesh is again subdivided into 8 identical children. This process continues until the desired spatial resolution is reached or when an octant – also called node – is completely empty. The resolution can be controlled by setting the maximum number of iterations that corresponds to a certain depth level. For example, starting from a root of 100 mm side, after $n = 10$ iterations the size of the edge of each leaf will be: $L = 100/2^{10} = 0.0976$ mm. It is evident that, thanks to this power law, the level of discretization is improving quickly with n . However, it is also important to consider that the number of nodes increases exponentially in relation to the level of depth. It can be shown that the theoretical maximum number of nodes is $N = \sum_{i=1}^n 8^{n-1}$. Octrees

therefore allow to obtain a denser data structure only where necessary, leaving the rest of the structure lighter. All nodes that are at the end of a branch – i.e. they have no children of their own – are called leaves. The set of all the leaves belonging to the last level of the tree represent the actual discretization of the mesh surface with the added value of including the description of the bulk. Figure 2 shows the octree structure on the left side and the set of leaves – obtained with seven iterations – on the right.

The construction of the octree starts with the calculation of the center of the root and its

²<https://www.creaform3d.com/>

³<http://www.danielgm.net/cc/>

⁴<http://www.meshlab.net/>

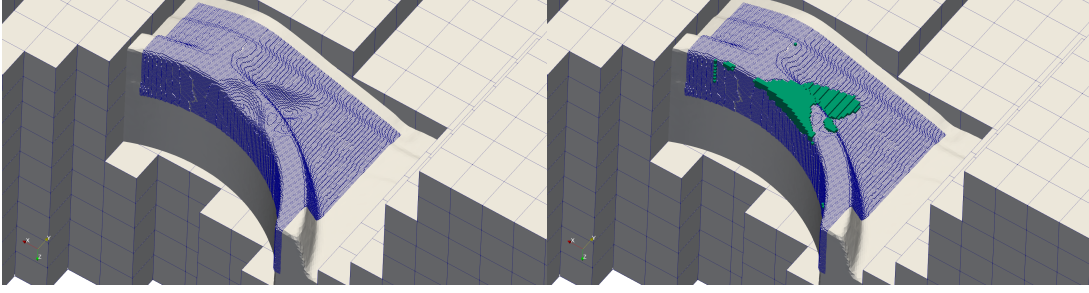


Figure 3: An example of higher resolution where is needed (left). The volume of the detected damage in green (right).

dimensions. In order to obtain two comparable octrees, position and dimensions of both octrees must match so that all the vertices of both meshes are taken into consideration.

Each vertex has coordinates – i.e. X_v, Y_v, Z_v – that will define its position in space. By checking the coordinates of all vertices it is possible to detect the maximum and minimum extension of the models along the three main axes $(X_{v,max}, X_{v,min}, Y_{v,max}, Y_{v,min}, Z_{v,max}, Z_{v,min})$. The center of the root node is calculated as $\mathbf{R}_c = \left\{ \frac{X_{v,max} + X_{v,min}}{2}, \frac{Y_{v,max} + Y_{v,min}}{2}, \frac{Z_{v,max} + Z_{v,min}}{2} \right\}$ while the size of the root side will be: $l = \max\{X_{v,max} - X_{v,min}; Y_{v,max} - Y_{v,min}; Z_{v,max} - Z_{v,min}\}$.

The construction of the octree proceeds with the division of the root node into 8 octants. Each triangle of the mesh that intersects even partially the volume of a child node is assigned to that node. In this way, each child node will have its own list of triangles which, at the next iteration, will be distributed among the new children. Note that the same triangle may belong to several octants of the same level. To check if a triangle belongs to an octant or not, it is sufficient to check if at least one vertex of the triangle is inside the volume of the node. DUOADD also checks if there are empty nodes and, in that case, they are not subdivided any further, thus keeping the data structure light.

A recursive routine continues in the subdivision of the nodes until the maximum level of recursion is reached. This level is defined by the user according to the sought resolution. This function assigns to each node newly created its position, size and a name programmatically created, according to the place occupied by the node in the octree data structure. In fact, one of the advantages of using a rigorous tree structure like the octree is that the position within the data structure is uniquely correlated to the geometric properties of the node itself – i.e. position and size of the node.

