

## Champs de repères pour les modèles CAO

François Protais, Etienne Corman, Nicolas Ray, Dmitry Sokolov

► **To cite this version:**

François Protais, Etienne Corman, Nicolas Ray, Dmitry Sokolov. Champs de repères pour les modèles CAO. Journées Françaises d'Informatique Graphique (JFIG2020), Nov 2020, Nancy, France. pp.1 - 5. hal-02992612

**HAL Id: hal-02992612**

**<https://hal.inria.fr/hal-02992612>**

Submitted on 6 Nov 2020

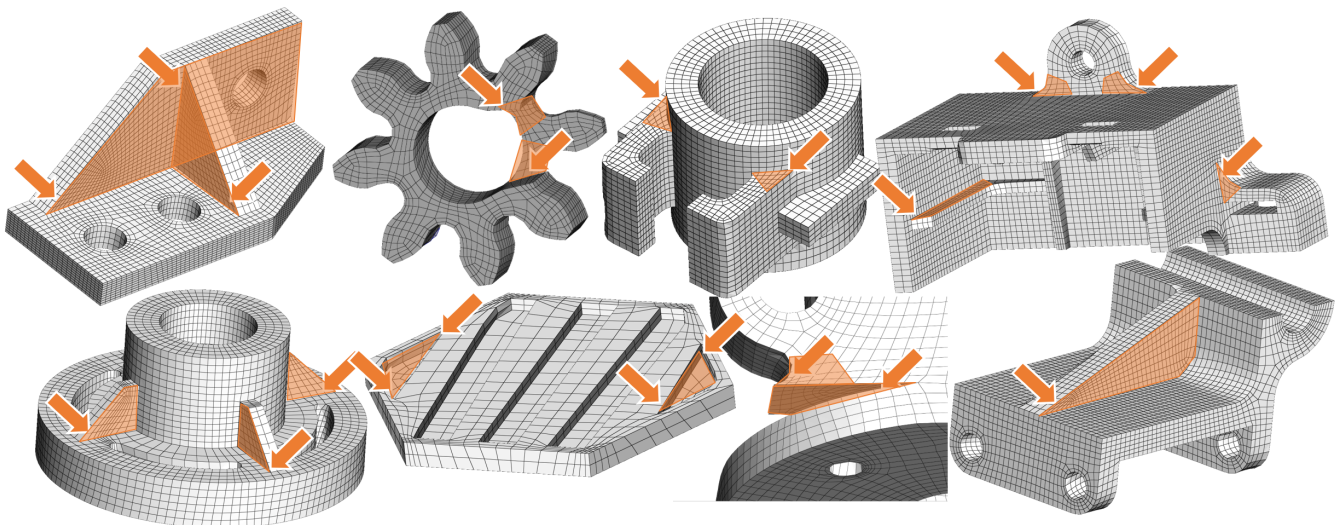
**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Champs de repères pour les modèles CAO

François Protais, Etienne Corman, Nicolas Ray et Dmitry Sokolov

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France



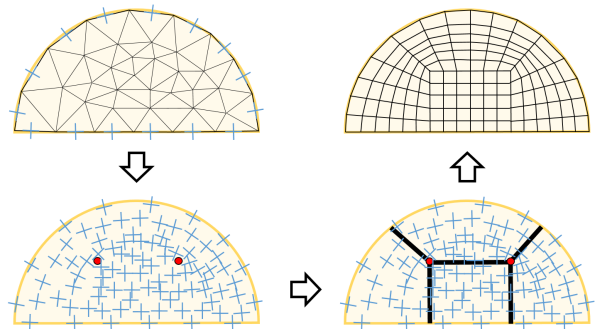
**Figure 1:** Maillage quadrangulaire de modèles CAO obtenus avec notre champ de repères. Le champ de repères le plus lisse, obtenu par une méthode usuelle, aurait une topologie incompatible avec un maillage quadrangulaire dans les zones montrées par les flèches en orange. L'impact de ces incohérences n'est pas local : cela produirait des quadrilatères dégénérés sur l'ensemble des zones surlignées en orange.

