Crypto-Compression 3D

Compte Rendu 6

Potin Clément, Fournier Romain
Master 2 IMAGINE
Université de montpellier
2021

Problématique

L'objectif est de développer un algorithme qui, de manière conjointe, compresse avec ou sans pertes et chiffre un objet 3D. Les performances de cette méthode seront ensuite mesurées en termes de taux de compression et de qualité de reconstruction de l'objet 3D.

Edgebreaker : améliorations et décompression

Comme nous l'avons décrit dans le compte rendu de la semaine précédente, nous travaillons actuellement sur l'implémentation de l'algorithme *Edgebreaker*, utilisé pour compresser la connectivité du maillage d'objets 3D.

Edgebreaker permet en effet, grâce à un parcours efficace des demi-arêtes du maillage, de représenter sa connectivité sous la forme d'une chaîne de caractères (appelée "CLERS", car composée des lettres C, L, E, R et S en fonction de la configuration dans laquelle se trouve le triangle actuel). Cette représentation induit donc que 2 à 3 bits par triangle sont nécessaires si on applique un codage entropique, comme le codage de *Huffman*, sur cette chaîne.

Cette semaine, notre travail s'est tout particulièrement porté sur l'amélioration de l'algorithme de compression *Edgebreaker*, déjà fonctionnel la semaine dernière. Celui-ci est désormais plus rapide, et implémente également une compression des positions des sommets du maillage, puisqu'ils sont désormais représentés par des vecteurs de correction.

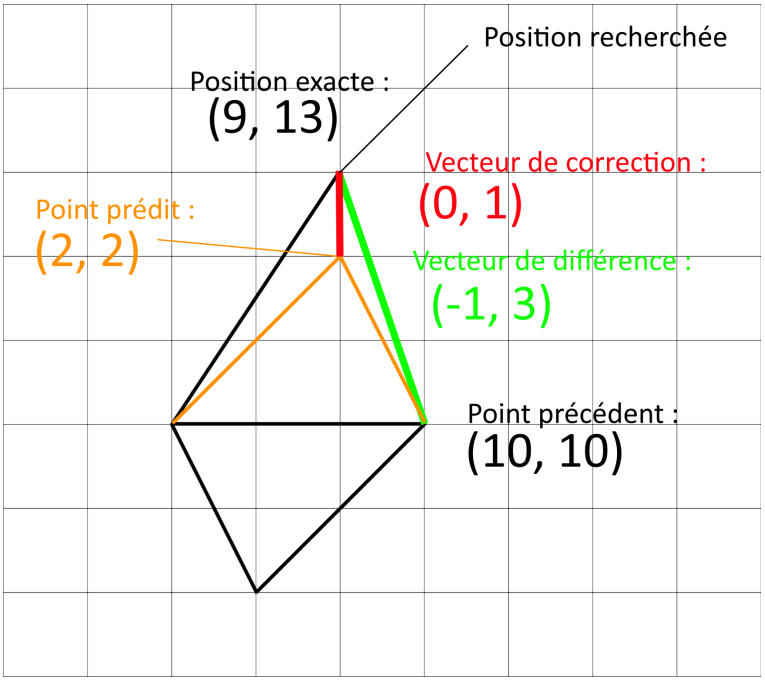
Vecteurs de correction

D'abord représentées par leurs positions exactes (ce qui induisait 3 nombres flottant (pour les coordonnées x, y et z) dont la plage de valeurs possible s'étendait sur l'entièreté de la boîte englobante de l'objet), puis par des vecteurs de différence, allant du sommet précédent au sommet actuel, pour n'obtenir que de petits vecteurs, les positions de nos sommets sont maintenant calculées à l'aide de vecteurs de prédiction et de correction.

Le principe est le suivant : à partir de la base du triangle courant et du sommet opposé au sommet recherché par rapport à cette base, on calcule un triangle symétrique par

rapport à cette base. Le vecteur de correction enregistré sera le vecteur allant du sommet prédit (le symétrique du triangle opposé par rapport à la base) au sommet réel.

Sous la forme d'un schéma, on aurait :



Position exacte, vecteur de différence et vecteur de correction

Ici, on cherche le sommet supérieur, dont la position exacte est (9, 13).

