

3D Printing onto Unknown Uneven Surfaces^{*}

Nils Bausch^{*} David P. Dawkins^{*} Regina Frei^{*}
Susanne Klein^{**}

^{*} *University of Portsmouth, Faculty of Technology, School of Engineering, Anglesea Building, Anglesea Road, Portsmouth, Hampshire, PO1 3DJ, United Kingdom*
(e-mail: work@nilsbausch.net; david.dawkins@port.ac.uk; work@reginafrei.ch)

^{**} *Security and Manageability Lab, Hewlett Packard Labs, Long Down Ave, Stoke Gifford, Bristol BS34 8QZ, United Kingdom*
(e-mail: susanne.klein@hpe.com)

Abstract: Since its inception, 3D printing has seen a wide area of applications, but a general approach to printing onto unknown objects has not been tackled yet. Nowadays 3D scanning technology can be used for reverse engineering. Multiple axis machines enable the creation of object layers at different deposition angles, and printing on uneven surfaces is achieved by conformal printing. In this paper, a new methodology is presented, which combines 3D scanning, multiple axis 3D printing, and conformal printing to create an affordable 3D printing system, which can deposit material onto a priori unknown uneven objects. A prototype system was developed, which can print a first layer on top of a previously unknown object. The creation of further layers is work in progress. The application areas for such a method could include repairing structures, product customization, printing security features on existing objects, adding functionality by, for example, printing antennas on items, and modifying prosthetics to fit individual patients.

Keywords: 3D printing, 3D scanning, conformal printing, sensors, robotics, control, mechatronic system

1. INTRODUCTION

3D printers generally create objects built from even horizontal layers. Only few very expensive and sophisticated systems are able to deposit material in uneven and non-horizontal layers. Most of these systems are used for printing electronic circuits and similar applications, and are hence suitable for high-precision small scale structures. Additionally, these systems require an electronic model of the object to be printed on, for the tool path to be generated in advance. This paper describes the progress made in a project focused on building a low-cost system for *conformal printing*, able to deposit layers of material onto any (unknown) object. The idea is to be able to work with unique objects, rather than working with large production series. To this purpose, the system combines 3D scanning, data processing and Fused Deposition Modelling (FDM) in uneven non-horizontal layers, using varying deposition angles.

The essence of *freeform manufacturing* is the ability to create shapes that are not derived from the usual geometrical elements like rectangles, cuboids, circles, spheres, cylinders and such, whilst still using numerically controlled tools. This requires the objects to be specified in other ways than

usual Computer Aided Design software does; for instance, using *point clouds*. One of the challenges is to specify a 3D object as a point cloud, which typically only describes the surface as an object. The other challenge is to generate a tool path based on the point cloud data.

The digitisation of an a priori unknown object into a point cloud can be realised with 3D scanning methods. The point cloud data can then be used for further processing and aid in printing onto an object's surface. Depending on the budget, commercially available 3D scanning solutions range from simple single-point to sophisticated line laser usually accompanied by a detector or camera module and a software package to convert the gathered data into 3D surface definitions. The accuracy of these scans is in part dependant on scanning speed, reflectivity, colour, and object orientation in regard to the laser and camera.

Turning objects into point clouds makes the object available for further processing and modifications in software programs. The ability to and create a digital object were crucial to this project, as this meant the transition from an a priori unknown object to a known object. Point cloud data is normally transformed into meshes, which is then used to create tool paths.

The purpose of this project is to develop a proof-of-concept system with reduced capabilities: only cylindrical and

^{*} This work is supported by the University of Portsmouth – Research Development Framework (RDF) 2015.

hemispherical surfaces will be considered. Once successful, randomly shaped surfaces will be included, and the idea of *on the fly* conformal printing may be applied to different additive manufacturing technologies, using a wider range of materials, achieving higher accuracy and faster production rates. At this stage, the focus is purely on feasibility.

The main steps achieved so far include: modifying a robotic platform for concurrent sensing and deposition of 3D printing material using several degrees of freedom; creating a fixation device able to hold diverse objects in place; control and programming of the platform; as well as sensor data acquisition and processing.

The paper is organized as follows: The state of the art in the research areas of interest is presented in section 2. A jargon-free introduction to the methodology in section 3 is followed by an in-depth description in section 4. The results and lessons learnt from the current system are described in section 5, followed by an outlook in section 6.

2. STATE OF THE ART

2.1 Conformal printing

It appears that there is currently no solution for conformal printing onto unknown uneven surfaces or 3D objects, at any scale. The only commercially available solutions for printing onto known uneven surfaces – aerosol jet printers (Paulsen et al., 2012) – are made for printing electronic circuits and requires inks to be transformed into aerosols. Starting at \$250k, the printer is not accessible for those with a smaller budget. Moreover, for most industrial applications, not only accuracy but also speed is critical. Hence the printing technology needs to be scalable for applications on bigger surfaces than those in electrical circuits while maintaining accuracy and repeatability.