## Introduction

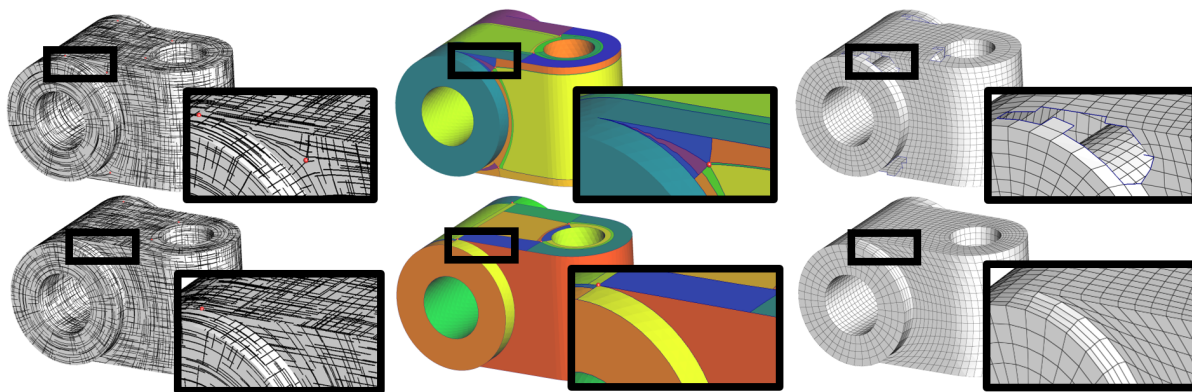
Les méthodes de génération de maillage quadrangulaire (*quad*) récentes se déroulent en deux étapes : tout d'abord, un champ de repères de guidage est calculé, ensuite une représentation paramétrique, ou paramétrage, représentant les quadrilatères est extraite. Le champ de repères définit l'orientation des quadrilatères à chaque point du domaine, tandis que l'étape de paramétrage détermine les positions des sommets des quadrilatères, voir Figure 2. Ainsi, les champs de repères sont des résultats intermédiaires dans le pipeline de génération de maillages quadrangulaires.

Un maillage quadrangulaire de haute qualité a généralement une faible distorsion et peu de sommets singuliers (valence  $\neq 4$ ). En règle générale, moins il y a de singularités dans le champ de repères, mieux c'est. Dans ce sens, les champs de repères sont produits par une optimisation numérique qui maximise la fluidité du champ, empêchant naturellement la création de singularités : au voisinage d'un point singulier, un champ a une courbure élevée qui est pénalisée par l'optimisation.

Les modèles CAO (conception assistée par ordinateur) sont



**Figure 2:** Partant d'un domaine triangulé, calculer un champ de repère permet de partitionner le domaine et d'en déduire un maillage quadrangulaire.

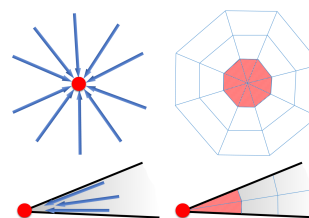


**Figure 3:** La production d'un champ de repères le plus lisse sur un modèle CAO n'est pas suffisante pour la segmentation par "motocycle" ou le maillage quadrangulaire (ligne supérieure). Notre algorithme permet de traiter ces configurations à bords vifs.

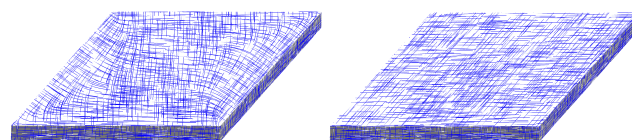
des surfaces particulières car beaucoup d'informations sont transportées par des arêtes vives. Ce réseau d'arêtes vives doit être préservé pendant le processus de remaillage. Ainsi, une arête caractéristique du maillage triangulaire d'entrée doit être une arête caractéristique du maillage quadrangulaire de sortie. Malheureusement, ces arêtes peuvent se rencontrer avec un angle aigu, ce qui rend impossible l'insertion d'un quadrilatère parfait dans un angle aussi aigu. En conséquence, le champ de repères calculé ne peut pas être transformé en un maillage quad et la méthode standard échoue. Les cas de défaillance typiques sont illustrés dans la partie gauche de la figure 8.