Lors de l'étape de compression, on pourrait :

- (En noir) Enregistrer la position exacte : (9, 13)
- (En vert) Calculer le vecteur de différence allant du sommet précédent (10, 10) au sommet actuel : (-1, 3)
- (En orange) Calculer le symétrique du triangle opposé pour prédire la position du sommet actuel : point prédit = (9, 12), et (en rouge) n'enregistrer que le vecteur de correction allant de ce sommet au sommet actuel : (0, 1)

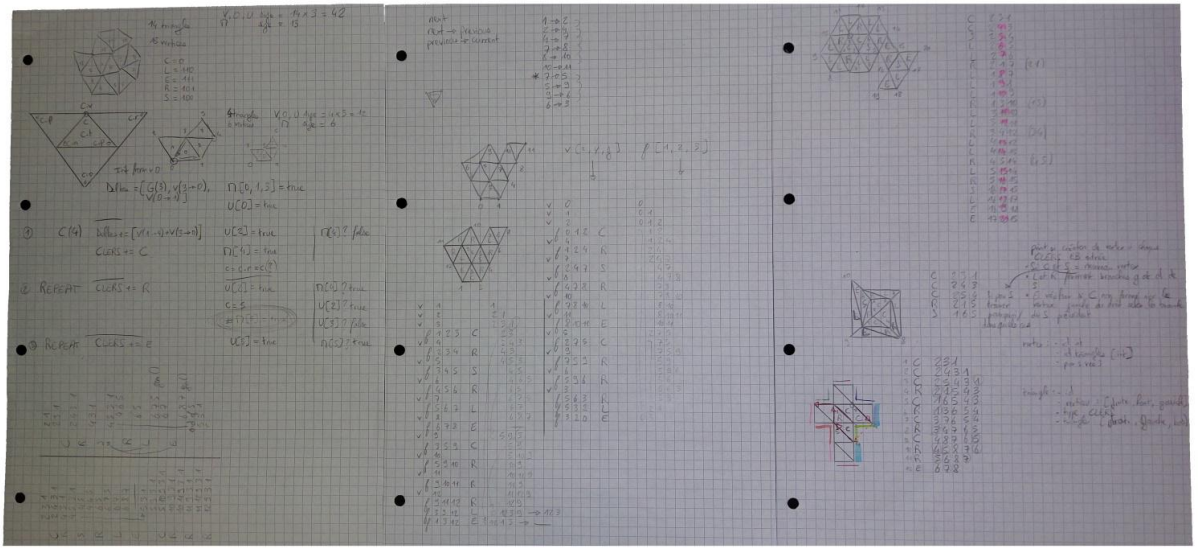
De cette façon, lors de la décompression, l'algorithme prédira la position des sommets en calculant un sommet symétrique au sommet opposé par rapport à la base du triangle (en orange), et y appliquera le vecteur de correction enregistré (en rouge).

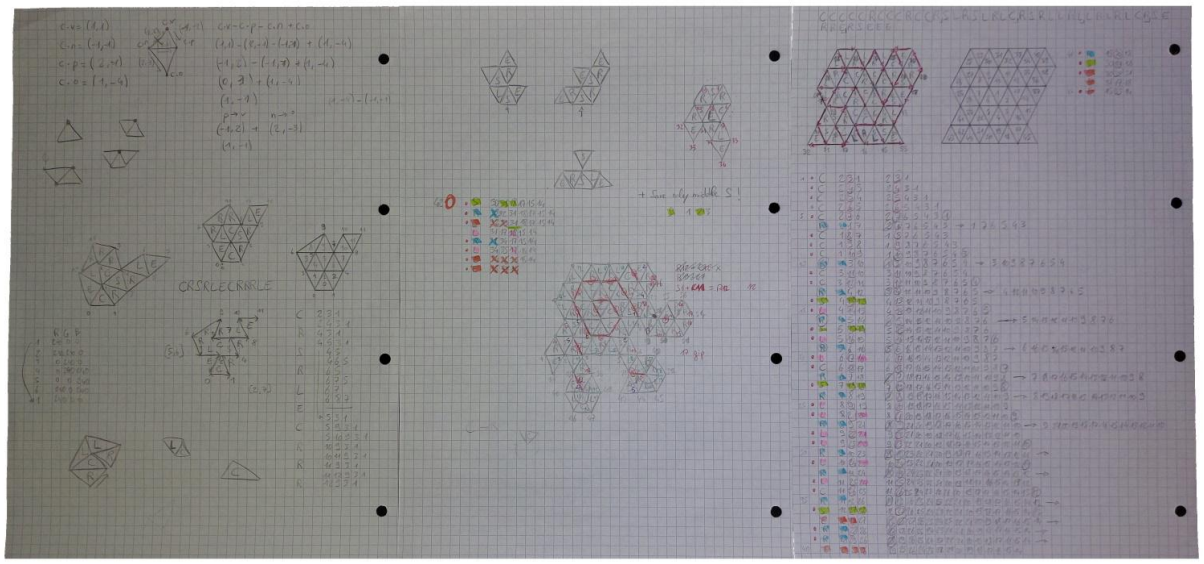
Ces vecteurs de correction, alors enregistrés dans la version compressée de notre maillage, sont :

