Doc Code: IDS
Document Description: Information Disclosure Statement Filed

PTO/SB/429 (08-12)
Approved for use through 07/31/2015. OMB 0651-0062
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no persons are required to respond to a collection of information unless it displays a valid OMB control number

THIRD-PARTY SUBMISSION UNDER 37 CFR 1.290

(Do **not** submit this form electronically via EFS-Web)

Application Number (required): 13/462,503

U.S. PATENTS AND U.S. PATENT APPLICATION PUBLICATIONS

Cite No.	Document Number Number-Kind Code[1]	Issue Date or Publication Date MM/DD/YYYY	First Named Inventor
	US-		
	US-		
	US-		
	US-		
	US-		
	US-		
	US-		
	US-		
	US-		

FOREIGN PATENTS AND PUBLISHED FOREIGN PATENT APPLICATIONS

Cite No.	Country or Patent Office and Document Number Country Code[2]-Number[3]-Kind Code[4]	Publication Date MM/DD/YYYY	Applicant, Patentee or First Named Inventor	Translation Attached
				☐
				☐
				☐
				☐
				☐
				☐
				☐
				☐
				☐

1. If known, enter kind of document by the appropriate symbols indicated on the document under WIPO Standard ST.16. See MPEP 901.04(a). 2. Enter the country or patent office that issued the document by two-letter country code under WIPO Standard ST.3. See MPEP 1851. 3. For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. 4. If known, enter kind of document by the appropriate symbols indicated on the document under WIPO Standard ST.16. See MPEP 901.04(a).

Doc Code: IDS
Document Description: Information Disclosure Statement Filed

PTO/SB/429 (08-12)
Approved for use through 07/31/2015. OMB 0651-0062
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no persons are required to respond to a collection of information unless it displays a valid OMB control number

THIRD-PARTY SUBMISSION UNDER 37 CFR 1.290

(Page 2 of 2)

Application Number (required): 13/462,503

NON-PATENT PUBLICATIONS (e.g., journal article, Office action)

Cite No.	Author (if any), title of the publication, page(s) being submitted, publication date, publisher (where available), and place of publication (where available)	Translation Attached	Evidence of Publication Attached
1	AMF Specification Draft v 0.45, published June 23, 2010 at http://ccsl.mae.cornell.edu/sites/default/files/AMF_V0.45.pdf	☐	☒
2	Vijay Chandru et al, Voxel-Based Modeling for Layered Manufacturing, pages 42--47, in IEEE Computer Graphics and Applications, November 1995, IEEE, Piscataway, NJ	☐	☐
3	Miller et al, Development of . . . Phantoms with Rapid Prototyping Technology, pp 4252-4257, IEEE Nuclear Science Symposium Conf. Rcd., 2007, IEEE, Piscataway, NJ	☐	☐
		☐	☐
		☐	☐
		☐	☐
		☐	☐
		☐	☐
		☐	☐
		☐	☐
		☐	☐
		☐	☐

STATEMENTS

The party making the submission is not an individual who has a duty to disclose information with respect to the above-identified application under 37 CFR 1.56.

This submission complies with the requirements of 35 U.S.C. 122(e) and 37 CFR 1.290.

☐ The fee set forth in 37 CFR 1.290(f) is submitted herewith.

■ The fee set forth in 37 CFR 1.290(f) is not required because this submission lists three or fewer total items and, to the knowledge of the person signing the statement after making reasonable inquiry, this submission is the first and only submission under 35 U.S.C. 122(e) filed in the above-identified application by the party making the submission or by a party in privity with the party.

Signature	s/Kit Walsh/	**Date**	2/28/2013
Name (Printed/Typed)	Kit Walsh	**Reg. No., if applicable**	

Examiner Signature*		**Date Considered**	

*EXAMINER: Signature indicates all items listed have been considered, except for citations through which a line is drawn. Draw line through citation if not considered. Include a copy of this form with next communication to applicant.

Privacy Act Statement

The **Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (*i.e.*, GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No:	13/462,503	Confirmation No.: 1020
Inventor(s):	Robert Steingart and David T. Chen	
Filed:	May 2, 2012	
Art Unit:	2127	
Examiner:	Laughlin, Nathan L.	
For:	Fabrication of Non-Homogeneous Articles Via Additive Manufacturing Using Three-Dimensional Voxel-Based Models	
Petitioners:	Electronic Frontier Foundation	

THIRD-PARTY PREISSUANCE SUBMISSION UNDER 37 C.F.R. § 1.290 CONCISE DESCRIPTION OF RELEVANCE

Cite No. 1 – AMF Specification Draft v. 0.45

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Dear Examiner Laughlin:

Listed on accompanying Form PTO/SB/429 are documents that may be considered material to the patentability of this application pursuant to 37 C.F.R. § 1.290. Copies of the patents or publications cited are enclosed, except as waived by 37 C.F.R. § 1.290(d)(3).

In accordance with 37 C.F.R. § 1.290(d)(2), Petitioners' undersigned representative submits the following concise description of relevance for the AMF reference, Cite No. 1 on Form PTO/SB/429:

The AMF reference discloses a modeling format for non-homogeneous 3D fabrication systems with materials specified by their physical properties (such as their attributes and color) or as composites thereof (including the capability to specify that a material is porous). *See* AMF at 1, 4–6. The method of manufacturing suggested by the AMF reference is similar to the technique for designing and manufacturing non-homogeneous 3D models by mapping physical properties to materials as disclosed in

¶¶ 34–45 of the Specification and recited by Claims 12–30 of the instant Application. Specifically, "[a]dditional material properties can be specified using the <metadata> element, such as . . . elastic properties for equipment that can control such properties." AMF at 4. This is analogous to the physical properties specified by Claims 12–30. The AMF draft allows the specification of the elastic modulus and Poisson ratio of materials. AMF at 10. The AMF draft also provides a specific tag defining "the color . . . to print if supported[.]" AMF at 8. Further, the AMF reference discloses texture maps defined as a 3D array of RGBA values specifying both color and transparency of each pixel. AMF at 5. This is analogous to the voxel map with physical properties defined for each voxel as specified by Claims 12–30. Other physical properties which may be specified include material mixes and gradients and porous materials. AMF at 5. AMF notes that "additive manufacturing technology is quickly evolving . . . to producing multi-material geometries in full color with functionally graded materials and microstructures." AMF at 1. This is analogous to the transfer function recited by claims 12–15 and 30, as it suggests that manufacturing systems are able to select materials based on function and color.

Should Examiner or the Office find that the above statement of relevance, or any portion thereof, is non-compliant with some requirement of 37 C.F.R. § 1.290, Petitioners respectfully request the third-party submission be entered if the error is of such minor character that it does not raise an ambiguity as to the content of the submission. *See* 70 Fed. Reg. 42,150, 42,168 (July 17, 2012).

Respectfully submitted,

ELECTRONIC FRONTIER FOUNDATION

By its counsel,

s/Kit Walsh/

Kit Walsh
Clinical Instructional Fellow, Cyberlaw Clinic,
Berkman Center for Internet and Society
Harvard Law School
23 Everett Street, 2nd Floor
Cambridge, MA 02138
Phone: (617) 495-7547
Fax: (617) 495-7641

Date: February 28, 2013

Designation: F XXXX – 10

Standard Specification for Additive Manufacturing File Format (AMF)[1]

1. Scope[1]

For the last three decades, the STL file format has been the industry standard for transferring information between design programs and additive manufacturing equipment. As additive manufacturing technology is quickly evolving from producing primarily single-material, homogenous shapes to producing multi-material geometries in full color with functionally graded materials and microstructures, there is a growing need for a standard interchange file format that can support these features. An STL file contains information only about a surface mesh, and has no provisions for representing color, texture, material, substructure, and other properties of the fabricated target object. This standard describes a framework for a interchange format to address the current and future needs of additive manufacturing technology.

The AMF file may be prepared, displayed, and transmitted on paper or electronically, provided the information required by this specification is included. When prepared in a structured electronic format, strict adherence to an XML schema is required to support standards-compliant interoperability. The Adjunct to this specification contains a W3C XML schema and Annex A1 contains an Implementation Guide for such representation.

This standard does not purport to address manufacturing safety and data security concerns, if any, associated with its use. It is the responsibility of the user of this standard to establish appropriate security, safety and health practices and determine the applicability of regulatory limitations prior to use.

This standard also does not purport to address any copyright and intellectual property concerns, if any, associated with its use. It is the responsibility of the user of this standard to meet any intellectual property regulations on the use of information encoded in this file format.

2.1 Contributors

This standard has been prepared based on a survey and consensus among stakeholders representing designers, equipment manufacturers, CAD software developers, and academicians. A list of contributors and supporters is provided in Appendix 2.

2. Key considerations

There is a naturally a tradeoff between the generality of a file format, and its usefulness for a specific purpose. Thus, features designed to meet the needs of one community may hinder the usefulness of a file format for other uses. In order to be successful across the field of additive manufacturing, this file format is designed to address the following concerns:

2.1 Technology independence: The file format shall describe an object in a general way such that any machine can build it to the best of its ability. It is resolution and layer-thickness independent, and does not contain information specific to any one manufacturing process or technique. This does not negate the inclusion of properties that only certain advanced machines support (for example, color, multiple materials, etc.), but these are defined in such away to avoid exclusivity.

2.2 Simplicity: The AMF file format is easy to implement and understand. The format can be read and debugged in a simple ASCII text viewer to encourage understanding and adoption. No identical information is stored in multiple places.

2.3 Scalability: The file format scales well with increase in part complexity and size, and with the improving resolution and accuracy of manufacturing equipment. This includes being able to handle large arrays of identical objects, complex repeated internal features (e.g. meshes), smooth curved surfaces with fine printing resolution, and multiple components arranged in an optimal packing for printing.

2.4 Performance: The file format should enable reasonable duration (interactive time) for read and write operations and reasonable file sizes for a typical large object. Detailed performance data is provided in the appendix.

---

[1] This specification is under the jurisdiction of ASTM Committee F42 on Additive Manufacturing Technologies and is the direct responsibility of Subcommittee F42.4 task group on Data Interchange.

2.5 Backwards compatibility: Any existing STL file can be converted directly into a valid AMF file without any loss of information and without requiring any additional information. AMF files are also easily converted back to STL for use on legacy systems, although advanced features will be lost. This format maintains the triangle-mesh geometry representation in order to take advantage of existing optimized slicing algorithm and code infrastructure already in existence.

2.6 Future compatibility: In order to remain useful in a rapidly changing industry, this file format is easily extensible while remaining compatible with earlier versions and technologies. This allows new features to be added as advances in technology warrant, while still working flawlessly for simple homogenous geometries on the oldest hardware.

3. Structure of this standard

Information specified throughout this standard is stored in XML format [1]. XML is an ASCII text file comprising a list of elements and attributes. Using this widely accepted data format opens the door to a rich host of tools for creating, viewing, manipulating, parsing and storing AMF files. XML is human-readable, which makes debugging errors in the file possible. XML can be compressed and/or encrypted if desired in a post-processing step using highly optimized standardized routines.

Another significant advantage of XML is its inherent flexibility. Missing or additional parameters do not present a problem for a parser as long as the document conforms to the XML standard. Practically, this allows new features to be added without needing to update old versions of the parser, such as in legacy software.

Precision

This file format is agnostic as to the precision of the representation of numeric values. It is the responsibility of the generating program to write as many or as few digits as are necessary for proper representation of the target object. However, a parsing program should read and process real numbers in double precision (64 bit).

Future amendments and additions

Additional XML elements can be added provisionally to any AMF file for any purpose but will not be considered part of this standard. An unofficial AMF element can be ignored by any reader, and does not need to be stored or reproduced on output. An element becomes official only when it is formally accepted into this standard.

Terminology

This section provides definitions of terms specific to this standard—these terms also include the common terms seen in many documents related to XML and Additive Manufacturing. See also Annex for definitions of additional terms specific to this specification.

Attribute—a characteristic of data, representing one or more aspects, descriptors, or elements of the data. In object-oriented systems, attributes are characteristics of objects. In XML, attributes are characteristics of elements.

Comments—all text comments associated with any data within the AMF not containing core relevant, technical, or administrative data, and not containing pointers to references external to the AMF.

Domain-specific applications—additional, optional sets of AMF data elements specific to such areas as novel additive manufacturing processes, enterprise workflow and supply chain management. Data sets for optional AMF domain-specific applications will be developed and balloted separately from this specification.

Extensible markup language (XML)—a standard from the WorldWideWeb Consortium (W3C) that provides for tagging of information content within documents, offering a means for representation of content in a format which is both human and machine readable. Through the use of customizable style sheets and schemas, information can be represented in a uniform way, allowing for interchange of both content (data) and format (metadata).

STL (file format)—a file format native to the stereolithography CAD software. This file format is supported by many software packages; it is widely used for rapid prototyping and computer-aided manufacturing. STL files describe only the surface geometry of a three dimensional object as a tessellation of triangles, without any representation of color, texture or other common CAD model attributes. The STL format specifies both ASCII and binary representations.

4. General structure

The AMF file begins with the XML declaration line specifying the XML version and encoding, for example:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Blank lines and standard XML comments can be interspersed in the file and will be ignored by any interpreter, for example

```
<!-- ignore this comment -->
```

The remainder of the file is enclosed between an opening `<amf>` element and a closing `</amf>` element. These elements are necessary to denote the file type, as well as to fulfill the requirement that all XML files have a single root element. The version of the AMF standard as well as all standard XML namespace declarations can be used, such as the Designed to identify the human language used. The unit system can also be specified (mm, inch, ft, meters or micrometers). In absence of a units specification, millimeters are assumed.

```
<AMF units="mm" version="1.0" xml:lang="en">
```

Within the AMF brackets, there are five top level elements:

1. `<object>` The object element defines a region or regions of material, each of which are associated with a material ID for printing. At least one object element must be present in the file. Additional objects are optional.
2. `<material>` The optional material element defines one or more materials for printing with an associated material ID. If no material element is included, a single default material is assumed.
3. `<texture>` The optional texture element defines one or more images or textures for color or texture mapping, each with an associated texture ID.
4. `<constellation>` The optional constellation element hierarchically combines objects and other constellations into a relative pattern for printing. If no constellation elements are specified, each object element will be imported with no relative position data. The parsing program can determine the relative positioning of the objects if more than one object is specified in the file.
5. `<metadata>` The optional metadata element specifies additional information about the object(s) contained in the file.

Only a single `object` element is required for a fully functional AMF file.

