

Canaux conducteurs dans des pièces imprimées par fabrication SLA/MSLA

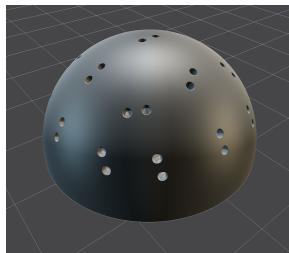
Rapport de stage de L3

Axel Stengel

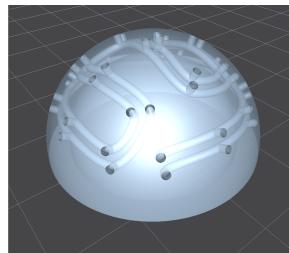
ENS Rennes

Nancy, France

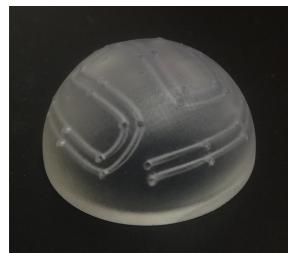
axel.stengel@ens-rennes.fr



(a) Modèle 3D



(b) Génération des canaux



(c) Pièce imprimée en 3D



(d) Pièce finale fonctionnelle

FIGURE 1. Conception d'une demi-sphère lumineuse (droite) à partir d'un modèle 3D (gauche) à l'aide de notre méthode

Résumé

Nous étudierons dans ce stage la faisabilité et les contraintes géométriques liées à la génération de canaux conducteurs au sein de pièces fabriquées par impression 3D résine. Le but est de créer algorithmiquement des réseaux de canaux de petite échelle ($< 1\text{mm}$) qui puissent être remplis de matériaux conducteurs après fabrication, afin de former un circuit électrique 3D pour diverses applications. Il s'agit de mener une étude expérimentale sur échantillons qui informe les contraintes d'un algorithme de type placement-routage spécialisé, implémenté dans le cadre du stage.

Keywords: Impression 3D, Circuit imprimé, Fabrication SLA/MSLA, Métal Liquide

1 Introduction

Dans l'industrie de l'électronique, l'invention des circuits imprimés ou PCB (Printed Circuit Board) a été une révolution. Ceux-ci sont utilisés partout, des ordinateurs aux voitures en passant par les fusées.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

INRIA Nancy, May 15 – July 28, 2023, Nancy, France

© 2023 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM..\$0.00

<https://doi.org/XXXXXX.XXXXXXXX>

Cependant la technologie vieille de plus de 60 ans garde toujours le même format : une plaque rigide. Or cela impose de nombreuses contraintes sur la conception d'objet contenant de l'électronique, notamment pour du prototypage rapide. De plus, la fabrication de PCB traditionnel demande des outils industriels couteux ce qui oblige à faire appel à un service externe spécialisé et qui assure une certaine qualité mais rajoute du temps et un coût lors de prototypage rapide. L'intérêt d'une méthode peu chère et reproductible par le grand public de créer des circuits électriques dans une pièce 3d quelconque se fait donc ressentir.

Au cours des dernières années, l'impression 3D est devenue un outil accessible au grand public que ce soit l'impression filament ou FDM (Fused Deposition Modeling) ou l'impression résine (technologies SLA), moins utilisée mais tout de même accessible en terme de prix et d'utilisation. Cette dernière permet un niveau de détails qui nous interresse particulièrement pour notre application. Cependant, il est encore impossible d'imprimer avec plusieurs résines à la fois, nous avons donc choisi de générer des canaux vides dans la pièce afin de les remplir de métal liquide.

L'objectif de ce papier est ainsi de présenter une méthode de génération de canaux dans une pièce 3D ainsi que le remplissage de ces dit canaux par du métal liquide afin de créer un réseau de canaux conducteurs pouvant connecter des composants électriques entre eux. Pour finir, nous allons présenter un algorithme de génération automatique de canaux dans une pièce.

2 Contexte

2.1 Impression 3D résine (fabrication SLA/MSLA)

L'impression 3D est une technique de fabrication additive, c'est à dire fabrication par ajout de matière, transformant un modèle 3D en un objet physique. Plusieurs technologies d'impression 3D existent actuellement, permettant d'imprimer différents matériaux à différentes précisions cependant elles se basent toutes sur un principe similaire : un logiciel appelé "slicer" ou "logiciel de découpe" qui prend en entrée un modèle 3D, principalement sous forme d'un maillage, et le découpe en tranches. Ces tranches sont transformées en instruction machine compréhensible par l'imprimante 3D et stockées dans un fichier. L'imprimante prend ensuite le fichier obtenu et imprime couche par couche l'objet voulu.