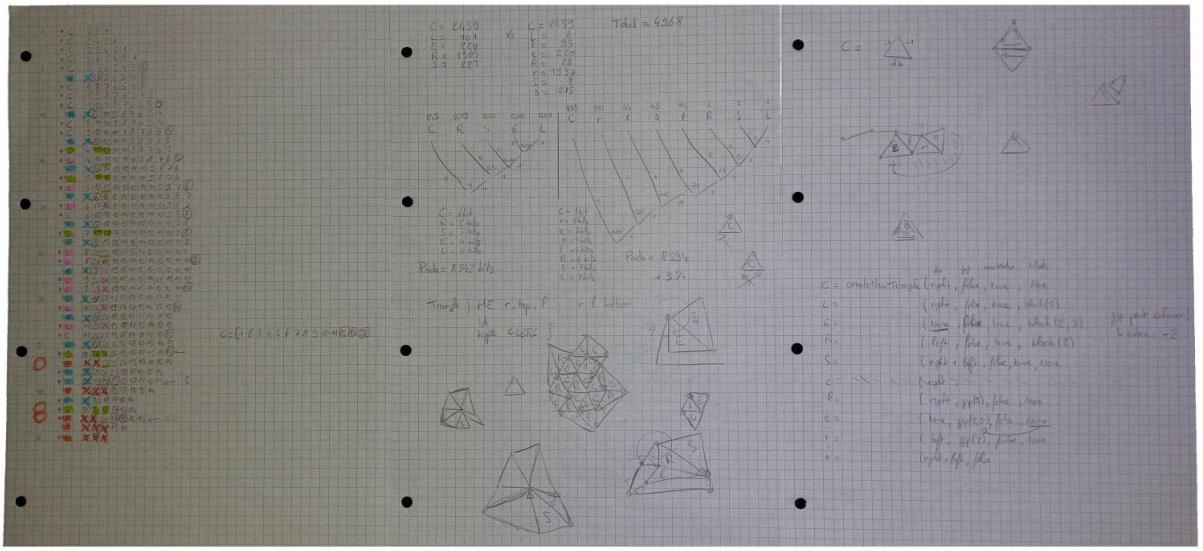
- Extrêmements petits par rapport à la position exacte des sommets (moins de bits sont nécessaires pour les représenter)
- Facilement compressibles davantage avec un codage de *Huffman* adapté à des vecteurs, puisque suite à la quantification, on s'assure d'avoir des sommets dont les positions seront sur la grille régulière définie pour cette quantification, et il y aura donc un bon nombre de vecteurs récurrents (exemple : si grille quantifiée avec un pas de 1 entre chaque point de la grille, une donnée de notre vecteur de correction peut valoir 0, 1, 2, etc, mais pas de nombres entre 0 et 1, 1 et 2, etc. Les tailles des triangles étant généralement très similaire dans un même maillage, beaucoup de vecteurs de correction seront alors identiques)