5. Geometry specification

The top level `<object>` element specifies a unique `id`, and contains two child elements: `<vertices>` and `<region>`. The `<object>` element can optionally specify a material.

The required `<vertices>` element lists all vertices that are used in this object. Each vertex is implicitly assigned a number in the order in which it was declared, starting at zero. The required child element `<coordinates>` gives the position of the point in 3D space using the `<x>`, `<y>` and `<z>` elements.

After the vertex information, at least one `<region>` element must be included. Each region encapsulates a closed volume of the object, Multiple regions can be specified in a single object. Regions may share vertices at interfaces but may not have any overlapping volume.

Within each region, the child element `<triangle>` shall be used to define triangles that tesselate the surface of the region. Each `<triangle>` element will list three vertices from the set of indices of the previously defined vertices. The indices of the three vertices of the triangles are specified using the `<v1>`, `<v2>` and `<v3>` elements. The order of the vertices must be according to the right-hand rule, such that vertices are listed in counter-clockwise order as viewed from the outside. Each triangle is implicitly assigned a number in the order in which it was declared, starting at zero.

```
<?xml version="1.0" encoding="UTF-8"?>
<amf units="mm">
  <object id="0">
    <mesh>
      <vertices>
        <vertex>
          <coordinates>
            <x>0</x>
            <y>1.32</y>
            <z>3.715</z>
          </coordinates>
        </vertex>
        <vertex>
          <coordinates>
            <x>0</x>
            <y>1.269</y>
            <z>2.45354</z>
          </coordinates>
        </vertex>
        ...
      </vertices>

      <region>
        <triangle>
          <v1>0</v1>
          <v2>1</v2>
          <v3>3</v3>
        </triangle>
        <triangle>
          <v1>1</v1>
          <v2>0</v2>
          <v3>4</v3>
        </triangle>
        ...
      </region>
    </mesh>
  </object>
</amf>
```

Figure 1. A basic AMF file containing only a list of vertices and triangles. This structure is compatible with the STL standard.

Smooth geometry

By default, all triangles are assumed to be flat and all triangle edges are assumed to be straight lines connecting their two vertices. However, curved triangles and curved edges can optionally be specified in order to reduce the number of mesh elements required to describe a curved surface. A curved triangle patch can be recursively subdivided into four triangles by the parsing program in order to generate a set of flat triangles at any desired resolution for manufacturing or display.

To specify curvature, a vertex can optionally contain a child element `<normal>` to specify desired surface normal at the location of the vertex. The normal should be unit length and pointing outwards. If this normal is specified, all triangle edges meeting at that vertex should be curved so that they are perpendicular to that normal and in the plane defined by the normal and the original straight edge.

When the curvature of a surface at a vertex is undefined (for example at a cusp, corner or edge), an `<edge>` element can be used to specify the curvature of a single non-linear edge joining two vertices. The curvature is specified using the tangent direction vectors at the beginning and end of that edge. The `<edge>` element will take precedence in case of a conflict with the curvature implied by a `<normal>` element.

Normals shall not be specified for vertices referenced only by planar triangles. Edge tangents shall not be specified for linear edges.

When interpreting normal and tangents, Hermite interpolation will be used. See Annex 3 for formulae for carrying out this interpolation.

The geometry shall not be used to describe support structure. Only the final target structure shall be described.

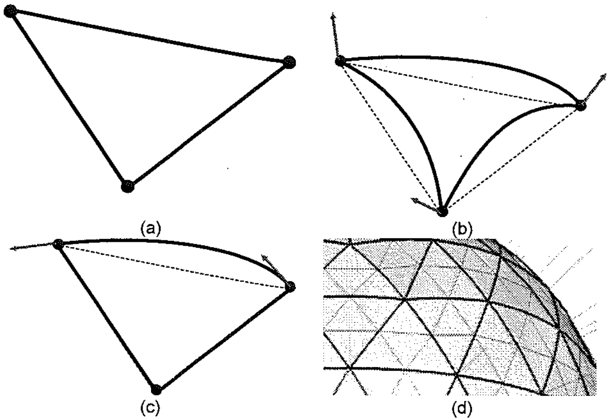
Restrictions on geometry

All geometry must comply with the following restrictions:

1. Every triangle must have exactly three different vertices
2. Triangles may not intersect, except at their common edges
3. Regions must enclose a closed volume
4. Regions may not overlap in their volume
5. Every vertex must be referenced by at least three triangles
6. Every pair of vertices must referenced by zero or two triangles per region
7. No two vertices can have identical coordinate

6. Material specification

Materials are introduced using the `<material>` element. Any number of materials may be defined using the `<material>` element. Each material is assigned a unique `id`. Geometric regions are associated with materials by specifying a `materialid` within the `<region>` element. Any number of materials may be defined. The `materialid` "0" is reserved for no material (void).



```
<?xml version="1.0" encoding="UTF-8"?>
<amf units="mm">
  <object id="0">
    <mesh>
      <vertices>
        <vertex>
          <coordinates >
            ...
          </coordinates >
          <normal>
            <nx>0</nx>
            <ny>0.707</ny>
            <nz>0.707</nz>
          </normal>
        </vertex>
        ...
        <edge>
          <v1>0</v1>
          <dx1>0.577</dx1>
          <dy1>0.577</dy1>
          <dz1>0.577</dz1>
          <v2>1</v2>
          <dx2>0.707</dx2>
          <dy2>0</dy2>
          <dz2>0.707</dz2>
        </edge>
        ...
      </vertices>

      <region materialid="0">
        <triangle>
          ...
        </triangle>
        ...
      </region>
    </mesh>
  </object>
</amf>
```

(e)

Figure 2. Specifying curvature: (a) A default (flat) triangle patch, (b) A triangle curved using vertex normals, (c) a triangle curved using edge tangents. (d) subdivision of a curved triangle patch into four curved sub-patches; (e) An AMF file containing curved geometry.

Material attributes are contained within each `<material>`. The element `<color>` is used to specify the RGBA appearance of the material in sRGB color space [2]. Additional material properties can be specified using the `<metadata>` element, such as the material name for operational purposes, or elastic properties for equipment that can control such properties. See Annex 1 for more information.

```
<?xml version="1.0" encoding="UTF-8"?>
<amf units="mm">
  <material id="1">
    <metadata type="Name">StiffMaterial</metadata>
  </material>
  <material id="2">
    <metadata type="Name">FlexibleMaterial</metadata>
  </material>
  <material id="3">
    <metadata type="Name">MediumMaterial</metadata>
    <composite materialid="1">0.4</composite>
    <composite materialid="2">0.6</composite>
  </material>
  <material id="4">
    <metadata type="Name">VerticallyGraded</metadata>
    <composite materialid="1">z</composite>
    <composite materialid="2">10-z</composite>
  </material>
  <material id="5">
    <metadata type="Name">Checkerboard</metadata >
    <composite materialid="1">
          floor(x+y+z%1)+0.5) </composite>
    <composite materialid="2">
          1-floor(x+y+z%1)+0.5) </composite>
  </material>

  <object id="0">
    <mesh>
      <vertices>
        ...
      </vertices>
      <region materialid="1">
        ...
      </region>
      <region materialid="2">
        ...
      </region>
    </mesh>
  </object>
</amf>
```

Figure 3. Homogenous and composite materials. An AMF file containing five materials. Material 3 is a 40/60% homogenous mixture of the first two materials. Material 4 is a vertically graded material; Material 5 has a periodic checkerboard substructure.

Mixed and graded materials and substructures

New materials can be defined as compositions of other materials. The element `<composite>` is used to specify the proportions of the composition, as a constant or as a formula dependent of the *x, y,* and *z* coordinates. A constant mixing proportion will lead to a homogenous material. A coordinate-dependent composition can lead to a graded material. More complex coordinate-dependent proportions can lead to nonlinear material gradients as well as periodic and non-periodic substructure. The proportion formula can also refer to a texture map using the `tex(textureid,x,y,z)` function (See Annex 1).

Any number of materials can be specified. Any negative material proportion value will be interpreted as a zero proportion. Material proportions shall be normalized to determine actual ratios.

Porous materials

Reference to `materialid "0"` (void) can be used to specify porous structures. The proportion of void can be either 0 or 1. Any fractional value will be interpreted as 1 (i.e. any fractional void will be assumed fully void).

7. Color specification

Colors are introduced using the `<color>` element by specifying the red, green, blue and alpha (transparency) values in sRGB color space [2]. The `<color>` element can be inserted at the material level to associate a color with a material, at the object level to color an entire object, at the region level to color an entire region, or at a vertex level to associate a color with a particular vertex.

Object color overrides material color specification; a region color overrides an object color, and vertex colors override region colors.

Graded colors and texture mapping

A color can also be specified by referring to a RGBA texture map or a grayscale map. The `<color>` element can then be used to refer to a coordinate within that `textureid`. A grayscale value will be used to modulate the base color of the region, object or material, or of white color if no base color is specified.

When the vertices of a triangle have different colors, the interior color of the triangle will linearly interpolate between those colors. The interpolation will be in the color space or in the texture space, depending on the way the color was specified.

Texture specification

The `<texture>` element can be used to associate a `textureid` with particular texture data. The texture map size will be specified and both 2D and 3D maps are supported. The data will be encoded string of bytes in Base64 encoding, as either RGBA values or as a grayscale values. RGBA values will be encoded as 4-byte quadruplets specifying each pixel color and transparency in the 0-255 range. Grayscale will be encoded as a string of individual bytes, one per pixel, specifying the grayscale level in the 0-255 range. The ordering of data will start with the top left corner and proceeding left to right then top to bottom. A 3D texture will specify the first layer initially and repeat for all subsequent depths. The data will be truncated or appended with zero values as needed to meet the specified texture size.

```
<?xml version="1.0"?>
<amf units="mm">

  <material id="1">
    <metadata type="Name">StiffMaterial</metadata>
      <color>
        <r>0</r>
        <g>0.2</g>
        <b>0.8</b>
      </color>
  </material>

  <texture id="1" width="10" height="26" type="rgba">
    TWFuIGlzIGRpc3RpbmdlaXNoZWQsIG5vdCB
    vbmx5IGJ5IGhpcyByZWFzb24sIGJ1dCBieS
    B0aGlzIHNpbmdlbGFyIHBhc3Npb24gZnJvb
    SBvdGhlciBhbmltYWxzLCB3aGljaCBpcyBh
    ...
  </texture>


  <object id="0">
    <mesh>
      <vertices>
        <vertex>
          <coordinates>
            ...
          </coordinates>
          <color textureid="1">
            <tx>5</tx>
            <ty>8</ty>
          </color>
        </vertex>
        ...
      </vertices>
      <region materialid="1">
        <color>
          <r>0.9</r>
          <g>0.9</g>
          <b>0.2</b>
          <a>0.8</a>
        </color>
        ...
      </region>
    </mesh>
  </object>
</amf>
```

Figure 4. Color Specification. Absolute color can be associated with a material, a region or a vertex. A vertex can also be associated with a coordinate in a color texture file.

8. Print Constellations

Multiple objects can be arranged together using the `<constellation>` element. A constellation can specify the position and orientation of objects to increase packing efficiency and to describe large arrays of identical objects. The `<instance>` element specifies the displacement and rotation an existing objects needs to undergo into its position in the constellation. The displacement and rotation are always defined relatively to the original position and orientation in which the object was originally defined. Rotation angles are specified in degrees and are applied first to rotation about the *x* axis, then about the *y* axis, and then about the *z* axis.

A constellation can refer to another constellation. However, recursive or cyclic definitions of constellations are not allowed.

When multiple objects and constellations are defined in a single file, only the top level objects and constellations are available for printing.

The orientation at which the objects will be printed will default to those specified in the constellation. The *z* axis is assumed to be the vertical axis, with the positive direction pointing upwards and zero referring to the printing surface. The *x* and *y* directions will correspond to the main build stage axes if a gantry positioning system is used.

```
<?xml version="1.0"?>
<amf units="mm">
  <object id="1">
      ...
  </object>
  <constellation id="2">
    <instance objectid="1">
      <deltay>5</deltay>
      <rz>90</rz>
    </instance>
    <instance objectid="1">
      <deltax>-10</deltay>
      <deltay>10</deltay>
      <rz>180</rz>
    </instance>
    ...
  </constellation>

</amf>
```

Figure 5. Print constellations. A constellation can assemble multiple objects together.

9. Meta-data

The `<metadata>` element can optionally be used to specify additional information about the objects, geometries and materials being defined. For example, this information can specify a name, textual description, authorship, copyright information and special instructions. The `<metadata>` element can be included at the top level to specify attributes of the entire file, or within objects, regions and materials to specify attributes local to that entity. Annex 1 lists reserved metadata types and their meaning.

```
<?xml version="1.0"?>
<amf units="mm">
  <metadata type="Description">Product 123</metadata>
  <metadata type="Author">John Smith</metadata>
  <metadata type="CAD">SolidX 2.2</metadata>
  <metadata type="Name">Part 1</metadata>
  <metadata type="Revision">1.3A</metadata>
        ...
  <object ObjectID="0">
     <metadata type="Name">Component 1</metadata>
        ...
  </object>
</amf>
```

Figure 6. Meta Data. Additional information can be stored about the object using the metadata element.

10. Compression and distribution

An AMF shall be stored either in plain text or be compressed. If compressed, the compression shall be in ZIP archive format [4] and can be done manually or at write time using any one of several open compression libraries, such as [5].

Both the compressed and uncompressed version of this file will have the AMF extension and it is the responsibility of the parsing program to determine whether or not the file is compressed, and if so to perform decompression during read.

Additional files may be included in the ZIP archive, such as manifest files and electronic signatures. However only the AMF file with the same name as the archive file will be parsed. Absence of a file with that name will constitute an error.

This standard does not specify any explicit mechanisms for ensuring data integrity, electronic signatures and encryptions.

11. Tolerances, textures, and additional information

It is recognized that there is additional information relevant to the final part that is not covered by this standard. This includes information such as surface texture and dimensional tolerances. Elements for this information are omitted from this revision in order to simplify initial adoption, but are expected to be added to future revisions of this standard.