One of the disadvantages of octrees is that the number of nodes increases exponentially as the number of levels increases. To prevent the octree from reaching unmanageable size, a procedure has been used that allows the level reached by the octree to be increased only in a certain area – e.g. the damage area. This allows the resolution to be higher only where it is really needed. Figure 3 shows a localized higher resolution that casts the focus of the analysis only on the damaged area.

The nodes that are present in the last level of recursion and that contain triangles of the mesh are marked as “object” and are the nodes that represent the triangulated surface.

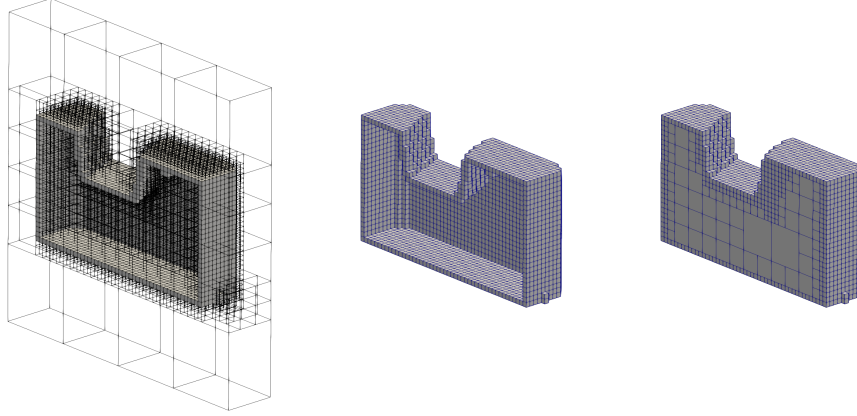


Figure 4: Empty structure (left and center). Filled octree (right).

3.2 Octree filling

The result of the previous phase is a complete octree where all the leaves completely inside or outside of the mesh are not marked as “object” because they do not intersect any triangle. To perform the Boolean operation between the two octrees, it is necessary to have the entire volume inside the triangulated surface to be marked as “object”, leaving only the external nodes marked “void”. The filling operation[19] is performed using a ray-triangle intersection algorithm[20]. This algorithm traces a ray from a point belonging to the root node to a point outside and counts the number of intersections of this segment with the triangulated surface of the mesh. If the number of intersections is odd it means that the starting point lays inside the mesh volume, otherwise the point is external. The routine repeats this procedure for all nodes not already marked as “object” using the center of the node itself as the starting point of the ray. Figure 4 shows the result of the filling operation. This filling operation could be computationally heavy because for each node one has to theoretically look for the intersections with each triangle of the mesh. To attenuate this problem a possible improvement is to point the ray along the vertical axis so that it is possible to select a priori the set of triangles that might have intersections, excluding all others from the analysis.

3.3 Boolean comparison

At this point the octree represents the actual volume enclosed by the mesh. The last problem to be solved before comparing the 3D scan octree with the octree of the original solid model is that the two data structures may be locally inconsistent due to the geometric differences of the starting meshes. In order to compare the two structures a recursive procedure has been written that, iterating simultaneously on both octrees, checks the presence of structure inconsistencies – i.e. different number of leaves or depth reached by each branch of the octrees – and corrects them creating two coherent structures.

Normally this operation does not involve a considerable weight increase of the octrees as the two models usually differ from each other only in the volume of the damage.

Once the two data structures have been made uniform, it is possible to carry out the comparison. This happens with another recursive routine that compares each node having the same position in both octrees and saves the result of the comparison in a new octree that has exactly the same data structure. During the comparison procedure only nodes that are not marked both “inside” or both “outside” are marked as “damage” and kept in the new octree. It is therefore a XOR boolean operation.

DUOADD performs this operation on both octrees at the same time, covering the entire data structure only once. The result of this operation is an octree in which only the nodes that actually represent the volume of the damage remain active.