Adams et al. (2011) present conformal printing of small electrical antennae onto hemispherical substrates using a system as previously mentioned.

Choi et al. (2011) modified an industrial FDM system to increase its degrees of freedom, allowing the machine to perform freeform manufacturing including complex curved shapes and deposition orientations other than horizontal. The system is also able to print onto existing parts. However, as with existing aerosol printers, the part surface needs to be known to generate the tool path.

2.2 Freeform manufacturing

The term freeform manufacturing is often used in conjunction with 3D printing and additive manufacturing (AM) technologies, as they allow the user to create freeform objects. However, these objects are always processed and built in (usually) flat horizontal layers, and will therefore not be reviewed here. Nevertheless, not only AM is used for creating objects with unusual shapes. Other technologies used include diamond cutters steered on paths defined by point clouds Scheiding et al. (2011). Heidrich et al. (2011) use laser ablation and polishing to create freeform optics.

To achieve narrower tolerances, freeform manufacturing using CNC machining has been combined with inspec-

tion and tool path generation (Lasemi et al., 2012). The machined surface is inspected to compare it with the designed surface to identify errors and then generate tool paths suitable to eradicate them. Three methods are frequently used to generate tool paths: iso-parametric, iso-planar and iso-scallop. Lasemi et al. (2012) introduce isophote partitioning as an alternative. Generally, the two most important parameters when generating tool paths for freeform surfaces are the path interval (also called side step / stepover) and the forward step. Optimizing a tool path also affects the print head alignment in relation to the object and can lead to a reduction in the number of commands and path necessary to carry out the same action.

2.3 3D scanning

Levoy et al. (2000) described a hardware and software solution to scan large statues. The researchers employed laser triangulation and laser time-of-flight rangefinders, and aligning, merging, and viewing underpinned by software. They developed their own software algorithms and hardware set ups and created renderings of large size (up to 250 GiB) per statue. The laser scanning methods took hours or days depending on the surface geometry and complexity.

Tam et al. (2013) surveyed 3D scanning technology for small objects, differing between rigid and non-rigid registration. Gathered data can be represented as point clouds or meshes commonly composed of triangular shapes of varying size, angle and orientation.

The DAVID laser scanner (Winkelbach et al., 2006; DAVID, 2015) utilises a technique called structured light triangulation (SLS). In principle, this method can use a single laser line, video camera, the object, and behind it a calibration panel with a 90° angle as a reference geometry, including printed control points for distance calculations. Multiple sweeps, either manually or automated, of the laser line across the object, will reflect light to the camera. The scanned object's surface geometry can then be calculated (Aydar et al., 2011), which is carried out by the DAVID4 software package. An advanced version of this replaces the laser with a projector, which projects a line pattern at close distance onto the object.

2.4 Point cloud data and mesh conversion

The output of a 3D scan, or a CAD system for CAM/CNC applications is in general a mesh or point cloud. From this data, a tool path is generated for the actual machining. Chen and Shi (2008) presented a method for tool path generation based on triangular meshes and the proposed method can be used to augment and automate freeform surface machining.

Chui et al. (2008) proposed a simple algorithm to generate a multi-axis tool path for machining. This direct method for constructing a 3D triangular mesh from the point cloud data created mesh points based on where the machining tool would contact the object.

3. METHODOLOGY

This section describes an ideal workflow to arrive at the desired outcome: 3D printing an object onto a previously unknown surface (object). The processes to achieve this can be broken down into following steps:

- (1) Place and fasten an *unknown* object into a 3D printer.
- (2) Scan the unknown object.
- (3) Extract point cloud data.
- (4) Superimpose the point cloud data with a to-be-printed *new* object.
- (5) Generate the toolpaths for the 3D printer.
- (6) Print the *new* object on top of the previously *unknown* object.

In the current state, many of these steps require manual intervention and the workflow is therefore not fully automated. The following subsections will detail the approach taken for each step, without going into minute detail. The detailed implementation is described in section 4.

3.1 Placing the object

An unknown object could be of any form and shape and therefore restrictions are necessary, to conform with the available overall space of the machine as well as the necessary printing space for the new object. The unknown object should be mounted appropriately, so that the scanning as well as printing either do not need any further human intervention or as less disturbance as possible. This is in order to minimise misalignment that could occur due to scanning and subsequent printing.

3.2 Scanning and point cloud data

The unknown object will be scanned inside the 3D printer and if necessary rotated to capture all the surfaces that will be printed on or to improve the scanning procedure. The scanning could be carried out by a line-laser or a pattern projected onto the object. A camera captures the reflected laser light from the object and a bespoke software system is then used to transform this input into point cloud data. Depending on the scanning method, accuracy and elapsed time can vary. Objects that were considered for scanning were limited to hemispherical and cylindrical shapes, as these shapes were uneven and did not have any concealed corners or edges, that could go undetected by the scanning method.