12. References

1. W3C Extensible Markup Language (XML) 1.0 (Fifth Edition) http://www.w3.org/TR/REC-xml/
2. A Standard Default Color Space for the Internet - sRGB, http://www.w3.org/Graphics/Color/sRGB
3. See sample code for implementation of an AMF/STL viewer and converter (BSD Open source) at: http://www.mae.cornell.edu/ccsl/amf
4. ZIP File Format Specification, PKWARE Inc, http://www.pkware.com/documents/casestudies/APPNOTE.TXT
5. See zip libraries such as info-zip (http://www.info-zip.org/

13. Annex 1 - AMF Elements

Element	Parent element(s)	Attributes	Multi elements?	Description
<amf>			No	Root XML element
		units		The units to be used. May be "in" for inches, "mm" for millimeters, "m" meters, "ft" for feet, or "micron" for micrometers.
<object>	<amf>		Yes	An object definition
		id		A unique ObjectID for the new object being defined
<color>	<object> <region> <material> <vertex>		No	The color to display the object in, and to print if supported
		textureid		The texture id of an existing texture if one is to be used as a color map
<r>, <g>, <b>, <a>	<color>		No	Red, Greed, Blue and Alpha (transparency) component of a color in sRGB space, values ranging from 0 to 1.
<mesh>	<object>		Yes	A 3D mesh hull
<vertices>	<mesh>		No	The list of vertices to be used in defining triangles
<vertex>	<vertices>		Yes	A vertex to be referenced in triangles
<coordinates>	<vertex>		No	Specifies the 3D location of this vertex
<x>, <y>, <z>	<coordinates>		No	X, Y, or Z coordinate, respectively, of a vertex position in space
<tx>, <ty>, <tz>	<color>		No	X, Y, or Z coordinate, respectively, within a texture map. The Z coordinate will be ignored if referring to a 2D texture map. Unspecified coordinates will assume a zero value.
<normal>	<vertex>		No	Specifies the 3D normal of the object surface at this vertex
<edge>	<vertices>		No	Specifies the 3D tangent of an object edge between two vertices
<dx1>,<dy1>,<dz1> <dx2>,<dy2>,<dz2>	<edge>		No	The normalized X, Y, or Z component (respectively) of the first or second edge direction vector
<nx>, <ny>, <nz>	<normal>		No	The normalized X, Y, or Z component (respectively) of a surface normal at a vertex
<region>	<mesh>		Yes	Defines a region from the established vertex list
		materialid		Which material id to use
<triangle>	<region>		Yes	Defines a 3D triangle from three vertices, according to the right-hand rule (counter-clockwise when looking from the outside)
<v1>, <v2>, <v3>	<triangle> <edge>			Index of the desired vertices in a triangle or edge
<texture>			Yes	Specifies an texture file or data to be used as a map. Lists a sequence of Base64 values specifying RGBA values for pixels from top to bottom left to right

		id		Assigns a unique texture id for the new texture
		width		Width (horizontal size, x) of the texture, in pixels.
		height		Depth (lateral size, y) of the texture, in pixels.
		depth		Height (vertical size, z) of the texture, in pixels.
		type		Encoding of the data in the texture. Allowed values are either "rgba" or "grayscale". In rgba mode, each sequence of four bytes in the data represent the four values of Red, Greed, Blue and Alpha (transparency) as values in the range of 0-255. If grayscale mode is used, each pixel is represented by one byte in the range of 0-255. If the texture is referenced using the tex function, these values are converted into a single floating point number (see Annex 2).
<material>	<amf>		Yes	An available material
		id		A unique material id. material ID "0" is reserved to denote no material (void) or sacrificial material.
<composite>	<material>		Yes	Compose existing materials. Value provides a numeric constant or mathematical function of coordinates x, y, z specifying the proportion of material of type MaterialID. If the value is negative, is assumed to be zero. The proportions are normalized to add up to 1, unless they are all zero, in which case no material is specified (void). Reference to MaterialID "0" implies reference to no material (void). The void material cannot be mixed. See Annex 2 for a list of allowable mathematical functions
		materialid		Reference to an existing material. Reference cannot be recursive or cyclic.
<constellation>	<amf>		Yes	A collection of objects or constellations with specific relative locations
		id		The Object ID of the new constellation being defined.
<instance>	<constellation>		Yes	An instance of an object or constellation to print
		objectid		The ObjectID of the existing object or constellation being instantiated. Recursive or cyclic references are not allowed.
<deltax>, <deltay>,<deltaz>	<instance>		No	The distance of translation in the x, y, or z direction, respectively, in the referenced object's coordinate system, to create an instance of the object in the current constellation
<rx>,<ry>,<rz>	<instance>		No	The rotation, in degrees, to rotate the referenced object about its x, y, and z axes, respectively, to create an instance of the object in the current constellation. Rotations shall be executed in order of x first, then y, then z.
<metadata>	<amf>, <object>, <region>, <material>, <vertex>		Yes	Specify additional information about an entity.
		type		The type of the attribute. Reserved types are: "Name" - The alphanumeric label of the entity, to be used by the interpreter if interacting with the user. "Description" - A description of the content of the entity "URL" - A link to an external resource relating to the entity "Author" - Specifies the name(s) of the author(s) of the entity "Company" - Specifying the company generating the entity "CAD" - specifies the name of the originating CAD software and version "Revision" - specifies the revision of the entity "Tolerance" - specifies the desired manufacturing tolerance of the entity in entity's unit system "Volume" - specifies the total volume of the entity, in the entity's unit system, to be used for verification (object and region only)

				"Elastic Modulus" - specifies the elastic modulus of the entity, in SI units (material only) "Poisson Ratio" - specifies the Poisson Ratio of the material, in SI units (material only)

14. Annex 2 - Mathematical operations and functions

Precedence	Operator	Description
1	()	Parentheses block
2	^	Power
3	*	Multiply
3	/	Divide
3	%	Modulus, including fractional
4	+	Add
4	-	Subtract
5	=	Equal*
5	<, <=	Less than (or equal to)*
5	>, >=	Greater than (or equal to)*
6	&	Intersection (Logical AND)*
6	|	Union (Logical OR)*.
6	\	Difference (Logical XOR)*
6	!	Negation (Logical NOT)*
6	sin(x)	Sine, radians
6	cos(x)	Cosine, radians
6	tan(x)	Tangent, radians
6	asin(x)	Arc sine, radians
6	acos(x)	Arc cosine, radians
6	atan(x)	Arc tangent, radians
6	floor(x)	Round down to nearest integer
6	ceil(x)	Round up to nearest integer
6	sqrt(x)	Square root
6	log(x)	Natural logarithm
6	exp(x)	Natural exponent
6	abs(x)	Absolute value
6	max(x,y)	Maximum value
6	min(x,y)	Minimum value
6	rand(x)	Generates a real (fractional) random number uniformly distributed in the range [0-x) (exclusive of x). The random seed is randomized when the AMF file is first opened.
6	tex(textureid,x,y,z) tex(textureid,x,y)	Returns a scalar value in the range [0-1] that interpolates the texture with the textureid at the coordinate (x,y,z) for 3D textures and (x,y) for 2D textures. if the texture is of type "grayscale", the range [0-1] corresponds to [0-255] in the texture data. If the texture is of type "rgba", the function will return the floating point number represented by the four bytes. Whole coordinate values refer to the center of the texture map pixels. If the values are fractional, linear interpolation shall be used. If the coordinates fall outside the texture, they shall wrap around. If the texture is two dimensional and a z coordinate is specified, the *z* coordinate shall be ignored.

* Logical operators returns a Boolean value of either 1 or 0, representing TRUE and FALSE, respectively. When processing non Boolean numbers as Boolean values, a zero value represents FALSE, and a nonzero value represents TRUE.

15. Annex 3 - Formulae for interpolating a curved triangular patch

Non-planar triangular patches with specified vertex normals or edge tangents shall be interpolated from their three endpoints and six tangent vectors and/or three vertex normals using interpolated Hermite curves, as follows.

1. For each of the three edges of the triangle (refer to Figure 7a):
    a. If the tangents t_0 and t_1 are specified explicitly in an `<edge>` element, skip this step, otherwise compute the tangent vectors t_0 and t_1 such that they are each perpendicular to the normal at their vertex and lay in the

plane defined by that normal and the two vertices. The following formula can be used. Given v_0, n_0 and v_1, n_1, define $d=v_1-v_0$, and

$$t_0 = |d|\frac{-(n_0 \times d)\times n_0}{\|(n_0 \times d)\times n_0\|},\quad t_1 = |d|\frac{-(n_1 \times d)\times n_1}{\|(n_1 \times d)\times n_1\|}$$

If n_0 is undefined or unspecified, set $t_0=d$

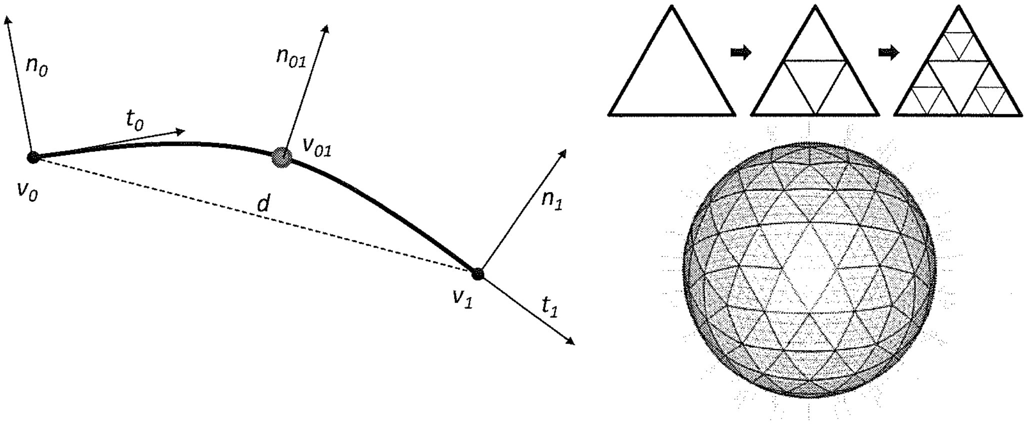
If n_1 is undefined or unspecified, set $t_1=d$

b. Compute the center point $v_{01}=h(0.5)$ using 2nd order Hermite curve interpolation

$h(s)=(2s^3-3s^2+1)v_0+(s^3-2s^2+s)t_0+(-2s^3+3s^2)v_1+(s^3-s^2)t_1$

c. Compute the center normal n_{01} by averaging the two end normals and renormalizing

2. Using the three new vertices and normals, split the triangle into four sub-triangles
3. Repeat the above procedure recursively for each triangle until the desired display or manufacturing precision is reached or until no significant improvement is obtained (refer to Figure 7b).



(a) (b)

Figure 7. Interpolating a curved triangle edge. (a) Notation used for the subdivision of a curve; (b) Triangles shall be divided recursively to the desired resolution as determined by the display or manufacturing tolerances. Example of a spherical surface represented with 320 triangles, each subdivided into 16 sub-triangles by using the above procedure recursively.

16. Appendix 1 - Performance

It is the goal of this standard to allow for interactive time performance for file read-write and for reasonable file sizes for typical large datasets. The table below summarizes performance statistics for a range of file sizes. The processing times refer to time used to read the file, parse the xml objects and construct an internal data structure [3]. Note that read and parse time are a relatively small portion of the time to process a file for fabrication (e.g. slicing).

16.1 File Size

Number of Triangles	Binary STL (uncompressed)	Binary STL (compressed)	AMF (uncompressed)	AMF (compressed)
1,016,388	49.6 Mb	25.3 Mb	205.9 Mb	12.2 Mb
100,536	4.9 Mb	2.3 Mb	20.1 Mb	1.2 Mb
10,592	518 K	249 K	2.1 Mb	129 K
1,036	51 K	20 K	203 K	12 K

16.2 Write time (seconds)

Number of Triangles	Binary STL (uncompressed)	Binary STL (compressed)	AMF (uncompressed)	AMF (compressed)
1,016,388	0.372	~3.4	6.8	15.5
100,536	0.038	0.038	0.79	1.78
10,592	0.005	0.005	0.11	0.21
1,036	0.001	0.001	0.06	0.06

16.3 Read and parse time (seconds)

Number of Triangles	Binary STL (uncompressed)	Binary STL (compressed)	AMF (uncompressed)	AMF (compressed)
1,016,388	0.384	~1.3	6.447	6.447
100,536	0.043	0.043	0.669	0.687
10,592	0.005	0.005	0.107	0.107
1,036	0.001	0.001	0.056	0.056

16.4 Accuracy (error calculated on unit sphere)

Number of Triangles	STL	AMF (with normals)
20	0.102673	0.006777
80	0.032914	0.000788
320	0.008877	8.28E-05
1,280	0.001893	1.01E-05
5,120	0.000455	1.95E-06
20,480	1.13E-04	4.51E-07
81,920	2.81E-05	1.11E-07
327,680	7.03E-06	2.75E-08
1,310,720	1.76E-06	6.87E-09

17. Appendix 2 - List of contributors and stakeholder liaisons

This standard has been prepared based on a survey and consensus among stakeholders representing designers, AM equipment manufacturers, CAD software developers, academicians, and end users. The following is an list of contributors and contact persons. Please contact hod.lipson@cornell.edu to be added to this list or to suggested people and companies who should be contacted about being added to the list. We are looking for representatives from all major stakeholders.