Dans ce stage, nous nous concentrons sur l'impression 3D résine et plus précisément les technologies SLA et MSLA (figure 1). Ces deux technologies reposent sur un même principe : le plateau de l'imprimante est plongé dans un bac de résine photopolymère, c'est à dire qui polymérise sous la présence d'UV, et une source de lumière vient activer et solidifier couche par couche la résine au fur et à mesure que le plateau monte. Pour l'impression SLA (Stereolithograph Apparatus), cette source de lumière est un laser qui est orienté par des miroirs mobiles afin de parcourir l'intégralité de la surface d'impression. L'impression MSLA (Masked Stereolithograph Apparatus) utilise quant à elle une source d'UV qui recouvre l'entièreté de la surface d'impression grâce à des lentilles et cette lumière est ensuite filtrée par un écran de type LCD (Liquid Crystal Display) afin d'imprimer une couche entière d'un seul coup.

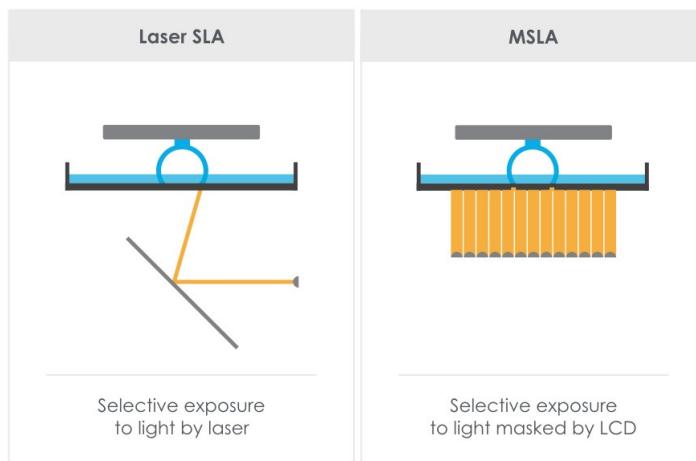


FIGURE 2. Schéma de fonctionnement des technologies SLA et MSLA (Dsazer25, CC BY-SA 4.0 <<https://creativecommons.org/licenses/by-sa/4.0/>>, via Wikimedia Commons)

Ces deux technologies ont été choisies par rapport à la technologie bien plus répandue et facile d'utilisation d'impression FDM (Fused Deposition Modeling) pour leur rapport précision/rapidité et finition des pièces. En effet, les imprimantes FDM font fondre un filament plastique et le déposent couche par couche à l'aide d'une buse afin de former l'objet final. Les contraintes physiques et mécaniques liées à cette technologie signifie que le temps d'impression augmente de manière linéaire avec la hauteur de couche (précision sur l'axe Z) et avec le diamètre de la buse (précision sur les axes X et Y). A l'inverse, les imprimantes résines ont une précision X-Y fixe (taille d'un pixel de l'écran ou diamètre du laser) et le temps d'impression est quasi-constant par rapport à l'épaisseur de couche (précision Z). Cela signifie que la précision d'une imprimante filament classique est de l'ordre de 100 à 200 µm sur l'axe Z et de l'ordre 400 µm de diamètre de buse alors que pour les imprimantes SLA le laser est de l'ordre de 70 µm de diamètre avec une épaisseur de couche de l'ordre de 50 µm et les imprimantes MSLA ont des écrans allant de 50µm jusqu'à 19 µm de largeur de pixel et des épaisseurs de couche de l'ordre de 20-50 µm. De plus, la surface des objets imprimés en FDM est très marquée par les lignes formés par les couches, la rendant très rugueuse alors qu'une pièce imprimée en résine a une surface bien plus lisse, même à hauteur de couche égale.