A problem resulting from the boolean XOR operation is that, in addition to the volume of damage, other disconnected nodes – unnecessary for the purpose of this work – are created. In fact, in some cases even very small differences between the STL models lead to differences in the octrees created. Disconnected and scattered knots are not useful for repair purpose and for this reason must be removed. Duoadd implements some algorithms for filtering and deleting those unnecessary nodes, discussed hereafter.

3.4 Filtering

Several methods of filtering the result have been implemented and tested but the two that gave the best results are:

- (a) filtering based on the number of neighbours of a node;
- (b) filtering based on the connection with other nodes along the main directions – i.e. not along the diagonals.

In both cases an algorithm has been written that finds the neighbours of a node[21] and controls their position and quantity. This routine is performed on each node marked as “damage” and, if it is not sufficiently connected with the other nodes, the node itself is deleted from the resulting volume. The first algorithm erodes the component especially in conditions of very scattered results. The second method is more conservative and in some cases does not remove all disconnected components. On the other hand, however, the second method returns a geometry more representative of the real damage and for this reason is the one that is used by default. The fact that few disconnected parts remain in the resulting octree does not represent a big issue because in the next steps they can be easily removed.

3.5 Octree representation

The data structure of the octree must be translated into a solid model to be used by Siemens NX. Each node is represented starting from its position inside the tree. Three different methods for exporting the octree have been implemented. The first uses the Visualization Toolkit (VTK) library[22] and outputs a set of cubes consisting of vertices, edges, faces and finally volumes. This type of representation is very efficient and allows a quick inspection of the results. The output obtained with this technique is not useful for the conversion of the octree into a solid CAD model but is therefore only used for debugging purposes. The second and third methods

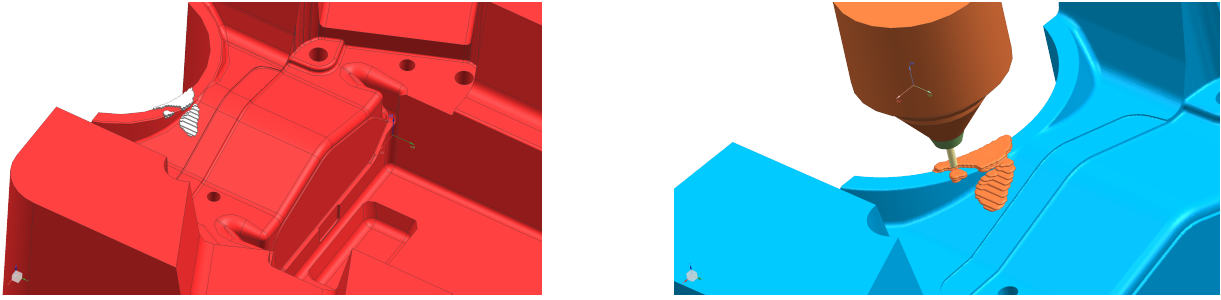


Figure 5: In red the original solid model while the white model represents the damage (left). Simulation of the additive operation (right).

use the OpenCascade (OCC) library[23] with which each node, representing the damage, is transformed into a cube object. The set of all the generated cubes can now be exported as a three-dimensional STEP [18] or IGES [24] model. The file thus obtained is the exact translation of the octree into a CAD model. The resulting solid will be a set of cubes disconnected from each other. These should be merged to use the total volume within the CAM software. To do that, the user can merge all cubes into one or more solids directly via any CAM software. However, it has been observed that this can take a long time if the number of cubes is high – i.e. tens of thousands of individual cubes can take hours to join. One of the reasons why this happens is that the union of many objects greatly increases the complexity of the resulting solid. Tests have shown that the joining operation remains very rapid until the solids to be joined are about one hundred. To overcome this problem, a routine has been written that combines 100 cubes at a time and puts the result in a new list of solids. The volumes resulting from the above operation are then joined together and will form a single solid. This STEP (or IGES) model is much lighter than the set of cubes of which it was composed in the beginning and is much more manageable within CAD/CAM software. With this routine, the time required for the joining operation has been drastically reduced.