Additive-Manufacturing Equipment Manufacturers contact persons

Name	Affiliation	Contact
Fjordén, Tony	Arcam	tony.fjorden@arcam.com
Napadensky, Eduardo	Objet	Eduardo.Napadensky@objet.com
Sprauer, Jean	Z Corp	jsprauer@zcorp.com
Lewis, Cathy	3D Systems	lewiscl@3DSystems.Com

Robb, Patrick	Stratasys	Patrick.Robb@stratasys.com
Semmler, Manfred	EOS	manfred.semmler@eos.info
Wigand, John	Solidscape	jwigand@solid-scape.com
Such, Alberto	HP	alberto.such@hp.com

CAD Software Companies contact persons

Name	Affiliation	Contact
Brans, Karel	Materialise	Karel.Brans@materialise.be
Martinez, Gonzalo	AutoDesk	gonzalo.martinez@autodesk.com
Ding, Joe	SolidWorks	jding@solidworks.com
Trainer, Asa	Parametric Technologies	ATrainer@ptc.com
Bacus, John	Google SketchUp	jbacus@google.com

Academia

Name	Affiliation	Contact
Hiller, Jonathan	Cornell University	jdh74@cornell.edu
Lipson , Hod	Cornell University	hod.lipson@cornell.edu
Stucker, Brent	University of Louisville	brent.stucker@usu.edu

Societies and organizations

Name	Affiliation	Contact
Wellington, Jane	Society of Manufacturing Engineers	JWellington@sme.org
Bowyer, Adrian	RepRap	adrian.reprap@googlemail.com
Malone, Evan	Nextfab Organization	evan.malone@nextfab.org
Lipton Jeffrey	Fab@Home	jil26@cornell.edu
Barkley, Jim	MITRE Corporation	jbarkley@mitre.org

End users: Designers, consultants, service providers, and other users

Name	Affiliation	Contact
Robert Schouwenburg	Shapeways	robert@shapeways.com
Overy Charles	LGM	cwho@lgm3d.com
Francis Rabuck	Bentley Systems	Francis.Rabuck@bentley.com

* Please contact hod.lipson@cornell.edu to be added to this list or to be removed from this list

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No:	13/462,503	Confirmation No.: 1020
Inventor(s):	Robert Steingart and David T. Chen	
Filed:	May 2, 2012	
Art Unit:	2127	
Examiner:	Laughlin, Nathan L.	
For:	Fabrication of Non-Homogeneous Articles Via Additive Manufacturing Using Three-Dimensional Voxel-Based Models	
Petitioners:	Electronic Frontier Foundation	

THIRD-PARTY PREISSUANCE SUBMISSION UNDER 37 C.F.R. § 1.290
AFFIDAVIT CONCERNING EVIDENCE OF PUBLICATION

Cite No. 1 – AMF Specification Draft v. 0.45

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Dear Examiner Laughlin:

Listed on accompanying Form PTO/SB/429 are documents that may be considered material to the patentability of this application pursuant to 37 C.F.R. § 1.290. Copies of the patents or publications cited are enclosed, except as waived by 37 C.F.R. § 1.290(d)(3).

Petitioners' undersigned representative submits the attached document, a true and correct copy of the web page http://creativemachines.cornell.edu/amf , as evidence of publication for the AMF reference, Cite No. 1 on Form PTO/SB/429. The web page document indicates on Page 1 that version 0.45 of the AMF Specification was published June 23, 2010.

The copy of the AMF Specification Draft submitted by Petitioners' undersigned representative is a true and correct copy of the document Petitioners' undersigned representative downloaded from the attached web page document's PDF link labeled June 23, 2010.

I hereby declare under penalty of perjury that the foregoing is true and correct.

Respectfully submitted,

ELECTRONIC FRONTIER FOUNDATION

By its counsel,

s/Kit Walsh/

Kit Walsh
Clinical Instructional Fellow, Cyberlaw Clinic,
Berkman Center for Internet and Society
Harvard Law School
23 Everett Street, 2nd Floor
Cambridge, MA 02138
Phone: (617) 495-7547
Fax: (617) 495-7641

Date: February 28, 2013

Cornell Creative Machines Lab

AMF

NOTE: All AMF files and downloads have been shifted to the amf wiki (http://amff.wikispaces.com/ (http://amff.wikispaces.com/)). All current information is available there. This page is preserved for historical reasons only.

This page contains information and resources regarding the AMF File format, intended to supercede the STL file format with features relevant for current and future Additive Manufacturing processes. This work is in conjunction with the current ASTM Committee *F42* on Additive Manufacturing Technologies, specifically the Design Task Group on File Formats, chaired by Hod Lipson (Contact (http://www.mae.cornell.edu/lipson/)).

Community

Join the google group to learn and discuss more about the stl 2.0 Proposal: http://groups.google.com/group/stl2 (http://groups.google.com/group/stl2)

ASTM Activity

AMF Draft Presentation

(Loughborough University, July 8, 2010)

Link to Powerpoint (http://ccsl.mae.cornell.edu/sites/default/files/AM2010_lipson.ppt) file. (*"Draft Additive Manufacturing Format"*)

AMF Specification Drafts

To officially comment on the current draft visit http://www.astm.org/DATABASE.CART/WORKITEMS/WK27506.htm (http://www.astm.org/DATABASE.CART/WORKITEMS/WK27506.htm) (you need to log in to view and officially comment)

To informally comment you can also use http://groups.google.com/group/stl2 (http://groups.google.com/group/stl2)

- These are old versions and should not be implemented. The current AMF spec can be obtained at http://www.astm.org/Standards/F2915.htm (http://www.astm.org/Standards/F2915.htm)
- V. 0.47: PDF (sites/default/files/AMF_V0.47.pdf) or DOC (sites/default/files/AMF_V0.47.doc) (Feb 3, 2011)
- V. 0.46: PDF (sites/default/files/AMF_V0.46.pdf) or DOC (sites/default/files/AMF_V0.46.doc) (Dec 1, 2010)
- V. 0.45: PDF (http://ccsl.mae.cornell.edu/sites/default/files/AMF_V0.45.pdf) or DOC (http://ccsl.mae.cornell.edu/sites/default/files/AMF_V0.45.doc) (June 23, 2010)
- V. 0.44: PDF (http://ccsl.mae.cornell.edu/sites/default/files/AMF_V0.44.pdf) or DOC (http://ccsl.mae.cornell.edu/sites/default/files/AMF_V0.44.doc) (June 14, 2010)
- V. 0.43: PDF (http://ccsl.mae.cornell.edu/sites/default/files/AMF_V0.43.pdf) or DOC (http://ccsl.mae.cornell.edu/sites/default/files/AMF_V0.43.doc) (June 11, 2010)
- V. 0.41: PDF (http://ccsl.mae.cornell.edu/sites/default/files/AMF_V0.41.pdf) or DOC (http://ccsl.mae.cornell.edu/sites/default/files/AMF_V0.41.doc) (May 26, 2010)
- V. 0.40: PDF (http://ccsl.mae.cornell.edu/sites/default/files/AMF_V0.4.pdf) or DOC (http://ccsl.mae.cornell.edu/sites/default/files/AMF_V0.4.doc) (May 24, 2010)

- Take the STL 2.0 survey: http://www.mae.cornell.edu/lipson/stl2.htm (http://www.mae.cornell.edu/lipson/stl2.htm) (still open)
- STL 2.0 Survey Results (as of Nov 9, 2009) and preliminary ideas: Presentation made to the ASTM F42 commitee (http://ccsl.mae.cornell.edu/sites/default/files/ASTM_presentation.ppt) Nov 10, 2009 by Hod Lipson

AMF Resources

STL to AMF converter

A standalone open source windows executable that views and converts stl files to amf (v 0.44) files and back. Only basic (stl equivalent) amf features are implemented. Note: This is pre-release software which is subject to change at any time, and may not be compatible with the future ASTM approved amf file spec.

Download

- AMF_Dev_44exe.zip (v. 0.44) 6-17-2010
- Source (VS 2008 project, requires QT 4.6)

Bugs and feature requests may be directed to Jon Hiller (jdh74@cornell.edu)

Sample Files

Download

Rook.amf*	40kb
Rook.stl (http://ccsl.mae.cornell.edu/sites/default/files/rook_0.STL) (binary)	180kb
Rook_Ascii.stl (http://ccsl.mae.cornell.edu/sites/default/files/Rook_Ascii.stl) (ascii)	976kb

(3682 Triangles)

Sphere20.amf*^	1k
Sphere80.amf*^	2k
Sphere320.amf*^	7k

*To view the human readable xml tags, open with an unzipping program. (In windows manually change the extension to *.zip)
^Includes information for curved-triangle processing

Icon

Icon to associate with AMF files

Download

AMF.ico (http://ccsl.mae.cornell.edu/sites/default/files/AMF.ico)

Other resources

- W3C Extensible Markup Language (XML) 1.0 (Fifth Edition) http://www.w3.org/TR/REC-xml/ (http://www.w3.org/TR/REC-xml/)
- A Standard Default Color Space for the Internet - sRGB, http://www.w3.org/Graphics/Color/sRGB (http://www.w3.org/Graphics/Color/sRGB)
- ZIP File Format Specification, PKWARE Inc, http://www.pkware.com/documents/casestudies/APPNOTE.TXT (http://www.pkware.com/documents/casestudies/APPNOTE.TXT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No:	13/462,503	Confirmation No.: 1020
Inventor(s):	Robert Steingart and David T. Chen	
Filed:	May 2, 2012	
Art Unit:	2127	
Examiner:	Laughlin, Nathan L.	
For:	Fabrication of Non-Homogeneous Articles Via Additive Manufacturing Using Three-Dimensional Voxel-Based Models	
Petitioners:	Electronic Frontier Foundation	

THIRD-PARTY PREISSUANCE SUBMISSION UNDER 37 C.F.R. § 1.290
CONCISE DESCRIPTION OF RELEVANCE

Cite No. 2 – Voxel-Based Modeling for Layered Manufacturing by Chandru et al.

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Dear Examiner Laughlin:

Listed on accompanying Form PTO/SB/429 are documents that may be considered material to the patentability of this application pursuant to 37 C.F.R. § 1.290. Copies of the patents or publications cited are enclosed, except as waived by 37 C.F.R. § 1.290(d)(3).

In accordance with 37 C.F.R. § 1.290(d)(2), Petitioners' undersigned representative submits the following concise description of relevance for the Chandru reference, Cite No. 2 on Form PTO/SB/429:

Chandru discloses a voxel-based design tool and modeling method for layered manufacturing. *See* Chandru at 42. Such a method is similar to the design application described in ¶ 20 of the Specification and recited by Claims 12-15 and 30 of the instant Application. Such a method is also similar to the 3D voxel-based model described in ¶¶ 14-20 of the Specification and recited by Claims 12-30 of the instant Application. Specifically, Chandru teaches a method for modeling non-homogeneous articles "with materials selectively placed at individual voxels." Chandru at 45. This method is analogous to the "one or more physical properties" associated with the voxel elements in the claimed invention. Chandru further teaches the use of layered manufacturing equipment to fabricate the designed models. Chandru at 42. This is analogous to the 3D printer/rapid prototyping machine recited by Claims 12-30 of the instant Application.

Chandru discloses the use of voxels at every stage of the design process. He discloses the memory requirements of voxel representations (Chandru at 46) and the use of voxels for mass calculation, interference detection, and constructive solid geometry at the design stage, noting that "a voxel-based system naturally provides a WYSIWYG interface." Chandru at 43–44. This is analogous to the voxel-based design application recited by Claims 12–15 and 30 of the instant Application.

Should Examiner or the Office find that the above statement of relevance, or any portion thereof, is non-compliant with some requirement of 37 C.F.R. § 1.290, Petitioners respectfully request the third-party submission be entered if the error is of such minor character that it does not raise an ambiguity as to the content of the submission. *See* 70 Fed. Reg. 42,150, 42,168 (July 17, 2012).

Respectfully submitted,

ELECTRONIC FRONTIER FOUNDATION

By its counsel,

s/Kit Walsh/

Kit Walsh
Clinical Instructional Fellow, Cyberlaw Clinic,
Berkman Center for Internet and Society
Harvard Law School
23 Everett Street, 2nd Floor
Cambridge, MA 02138
Phone: (617) 495-7547
Fax: (617) 495-7641

Date: February 28, 2013

Voxel-Based Modeling for Layered Manufacturing

Vijay Chandru, Swami Manohar, and C. Edmond Prakash
Indian Institute of Science

Several manufacturing technologies that support rapid prototyping have recently become available both in research laboratories and in the commercial marketplace. These technologies—variously called layered manufacturing, additive manufacturing, and stereolithography—allow a part, prototype, or tool to be built by laying down material in a gradual, controlled way. By contrast, traditional manufacturing methods depend on removing material (as in milling and turning) or deforming it (as in casting and molding). In this article, we use the term layered manufacturing (LM) to denote any of the technologies that support the fast-developing field of rapid prototyping.[1,2]

Layered manufacturing technologies have revolutionized the prototyping of complex geometric designs, but still employ traditional CAD tools. A voxel-based approach is under development in a modeling tool called G-WoRP.

Software efforts in this area have focused so far on ensuring compatibility between existing CAD tools and the new manufacturing process. Because existing CAD tools are geared toward the design of parts manufactured by traditional methods, they do not help designers exploit the expanded design space offered by LM technologies. We are not aware of any work attempting to redefine the role of CAD tools in the new context. Project Maxwell[3] does try to finesse the CAD-tool-based design phase by using shape optimization techniques to go automatically from the functional specification to geometric representation. This approach obviously depends on the ability of LM technologies to manufacture arbitrary shapes.

We believe that a voxel-based approach to geometric modeling has several features that make it close to ideal for exploiting the new technologies of layered manufacturing.

Voxel-based geometric modeling

A voxel represents a volume element in volume graphics, just as a pixel denotes a picture element in raster graphics. *Voxelization* is the process of converting a geometrically represented 3D object into a voxel model. Kaufman[4] proposed that graphics is ready to shift paradigms from 2D raster graphics to 3D volume graphics with implications similar to those of the earlier shift from vector to raster graphics. Volume graphics, voxelization, and volume rendering have attracted considerable research in recent years. All of this work, however, has been directed at the informative display of volume data. We propose a voxel-based approach to geometric modeling for the new LM technologies.

The current range of rapid prototyping machines can be broadly classified based on the way they add material to an object under fabrication:

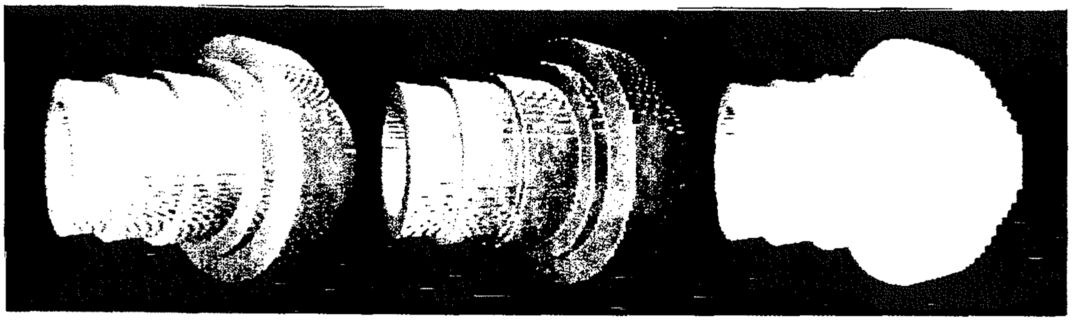
- Sequential/vector-based systems create layers by the sequential formation (by solidification or deposition) of the contours in the object's cross sections. Solid interiors are obtained with a hatching or filling-in operation.
- Parallel/image-based systems use masks to create successive layers of the component. Either a light source solidifies a photopolymer or a sprayer deposits a material on surfaces exposed by the mask. The advantage of this approach is that geometric complexity does not affect the time it takes to complete a layer. Each mask is simply a slice of the object—essentially, the image of the object's cross section.