3.3 Superimposing object data

The previously unknown object data - available as point cloud data after scanning and turned into a triangular mesh - is then superimposed with an existing design for an object that is supposed to be 3D printed, see fig. 1. The remaining parts to be printed of the new object are determined by using a boolean intersection operation, whereby areas of intersection will form the boundaries from where parts of the new object remain or are deleted accordingly. The mathematical description of this boolean operation is:

$$\text{scanned} \cap \text{new}_{\text{complete}} = \text{new}_{\text{intersected}} \quad (1)$$

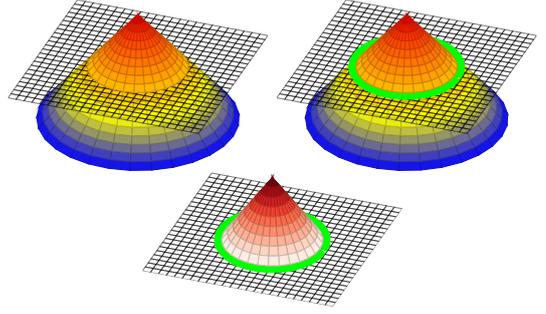


Fig. 1. Superimposing a scanned and new object: the black mesh represents the surface of a scanned object, the coloured cone is the surface of a new object to be printed on top. Clockwise from top left: both objects are arranged to intersect; the green line marks the point of intersection; only the top part of the cone is left to be printed.

This step requires a second bespoke software system, which will carry out superimposing of the objects.

3.4 Generating tool path data and printing

The second bespoke software system is then used to generate the necessary tool path data required to control the 3D printer. For reasons of simplicity, only printing on top of an object was considered, ensuring that the print head does not collide with other printer components.

4. SETUP AND DEVELOPMENT

Before going into the details of the methodology, the selected 3D printer and initial setup is introduced.

Several commercially available 3D printing systems were identified as suitable for this project. These include the ‘5AxisMaker’, ‘The BigOne’ and the ‘Ormerod’. The Prusa RepRap i3 was chosen for several reasons: to shorten the process of building the prototype, improve its ease of use, keep costs low and to benefit from the possibility to modify pre-existing firmware. The printer was converted to include seven Degrees Of Freedom (DOF) for FDM. The DOF include linear movements along the X , Y , and Z axes as well as angular movements around these in α , β , and γ direction and the extruder output. Further benefits of using RepRap 3D printers are open sourced CAD design files, compatible open source firmware, and hardware with a wealth of available information, backed by a large community.

4.1 Axes and DOF

The conversion to a 7 DOF machine was realised by creating new and adjusting existing parts of the printer. The new parts were designed using FDM rapid prototyping methods, resulting in a stable design with a relatively high rigidity. The machine layout was adjusted to house another three stepper motors for the additional DOF. The controller hardware driving the stepper motors was replaced with a Duet and Duex4, which could control up to 9 DOF (X , Y , Z , α , β , γ , plus three extruders) and the firmware replaced with RepRap DC42 and adjusted to operate the 3 rotary DOF about α , β , and γ .

As the system uses the Cartesian coordinate system, the elemental rotations about axis α , β , and γ are obtained using the arctangent function. New coordinates, for example x' and y' can be decomposed from the rotational axis' angles and then the offsets obtained from two dimensional rotational matrices:

$$R(\theta) = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2)$$

If 5 DOF for a single transformation were used, the new coordinates and angles could be easily parsed to G-code format, which is a simple and effective method of communication for CNC and 3D printers alike (Bourell et al., 2011; Brown and De Beer, 2013). Although the prototype 3D printer has 6 DOF available in the current implementation (the 7th being material extrusion), it only supports 5 concurrent DOF at this stage. This means that one of the rotary axes must always be in its home position to ensure a perpendicular orientation of the print head in relation to the object surface. Currently the user specifies which rotational axes to use, and defaults to α and β if none are selected.

The minimum distance to travel on the linear axes X , Y , and Z per stepper motor step are $12.5\mu\text{m}$, $12.5\mu\text{m}$ and $0.25\mu\text{m}$ respectively. The minimum degrees per step on the rotary axes α , β , and γ are 0.012° , 0.056° and 0.111° respectively.

The axis speed varies, with the slowest maximum speed being the Z axis at 2mm s^{-1} (120mm min^{-1}) with stable operation, and an acceleration of 50mm s^{-2} . The fastest axis is the X axis, with a speed of 420mm s^{-1} ($25\,000\text{mm min}^{-1}$) over long lengths $>100\text{mm}$, and an acceleration of 2500mm s^{-2} . The feed-rate, a G-code parameter, specifies the maximum speed at which the motors can move per command, which must be less than the permitted maximum speed and is heavily dependant on the distance travelled as well as the acceleration used. The stepper motor control maintains a velocity that ensures all motors arrive at the desired destination at the same time.