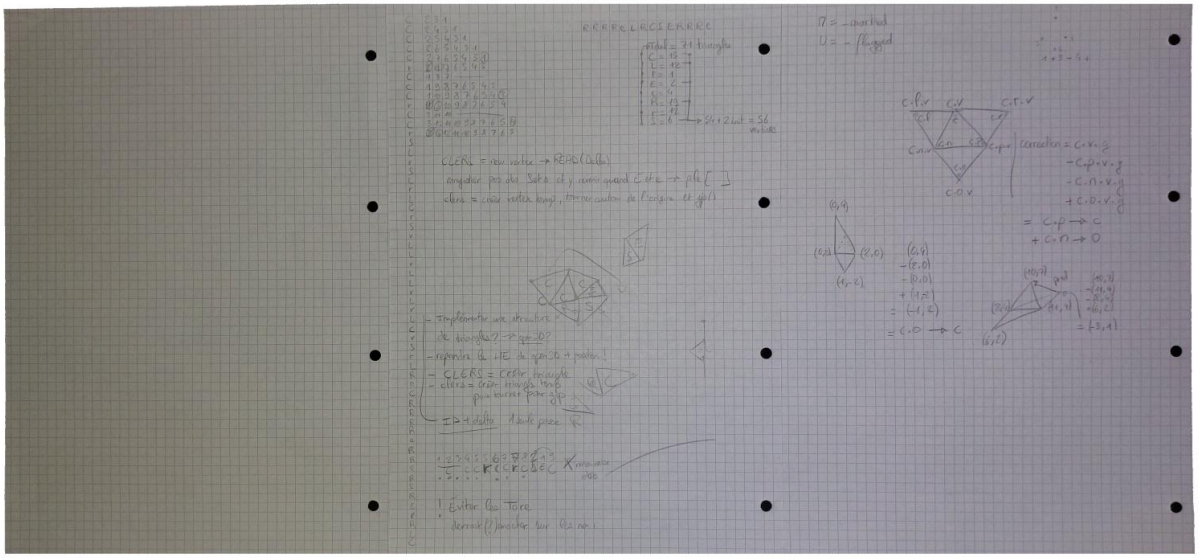
Décompression avec Edgebreaker

En parallèle de ce travail d'amélioration de notre algorithme et de sa préparation à la décompression, nous avons travaillé sur de nombreuses tentatives d'implémentation d'un algorithme de décompression. Ces tentatives, pour l'instant infructueuses, se basaient sur des articles de recherche portant sur *Edgebreaker* (notamment [6, 8]) ainsi que sur des intuitions personnelles. Ici, quelques unes de nos expérimentations sur papier :



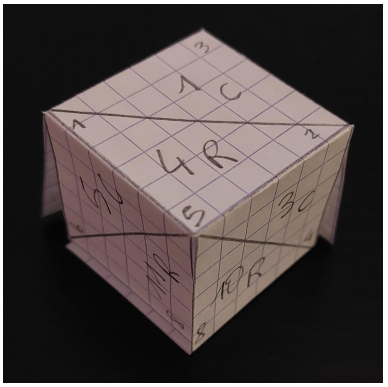






Expérimentations sur la décompression d'Edgebreaker

Nous avons également réalisé des tests sur des modèles papiers en 3D :



Compression/décompression d'Edgebreaker sur un cube

Finalement, nous sommes en train de retravailler les algorithmes présentés dans [8] afin de les adapter pour leur permettre de fonctionner avec les données et structures de données avec lesquelles nous travaillons pour les autres modules de ce projet (lecture/écriture de fichiers .obj/.rfcp, visualisation sur ordinateur, quantification, codage de *Huffman*, chiffrement, ...), et nous espérons finaliser l'implémentation d'*Edgebreaker* et lier ce module au reste d'ici quelques jours.

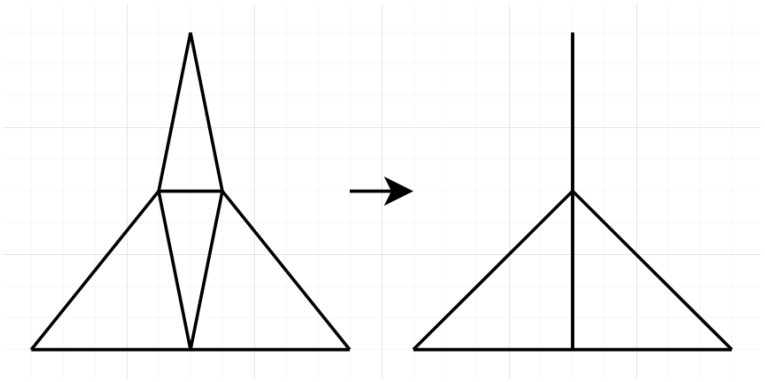
Quantification et maillages non-manifolds

Un obstacle auquel nous nous sommes heurtés cette semaine est l'apparition de maillages non-manifolds lors de l'étape de quantification. Un maillage non-manifold est un maillage ne respectant pas la règle : "une arête ne peut pas lier moins d'une et pas plus de 2 faces". Autrement dit, un maillage non-manifold comporte au moins une arête "seule" (ne liant aucune face) ou liant 3 faces ou plus.