Ces arêtes vives problématiques sont omniprésentes dans les modèles de CAO car ils apparaissent naturellement à partir des filets et des chanfreins – dessins courants pour les pièces mécaniques. Dans l'ensemble de données ABC [KMJ\*19], une collection d'un millions de modèle CAO, nous avons constaté que plus de 65% des modèles ont des arêtes vives potentiellement problématiques. Dans Thingi10k [ZJ16], qui n'est pas spécialisée dans les modèles CAO, environ 35% de l'ensemble des données présentent au moins un coin pointu. Ainsi, les algorithmes actuels seraient incapables de produire des maillages quad sur une grande partie de ces ensembles de données. Dans cet article, nous proposons une solution pour mailler des surfaces, contenant des arêtes vives, avec des quadrilatères, comme l'illustre la figure 3.

**Contribution** Notre solution produit automatiquement des champs de repères orthogonaux de haute qualité sans singularités d'index 1 sur les modèles CAO (voir figure 4). La topologie du champs est obtenue en optimisant sa régularité, en restant aligné avec les arêtes vives. De plus, nous réduisons encore la rotation totale du champ en relaxant son orthogonalité (Figure 5), pour une meilleure correspondance avec l'orientation attendue des quadrilatères. Toutes ces contributions sont faciles à mettre en œuvre et ont un impact léger sur les performances : nous résolvons simplement deux systèmes linéaires en utilisant les outils d'optimisation numérique standard pour la génération du champ de repères. En outre, nous proposons une solution honnête pour le calcul de *motorcycle graphs* [EGKT08], qui a passé avec succès tous nos tests sur des géométrie difficiles, et qui est beaucoup plus simple que



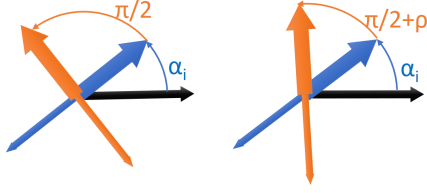
**Figure 4:** Un indice 1 singularité est un pôle cartographique (en haut à gauche). Les quadrilatères obtenus en traçant des lignes de force sont dégénérés en triangles sur le pôle (en haut à droite). Le champ le plus lisse contraint par un coin pointu présente le même type de singularité, placé sur la limite (en bas à gauche et à droite).



**Figure 5:** Les champs orthogonaux (à gauche) ne sont pas toujours la meilleure option pour un maillage quad. L'assouplissement de cette contrainte (à droite) donne de bien meilleurs résultats.

la solution robuste actuelle [RS14]. Les champs de sortie peuvent être facilement utilisés pour calculer un paramétrage globale [KNP07, BZK09] pour la génération de maillage quad, ou pour diviser la surface en régions quadrangulaires en traçant des *motorcycle graphs* non dégénérés [MPZ14, CBK15], comme détaillé dans la section 3.

Dans la suite de ce papier, après un court rappel de l'état de l'art, nous détaillons dans la section 1 notre approche pour éviter l'apparition d'indice 1 sur le bord dans les champs de repères. Ensuite, nous optimisons la courbure du champs en relaxant l'orthogonalité dans la section 2. Enfin, dans la section 3, nous montrons la qualité de nos champs en générant des maillages quadrilatères ainsi que des *motorcycle graphs*.



**Figure 6:** *Gauche* : Un repère orthogonal est un ensemble de quatre vecteurs unités associés aux angles  $\{\alpha, \alpha + \pi/2, \alpha + \pi, \alpha + 3\pi/2\}$  par rapport au vecteur référence (en noir). *Droite* : un repère non-orthogonal a un degré de liberté en plus,  $\rho_i$ , éloignant la seconde branche de l'orthogonalité.