4.2 3D scanner and GUI

The 3D scanner was integrated into the new 3D printer design and consists of a blue line laser and a monochrome camera, backed by the DAVID4 software. Sweeping motions of the laser line across an object's surface are recorded by the camera and processed to form point clouds in the DAVID4 software.

A new host controller software Graphical User Interface (GUI) was developed in C# to simplify setup and communication with the Prusa i3 printer. Other controller software is available, but most included safety features that prevented non-standard operation: For example, when executing print manoeuvres, the Z axis will not advance downwards, or in other cases the G-code format string was not recognised. The GUI control software also enables easy 'one click' operations to setup/perform for the printer: scanning, printing, slicing, homing, serial enumeration, offsetting, testing, and manoeuvring.

4.3 Placing the object and calibration

Due to the underlying technique of structured light triangulation, and how the DAVID4 software determines the surface geometry using a line laser, a removable calibration panel was incorporated into to the printer design. This panel remains in place throughout the scanning procedure. The main advantage of using this type of 3D scanner and software is the possibility to reproduce the surface geometry quickly and to a high standard precision and repeatability, $\pm 0.02\text{mm}$, using the modified Prusa i3's rotational and linear axes. After initial calibration using the calibration panel and the DAVID4 software, the user places an object into the printer's gripper and secures it by fastening the bolts.

4.4 Scanning and point cloud data

Once the calibration of the scanner and object placement is complete, the user can semi-automate the scanning procedure by using the custom GUI host program to connect, scan, manipulate, and send toolpaths. After the user clicked 'setupScan' the printer returns to its home position and then moves to a suitable location where the user can either start a manual scan (where the user controls the laser scanner with manual G-code commands) or an automated 'one click scan', providing DAVID4 is set up as well. If the automated scan method has been activated the printer will traverse across the X axis while performing small rotations around the Y axis. This method yields better results from the laser scan than the alternative procedure, which was to just traverse across the X axis at a fixed angle, see section 5.4.

After each scanning procedure has finished, the user can check for satisfactory surface geometry, by visual inspection of the data that was recorded by the DAVID4 software. The user then decides whether more/better surface data has to be captured, in which case the scanning procedure is repeated. Rerunning the scan can enhance the already captured data. Once multiple scans were performed, the DAVID4 software fused these different scans into one coherent object.

Assuming satisfactory surface geometry was captured, the user must then ensure no irrelevant artefacts have been captured. This is done by visual inspection of the scanned object within the DAVID4 software. Wrongly included artefacts are selected and the points representing them removed.

The point cloud data obtained from the scanning is smoothed and averaged during this process using DAVID4 software, which is necessary to account for reflected scattered light from the object surface. Tests showed, see section 5.4, that by selecting the appropriate smoothing and averaging filters, DAVID4 was able to reproduce basic geometry to a high standard. The aperture setting on the camera may need to be adjusted to suit the ambient light conditions; similarly the absorption, reflectivity and emissivity of the object being scanned affect the camera's ability to pick up the laser light. For example, luminous yellow objects prevent the scanner from working efficiently with a blue line laser. This in turn affects the interpreted

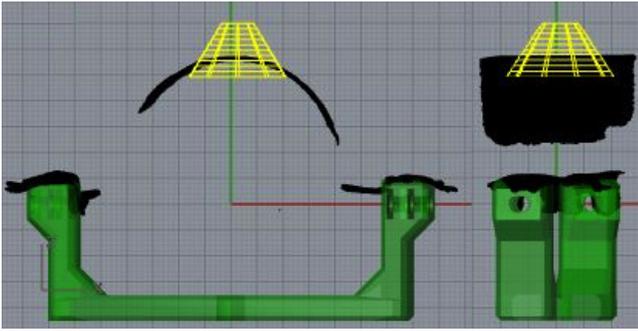


Fig. 2. Alignment of the green reference gripper with the black scanned object and the yellow cone-shaped surface mesh of the object to-be printed viewed from the ZX and ZY plane.

laser line produced via the software, meaning less of the surface is captured.

At this stage in the project, the scanned objects are limited to simple shapes to allow fast prototyping and precision testing. Moreover, the printer and scanner currently has limitations to the angles that can be scanned, brought about by the calibration panel remaining in place throughout the scan, limiting the movement of the object. Another limitation is the print head size, which hinders approaches to concave corners with sharp angles.

Ultimately, after the surface geometry has been captured, the user exports the file as a high density STereoLithography (STL) mesh file, with facets selected in DAVID4 no greater than 0.5 mm.

4.5 Superimposing object data

The superposition (or positioning of the object to-be printed with reference to the scanned object) is carried out in Rhino/Grasshopper, allowing for easy manipulation and control. The imported geometries could be, for example, complex planes, complex meshes, surfaces, or point clouds. The main function of Rhino/Grasshopper is to slice the to-be printed object and obtain tool path coordinates, including facets and vertices normals, lengths and volume, as a part of the object surface geometry. The overall outcome is a tool path consisting of positional coordinates, length of the line segment, and the vertices' normal coordinates, and from this the volumetric extrusion can be computed.