2.2 Circuits imprimés (PCB)

Une grande majorité des circuits électroniques actuels se présente sous la même forme : les circuits imprimés auxquels sont soudés les composants nécessaires. Les circuits imprimés ou PCB (Printed Circuit Board) sont des plaques composées d'une alternance de couches d'un matériau isolant (généralement de la résine époxyde renforcée à la fibre de verre) et de fines couches de cuivre. Ces couches de cuivre sont gravées par un procédé chimique afin de créer des pistes conductrices pouvant relier les composants entre eux. La connection entre les différentes couches du circuit imprimé se fait à l'aide de vias et la connection entre les composants et le circuit s'effectue en soudant les composants soit à des pastilles de cuivre sur la surface du circuit (SMD ou Surface Mounted Device), soit, pour les composants à patte, à travers un trou d'insertion qui traverse le circuit [13].

Au fur et à mesure que la technologie a évolué et que les composants se sont miniaturisés, dessiner les pistes conductrice à la main n'était plus une option. Des outils professionnels ont alors été mis en place afin de créer les pistes nécessaires pour un PCB à l'aide de logiciels de CAD[1][4]. Le routage automatique des pistes d'un PCB est cependant un problème NP-complet [14] ce qui mène à de nombreux algorithmes d'approximation pour le routage automatique de PCB [17].

La fabrication de PCB est une technologie mature qui est utilisée à travers de nombreuses industries. Celle-ci présente cependant une restriction majeure : le substrat isolant du circuit imprimé est traditionnellement plat et rigide [13]. Pour cette raison, de nombreux projets de recherche s'intéressent à la fabrication de circuits imprimés sur des surfaces courbes [16]. Si certains s'intéressent au pliage de PCB pour former des formes 3D [6], d'autres se concentrent sur la déposition de pistes conductrices sur une surface courbe [8]. Cependant, une grande partie de ces méthodes demande de l'équipement spécialisé à un coût élevé et/ou limite l'adaptabilité du dispositif.

2.3 Métal liquide

L'idée de créer des circuits imprimés avec du métal liquide à la place des pistes de cuivre n'est pas nouvelle. En effet, dans le domaine de l'électronique flexible, du métal liquide est souvent utilisé sur des surfaces élastiques, ce qui donne lieu à des circuits électroniques étirables et pliables sans grand changement aux propriétés électriques du circuit [7] [2].

Le métal liquide utilisé ici est en réalité un alliage basé sur du Gallium dans la majorité des cas [3]. En effet les alliages à base de Gallium ont de nombreux avantages par rapport aux alternatives à base de Mercure, principalement les alliages à base de Gallium ne sont pas toxiques et sont même bio-compatibles contrairement au Mercure qui lui est extrêmement toxique et pose de grands risques au moindre contact avec la peau. De plus la température de fusion des alliages de Gallium peut être contrôlée par la proportion et la constitution des autres composants comme l'Etain ou l'Indium [15] [5].

2.4 IceSL

Afin de générer les canaux dans le modèle 3D, il est nécessaire d'utiliser un logiciel de modélisation. Celui utilisé dans ce papier est IceSL[12], un logiciel développé et maintenu par l'équipe MFX. IceSL permet à la fois de modéliser nos modèles 3D mais aussi de les trancher pour pouvoir les imprimer. Dans IceSL, la modélisation s'effectue à l'aide de scripts écrits en Lua accompagné un module spécial contenant les différentes classes et fonctions de modélisation.

3 Génération des canaux

Dans ce stage, nous imprimons des pièces comportant des canaux. Nous présentons dans cette partie la génération des canaux. Dans un premier temps nous expliquons les contraintes liées à la forme des canaux et la forme choisie, puis nous présentons la génération de la géométrie à l'aide

d'IceSL. Afin de faciliter le procédé, et permettre l'utilisation de ce dernier par des personnes externes, nous avons inclu toutes les fonctions nécessaires dans un module Lua `channel_gen` qui peut être importé dans IceSL.

3.1 Courbes et splines

Les canaux ayant pour but d'être remplis de métal liquide, certaines contraintes physiques s'appliquent à la forme de ces derniers. En effet, les canaux doivent être d'abord vidés de toute la résine résiduelle puis être remplis de métal liquide.

Dû à la viscosité de la résine et le comportement unique du métal liquide à l'échelle considérée [9], nous avons déterminé que la présence d'angles, notamment d'angles aigus, pourrait résulter en la présence d'une section de résine dans le canal, ce qui empêcherait un bon contact électrique.