Though the current installed base of sequential/vector-based systems far exceeds that of parallel/image-based systems, we believe the latter will dominate over the long term. A voxel-based approach is well suited to parallel systems, and the remainder of this section briefly sketches aspects of geometric modeling that can benefit from adopting this approach.

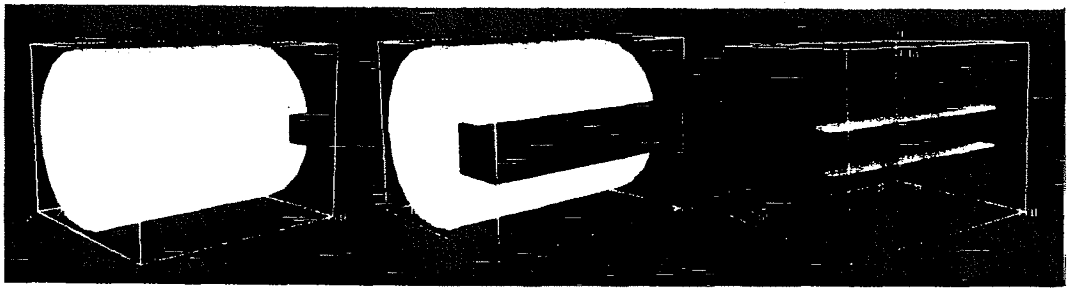
Visual correspondence

The volume-rendered image of a voxel model has direct visual correspondence to the object fabricated using LM equipment. Figure 1 shows an object voxelized using three different resolutions along the z-axis. (We obtained these and other images in this article using Bob and VolVis, two public-domain volume-rendering pack-

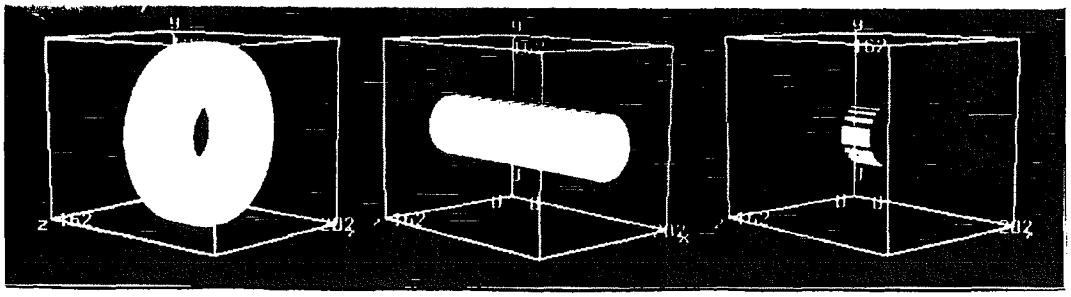
0272-17-16/95/$4.00 © 1995 IEEE



1 Object voxelized using three different resolutions along the z-axis.



2 Interference in a cylinder and key assembly detected by the difference volume.



3 Tolerance variations in washer and peg assembly visible in the interference volume.

ages that employ direct volume rendering.)

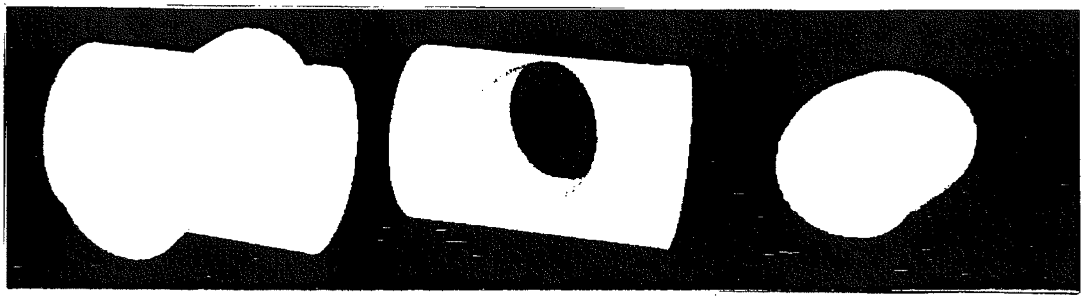
If we assume that the z-axis corresponds to the vertical axis in the LM equipment, then the z-axis step size corresponds to the layer thickness. In Figure 1, the step size or layer thickness on the left is half that in the middle, which is half that on the right. The images show the surface finish as it would appear in the actual object fabricated from the voxel models. Thus, a voxel-based system naturally provides a WYSIWYG interface, whereas a traditional geometry-based modeler displays a smooth, shaded object that gives the designer no feedback on the actual surface finish of the object after fabrication.

Estimating mass properties

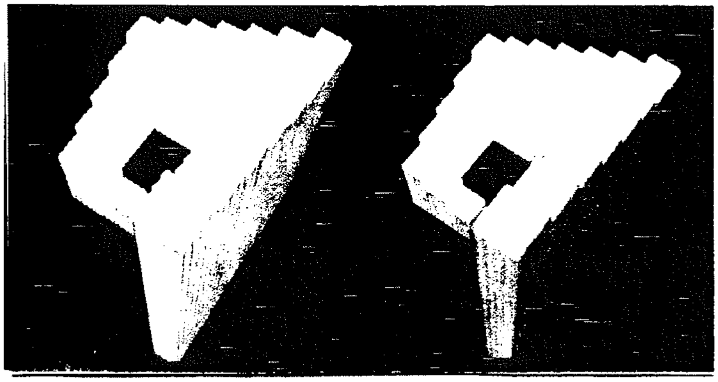
The voxel model lets designers evaluate the mass properties of the modeled object. For instance, the total material volume can be obtained as a simple sum of all the nonzero voxels of the volume buffer, suitably scaled by the parameters of the LM technology (layer thickness, horizontal resolutions, and so forth). This measure is a reasonably accurate estimate of the actual volume of the object, since each voxel translates to a precisely quantifiable unit of material deposited during fabrication. This measure can be applied independent of the object's topology. Prakash and Manohar[5] present a set of volume measures that enable the designer to easily estimate various physical properties of the modeled object. Here we outline two simple applications of these volume measures.

Interference detection. Detecting interference among the components of an assembly is a major problem in computer-aided design of mechanical parts. Analytical methods for computing such interference have been proposed,[6] but they are complicated and do not easily generalize to arbitrary geometries. A volume-error measure—the intersection volume of the two voxel volumes A and B—gives a direct and simple quantification of the interference. Figure 2 displays the interference between a cylinder and sliding key. The complete assembly displayed in the middle does not indicate any problem, but the difference volume on the right clearly shows the extent and location of the interference. This measure can be applied irrespective of the assembly's complexity and is another advantage of a voxel model.

Tolerancing. Tolerancing is an important aspect of mechanical design (for example, see ElMaraghy et al.[6]). In a voxel-based modeler, the intersection of two objects due to variations in tolerance can be visualized as a 3D interference volume itself. Figure 3 shows this in a wash-



4 Voxel-based CSG modeling of union, set difference, and intersection.



5 Staircase voxelized at 1 sample per voxel (left) and 64 samples per voxel (right).

er and peg with tolerance variations in size (diameter of the peg) and in position (axis of the peg). The interference volume due to these tolerance variations is shown on the far right. Tarbox and Gottschlich[7] provide an excellent treatment of the use of voxel models for automated visual inspection, addressing issues of registration and the use of volumetric set operations.

CSG modeling

The suitability of a voxel-based approach to constructive solid geometry (CSG) operations is well documented (for example, see Kaufman[4]). Figure 4 shows three CSG objects rendered using this approach. The union of two off-axis cylinders appears on the left; the set difference is in the middle and their intersection is on the right. Computation of the Boolean operations reduces to voxel-by-voxel logical operations.

Determining layer thickness

The fabrication time for image-based LM technologies depends primarily on the number of layers. As seen with reference to Figure 1, the number of layers determines the smoothness of the surface as well as the fidelity between the modeled and fabricated objects. Reducing the number of layers in fabricating a given model exactly parallels the aliasing problem in 2D graphics. Thus, you can apply antialiased voxelization algorithms when converting the geometric description of an object to a voxel model.

We have developed an antialiasing algorithm[5] that is a simple extension of the accumulation buffer algorithm for 2D polygons. Figure 5 shows a staircase voxelized with one sample per voxel on the left and 64 samples per voxel on the right. In both cases the resolution of the volume buffer is the same.

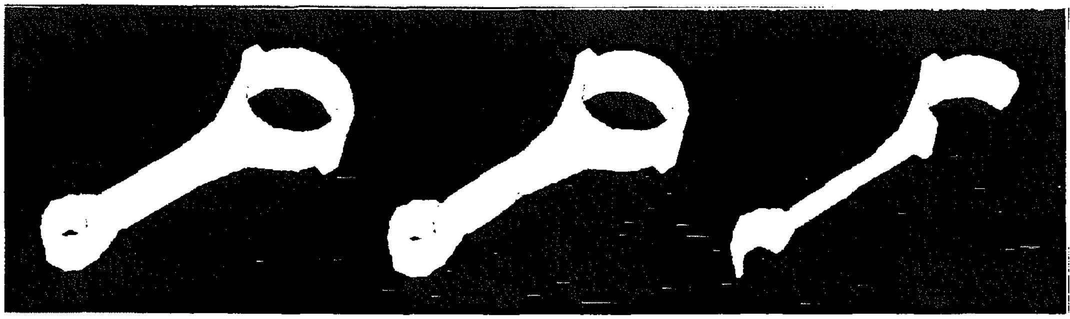
Voxel-level analysis

Figure 6 shows the volume-rendered image of a connecting rod modeled as a voxel array of dimension 200 × 80 × 30. The image in the middle shows the results of free-body vibrational analysis on the model, and a cross section of this image appears on the right. Each voxel has information about the three components of the vibration. By visualizing the analysis results at the voxel level, it is possible to determine critical regions where the vibrations might exceed a specified threshold.

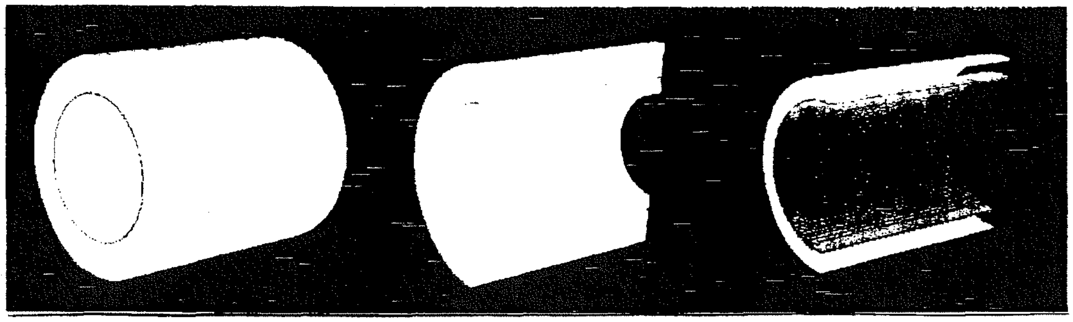
Such analyses and visualizations are not exclusive to voxel-based design tools. In fact, we obtained the connecting rod data in Figure 6 by voxelizing a polyhedral model of the object and the analysis data by applying finite-element analysis on the polyhedral model. The powerful aspect of voxel-based modeling is that the designer can selectively modify individual voxels so that the resulting object meets the design specifications. Conventional modeling does not support this capability. Achieving it in voxel-based modeling, however, requires solutions to two problems: computational analysis of voxel-based models and interactive volume sculpting (discussed later under "Research issues in voxel-based modeling").

The conventional design and analysis cycle does not consider whether a component is intended for fabrication using LM equipment. Consequently, the component is designed using any one of the widely available mechanical CAD packages. The component is modeled, then analyzed using a finite-element analysis module. The results are displayed, the design is iterated to account for analysis results, and the model is output in the .STL format for fabrication. The LM equipment takes the .STL file and generates a set of slices by orienting the object based on constraints and process parameters unique to that LM equipment. The component designer has no information about the choices made during the slicing step. Because of the discretization inherent to LM, the resulting component could have properties different from what the CAD analysis predicted.

A voxel-based approach, in contrast, eliminates the need for an intermediate format as well as for a post-processing step beyond the designer's control. Thus, the results of a voxel-based analysis enable the designer to modify the model appropriately.



6 Variational analysis of a connecting rod.



7 Voxel model of a composite object.

Designing composites

Given solutions to voxel-level computational analysis and interactive sculpting, the voxel-based approach can exploit a major capability of LM equipment: the fabrication of composite objects. The range of materials that current commercial LM systems handle is limited but growing. It is very likely that in the near future, LM technology will mature to fabricate a single component from multiple materials. Conventional design tools are not oriented to the design of composite objects. Specialized tools are used in areas like the aircraft industry where composite materials play a major role. However, advances in LM technologies promise to bring composites into the domain of the average mechanical component. Figure 7 shows a simple composite object created using a voxel model. There are two layers of a different material (blue and red in the figures) reinforcing the bulk material (in yellow).

A voxel-based modeler can ultimately provide the capability to design a composite object with materials selectively placed at individual voxels. There is no need to compute the complex geometries of the interleaved materials because each slice of the voxel buffer can be directly read out during fabrication and several masks per layer can be created to deposit the different materials. Such capabilities will be indispensable as the technology of microelectromechanical systems matures. A voxel will then be of molecular dimensions.

Estimating surface properties

Estimating the surface properties of objects from a voxel model poses another challenge. A solution is feasible because of the direct relationship between a voxel and the basic additive resolution of the LM equipment. This relationship implies that the surface area of the resulting object can be estimated by identifying the exposed voxels in the model, adding the area contributed by the voxel faces on the boundary, and using suitable filters to simulate the effects of merging and coagulation behaviors in the real material. Properties such as friction coefficients, surface roughness, and contact area between interacting parts to be estimated.