Ce problème était, jusque-là, passé inaperçu car nous n'avions pas encore lié le module de Quantification au module *Edgebreaker*, et donc pas encore converti notre maillage en une structure digestible par cet algorithme.

Lors de la quantification, alors que nous déplaçons simplement les points sans toucher au maillage, un cas spécial est apparu : les faces à aire nulle. Ces "duplications" des sommets du maillage (car 2 sommets proches sont placés au même endroit après quantification) créent par endroit des configurations rendant le maillage non-manifold, ou pouvant scinder le maillage en 2.

Le schéma ci-dessous illustre un cas de maillage non-manifold après quantification :

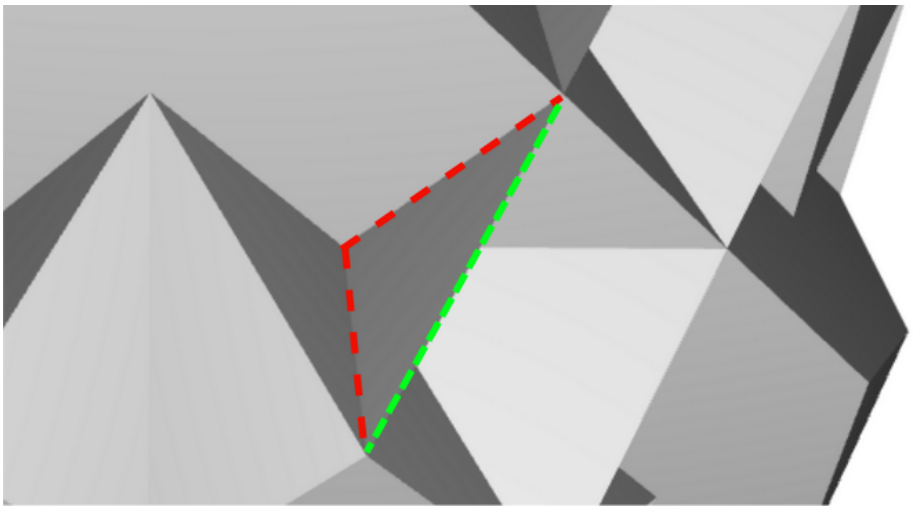


Cas de "duplication"

Certains triangles peuvent se retrouver "écrasés", et l'apparition d'une mauvaise topologie est possible. *Edgebreaker*, ainsi que la librairie que nous utilisons pour stocker le maillage, ne supportent pas ce genre de cas.

Pour pallier ce problème, nous avons implémenté un algorithme de simplification des sommets et triangles, permettant de retirer du maillage les points doublons et donc les faces posant problème. Cet algorithme a en effet réglé le problème des cas de duplication, mais certains de nos maillages restaient non-manifolds après quantification.

Une inspection plus approfondie nous a permis d'identifier une seconde configuration problématique : les pincements. Ci-dessous, une capture d'écran sur laquelle on peut observer des arêtes à trois faces voisines (en rouge), et une arête n'ayant qu'un seul voisin (en vert) :



Cas de "pincement"

Dans ce cas, une simplification n'a aucun effet car le triangle existe bien. Nous pensions, pour traiter ces configurations, parcourir nos triangles et supprimer ceux ayant au moins une arête donnant sur plus de deux faces. Cela permettrait d'éliminer les cas de pincement sur nos maillages quantifiés, et nous espérons que les maillages obtenus seront alors adaptés à *Edgebreaker*.