By importing both the object to-be printed and the previously scanned object into Rhino as STL files, one can manipulate the scanned object to match up with the printer's grippers, which are used as an alignment (see fig. 2). Importing the gripped object into the workflow of Rhino allows for visual inspection and alignment, where the home (origin) position is unified between the software and printer. The tool path is generated relative to these coordinates. Hence aligning the scanned part with the printer's gripper in Rhino is essential and currently manually operated.

When the scanned object is in position, the user needs to define a suitable location for the object to-be printed, or 'new' object, such that it intersects the scanned object. Before a boolean intersection can be performed, two volumetric objects are created: the 'new' object's envelope

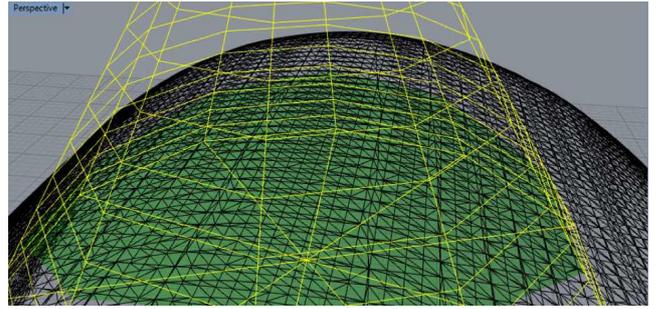


Fig. 3. The green intersection plane as a result of superimposing the yellow object to-be printed and the black previously scanned object.

forms the first object, and a combined stack of layers from the scanned object's surface forms the second object. A bounding box enveloping the 'new' object is created to capture height, width, and depth. The number of layers in the stack is dependant on the printed layer height, which depends on the 3D printer setup, and the 'new' object's bounding box as limits. Once the layer stack, now representing a volume, is created, a boolean intersection is performed, whereby the common parts of both objects are kept and results, in this case, in a sliced mesh that conforms to the object's surface and forms the new base of the object to-be printed, as can be seen in fig. 3. The described method can currently only operate with scanned surfaces that have no holes in the area of intersection with the to-be printed object, which was visually verified by the user.

The remaining components contain the data parameters of the intersection line's geometry and position (including length, extrusion width and layer height to calculate extrusion volume) and additional infill data was generated as support. Distinct differences to traditional 3D printing were addressed in intermediate steps. At first, the normal vectors of each point to be printed are calculated to ensure perpendicular material deposition. Then a print manoeuvre tool path is created and amended with so-called ISO-hops, non-print manoeuvres, including advancing to a next layer, designed to avoid collisions with the scanned object or previously printed paths. In a last step, polylines consisting of closest points combine print and non-print manoeuvres are smoothed by distributing points with a minimal move distance to eliminate jagged print head movements. A simulated 3D printer output can be seen in fig. 4.

4.6 Generating tool path data and printing

A second bespoke software program in Python was created to generate the tool path data required to control the 3D printer. This program converts the exported files from Rhino/Grasshopper into a usable form for 3D printing and the GUI program uploads the resulting G-code to the 3D printer. This step translates the generated tool path's coordinates into the one used by the 3D printer.

The Rhino/Grasshopper exported tool path contains lists with the appropriate tool path and data of the normal vectors. The tool path coordinates (polycurve coordinates) are looked up in the decomposed mesh list which also

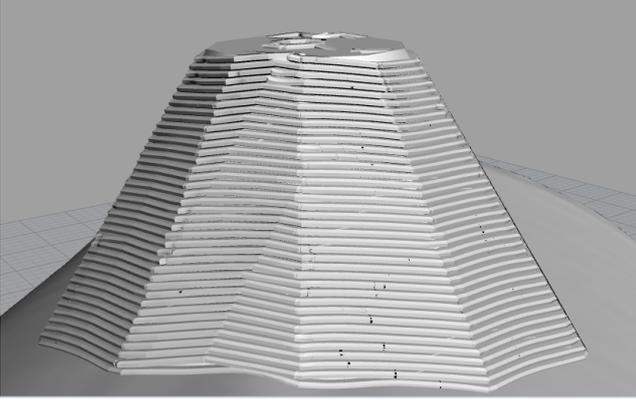


Fig. 4. A simulated toolpath output used for development purposes, showing the scanned object as smooth surface and the layers of the ‘new’ object printed on top.

contains the coordinates of all the points in the interested mesh vertices and their normal coordinates. Normal coordinates are then extracted. These are the normals for each vertex related to the point on the tool path. Each time a match is found, *atan2*, a modified arctangent function of two parameters X and Y is used to obtain the angle of the vertices’ normal vector. The normal vectors are then mapped to the printer’s geometry and orientation for each rotational axis by adding or subtracting π or $\frac{\pi}{2}$, depending on the quadrant found by *atan2*.