Generating slices

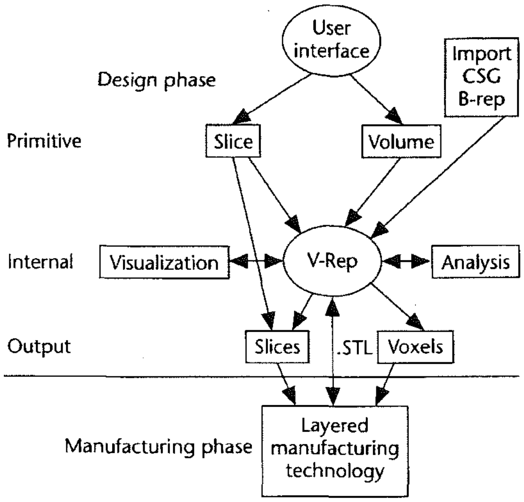
As pointed out earlier, image-based LM systems do not require a slicing step with the voxel-based approach, since the slices in three orthogonal directions are directly available from the voxel volume. However, vector-based LM equipment requires a translation step to extract contours and enclosed regions from the image data of each slice. The obvious drawback is the size of the data, but image compression techniques can mitigate this problem.

Reverse engineering

The combination of LM technologies and volume scanning devices (like CT and MRI) is a powerful reverse-engineering platform. A voxel-based modeling package can import the scanned volume data, perform voxel-level modifications, and then directly fabricate the object, or its tooling, using LM equipment. Custom prostheses, replicas of archaeological artifacts, and retooling for components where a sample is available but the design information is lost are some areas that greatly benefit from this combination.

Research issues in voxel-based modeling

Several bottlenecks, some computational and some algorithmic, must be overcome to realize the potential offered by voxel-based geometric design. We briefly sketch them here and indicate current research and development efforts aimed at their elimination.



8 **G-WoRP architecture.**

Memory

The memory requirements of voxel models are enormous. To store a model of reasonable resolution, say 400 × 400 × 400, in raw form (that is, just as a 3D array of voxels) requires 64 megabytes of space. You can trade computation time for storage space by one of the following means:

- Store the voxel array in compressed form and use algorithms that will operate directly on the compressed data.
- Convert the voxel array into a more compact representation and reconvert into voxels when required. There are several candidate representations: octrees, wavelets, shells, alpha-shapes, and spheres (see Ranjan and Fournier[8] and the references therein).
- Retain the original geometric representation and use voxelization algorithms when necessary. This is especially valuable, since engineering design has a large library of components that can be imported into the voxel-based modeler.

Rendering complexity

To be an effective design tool, a voxel-based system should be able to update the display at interactive rates. Current graphics rendering systems cannot provide rendering performance levels on voxel models comparable to their polygon-rendering performance. This situation, however, will likely change in the near future as current research focused on parallel algorithms and hardware support for volume rendering resolves the problems.

Interactive volume sculpting

One premise underlying a voxel-based modeling and design system is the availability of a powerful interaction paradigm that gives a designer freedom to realize arbitrary shapes. Without this capability, called *interactive sculpting*, most of the advantages of a voxel-based design for rapid prototyping are nullified. Interactive sculpting is currently a highly active research area.[9] Voxel-based sculpting, however, has not received much attention. In a pioneering paper, Galyean and Hughes[10] presented a detailed vision of a voxel-based sculpting system that is a logical extension of a 2D paint program. They described the system as a coarse modeling/sculpting tool and, due to the limited volume resolutions, not a precision modeling tool. No follow-up work on the system has been reported, but our current research platform, described in the next section, extends this approach with a specific focus on layered manufacturing.

Other issues

We have discussed the importance of computational analysis of voxel-based models (see earlier section, "Voxel-level analysis." Efficient algorithms for LM process planning is another challenging research issue, which we address in the next section.

G-WoRP

G-WoRP, a *geometric workbench for rapid prototyping*, is a work in progress to create modeling software tuned toward fabrication with LM equipment. We briefly outline it here and describe it in detail elsewhere.[11]

Figure 8 presents an overview of the G-WoRP architecture. The user interacts with the workbench through an input layer that provides two major primitives—the *slice* and the *voxel*—and the operations that support them. An import facility also permits designs from other CAD systems.

Central to the internal layer is *V-Rep*, a new representation scheme that provides an efficient interface among the various G-WoRP modules. The output layer gives the part description in a form suitable to the actual LM technology employed. It also supports a process description for manufacturing the part using traditional processes.

The design and manufacturing phases do not require an explicit process-planning step because the design description of the part closely resembles the input description needed by the LM equipment. In most LM technologies, the process-planning steps reduce to the following:

- Proper orientation of the workpiece with respect to the machine, based on factors such as production time (which depends on the number of layers), accuracy, and avoidance of overhangs and trapped liquid.
- Modification of the model to compensate for shrinkage, warpage, and so on.
- Possible design of support structures for overhanging parts in some LM technologies. (Others may create the support structure automatically as part of the manufacturing process.)

G-WoRP integrates these process-planning steps in the design phase itself. Since the design develops in terms of slices or voxels, the designer can get immediate feedback about manufacturability. Further, the choice of primitives in G-WoRP obviates the requirement for a slicing processor. The manufacturing process information is made available to the internal layer to enforce constraints on the design as well as to assist in the process-planning steps.

The G-WoRP modeling paradigms based on the slice primitive are closely related to the sweep representations of geometric modeling, which have been thoroughly addressed in the literature. It is the voxel primitive that sets G-WoRP apart from traditional geometric modeling software. Clearly, building up a complex part (or even a simple part) one voxel at a time is tedious, if not totally impractical. We therefore provide several operations that let the designer work with a large chunk of voxels. However, the designer can still choose to modify a few voxels, one at a time if necessary.

The workbench is being implemented in C++ and OpenGL. We have completed the following modules: the slice primitive and its operations, the import facility for CSG models, a new voxelization algorithm with antialiasing for polyhedrally defined objects, and volume measures and their visualization. Our current focus is on voxel operations that include generalized sculpting tools; creation, manipulation, and copying of subvolumes; conversion between representations; and operations on individual layers of a volume. We are also exploring the use of virtual reality interfaces for sculpting.

Conclusion

The voxel-based approach for geometric modeling offers a powerful methodology for the new rapid prototyping technologies. It has several advantages over conventional modeling methods, stemming chiefly from the close resemblance between a voxel model of an object and the object fabricated using an LM technology. G-WoRP is our research vehicle, currently under development, for tackling the several challenging problems that remain to be solved. The design of an interactive environment for voxel sculpting is the critical factor that will bring out the full power of the voxel-based approach to geometric modeling and is the focus of our current efforts. ■

Acknowledgment

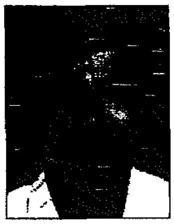
This research is supported in part by grants from the Department of Science and Technology, Government of India, and the All India Council of Technical Education.

References

1. P.F. Jacobs, *Rapid Prototyping and Manufacturing—Fundamentals of Stereolithography*, Soc. of Manufacturing Engineers SME-CASA, Dearborn, Mich., 1992.
2. M. Burns, *Automated Fabrication*, Prentice-Hall, Englewood Cliffs, N.J., 1993.
3. D. Dutta et al., "Project Maxwell: Towards Rapid Realization of Superior Products," *Proc. 1992 Solid Freeform Fabrication Symp.*, Univ. of Texas at Austin, 1992.
4. A. Kaufman, D. Cohen, and R. Yagel, "Volume Graphics," *Computer*, Vol. 26, No. 7, July 1993, pp. 51-64.
5. C.E. Prakash and S. Manohar, "Error Measures and 3D Anti-Aliasing for Voxel Data," *Proc. Pacific Graphics 95*, World Scientific Press, Singapore, 1995, pp. 225-239.
6. W.H. ElMaraghy et al., "Intersection Volumes and Surface Areas of Cylinders for Geometrical Modeling and Tolerancing," *Computer-Aided Design*, Vol. 26, Jan. 1994, pp. 29-45.
7. G.H. Tarbox and S.N. Gottschlich, "IVIS: An Integrated Volumetric Inspection System," *Computer Vision and Image Understanding*, Vol. 61, No. 3, May 1995, pp. 430-444.
8. V. Ranjan and A. Fournier, "Volume Models for Volumetric Data," *Computer*, Vol. 27, No. 7, July 1994, pp. 28-36.
9. Special Issue on Interactive Sculpting, *ACM Trans. Graphics*, Vol. 13, No. 2, Apr. 1994.
10. T.A. Galyean and J.F. Hughes, "An Interactive Volumetric Modeling Technique," *Computer Graphics* (Proc. Siggraph), Vol. 25, No. 4, July 1991, pp. 267-274.
11. V. Chandru and S. Manohar, "G-WoRP: A Geometric Workbench for Rapid Prototyping," *Proc. ASME Int'l Mechanical Engineering Congress*, ASME, New York, 1994.

***Vijay Chandru** is an associate professor in the Department of Computer Science and Automation at the Indian Institute of Science in Bangalore. His research interests are in geometric modeling, computational logic, and combinatorial optimization. Chandru obtained his doctorate at the Massachusetts Institute of Technology in 1982.*

***Swami Manohar** is an assistant professor in the Department of Computer Science and Automation and an associate faculty member of the Supercomputer Education and Research Center at the Indian Institute of Science. His research and teaching interests are in parallel architectures and algorithms, computer graphics, and virtual reality. Manohar received his BE (ECE) from the Government College of Technology, Coimbatore, India, in 1982, his ME (EE) from the Indian Institute of Science in 1984, and his MS and PhD in computer science from Brown University in 1987 and 1989, respectively.*



***C. Edmond Prakash** is a scientific officer at the Supercomputer Education and Research Center, Indian Institute of Science. His research interests are in volume graphics, scientific visualization, and parallel algorithms for graphics. Prakash received his BE (Mech.) from Annamalai University, India, in 1986 and his ME (Engg. Design) from Anna University, India, in 1988.*

Readers may contact Manohar at the Dept. of Computer Science and Automation, Indian Institute of Science, Bangalore 560012, India, e-mail manohar@csa.iisc.ernet.in.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No:	13/462,503	Confirmation No.: 1020
Inventor(s):	Robert Steingart and David T. Chen	
Filed:	May 2, 2012	
Art Unit:	2127	
Examiner:	Laughlin, Nathan L.	
For:	Fabrication of Non-Homogeneous Articles Via Additive Manufacturing Using Three-Dimensional Voxel-Based Models	
Petitioners:	Electronic Frontier Foundation	

THIRD-PARTY PREISSUANCE SUBMISSION UNDER 37 C.F.R. § 1.290 CONCISE DESCRIPTION OF RELEVANCE

Cite No. 3 – "Development of Anatomically Realistic PET and PET/CT Phantoms with Rapid Prototyping Technology" by Miller et al.

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Dear Examiner Laughlin:

Listed on accompanying Form PTO/SB/429 are documents that may be considered material to the patentability of this application pursuant to 37 C.F.R. § 1.290. Copies of the patents or publications cited are enclosed, except as waived by 37 C.F.R. § 1.290(d)(3).

In accordance with 37 C.F.R. § 1.290(d)(2), Petitioners' undersigned representative submits the following concise description of relevance for the Miller reference, Cite No. 3 on Form PTO/SB/429:

Miller discloses a method for fabricating articles using rapid prototyping technology. *See* Miller at 4252. Such a method is similar to the technology described in ¶¶ 14–20 of the Specification and recited by Claims 12–30 of the Application. Specifically, Miller discloses fabrication using "a three-dimensional binary mask of the phantom geometry using cubic voxels." Miller at 4253. This is analogous to the voxel map recited by claims 12–30. Miller discloses that "[b]y adding radioactive dyes to the binder, we can produce parts with highly detailed distributions of radioactivity[,]" and "[t]he phantoms we have printed to date all contained ^{18}F." Miller at 4253, 4256. These distributions of radioactivity are analogous to the varying physical properties recited by claims 12–30 of the Application. Further, Miller discloses that he is in the process of developing "intentionally non-uniform phantoms" printed "not only with multiple colors, but also multiple tracer concentrations." Miller at 4254. These distributions of color and levels of radioactivity are also analogous to the varying physical properties recited by claims 12–30 of the Application.

Should Examiner or the Office find that the above statement of relevance, or any portion thereof, is non-compliant with some requirement of 37 C.F.R. § 1.290, Petitioners respectfully request the third-party submission be entered if the error is of such minor character that it does not raise an ambiguity as to the content of the submission. *See* 70 Fed. Reg. 42,150, 42,168 (July 17, 2012).

Respectfully submitted,

ELECTRONIC FRONTIER FOUNDATION

By its counsel,

s/Kit Walsh/

Kit Walsh
Clinical Instructional Fellow, Cyberlaw Clinic,
Berkman Center for Internet and Society
Harvard Law School
23 Everett Street, 2nd Floor
Cambridge, MA 02138
Phone: (617) 495-7547
Fax: (617) 495-7641

Date: February 28, 2013

Development of Anatomically Realistic PET and PET/CT Phantoms with Rapid Prototyping Technology

Michael. A. Miller and Gary D. Hutchins, *Member, IEEE*

***Abstract*—The utility of PET and PET/CT in research and diagnosis of cancer, cardiac and neurological disorders has been widely demonstrated. Phantoms with well defined geometries that accurately model radiolabeled tracer concentrations and photon attenuation coefficients are suited for characterization of imaging systems, but not as well suited for evaluating methods sensitive to detailed anatomical structure, such as algorithms for monitoring tumor response. An ideal phantom would have the shape and activity distribution of a realistic tumor and would be useful in evaluation of automated image analysis systems. Such a phantom, imaged at sites involved in clinical trials, would be valuable for evaluating consistency and accuracy. We have developed a method of creating such phantoms by incorporating radioactive tracer as dye for a cellulose powder based rapid prototyping system. This allows us to create phantoms with spatial resolution limited only by the stereolithography printer system (slice thickness is 0.18 mm, printing resolution is 600 dpi). We have evaluated the method by printing several small phantoms with 18F and measuring activity in a gamma counter. The relative standard deviation of the activity of multiple identical phantoms was 2%. Activity in unlabeled parts was less than 2% of adjacent labeled regions. We have created and printed realistic phantoms based on the SPL human brain atlas [1, 2], the Paxinos & Watson rat brain atlas [3] and from PET/CT images of human lung nodules, showing that this is a practical method for making complex radioactive phantoms that model real anatomy. We are proceeding with further development to allow us to produce phantoms with multiple activity concentrations, tunable photon attenuation coefficients and long lived isotopes.**

I. INTRODUCTION

THE utility of positron emission tomography (PET) and x-ray computed tomography (CT) in research and diagnosis of cancer, cardiac and neurological disorders has been well demonstrated. The availability of highly detailed, anatomically correct imaging phantoms is expected to enable the continued expansion and improvement of *quantitative* PET and PET/CT studies. This paper introduces a method of creating such phantoms using rapid prototyping technology (3D printing). We first give a summary of phantom characteristics and the limitations imposed by traditional fabrication methods. We then describe our method of fabricating phantoms with 3D printing. Finally we present results of our preliminary quantitative evaluation of these phantoms along with examples of the phantoms.