Nous avons ainsi choisi d'utiliser des splines cubiques pour la forme des canaux. Nous avons implémenté 2 splines cubiques différentes :

- La spline de Catmull-Rom est une spline qui interpole tous ses points de contrôle de façon continue et même C^1 ce qui assure un canal remplissable. Cette spline est utilisée de base dans notre module si aucune spline est spécifiée car elle est facile d'utilisation et intuitive à comprendre.
- La spline de Bezier est une spline qui permet plus de contrôle sur la forme de la courbe mais nécessite un paramètre de tangente en plus pour chaque point d'interpolation ce qui complexifie un peu son utilisation. Nous l'avons tout de même implémenté car ce contrôle supplémentaire peut être nécessaire dans certains cas.

3.2 Génération de la géométrie

Pour générer les canaux, il suffit donc juste de fournir une liste de points de passage, la largeur voulue et les subdivisions radiales et longitudinales du canal pour obtenir un tube ayant la forme de la spline de Catmull-Rom passant par les points de contrôle définis. Nous pouvons ensuite faire la différence entre ce volume et le maillage 3D de l'objet voulu pour obtenir un canal à l'intérieur du modèle.

La géométrie des canaux est générée dans le logiciel IceSL et utilise des fonctions de base d'IceSL. Un des avantages de IceSL est la gestion des opérations booléennes entre les différents volumes puisque aucune opération sur le maillage 3D n'est faite sur le coup mais seulement à l'export de l'objet. Cela nous permet donc de faire des différences booléennes entre le maillage 3D d'origine et les canaux sans avoir à faire un compromis entre la qualité des canaux et les performances du logiciel.

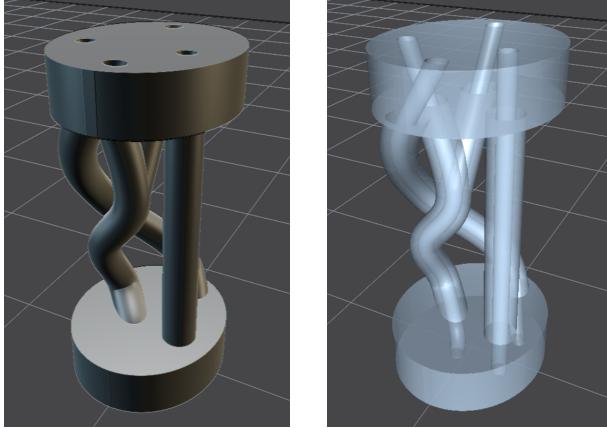


FIGURE 3. Modèle de test du module en vue solide (gauche) et vue transparente (droite)

4 Algorithmme de routage

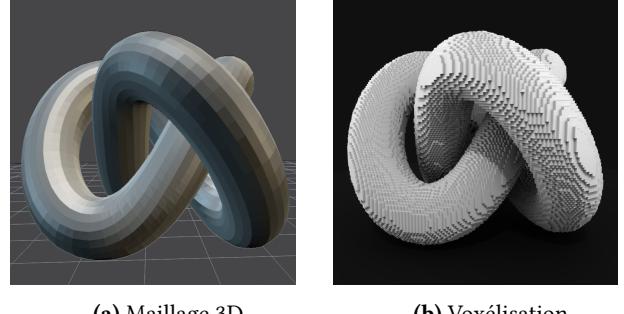
S'il est parfois nécessaire de définir à la main les points de contrôle des courbes, dans de nombreux cas, l'ajout d'une troisième dimension facilite grandement l'autoroutage des canaux par rapport à des PCB classiques. Nous présentons ici un algorithme dérivé de A*.

4.1 Attendus de l'algorithme

Le but d'algorithme est de connecter entre-eux les différents points d'une même piste tout en vérifiant que deux pistes différentes ne s'intersectent pas. Ainsi, étant donné plusieurs ensembles de point sur la surface du modèle, l'algorithme doit pouvoir calculer des parcours connectant chaque point à l'intérieur d'un ensemble sans que ceux-ci intersectent les parcours générés entre les points d'autres ensembles. Dû à la méthode de génération des canaux, l'algorithme doit retourner pour chaque parcours la liste de points de contrôle de la spline Catmull-Rom correspondante.