4 CAM OPERATIONS AND MACHINING

The solid STEP file obtained in the previous step can be used in CAM to create the toolpaths necessary for the deposition of the new material on the damaged component and, subsequently, for finishing operations. The solid created by DUOADD is already located in space according with the same coordinate system as the original solid model. This is very important because it would be difficult to place a solid model without having precise references. Figure 5 shows the CAD model of the damage automatically placed in the exact position into workspace and the simulation of the additive operation. Thanks to DUOADD instead we can move both solids and position them simultaneously in the workspace avoiding misalignments. As with the normal NX workflow, the solid to be added to the component must be expanded to ensure that sufficient material is brought in. Then it is possible to proceed with the creation of toolpaths for both deposition and finishing operations.

The CAM-generated part program is then used by the hybrid CNC machine – i.e. the DMG MORI Lasertec 65 3D – to add the missing material and remove the excess material. The



Figure 6: The new material added to fill the damage (left). The worn area after the finishing operation (right).

DLD process can deposit material with a deposition rate of 10-25 g/min, which means that, in a few minutes, the material required for filling the damage is deposited. Subsequent milling operations return the component to its original shape, restoring its functionality. Figure 6 shows the actual additive operation on the left, and the removal of the surplus material on the right.

5 CONCLUSIONS

In this work, the problem of repairing damaged objects has been addressed using an effective and innovative approach. The workflow designed and implemented within DUOADD has proven to be repeatable and robust.

With just a few steps, DUOADD allows to detect the volume of the damage in a quick and easy way. It returns the solid model of the damage ready to be used in the CAM software. The speed of the software is strongly dependent on the level of resolution that the user wants to obtain and on the number of triangles of which the meshes are composed, but usually the whole process takes a few tens of seconds. Currently the only two possible alternatives to DUOADD are to manually build the solid model to be used in CAM with a CAD software and then use the DLD machine, or to manually add the material through welding, but these procedures take much longer and are not repeatable.

Authors successfully use DUOADD to repair a die for injection molding but there are many other applications that could benefit from the use of this software – e.g propeller/turbine blades restoration or the refurbishment of artistic/historic artifacts.

Many improvements are expected in the near future such as the binary encoding of the octree structure and the smoothing of the shape of the exported solid body.

ACKNOWLEDGMENT

The authors would like to thank the ProM facility laboratory⁵ for strongly supporting this work.

⁵<https://www.promfacility.eu/>

REFERENCES

- [1] M. Labudovic, D. Hu, and R. Kovacevic, “A three dimensional model for direct laser metal powder deposition and rapid prototyping,” *Journal of Materials Science*, vol. 38, no. 1, pp. 35–49, 2003. [Online]. Available: <http://dx.doi.org/10.1023/A:1021153513925>
- [2] G. Lewis, J. Milewski, D. Thoma, and R. Nemec, “Properties of near-net shape metallic components made by the directed light fabrication process,” in *Proceedings to the Solid Freeform Fabrication Symposium, Univ. of Texas at Austin*. Citeseer, 1997.
- [3] J. Mazumder, J. Choi, K. Nagarathnam, J. Koch, and D. Hetzner, “The direct metal deposition of H13 tool steel for 3-D components,” *JOM*, vol. 49, no. 5, pp. 55–60, 1997. [Online]. Available: <http://dx.doi.org/10.1007/BF02914687>
- [4] B. S. a. Ian Gibson, David Rosen, *Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing*, 2nd ed. Springer-Verlag New York, 2015.
- [5] N. Hopkinson, R. Hague, and P. Dickens, *Rapid manufacturing: an industrial revolution for the digital age*. John Wiley & Sons, 2006.
- [6] A. Pinkerton, W. Wang, and L. Li, “Component repair using laser direct metal deposition,” *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 222, no. 7, pp. 827–836, 2008.
- [7] J. H. Jang, B. D. Joo, S. M. Mun, M. Y. Sung, and Y. H. Moon, “Application of direct laser melting to restore damaged steel dies,” *Metals and Materials International*, vol. 17, no. 1, pp. 167–174, 2011.
- [8] L. Han, K. M. Phatak, and F. W. Liou, “Modeling of laser deposition and repair process,” *Journal of Laser Applications*, vol. 17, no. 2, pp. 89–99, 2005.
- [9] J. M. Wilson, C. Piya, Y. C. Shin, F. Zhao, and K. Ramani, “Remanufacturing of turbine blades by laser direct deposition with its energy and environmental impact analysis,” *Journal of Cleaner Production*, vol. 80, pp. 170–178, 2014.
- [10] J.-Y. Hascoët, S. Touzé, and M. Rauch, “Automated identification of defect geometry for metallic part repair by an additive manufacturing process,” *Welding in the World*, vol. 62, no. 2, pp. 229–241, 2018.
- [11] J. L. Rickli, A. K. Dasgupta, and G. P. Dinda, “A descriptive framework for additive remanufacturing systems,” *International Journal of Rapid Manufacturing*, vol. 4, no. 2-4, pp. 199–218, 2014.
- [12] E. Dyllong and C. Grimm, “A Reliable Extended Octree Representation of CSG Objects with an Adaptive Subdivision Depth,” in *Proceedings of the 7th*