### Travaux connexes

La génération de champs de repères est un domaine de recherche actif de la dernière décennie. Ils ont été introduits dans l'infographie pour placer des repères en rendu non photoréaliste [HZ00]. Cependant, l'application principale est le maillage quad par paramétrage global qui a été découvert quelques années plus tard [RLL\*06], et amélioré dans [BZK09, KNP07]. Les améliorations récentes les plus significatives sont consacrées à la robustesse, à commencer par [BCE\*13]. L'ingrédient clé des travaux suivants [MPZ14, CBK15] est basé sur des *motorcycle graphs* : une décomposition en parcelle de forme quadrilatère de la surface obtenue en traçant des lignes dans le champ de repères. Notre méthode rend cette décomposition plus robuste sur les modèles CAO, comme indiqué dans la section 3. Pour un état de l'art complet sur les champs de repères et leurs applications, nous invitons à la lecture de [VCD\*17].

### 1. Calcul de champ orthogonal

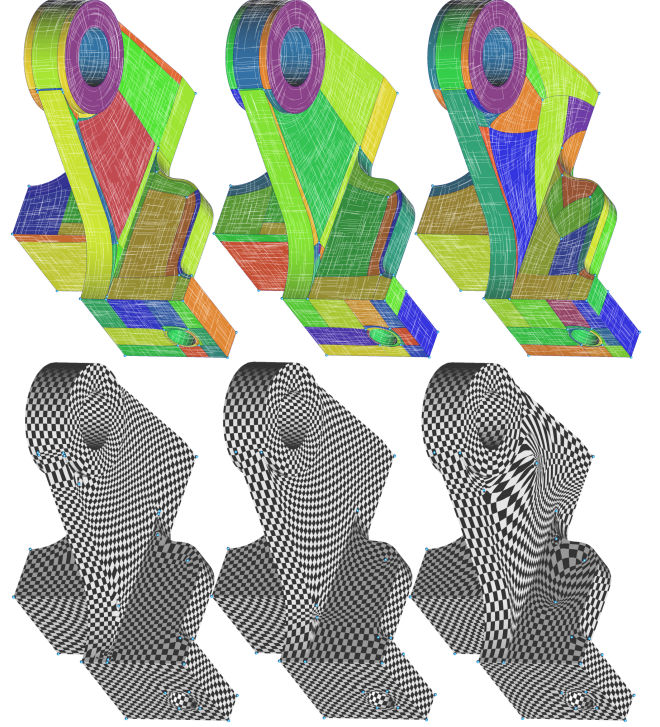
Notre pipeline prend comme entrée un maillage triangulaire avec des arêtes du bord identifiées, et calcule un champ de repères constant par triangle, non orthogonal, aligné avec les arêtes du bord. L'objectif est de calculer un champ de repères avec la rotation totale la plus faible ; le champ ne doit pas présenter de singularités d'indice 1 dans les coins aigus. Pour simplifier la notation, les arêtes caractéristiques seront considérées comme des bords.

Un repère sur un triangle consiste en un ensemble de quatre vecteurs unités tangents  $\{v, v^\perp, -v, -v^\perp\}$ . On donne à chaque triangle  $i$  un vecteur de référence  $b_i$  et on décrit le repère comme l'angle de rotation  $\alpha_i$  du vecteur de référence  $b_i$  autour de la normale. Le champ de repères est alors représenté comme un ensemble d'angles  $\{\alpha_i, \alpha_i + \pi/2, \alpha_i + 2\pi/2, \alpha_i + 3\pi/2\}$ , voir Fig. 6–gauche.

La rotation dans un champ de repères orthogonal est communément mesurée par la rotation  $r_{ij}$  entre deux repères de triangles adjacents  $i, j$  :

$$r_{ij} = -\alpha_i + \alpha_j + c_{ij} + p_{ij}\pi/2, \quad (1)$$

où  $c_{ij}$  donne la différence (après aplatissage des triangles) entre les vecteurs de référence  $b_i$  et  $b_j$ .  $-\alpha_i + \alpha_j + c_{ij}$  est alors la rotation entre les deux repères adjacents, et  $p_{ij}$  est l'entier qui signifie l'invariance des repères par rotation de  $\pi/2$ . L'approche usuelle