4.2 Présentation de l'algorithme

Pour appliquer l'algorithme A*, il est nécessaire de se placer dans un espace discret avec des cases accessibles et non accessibles. Pour cela nous faisons appel à un voxelisateur, un programme qui prend en entrée un maillage 3D et retourne une voxélisation du maillage, c'est à dire une grille en 3 dimensions dont la valeur des cases indique l'emplacement de la case par rapport au modèle : si la case est à l'intérieur, à la surface ou à l'extérieur du modèle. Cela permet d'obtenir une discréttisation qui approxime le modèle 3D.



(a) Maillage 3D

(b) Voxélisation

FIGURE 4. Approximation (gauche) par voxel d'un maillage 3D (droite)

Notre algorithme se déroule de la façon suivante.

Algorithm 1 Algorithme de routage

```

function ROUTING(mesh, plist, width)
    voxs  $\leftarrow$  VOXEL(mesh)
    for i  $\leftarrow$  1, #plist do
        trace  $\leftarrow$  {}
        for j  $\leftarrow$  2, #plist[i] do
            start_point  $\leftarrow$  plist[i][j - 1]
            end_point  $\leftarrow$  plist[i][j]
            path  $\leftarrow$  A*(voxs, start_point, end_point)
            path  $\leftarrow$  SIMPLIFYPATH(path, voxs)
            trace  $\leftarrow$  trace + path
            PRINT(path)
        end for
        FILLVOX(voxs, trace, width)
    end for
end function
```

Les fonctions auxiliaires utilisées sont :

- VOXEL(*mesh*) correspond au voxelisateur, la fonction prend donc le maillage *mesh* en entrée et retourne une approximation en voxel de ce dernier.
- A*(*voxs, start_point, end_point*) est l'exécution de l'algorithme A*, la fonction retourne la liste de cases correspondant au chemin trouvé par l'algorithme A*. L'heuristique utilisé ici est la distance de Manhattan
- SIMPLIFYPATH(*path, voxs*) est une fonction qui permet de simplifier le chemin trouvé, c'est à dire qui ne garde que les points nécessaires dans *path* et maximise les lignes droites dans le chemin.
- FILLVOX(*voxs, trace, width*) permet de marquer les cases dans la piste *trace* de largeur *width* comme inaccessible une fois que tous les segments de cette même piste ont été calculés.

L'algorithme de routage présenté ici a été codé en C++ et se base sur le code du voxelisateur VoxSurf développé par Sylvain Lefebvre [11]. Tout comme VoxSurf, l'algorithme

utilise la bibliothèque LibSL [10], aussi développé par Sylvain Lefebvre, qui permet ici d'avoir accès à de nombreuses fonctions et objets utiles pour gérer de la géométrie en 3 dimensions.

Tout comme le voxélisateur utilisé dans l'algorithme, notre programme produit aussi un fichier .vox qui contient la voxélisation du modèle avec des matériaux différents pour différencier différentes zones du modèle telles que la surface du modèle, la coquille interieure, les pistes et le périmètre de sécurité autour des pistes. Cela permet de facilement visualiser les pistes avant de les ajouter au modèle via IceSL.

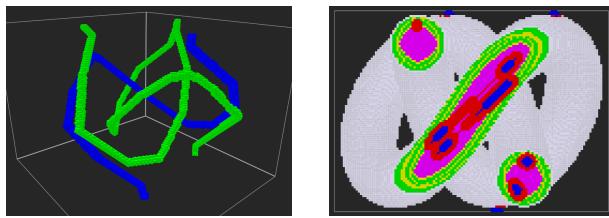


FIGURE 5. Pistes calculées par l'algorithme (droite) et coupe transversale du voxel après routage (gauche) avec : la surface du modèle en blanc, la coquille interieure en jaune et vert, les pistes en bleu et le perimètre de protection autour des pistes en rouge.

5 Impression et remplissage des canaux

Enfin, une fois la géométrie générée, il reste à réaliser l'objet physique. Nous présentons dans cette partie les deux étapes du procédé : l'impression des modèles et le remplissage des canaux.

5.1 Impression des modèles

Nous avons fait le choix ici d'imprimer en utilisant des résines transparentes afin de pouvoir observer les canaux pour des défauts d'impression ou autres obstructions.

Les canaux sont de petite taille (< 1 mm de diamètre) et sont donc très sensibles à la calibration de l'imprimante. Notamment il est primordial de ne pas surexposer la résine car le *bleeding* (terme utilisé pour signifier la polymérisation de résine au-delà de la zone d'exposition dû à une surexposition) peut boucher facilement les canaux (figure 6).