Using the anti-clockwise rotation matrix in eq. (2) the offsets are then determined (eqs. (5) and (6)) as each axis’ rotational angle was known and mapped to the printers rotational axis; by determining the quadrant in which the normal vector lies, the algorithm ensures the printer rotates in the direction closest to align to the object. The angle about the rotational axis γ was calculated with eq. (4).

$$\psi = \text{atan2}(y_{\text{normal}}, x_{\text{normal}}) \quad (3)$$

$$\gamma = \begin{cases} -\psi & \text{if } 0^\circ \leq |\psi| \leq 90^\circ, \\ 180^\circ - \psi & \text{if } 90^\circ < \psi \leq 180^\circ, \\ -180^\circ - \psi & \text{if } -90^\circ > \psi \geq -180^\circ \end{cases} \quad (4)$$

$$x_{\text{offset}} = x' - x \quad (5)$$

$$y_{\text{offset}} = y' - y \quad (6)$$

At this point, it suffices to determine the change of X , Y , and Z , after which, depending on the users selection of which rotary axes to use, the sum of the user selected rotational offset coordinates are added to the relative coordinates on the tool path.

By iterating over each point on the tool path in this manner, the G-code is then saved before being processed through the printer’s GUI program to upload it.

5. RESULTS AND INTERPRETATION

Subsequently, practical aspects of implementation are described. A range of encountered issues are reported and current / future improvements explained.

5.1 Axes and DOF

Figure 5 depicts the outcome of the Prusa i3’s conversion to a 7 DOF printer. All coloured objects were designed

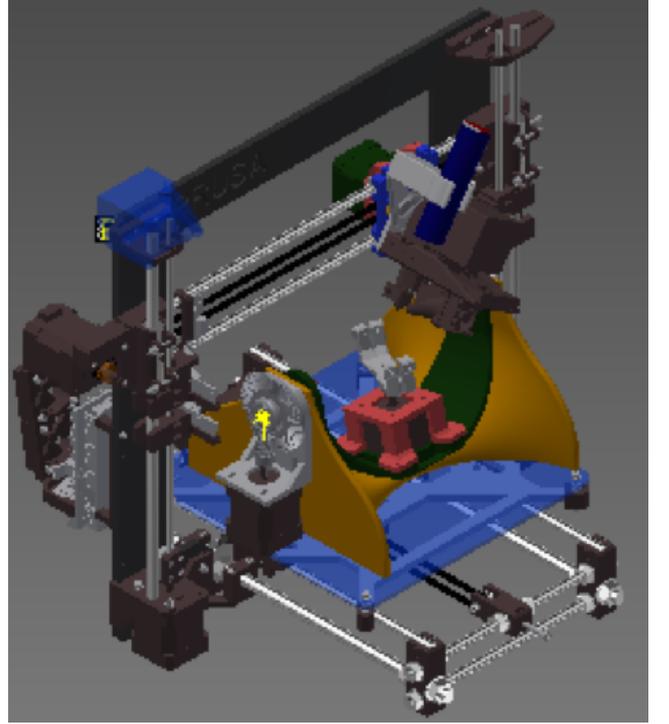


Fig. 5. Modifications (in colour) made to the Prusa i3 to support 7 DOF.

to create the multi-DOF capabilities. Initially all of these parts were made by FDM to reduce the time from design to manufacturing for the necessary changes to the Prusa i3’s original design. This includes the rotational β axis’ coupling to the stepper motor. However, the material’s low rigidity caused the print head to oscillate during normal print manoeuvres. The affected parts were then replaced with machined aluminium parts.

Initial concerns with the α axis not providing enough torque were overcome with the addition of a gear set with a ratio of 4.6:1. The α axis is driven by one stepper motor installed by the side of the orange base plate (see fig. 5) and actuates the green curved ‘swing’, which carries the motor for the γ axis with the gripper and object. The axis is able to be held securely at 90° angles using only 50% of the maximum current. The curved design of this axis gives it a relatively high stability, and there is no significant warping or angular drift between the motorised side of the α axis and the non-motorised side.

5.2 3D scanner and GUI

The 3D scanning camera was mounted onto the frame (light blue top left in fig. 5) and the manually switched blue line laser was mounted onto the X axis (dark blue top right in fig. 5). Tests were performed to determine at what height the laser would need to be mounted. A distance of 100 mm showed the cleanest and clearest part of the laser line drawn on a flat surface with a length of 173 mm and therefore resulted in a good coverage of objects mounted in the gripper. As the distance of the line laser to the object is small, once the laser line focus is calibrated, no further interaction is necessary. For a fully automated 3D printer, the focus should be fixed, and the line laser controlled by the host controller software.