**Figure 7:** Impact de la localité de  $\omega_{ij}$  sur un *motorcycle graph* et une paramétrisation globale. *Gauche* : Fixer localement  $\omega$  sur les champs de repères maintient une grande divergence près des angles vifs. *Droite* : le correctif global perturbe le champ partout, en distribuant une courbure supplémentaire très loin des angles aigus. *Middle* : Notre solution permet d'obtenir un bon compromis.

pour obtenir le champs orthogonal le plus lisse est de minimiser la rotation totale du champs :

$$\arg \min \sum_{ij} r_{ij}^2 \quad \text{sous des contraintes de bord.} \quad (2)$$

Malheureusement, cette approche donne souvent des champs de repères incompatible avec un maillage quad (voir Fig 4). Pour éviter cela, nous proposons d'étudier une énergie un peu modifiée :

$$\arg \min \sum_{ij} (r_{ij} - \omega_{ij})^2 \quad \text{sous des contraintes de bord.} \quad (3)$$

avec  $\omega_{ij}$  une rotation objectif qui permet d'éviter les index 1. Ce  $\omega_{ij}$  est déduit près du bord lorsque les contraintes choisies induiraient directement un index 1. On peut décider de le propager vers l'intérieur pour diminuer la rotation par rapport à (2). La figure 7 montre différents choix : aucune propagation introduit une singularité très proche du bord et donc une forte courbure, à l'inverse propager globalement  $\omega_{ij}$  introduit de la courbure inutile loin des points d'intérêt. Une légère propagation locale permet d'obtenir un bon compromis.

## 2. Relaxation de l'orthogonalité

À ce stade, nous avons généré un champ de repères orthogonaux maillable avec des quads en minimisant  $\sum_{ij}(r_{ij} - \omega_{ij})^2$ . Cependant, sa géométrie peut être optimisée davantage en relâchant la contrainte d'orthogonalité (voir Figure 5). Nous introduisons pour chaque repère un angle  $\rho_i$  qui ajoute de la non-orthogonalité (voir Fig 6).

Pour comparer la rotation, (1) se scinde maintenant en deux parties :

$$\begin{cases} r_{ij}^0 = -\alpha_i + \alpha_j + c_{ij} + p_{ij}\frac{\pi}{2} + (p_{ij} \bmod 2)\rho_j \\ r_{ij}^1 = -\alpha_i - \rho_i + \alpha_j + c_{ij} + p_{ij}\frac{\pi}{2} + (1 - (p_{ij} \bmod 2))\rho_j \end{cases} \quad (4)$$

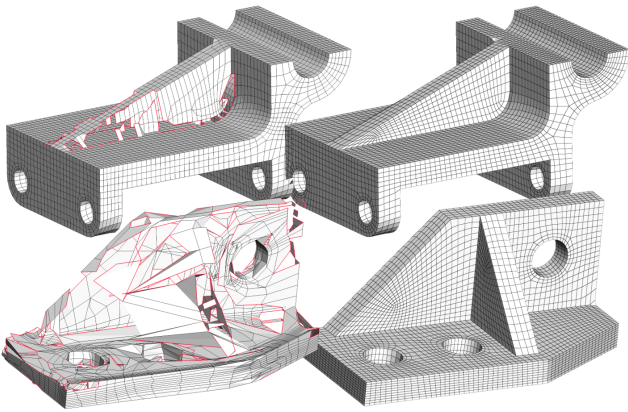
où chacune des branches du repère de  $i$  s'associe avec une des branches du repères de  $j$ . Dans ce cadre, minimiser la rotation totale s'exprime donc de la façon suivante :

$$\arg \min \sum_{ij} \left( r_{ij}^0 \right)^2 + \left( r_{ij}^1 \right)^2 \quad \text{sous des contraintes de bord.} \quad (5)$$