Une fois le modèle imprimé, il est nécessaire de nettoyer la pièce et de vider les canaux de la résine résiduelle. Pour cela nous avons utilisé un compresseur à air avec une aiguille. Il est important de faire cela avant de traiter la pièce aux UV sinon la résine à l'intérieur des canaux risque de polymériser et de boucher les canaux.



FIGURE 6. Défaut d'impression dû à une surexposition de la résine

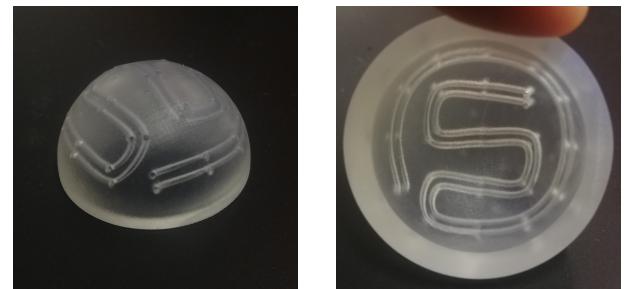


FIGURE 7. Pièce imprimée, nettoyée, évidée et traitée vu d'en dessous (gauche) et de profil (droite)

5.2 Remplissage des canaux et mise en place des composants

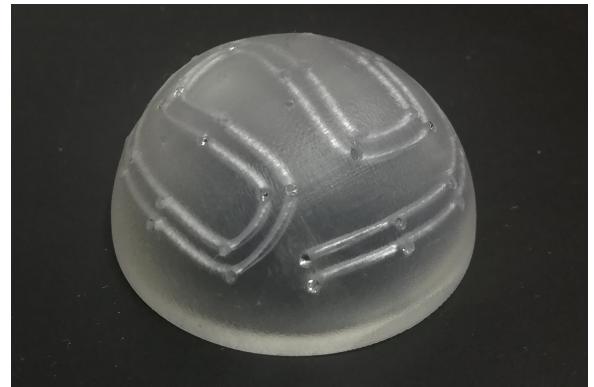


FIGURE 8. Pièce après remplissage des canaux avec du Galinstan

Une fois la pièce imprimée, il faut remplir les canaux de métal liquide. L'alliage utilisé ici est du Galinstan (68.5% Gallium, 21.5% Indium et 10% Etain). L'injection du métal liquide est effectuée à l'aide d'une seringue avec une aiguille de 0.8mm. Pour le remplissage d'une trace, tous les points

de sortie sont couverts de scotch électrique à part 2, un pour l'injection et l'autre pour la sortie d'air.

Pour connecter les différents composants à la pièce, deux méthodes sont disponibles. Pour les composants à pâtes, il suffit d'insérer les pâtes dans les trous et de fixer le composant à l'aide d'une colle non conductrice. Pour les autres composants (SMDs ou Surface Mounted Devices), il est possible de créer un bouchon d'époxy conductrice au niveau de la connection et de souder le composant de manière traditionnelle dessus. Notons cependant que nous n'avons pas eu le temps de tester cette dernière méthode.

6 Résultats

Avec tous les outils maintenant à notre disposition, nous sommes en mesure de fabriquer des objets et prouver le concept.

Dans un premier temps, nous avons imprimé le modèle test pour vérifier le concept (figure 9) et nous remarquons que les canaux conducteurs se comportent comme on le souhaite et la LED bleue s'allume.

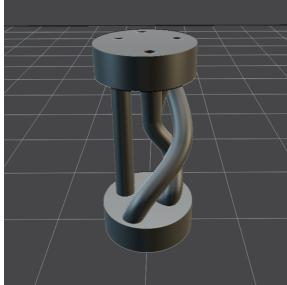


FIGURE 9. Modèle test (gauche) fini avec une LED bleu et une LED rouge (droite)

Une fois le concept validé, nous avons pu passer à des objets plus complexes. Notamment des objets qu'il serait impossible, ou en tout cas compliqué, à reproduire avec un circuit imprimé classique. Nous avons donc décidé de modéliser une demi-sphère avec 21 encoches pour LED uniformément distribués autour de la demi-sphère (figure 10).