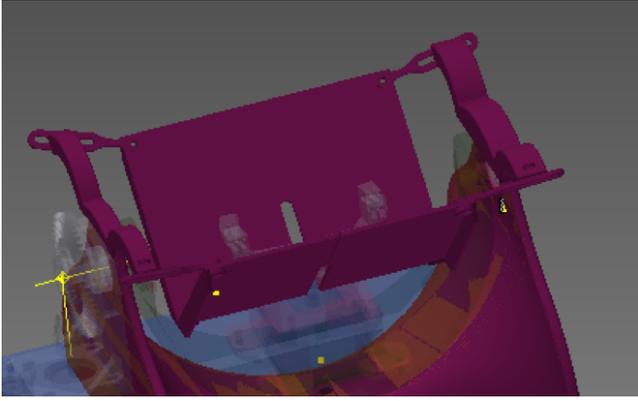


Fig. 6. The calibration panel mounted onto the printer, showing the gripper in light grey in between.

The GUI software, developed to communicate with the modified printer, is work in progress. The current features include: Connect, open, display available COM ports, print, home, home offset, go to home offset, setup scan, scan, slice (including which additional DOF to print with), Cartesian increments in all directions, send line command, recall previous command, display of current G-code file to send, printer communication terminal, and temperature demand. The software addresses, for example, discrepancies due to bugs found in the firmware and overcomes restrictions imposed by other 3D printer control software. A future version will include the following features: 3D printing progress, displaying of the tool path, sliced layer display for error checking, printing simulation, and current temperature graph.

5.3 Placing the object and calibration

The calibration panel for the 3D scan was redesigned for easy removal and mounting for the scan procedure, see fig. 6. The DAVID4 software uses a patterned geometry to correctly determine the size of the scanned object and must be precisely arranged at a 90° angle to determine the correct object surface distances. As the calibration pattern is not in the centre, it allowed space for a small cutaway enabling rotation about the Z axis, α , but only if the gripped object is small enough.

Three manual grippers were designed for object sizes from approximately $120 \times 60 \times 50$ mm to $40 \times 20 \times 10$ mm. Furthermore, due to the gripper arrangement, the current to drive the α axis' stepper motor decreased due to a shift in the centre of mass for that axis. In the future, the construction of the gripper could be further improved, ideally with automated object gripping.

5.4 Scanning and point cloud data

While scanning, preliminary tests were performed. The mounted laser was tested with two different methods of scanning: The first method was a linear sweep, with the laser angled at -30° on the β axis. The X carriage would traverse and the camera would erroneously capture artefacts and produce steps within the surface topology, which occurred at all feed rates. As the linear scanning method produced anomalies, a second method consisting of a combination of a rotary and linear sweep was used

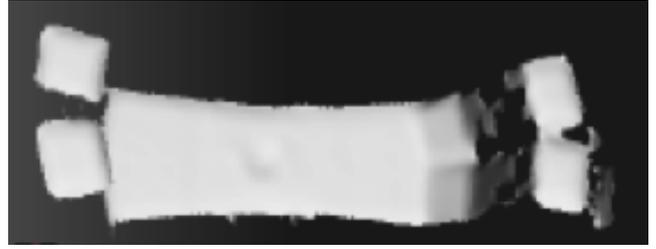


Fig. 7. Detected gripper surface after 3D scanning with a combined linear and rotary line laser sweep.

instead. Figure 7 shows the results after the interpreted geometry was smoothed and averaged in DAVID4.

The camera exposure setting needs to be such that it removes noise and is dependent on the ambient light, laser brightness, distance, beam width, and surface type. A shorter exposure is usually better for noise removal but in turn less of the object is captured. Tests in the current setup found $1/640$ s to be ideal for fast scans for simple shapes, whereas for slower more detailed scans with many passes, the exposure can be set to $1/4480$ s.

5.5 Superimposing object data

The alignment of the scanned part with the reference gripper in Rhino was carried out manually and was a task that required time and patience. In the future, an algorithm could be developed to automate the alignment of scanned part with gripper in a Grasshoppers workflow. Here the rotation tools and detection of the grippers shape could be linked together in such a way that the scanned object gets placed using a 'best fit' method, with minimal user interaction. In comparison to aligning the scanned part with the reference gripper, the orientation and positioning of the object to-be printed is easy to execute.

The perimeter, that is edge of the intersection, is currently generated by specifying each point on the tool path from vertices of the mesh line. Therefore high density meshes have to be used due to the nature of operation. The overall results were accurate.

5.6 Generating tool path data and printing

Once the tool path perimeter has been generated, the print head comes within printing distance of the object, which is 0.2 ± 0.04 mm. A downside of the current printer firmware are jerky movements relating to the acceleration change rate. With a route / path planner, this could be limited. The implementation in another firmware (Marlin) uses the instantaneous change in velocity (the initial and final speed of a vector in path planning) to generate the same principle, which could be adopted for future firmware improvements.