- International Conference on Parallel Processing and Applied Mathematics*, ser. PPAM'07. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 1341–1350. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1786194.1786353>
- [13] H. Samet, “An Overview of Quadrees, Octrees, and Related Hierarchical Data Structures,” in *Theoretical Foundations of Computer Graphics and CAD*, R. A. Earnshaw, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988, pp. 51–68.
- [14] Y. Zhang, Z. Yang, G. He, Y. Qin, and H.-c. Zhang, “Remanufacturing oriented geometric modelling for the damaged region of components,” *Procedia CIRP*, vol. 29, pp. 798–803, 2015.
- [15] E. Bagci, “Reverse engineering applications for recovery of broken or worn parts and re-manufacturing: Three case studies,” *Advances in Engineering Software*, vol. 40, no. 6, pp. 407–418, 2009.
- [16] R. Liu, Z. Wang, T. Sparks, F. Liou, and C. Nedic, “Stereo vision-based repair of metallic components,” *Rapid Prototyping Journal*, vol. 23, no. 1, pp. 65–73, 2017.
- [17] P. Cignoni. MeshLab. [Online]. Available: <http://meshlab.sourceforge.net/>
- [18] I. 110303-21:2016, *STEP-file, ISO 10303-21 – Industrial automation systems and integration – Product data representation and exchange – Part 21: Implementation methods: Clear text encoding of the exchange structure*. ISO, Geneva, Switzerland, March 2013. [Online]. Available: <https://www.iso.org/standard/63141.html>
- [19] S.-N. Yang and T.-W. Lin, “A new linear octree construction by filling algorithms,” in *[1991 Proceedings] Tenth Annual International Phoenix Conference on Computers and Communications*, Mar 1991, pp. 740–746.
- [20] T. Akenine-Möller, “Fast 3D triangle-box overlap testing,” *Journal of graphics tools*, vol. 6, no. 1, pp. 29–33, 2001.
- [21] G. Schrack, “Finding neighbors of equal size in linear quadrees and octrees in constant time,” *CVGIP: Image Understanding*, vol. 55, no. 3, pp. 221 – 230, 1992. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/104996609290022U>
- [22] W. M. Schroeder, K. Martin, and B. Lorensen, “The Visualization Toolkit: An Object Oriented Approach to 3D Graphics 3rd Edition, Kitware,” *Inc. Publisher*, 2003.
- [23] J. Russell and R. Cohn, *Open Cascade Technology*. Book on Demand, 2012. [Online]. Available: <https://books.google.ro/books?id=thWNMQEACAAJ>
- [24] M. Kahrs, “The Heart of IGES,” *Softw. Pract. Exper.*, vol. 25, no. 8, pp. 935–946, Aug. 1995. [Online]. Available: <http://dx.doi.org/10.1002/spe.4380250806>