Cependant, le routage des deux derniers exemples a été fait à la main, soit par placement des points manuellement (figure 9) soit par calcul direct de la position des points (figure 10). Il nous faut donc un modèle pour tester notre algorithme de routage automatique. Malheureusement, par manque de temps, ce modèle n'a pas pu être imprimé correctement.

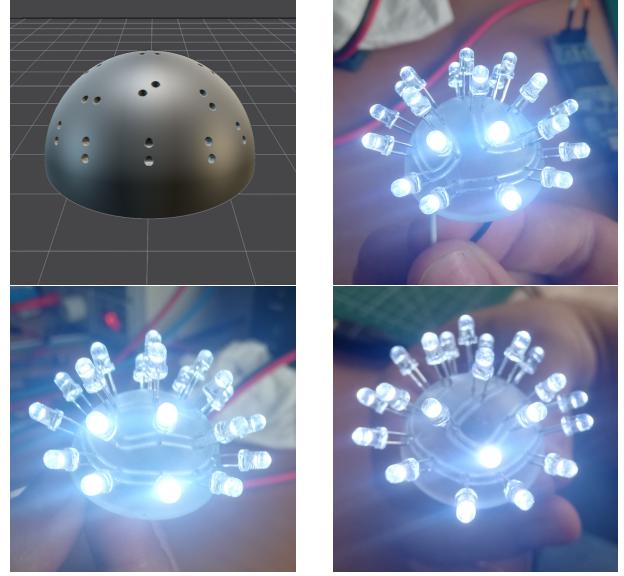


FIGURE 10. Demi-sphère lumineuse (haut gauche) finie avec des LED blanches

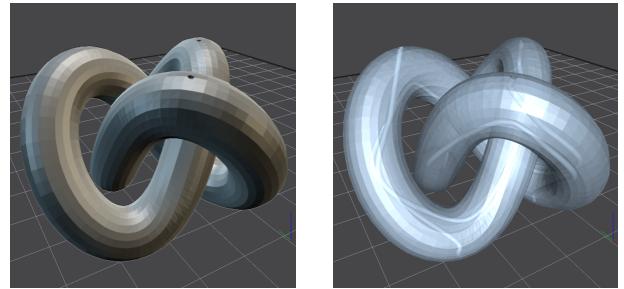


FIGURE 11. Maillage 3D du modèle en vue solide (gauche) et en vue transparente (droite)

7 Discussions et travaux futurs

Le principal problème ici est un manque de temps pour réaliser plus d'exemples et de modèles, notamment des modèles avec des réseaux de canaux plus complexes, avec des composants différents. De même, nous n'avons pas eu l'occasion d'imprimer des modèles avec des canaux auto-générés par notre algorithme de routage.

L'algorithme de routage est fonctionnel mais en est qu'à sa première itération. Nous avons pas eu le temps d'optimiser le post-traitement des parcours ou de tester différentes heuristiques pour l'algorithme A* qui pourrait nous donner de meilleurs canaux, notamment lorsque l'on considère plusieurs pistes séparées puisque l'algorithme actuel ne considère pas la largeur de la section du modèle qu'il traverse et donc peut bloquer le passage pour d'autres pistes. De même, la gestion de piste à plus que 2 sorties est très

rudimentaire et nécessite plus de travail.

Enfin, d'un point de vue électronique, une étude plus poussée des caractéristiques électriques des circuits est nécessaire si cette méthode veux être utilisée pour de l'électronique de précision.

8 Conclusion

Nous présentons dans ce papier une méthode de génération et fabrication de circuits imprimés en 3 dimensions. Cette méthode développée durant ce stage dans l'équipe MFX de l'INRIA Nancy permet de prendre un objet 3D quelconque et de générer des canaux à l'intérieur. Cette pièce peut ensuite être imprimée à l'aide d'une imprimante résine et les canaux peuvent être remplis de métal liquide afin de servir de pistes conductrices.

Une fois fini, ces circuits imprimés en 3D peuvent être utilisé comme un outil supplémentaire de conception électronique et permettent des formes et des objets qui étaient difficiles voir même impossible à réaliser avec des circuits imprimés classiques. Cela peut notamment aider grandement pour du prototypage rapide et peu coûteux puisque cette méthode utilise très peu de matériel cher ou nécessitant une grande expertise.

Le travail effectué dans ce stage a permis de vérifier le concept d'utiliser du métal liquide dans des pièces imprimées en 3D pour réaliser des circuits imprimés.