Printing extensive lengths with a single G-code command whilst doing rotary movements does currently not lead to ideal results. If the movement was to follow a straight line whilst rotating around the β axis, the print head nozzle could dip below or raise above the selected line. This led to the design decision to use very small increments between tool path points, which in turn prevents the nozzle from

crashing into the object or going off path. Again, the firmware could be modified to automatically generate the offsets for the print head if a specific G-code command was to be used. This way the print head tip would always remain at the specified Cartesian coordinate.

Also in the future, after calibration of the printer, the print head should follow a ‘keep out’ boundary box during the movements from its home position to the start of tool path which would prevent object collisions.

6. CONCLUSION

Whilst conformal printing onto previously unknown objects is theoretically feasible because all required technologies are available, it is a challenge to compose an affordable system that combines all involved elements. This paper reports on the progress made in an ongoing project aimed at creating a desk-top printer with all required capabilities. After explaining the adopted methodology, technical details of the solution are described, and practical issues are presented together with workarounds. The prototype is able to scan an unknown object to gather point cloud data, superimpose the data with the object to be created on top, generate a tool path, and print the first layer of the new object onto the existing object. Currently, the project team is working on the subsequent layers and technical refinements. Future work will include filling in the currently produced perimeter, much in the same way of conventional slicing, using iso-curves, the utilisation of a SLS projector scanner, and the inclusion of Euclidean / Tait-Bryan rotational angle calculations to utilise all 6 DOF.

ACKNOWLEDGEMENTS

Our sincere thanks go to the DC42 firmware developer David Crocker from the RepRap community.

REFERENCES

- Adams, J., Duoss, E., Malkowski, T., Motala, M., Ahn, B., Nuzzo, R., Bernhard, J., and Lewis, J. (2011). Conformal printing of electrically small antennas on three-dimensional surfaces. *Advanced Materials*, 23(11), 1335–1340.
- Aydar, U., Akyol, O., and Duran, Z. (2011). A low-cost laser scanning system design. In *XXIIIth International CIPA Symposium, Prague*.
- Bourell, D., Stucker, B., Chen, Y., Zhou, C., and Lao, J. (2011). A layerless additive manufacturing process based on cnc accumulation. *Rapid Prototyping Journal*, 17(3), 218–227.
- Brown, A.C. and De Beer, D. (2013). Development of a stereolithography (stl) slicing and g-code generation algorithm for an entry level 3-d printer. In *AFRICON, 2013*, 1–5. IEEE.
- Chen, T. and Shi, Z. (2008). A tool path generation strategy for three-axis ball-end milling of free-form surfaces. *Journal of Materials Processing Technology*, 208(1–3), 259 – 263. doi: <http://dx.doi.org/10.1016/j.jmatprotec.2007.12.142>.
- Choi, J., Medina, F., Kim, C., Espalin, D., Rodriguez, D., Stucker, B., and Wicker, R. (2011). Development of a mobile fused deposition modeling system with enhanced manufacturing flexibility. *Journal of materials processing technology*, 211(3), 424–432.
- Chui, K., Chiu, W., and Yu, K. (2008). Direct 5-axis toolpath generation from point cloud input using 3d biarc fitting. *Robotics and Computer-Integrated Manufacturing*, 24(2), 270 – 286. doi: <http://dx.doi.org/10.1016/j.rcim.2006.11.004>.
- DAVID (2015). DAVID - SLS-2. URL <http://www.david-3d.com/en/products/sls-2>.
- Heidrich, S., Willenborg, E., and Richmann, A. (2011). Development of a laser based process chain for manufacturing freeform optics. *Physics Procedia*, 12, 519–528.
- Lasemi, A., Xue, D., and Gu, P. (2012). A freeform surface manufacturing approach by integration of inspection and tool path generation. *Int. Journal of Production Research*, 50(23), 6709–6725.
- Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., et al. (2000). The digital michelangelo project: 3d scanning of large statues. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 131–144. ACM Press/Addison-Wesley Publishing Co.
- Paulsen, J., Renn, M., Christenson, K., and Plourde, R. (2012). Printing conformal electronics on 3d structures with aerosol jet technology. In *IEEE Future of Instrumentation International Workshop (FIIW), 2012*, 1–4.
- Scheidung, S., Yi, A., Gebhardt, A., Li, L., Risse, S., Eberhardt, R., and Tünnermann, A. (2011). Freeform manufacturing of a microoptical lens array on a steep curved substrate by use of a voice coil fast tool servo. *Optics express*, 19(24), 23938–23951.
- Tam, G., Cheng, Z., Lai, Y., Langbein, F., Liu, Y., Marshall, D., Martin, R., Sun, X., and Rosin, P. (2013). Registration of 3d point clouds and meshes: a survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics*, 19(7), 1199–1217.
- Winkelbach, S., Molkenstruck, S., and Wahl, F.M. (2006). Low-cost laser range scanner and fast surface registration approach. In *Pattern Recognition*, 718–728. Springer.