Phantoms used to evaluate quantitative performance of imaging systems must have well defined geometries made up of regions containing radioactivity and/or contrast material. These geometries are relatively easy to manufacture using traditional methods. Radioactivity is typically introduced into the phantom by filling hollow volumes with activity in a liquid mixture. In many cases this is a water mixture, while in some it is an epoxy mixture that will set to make a solid phantom. For example, a hot tumor is often modeled as a spherical volume contained in a uniform background. Images of such phantoms lend themselves to analysis that provides reproducible measurements of system characteristics.

Some characteristics of such idealized geometries lead to phantoms that significantly differ from realistic anatomy. Anatomical and pathological structures do not occur in idealized shapes — for example, no tumor is truly spherical. Fillable objects have walls separating them from their surroundings. Making these walls small enough to be below the spatial resolution limits of imaging systems usually makes them fragile and difficult (or impossible) to manufacture. For nuclear medicine and PET phantoms, these walls do not contain radioactive material, resulting in cold boundaries between regions. In large phantoms, such as those modeling whole human organs, these cold boundaries are often negligible. In small phantoms, such as those modeling small animal organs and human organ structures, the wall thickness may often be considerably larger than the object being modeled. This lack of realistic shapes and small scale structures results in phantoms that are unsuited for evaluating imaging techniques sensitive to this level of detail. The ability to make phantoms with realistic geometries that match the detailed structure of the bodies and organs they are intended to model would allow us to investigate resolution effects, and correct for them, in a way that is otherwise impossible. The 3D printing method that we present here provides just that ability.

II. PHANTOM MAKING WITH 3D PRINTING

3D printing (3DP) is a form of rapid prototyping technology that creates three dimensional objects by solidifying layers of deposited powder using a liquid binder [4]. The printer we are using, a Spectrum Z510 from Z Corporation, is capable of printing parts as large as 254 mm × 356 mm × 203 mm (10" × 14" × 8"). Parts are built up by spreading a cellulose-based powder (zp15e, Z Corporation) layer upon layer, each 0.18 mm thick (0.007"), as illustrated in Fig 1. Each layer is

This work was supported in part by the Indiana Genomics Initiative (INGEN, supported in part by Lilly Endowment, Inc.)

Authors are from the Department of Radiology, Indiana University School of Medicine, Indianapolis, IN 46202 USA (telephone: 317-278-0141, e-mail: mmiller3@iupui.edu, gdhutchi@iupui.edu).

selectively hardened by applying a colored pattern of binder with inkjet print heads. The printer resolution is 600×540 dpi on each layer. Printing speed is 2-4 layers per minute, 20-40 mm per hour.

Phantoms are printed from digital models that are created either with standard CAD software, from segmented image data, or a combination of both. For example, a part might be created from a surface model generated from a segmented chest CT image, or from an MR image of the brain. Printing parameters are set to produce a uniform binder concentration throughout the solidified portions of the phantom. By adding radioactive dyes to the binder, we can produce parts with highly detailed distributions of radioactivity.

To date all printing has been done with ^{18}F in zb 60 binder (Z Corporation). Starting with a 740 MBq dose of [^{18}F]FDG (20 mCi) in 500 mL of binder (1.5 MBq/mL), the resulting phantom activity concentration is approximately 60 kBq/mL at end of printing when phantoms are ready to be imaged. Several small, uniform spherical phantoms are included with each print job. They are assayed in a gamma counter to measure the activity concentration and to evaluate uniformity.

III. RESULTS AND EXAMPLE PHANTOMS

Activity concentrations were determined by assaying 500 μL binder samples and 5 mm diameter spherical phantoms. After decay-correcting the results to the time at which the print job was completed and the phantoms were ready to image, the activity concentration in three spherical phantoms was 63.3 ± 1.5 kBq/mL (mean $\pm$ standard deviation). Activity concentration in four binder samples was 517 ± 32 kBq/mL resulting in a part:binder activity ratio of 0.123 ± 0.008. This ratio is consistent across multiple print jobs.

A. Image Quality Phantom

A PET image of an example phantom is presented in Fig. 2. This phantom is similar to the NEMA image quality phantom [6], with hot spheres 10, 13, 17 and 22 mm in diameter. It was created by generating a three-dimensional binary mask of the phantom geometry using cubic voxels 1 mm on a side. This mask loaded into Slicer (www.slicer.org) and thresholded to create surface models representing hot and cold spheres and the shell surrounding the phantom body. In the printed phantom, the shell serves as a container to hold the unbound/unradiolabeled powder. The models were exported to vtk files and converted to stereolithography format (stl) using the VTK libraries (www.vtk.org). STL files are directly readable by the printer software. Overall phantom thickness was 41 mm. Total time to print this phantom was approximately 1.5 hours. The image was acquired with a Biograph-16 HIREZ PET/CT (Siemens Medical) and reconstructed with our standard whole body protocol: 3 minutes per bed position, OSEM2D with two iterations and eight subsets.

B. Human Brain Phantom

A human brain phantom was created using data from the SPL brain atlas [1, 2]. Brain structures were segmented and converted to VTK surface models using Slicer. The resulting surface models, shown with the phantom in Fig. 3, were converted from vtk to stereolithography format and used as input to the printer. Cortical structures, the cerebellum and striata were labeled with 53 kBq/mL of [^{18}F]FDG (decay corrected to the start of the image acquisition). The phantom consisted of a 15 mm thick slab with caps above and below to contain the unbound powder surrounding the hot structures. The phantom was imaged on a Biograph-16 HIREZ PET/CT (Siemens Medical) using our standard PET/CT adult brain imaging protocol (PET image: 10 minute acquisition, OSEM2D with 2 iterations, 8 subsets, CT image: 3 mm slice thickness, 120 kVp, reconstruction with H31s convolution kernel). PET and CT images of the phantom are presented in Fig. 4. Note that while there is good contrast in the CT image, the actual CT numbers range from -600 to -500.

C. Human Lung Phantom With Nodule

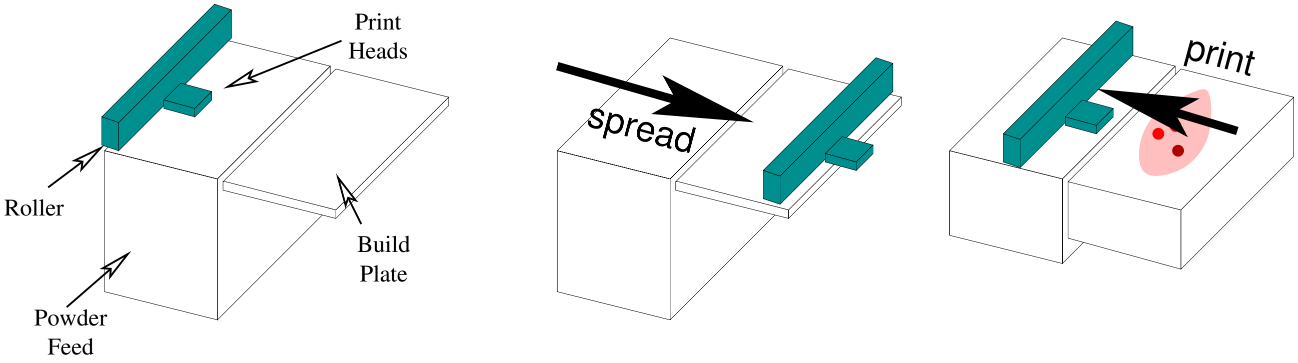
Our experience with the brain phantom showed that the x-ray attenuation coefficient of unbound powder is similar to that of lung, indicating that this method may be useful for making realistic PET/CT lung phantoms. A lung phantom was created using data from a clinical PET/CT study acquired with the Biograph-16 HIREZ PET/CT. CT data was segmented to represent soft tissue, the lung nodule, and bone. The digital model and photographs of the phantom shown in Fig. 5 illustrate the capability of the method for producing small anatomical structures.

D. Rat Brain Phantoms

A digital brain map created from the rat brain atlas by Paxinos and Watson [3] was used to produce rat brain phantoms. Following procedures similar to those used for the human brain phantoms, we created surface models as illustrated in Fig. 6. Multiple FDG-labeled phantoms were printed with activity in the cortex, striata and cerebellum. Phantoms with certain structures left cold were printed to model metabolic ablation studies. Fig. 7 shows images of the rat brain phantom along with images of phantoms printed with no activity in the right striatum and right motor cortex. The PET images were acquired on the Indiana Small Animal PET scanner (IndyPET-III) [7] and reconstructed using filtered backprojection. The capability to make phantoms that accurately reproduce very small structures such as these allows us to perform highly detailed investigations of quantitative imaging. These phantoms have been used to evaluate the quantitative accuracy of reconstruction methods as reported elsewhere in this proceedings [8].

IV. DISCUSSION

The use of 3D printing technology for producing imaging phantoms has allowed us to create structures with a degree of detail that was previously unachievable. Initial measurements indicate that the method allows us to produce phantoms with concentration variations of less than or equal to 2%, which is better than or similar to what is achievable with currently



Components of the 3D printer. Spreading a layer of powder. Printing a cross sectional layer.

Fig. 1. Schematic illustrating the principles of 3D printing. Powder layers are prepared by lowering the build plate by 0.007”, raising the powder feed piston and spreading powder with the roller. The powder is selectively hardened by applying binder with the inkjet print heads [5]. The process is repeated layer by layer until the entire part has been produced.

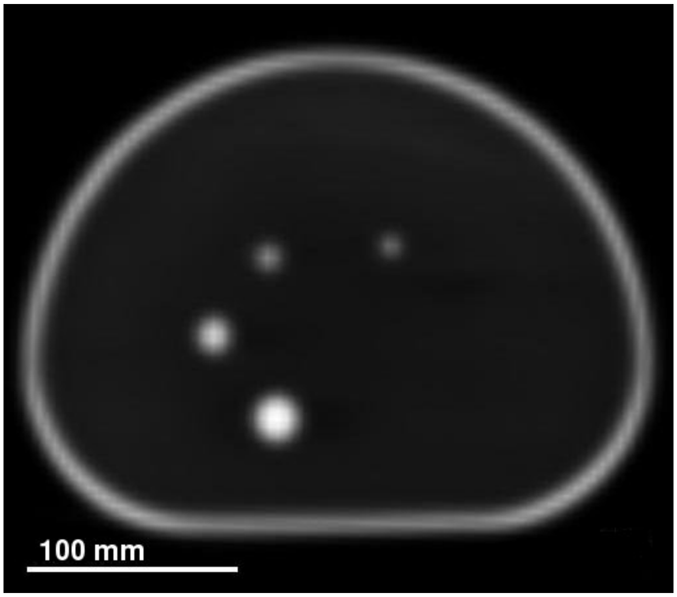


Fig. 2. Torso-sized phantom, similar to NEMA NU-2 2007 image quality phantom.

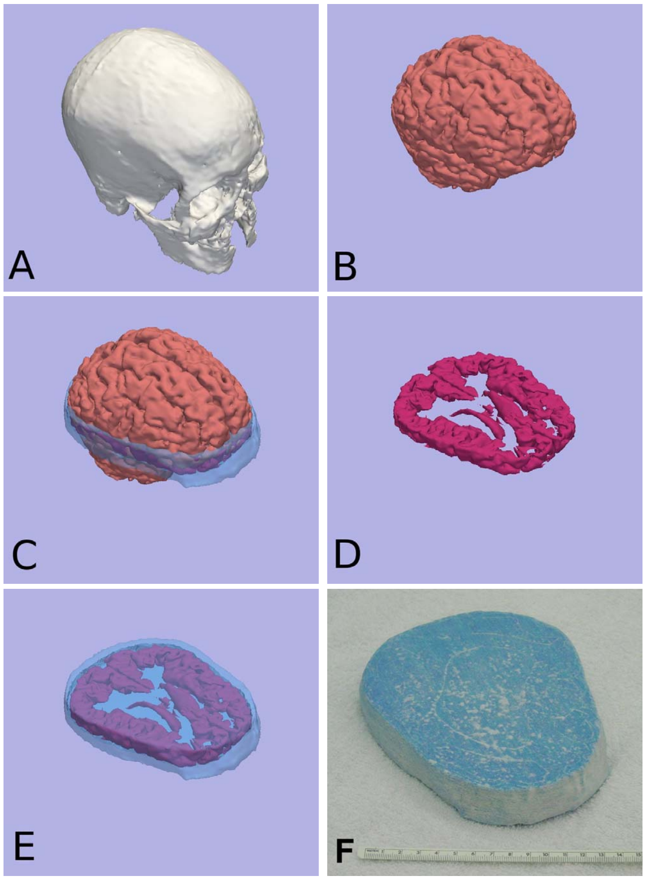


Fig. 3. Human brain phantom. Panels A-E show the digital models used to create the phantom. A, B and C show the relative positions of the skull, brain and phantom slab. D and E show the radiolabeled portion of the phantom and the case. Panel F is a photograph of the phantom.

available epoxy-based solid phantoms. We are continuing to investigate phantom uniformity. In addition, we would like to be able to print intentionally non-uniform phantoms. While the Spectrum Z510 printer is a color printer, using clear, cyan, magenta and yellow binders to produce colors, it prints color only on the surface and applies clear binder to provide structure to the interior of parts. Colored binder is applied only to the outer 1-2 mm of part edges, making it impossible to print color throughout the part. We are working with the manufacturer to write software that will allow us to print multiple colors throughout the part volume. This will allow us to print not only with multiple colors, but also multiple tracer concentrations, resulting in phantoms that more accurately reproduce physiologically relevant tracer distributions. For example, phantoms could be printed that include a background activity concentration and hot spots representing hypermetabolic regions such as tumors, myocardium or brain.