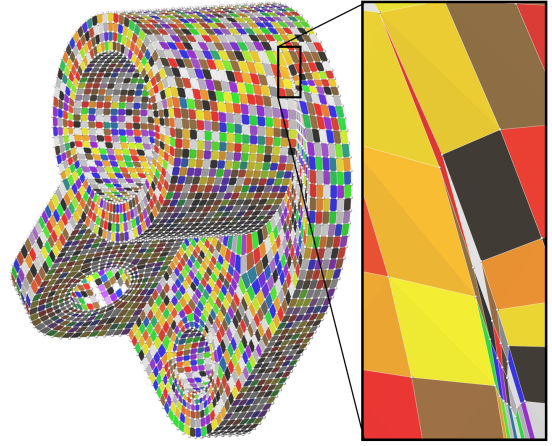
Or, dans l'expression de cette énergie, rien n'empêche la dégénérescence des repères (lorsque  $\rho_i = \pm\pi/2$ ). Pour éviter toute dégénérescence, nous partons de la solution obtenue dans la section 1, et nous ajoutons un léger terme pour favoriser l'orthogonalité du champ :

$$\arg \min \left\{ \sum_{ij} \left[ \left( r_{ij}^0 \right)^2 + \left( r_{ij}^1 \right)^2 \right] + \varepsilon \sum_i \rho_i^2 \right\},$$

où  $\varepsilon = 10^{-2}$ . En plus de cela, une fois l'énergie minimisée, nous vérifions que  $|\rho_i| < \alpha$ , où  $\alpha$  est l'écart maximal par rapport à l'orthogonalité. Si un repère est trop déformé, nous le corrigeons localement en appliquant la plus petite modification possible à  $\rho_i$ .



**Figure 8:** Maillages quads obtenus à partir de champs de repères classiques (à gauche) et avec nos champs de repères (à droite). Dans le premier modèle, une grande parcelle est dégénérée. Dans le second exemple, le paramétrage global a échoué pour le champ de repères classique, alors qu'il fonctionne parfaitement avec notre champ non orthogonal.



**Figure 9:** Notre méthode produit un maillage quad aligné avec des arêtes vives même en présence de coins extrêmement aigus (gros plan).

## 3. Résultats et application

Le calcul du champs est fait en utilisant le "vecteur de représentation", voir [RSL16], et notre méthode consiste donc en la résolution de deux systèmes linéaires. Bien qu'en pratique 3 à 5 fois plus lent que le calcul d'un champ classique, cela reste largement négligeable par rapport au calcul d'un paramétrage global. Par exemple, sur le modèle de la figure 7 avec 35k triangles, le calcul du champ a mis 3 sec., contre 155 pour le paramétrage.

**Maillage quad** Nous calculons un paramétrage global en utilisant une méthode similaire à [CBK15], en utilisant nos champs. Ensuite, nous utilisons [EBCK13] pour extraire le quadrillage. Il faut noter que les contraintes du bord rendent le paramétrage global beaucoup plus difficile à obtenir, mais nos maillages sont très fidèles aux arêtes vives du bord (cf. Fig 8 et 9).

**Motorcycle graph** Les *motorcycle graph* sont primordiales dans les méthodes robustes de générations de quad [MPZ14, CBK15, KLF12]. Malheureusement, les solutions robustes pour les tracer [MPZ14, RS14] ne sont pas adaptées à notre représentation. Le principal problème est lorsque deux *motorcycles* fusionnent, ce qui arrive le long d'arêtes "sink". Nous détectons de telles arêtes, et nous résolvons le problème en modifiant très localement une branche du champ, ce qui a peu de chance de créer une nouvelle arête "sink". Dans nos tests, cette méthode s'est montré efficace même sur des angles extrêmement aigus, voir figure 9.

## Conclusion

Ce travail est une solution honnête pour générer des champs de repères sur des modèles CAO. Il est robuste aux angles aigus, plus lisse que les champs de repères classiques en relaxant l'orthogonalité, et peut être facilement coupé en lançant des *motorcycles*. La production d'un maillage quad à partir de ces champs reste un problème difficile, en particulier pour les modèles CAO où le maillage doit capturer les courbes caractéristiques par des arêtes des quads.