Lien du Git

Notre travail sera mis à jour au lien suivant :

https://github.com/Romimap/3D-CryptoCompression/

Références

Demos & Softwares

1. Vladimir Agafonkin, "Edgebreaker, the Heart of Google Draco" : https://observablehq.com/@mourner/edgebreaker-the-heart-of-google-draco

2. Google Draco: https://github.com/google/draco

Papers

3. Michael Deering, "Geometry Compression", sun Microsystems, 1995 : http://web.cse.ohio-state.edu/~shen.94/Su01_888/deering.pdf

4. Chandrajit L Bajaj, Valerio Pascucci, Guozhong Zhuang, "Single Resolution Compression of Arbitrary Triangular Meshes with Properties", University of Texas, 1997: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.14.946&rep=rep1&type=pdf

5. Mike M. Chow, “Optimized Geometry Compression for Real-time Rendering”, Massachusetts institute of Technology, May 1997:
https://www.semanticscholar.org/paper/Optimized-geometry-compression-for-real-time-Chow/39babe9519e6b58bae4350e4f57be86dc45ce6f9

6. Jarek Rossignac, “Edgebreaker: Connectivity compression for triangle meshes”, Georgia Institute of Technology, 1999 :
https://www.cc.gatech.edu/~jarek/papers/EdgeBreaker.pdf

7. Daniel Cohen-Or, David Levin, Offir Remez, “Progressive Compression of Arbitrary Triangular Meshes”, Tel Aviv University, 1999:
https://www.tau.ac.il/~levin/vis99-dco.pdf

8. Jarek Rossignac, Alla Safonova, Andrzej Szymczak, “3D Compression Made Simple: Edgebreaker on a Corner-Table”, College of Computing and GVU Center, Georgia Institute of Technology, 2001:
https://www.researchgate.net/publication/3896746_3D_compression_made_simple_Edgebreaker_with_ZipandWrap_on_a_corner-table

9. Pierre Alliez, Mathieu Desbrun, “Progressive Compression for Lossless Transmission of Triangle Meshes”, University of Southern California, February 2002:
https://www.researchgate.net/publication/2534417_Progressive_Compression_for_Lossless_Transmission_of_Triangle_Meshes

10. Jarek Rossignac, “3D mesh compression”, College of Computing and GVU Center Georgia institute of Technology, January 2003:
https://www.researchgate.net/publication/27521282_3D_Mesh_Compression

11. Esam Elsheh, A. Ben Hamza, "Secret sharing approaches for 3D object encryption", Concordia Institute for Information Systems Engineering, Concordia University, Montréal, QC, Canada, 2011:
https://www.sciencedirect.com/science/article/abs/pii/S095741741100724X

12. In-Ho Lee and Myungjin Cho, "Optical Encryption and Information Authentication of 3D Objects Considering Wireless Channel Characteristics", Department of Electrical, Electronic, and Control Engineering, Hankyong National University, Ansung 456-749, Korea, October 2013:
https://www.osapublishing.org/DirectPDFAccess/766AED72-4C7E-47EE-BB28209C333886A8_276786/josk-17-6-494.pdf

13. Marc Éluard, Yves Maetz, and Gwenaël Doërr, "Geometry-preserving Encryption for 3D Meshes", Technicolor R&D France, November 2013:
https://www.researchgate.net/profile/Gwenael-Doerr/publication/273257218_Geometry-preserving_Encryption_for_3D_Meshes/links/54fc4b660cf2c3f52422a624/Geometry-preserving-Encryption-for-3D-Meshes.pdf

14. Xin Chen, Jingbin Hao, Hao Liu, Zhengtong Han and Shengping Ye, “Research on Similarity Measurements of 3D Models Based on Skeleton Trees”, School of Mechatronic Engineering, China University of Mining and Technology, Daxue Road 1, Xuzhou 221116, China, State Key Laboratory of Materials Forming and Mould Technology, Huazhong University of Science and Technology, Luoyu Road 1037, Wuhan 430074, China, April 2017:
https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwi91fmtwZj0AhUI2BoKHYcKA1AQFnoECAMQAQ&url=http%3A%2F%2Fwww.mdpi.com%2F2073-431X%2F6%2F2%2F17%2Fpdf-vor&usg=AOvVaw0lfo-8VxxYFuVQTnyCt6JI

15. Ying Zhou, Lingling Wang, Lieyun Ding, Cheng Zhou, “A 3D model Compression Method for Large Scenes”, Huazhong Univ. of Science and Technology, 2018:
https://www.iaarc.org/publications/fulltext/ISARC2018-Paper207.pdf