Another area where we are working to improve the phantoms is in phantom density. The density of the cellulose-based zp15e powder is approximately 0.5 g/mL, with an x-ray attenuation coefficient that results in CT numbers near -600. After printing, while the bound parts of the phantoms

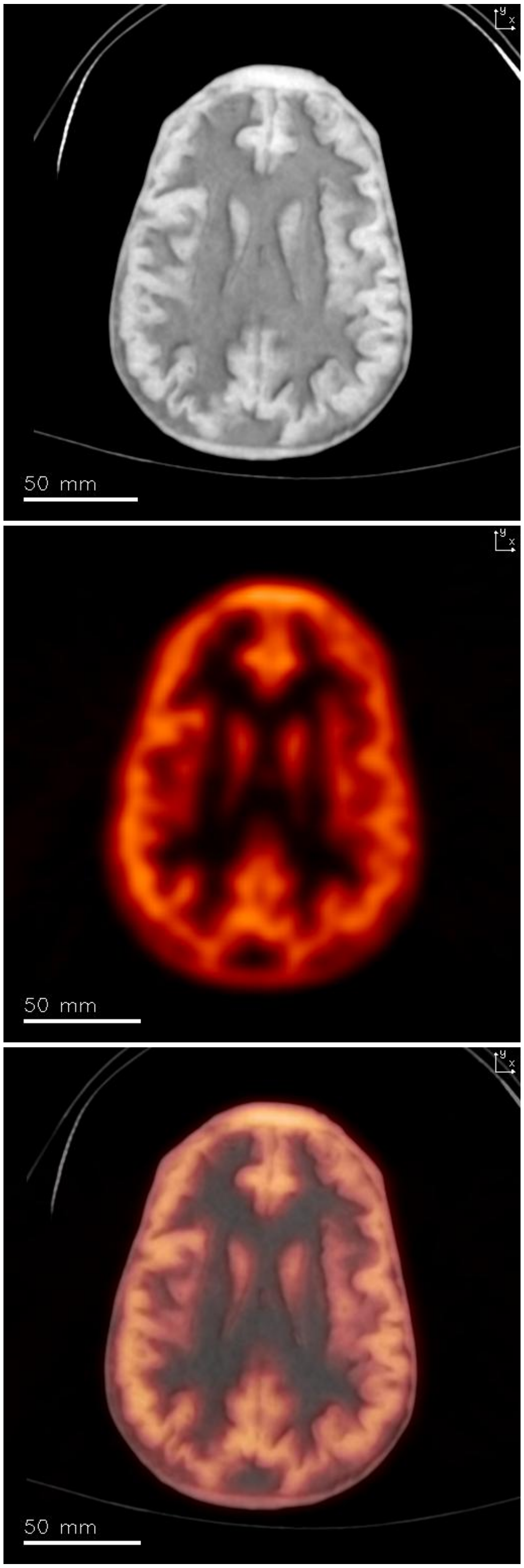


Fig. 4. CT, PET and fused PET/CT images of the human brain phantom.

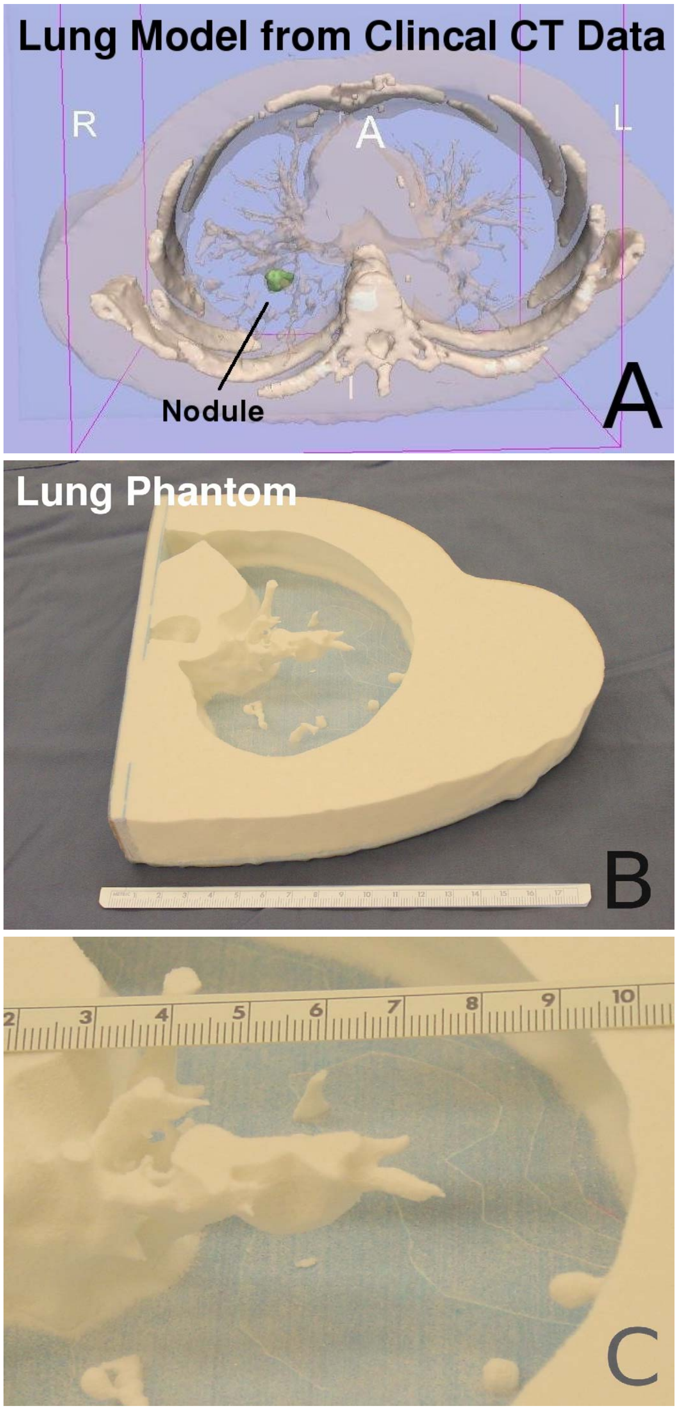


Fig. 5. Human lung phantom. Panel A: surface model of the phantom showing soft tissue, bone and the lung nodule (green). Panels B and C: photographs of the phantom with the lower cap, nodule and lung material removed, showing the anatomical structure. The rule is marked in centimeters.

are still moist, the CT number increases to about -500. We are taking two approaches to bring the phantom density closer to that of real tissues. One approach is to develop powder that functions similarly to the zp15e, but has a density close to that of water. This would allow us to embed our printed activity concentrations within phantoms with tissue equivalent density, and therefore tissue equivalent photon scatter and attenuation characteristics. The other approach we are considering is to develop pigments with massive additives (bismuth nano-particles for example) that will allow us to reach CT numbers similar to that of soft tissues and bone. Similarly to the color/activity concentration mixing, this would allow us to print phantoms with CT numbers to simulate lung (HU ≈ -600, unbound

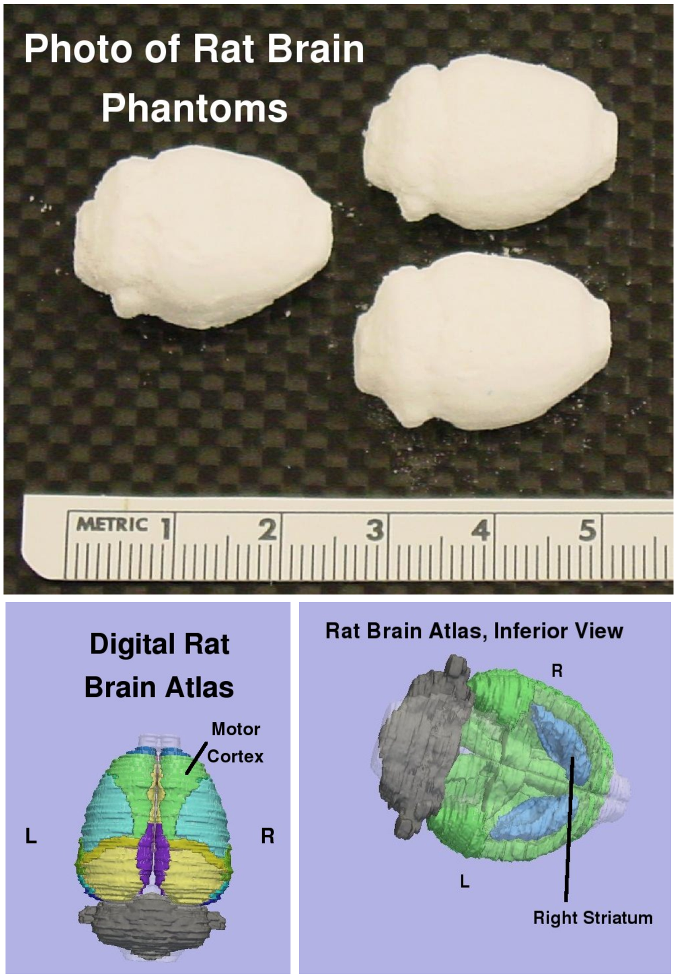


Fig. 6. Rat brain phantoms (upper photograph) and digital models showing the radiolabeled structures.

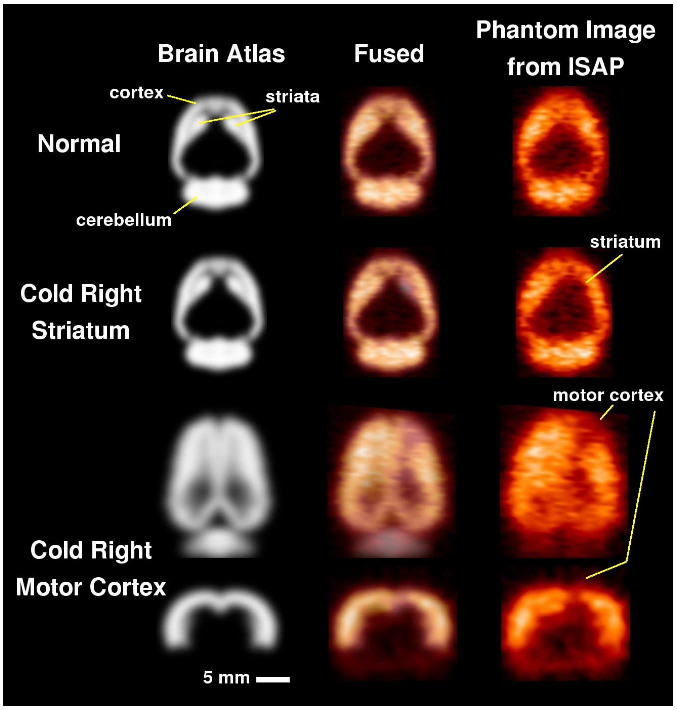


Fig. 7. PET images of the rat brain phantoms. The left column shows the digital model used to create the phantoms, blurred with a 1.25 mm FWHM Gaussian kernel to represent the resolution of the IndyPET-III scanner. The right column contain PET images acquired with the IndyPET-III scanner. The center column shows the left and right columns fused. The PET images in the top row are of a phantom with radiolabeled cortex, striata and cerebellum. The PET images in the second row are of a phantom with radiolabeled cortex, left striatum and cerebellum, but with the right striatum unlabeled. The PET images in the bottom two rows are of a phantom with radiolabeled cortex, striata and cerebellum, but with the right motor cortex unlabeled.

powder with minimal binder), soft tissue (HU $\approx$ 0, powder bound with binder carrying massive pigment) and bone ((HU $\approx$ 1000, powder bound with binder carrying massive pigment). Pigments for inkjet printing are considerably more problematic than dyes, especially compared to dyes at trace concentrations such as FDG. Inks are constrained by viscosity and particulate size limits of inkjet print heads [9]. We are working to develop a modified print head that will allow us to continuously mix relatively massive binders without clogging the jets.

We are also working towards printing with long-lived isotopes. The phantoms we have printed to date all contained ^{18}F. With a relatively short half life of 109.77 minutes, these phantoms can only be imaged for a few hours before the activity decays below a useful level. Ultimately we intend to print phantoms with longer lived isotopes, such as ^{68}Ge with a half life of 270 days, so that the resulting phantoms can be imaged repeatedly and shared among institutions. For example, phantoms created from patient data could be imaged at multiple sites, effectively allowing a group patients to be imaged at all sites in a given clinical trial. If the realistic phantoms model the particular patient anatomy and tracer uptake patterns that are being imaged in the trial, this would allow us to directly measure site to site variances. Note that suitable isotopes are not limited to positron emitters, so printed phantoms can potentially be made for SPECT and other modalities. We foresee no technical problems with printing with long-lived isotopes, although there are radiation safety considerations that must be addressed.

V. CONCLUSIONS

We have developed a new method for manufacturing imaging phantoms using 3D printing technology. Phantoms produced with this method reproduce anatomic detail at the submillimeter level. Phantom uniformity is equal or better to other available techniques which cannot approach this level of detail. The boundaries between regions of different activities have zero thickness, which is a great advantage when making phantoms which test systems at the limits of resolution, especially for small animal imaging. Further enhancement of the method is expected to result in phantoms that contain multiple activity concentrations of long-lived isotopes in tissue-equivalent densities. Such phantoms, when used at one site or shared among many, have the potential to allow us to do quantitative evaluations of imaging systems that have so far been impossible.

REFERENCES

[1] M. E. Shenton *et al.*, “Harvard brain atlas: A teaching and visualization tool,” in *Biomedical Visualization*, 1995.

[2] R. Kikinis *et al.*, "A digital brain atlas for surgical planning, model driven segmentation and teaching," *IEEE Transactions on Visualization and Computer Graphics*, vol. 2, no. 3, pp. 232–241, Sep. 1996.

[3] G. Paxinos and C. Watson, *The Rat Brain in Stereotaxic Coordinates*, 4th ed. Academic Press, 1998.

[4] Z. Corporation, "Z corporation 3D printing technology: Fast, affordable and uniquely versitile," Z Corporation, Tech. Rep., 2007.

[5] *Spectrum Z510 User Manual*, Z Corporation, Burlington, MA, US, Sep. 2005.

[6] *NEMA NU 2-2007: Performance Measurements of Positron Emission Tomographs*, National Electrical Manufactures Association, 2007.

[7] N. Rouze, V. Soon, J. Young, S. Siegel, and G. Hutchins, "Initial evaluation of the indiana small animal pet scanner," *Nuclear Science Symposium Conference Record, 2005, IEEE*, vol. 4, pp. 2394–2398, 23-24 Oct 2005.

[8] V. C. Soon, M. A. Miller, and G. D. Hutchins, "A non-iterative method for emission tomographic image reconstruction with resolution recovery," in *Nuclear Science Symposium Conference Record, 2007, IEEE*, 2007.

[9] P. Calvert, "Inkjet printing for materials and devices," *Chem Mater*, vol. 13, no. 10, pp. 3299–3305, 2001.