## References

- [BCE\*13] BOMMES D., CAMPEN M., EBKE H.-C., ALLIEZ P., KOBBELT L.: Integer-grid maps for reliable quad meshing. *ACM Trans. Graph.* 32, 4 (July 2013), 98:1–98:12. URL: <http://doi.acm.org/10.1145/2461912.2462014>, doi:10.1145/2461912.2462014. 3
- [BZK09] BOMMES D., ZIMMER H., KOBBELT L.: Mixed-integer quadrangulation. *ACM Trans. Graph.* 28, 3 (July 2009), 77:1–77:10. URL: <http://doi.acm.org/10.1145/1531326.1531383>, doi:10.1145/1531326.1531383. 2, 3
- [CBK15] CAMPEN M., BOMMES D., KOBBELT L.: Quantized global parametrization. *ACM Trans. Graph.* 34, 6 (Oct. 2015). URL: <https://doi.org/10.1145/2816795.2818140>, doi:10.1145/2816795.2818140. 2, 3, 4
- [EBCK13] EBKE H.-C., BOMMES D., CAMPEN M., KOBBELT L.: Qex: robust quad mesh extraction. *ACM Transactions on Graphics (TOG)* 32, 6 (2013), 1–10. 4
- [EGKT08] EPPSTEIN D., GOODRICH M. T., KIM E., TAMSTORF R.: Motorcycle graphs: Canonical quad mesh partitioning. In *Proceedings of the Symposium on Geometry Processing (Goslar, DEU, 2008)*, SGP '08, Eurographics Association, pp. 1477–1486. 2
- [HZ00] HERTZMANN A., ZORIN D.: Illustrating smooth surfaces. In *PROCEEDINGS OF SIGGRAPH 2000* (2000), pp. 517–526. 3
- [KLF12] KOWALSKI N., LEDOUX F., FREY P.: A PDE based approach to multi-domain partitioning and quadrilateral meshing. 4
- [KMJ\*19] KOCH S., MATVEEV A., JIANG Z., WILLIAMS F., ARTEMOV A., BURNAEV E., ALEXA M., ZORIN D., PANOZZO D.: Abc: A big cad model dataset for geometric deep learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019). 2
- [KNP07] KAELBERER F., NIESER M., POLTHIER K.: QuadCover - Surface Parameterization using Branched Coverings. *Computer Graphics Forum* (2007). doi:10.1111/j.1467-8659.2007.01060.x. 2, 3
- [MPZ14] MYLES A., PIETRONI N., ZORIN D.: Robust field-aligned global parametrization. *ACM Trans. Graph.* 33, 4 (July 2014). URL: <https://doi.org/10.1145/2601097.2601154>, doi:10.1145/2601097.2601154. 2, 3, 4
- [RLL\*06] RAY N., LI W. C., LÉVY B., SHEFFER A., ALLIEZ P.: Periodic global parameterization. *ACM Trans. Graph.* 25, 4 (Oct. 2006), 1460–1485. URL: <http://doi.acm.org/10.1145/1183287.1183297>, doi:10.1145/1183287.1183297. 3
- [RS14] RAY N., SOKOLOV D.: Robust polylines tracing for n-symmetry direction field on triangulated surfaces. *ACM Transactions on Graphics* (2014). 2, 4
- [RSL16] RAY N., SOKOLOV D., LÉVY B.: Practical 3d frame field generation. *ACM Trans. Graph.* 35, 6 (Nov. 2016). URL: <https://doi.org/10.1145/2980179.2982408>, doi:10.1145/2980179.2982408. 4
- [VCD\*17] VAXMAN A., CAMPEN M., DIAMANTI O., BOMMES D., HILDEBRANDT K., TECHNION M. B.-C., PANOZZO D.: Directional field synthesis, design, and processing. In *ACM SIGGRAPH 2017 Courses* (New York, NY, USA, 2017), SIGGRAPH '17, Association for Computing Machinery. URL: <https://doi.org/10.1145/3084873.3084921>, doi:10.1145/3084873.3084921. 3
- [ZJ16] ZHOU Q., JACOBSON A.: Thingi10k: A dataset of 10,000 3d-printing models. *arXiv preprint arXiv:1605.04797* (2016). 2