Cependant, seulement des circuits simples ont pu être imprimés par manque de temps, la prochaine étape logique sera donc d'essayer de réaliser des circuits plus complexes.

Références

- [1] Autodesk. 1988. EAGLE. <https://www.autodesk.com/products/eagle/overview>
- [2] Michael D. Bartlett. 2021. Liquid assets for soft electronics. *Nature Materials* 20 (2021), 714–715. Issue 6. <https://doi.org/10.1038/s41563-021-00939-y>
- [3] Guyue Bo, Long Ren, Xun Xu, Yi Du, and Shixue Dou. 2018. Recent progress on liquid metals and their applications. *Advances in Physics : X* 3, 1 (2018), 1446359. <https://doi.org/10.1080/23746149.2018.1446359> arXiv:<https://doi.org/10.1080/23746149.2018.1446359>
- [4] Jean-Pierre Charras. 1992. KiCad. <https://www.kicad.org/>
- [5] D. S. Evans and A. Prince. 1978. Thermal analysis of Ga-In-Sn system. *Metal Science* 12, 9 (1978), 411–414. <https://doi.org/10.1179/030634578790434025> arXiv:<https://doi.org/10.1179/030634578790434025>
- [6] Marco Freire, Manas Bhargava, Camille Schreck, Pierre-Alexandre Hugron, Bernd Bickel, and Sylvain Lefebvre. 2023. PCBend : Light Up Your 3D Shapes With Foldable Circuit Boards. *ACM Transactions on Graphics* (2023). <https://doi.org/10.1145/3592411>
- [7] Seung Hee Jeong, Anton Hagman, Klas Hjort, Magnus Jobs, Johan Sundqvist, and Zhigang Wu. 2012. Liquid alloy printing of microfluidic

stretchable electronics. *Lab Chip* 12 (2012), 4657–4664. Issue 22. <https://doi.org/10.1039/C2LC40628D>

- [8] Hyun Chul Jin, John R. Abelson, Martin K. Erhardt, and Ralph G. Nuzzo. 2004. Soft lithographic fabrication of an image sensor array on a curved substrate. *Journal of Vacuum Science and Technology B : Microelectronics and Nanometer Structures* 22, 5 (Sept. 2004), 2548–2551. <https://doi.org/10.1116/1.1795249>
- [9] Sangyun Jung, Sejin Choi, Jongwon Lee, and Wonjung Kim. 2019. Capillary Flow of Gallium Based Liquid Metal with Surface Oxide. In *APS Division of Fluid Dynamics Meeting Abstracts (APS Meeting Abstracts)*. Article P36.006, P36.006 pages.
- [10] Sylvain Lefebvre. 2006. LibSL. <https://github.com/sylefeb/LibSL/>
- [11] Sylvain Lefebvre. 2018. VoxSurf. <https://github.com/sylefeb/VoxSurf/>
- [12] Sylvain Lefebvre, Salim Perchy, Cédric Zanny, and Pierre Bedell. 2013. IceSL. <https://icesl.loria.fr/>
- [13] Jens Lienig and Juergen Scheible. 2020. *Fundamentals of Layout Design for Electronic Circuits*. Springer International Publishing. <https://doi.org/10.1007/978-3-030-39284-0>
- [14] Qiang Ma and Martin D. F. Wong. 2012. NP-Completeness and an Approximation Algorithm for Rectangle Escape Problem With Application to PCB Routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 31, 9 (2012), 1356–1365. <https://doi.org/10.1109/TCAD.2012.2193581>
- [15] B Predel. 2013. Ga-In (Gallium-Indium) phase equilibria, crystallographic and thermodynamic data of binary alloys. *Dy-Er... Ir-Y : Supplement to Subvolumes IV/5B, IV/5C and IV/5D* (2013), 86–86.
- [16] Steven I. Rich, Zhi Jiang, Kenjiro Fukuda, and Takao Someya. 2021. Well-rounded devices : the fabrication of electronics on curved surfaces – a review. *Mater. Horiz.* 8 (2021), 1926–1958. Issue 7. <https://doi.org/10.1039/D1MH00143D>
- [17] Lee Ritchey. 2000. PCB routers and routing methods. *Electronic Engineering (London)* 72 (05 2000), S93–S95.