

# Calcul efficace d'interfaces dans des assemblages

François JOURDES, Georges-Pierre BONNEAU, Stefanie HAHMANN, Jean-Claude LEON et François FAURE

Laboratoire Jean Kuntzmann (Université de Grenoble, CNRS) et Inria

## Résumé

La préparation de modèles CAO d'assemblages complexes à des fins de simulation est un processus long et très fastidieux, puisque de nombreuses tâches telles que le maillage et l'idéalisation sont toujours remplis effectuées manuellement. Pour cela, la détection et l'extraction des interfaces géométriques entre les composants de l'ensemble sont d'une importance capitale non seulement pour les objectifs de la simulation, mais aussi pour toutes les transformations de forme nécessaires, telles que des idéalizations ou des suppressions de détails. Il s'agit d'une tâche répétitive, en particulier lorsque des assemblages complexes doivent être traités. Cet article propose une méthode pour générer rapidement et de manière entièrement automatique une description géométrique précise des interfaces dans les modèles de CAO B-Rep d'assemblages. L'approche combine une technique de ray-casting accélérée par GPU couramment utilisée en informatique graphique avec un algorithme d'extraction de courbes à base de graphes. Non seulement il est capable de détecter un grand nombre d'interfaces de manière efficace, mais il fournit également une géométrie NURBS précise des interfaces, qui peut être stockée dans un simple fichier STEP en vue de post-traitement. Nous démontrons notre approche sur des exemples tirés de l'industrie aéronautique et automobile.

**Mots-clés :** Modélisation géométrique, assemblages, calcul d'interfaces, GPU, B-Rep, CAO.

## 1. Introduction

L'utilisation quasi-systématique de logiciels de CAO depuis plusieurs décennies dans les industries a conduit à des représentations communes des produits sous la forme d'assemblage de composants, également désigné par maquette numérique ou digital mock up (DMU) de plus en plus complexes, allant de dizaines de composants pour les produits simples à des milliers de composants pour une voiture et jusqu'à des centaines de milliers de composants pour un avion.

La détection et le calcul des interfaces entre les composants de modèles CAO B-Rep est une étape clé pour l'analyse efficace des assemblages. Une interface entre deux composants est une partie de surface commune aux deux composants et peut être vue comme l'empreinte d'une surface sur l'autre et vice-versa, ainsi qu'illustré à la Figure 1. La connaissance des interfaces est essentielle pour plusieurs traitements de la DMU comme par exemple l'idéalisation de composants, la simulation par éléments finis ou encore la simulation en réalité virtuelle. Plus généralement, les inter-

faces géométriques de composants apparaissent comme des entités clés pour traiter efficacement les assemblages. Cependant, ces entités ne sont pas facilement disponibles dans les DMU et un traitement de la géométrie spécifique est nécessaire pour obtenir les interfaces souhaitées. En fonction du nombre de composants dans un assemblage ce traitement peut impliquer des mois-homme de travail.

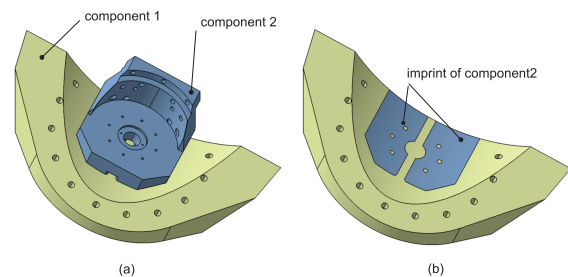


Figure 1: deux composants voisins et leur interface (remerciements ANTECIM). (a) les deux composants dans leur position d'assemblage. (b) l'empreinte du composant 2 sur composant 1 représentant l'interface géométrique précise définie par une surface cylindrique.

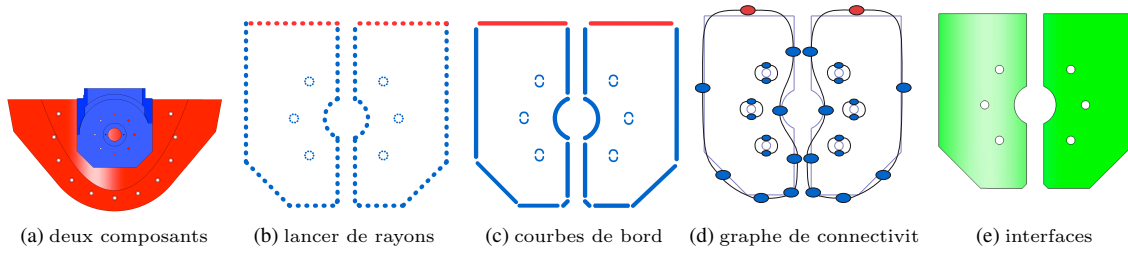


Figure 2: l'algorithme en 4 étapes.

Sur la base de l'état de l'art actuel (littérature académique et logiciels industriels) concernant le traitement des interfaces en CAD/CAM, nous pouvons faire les observations suivantes. Premièrement, la plupart des travaux antérieurs ne comporte pas de description précise des interfaces géométriques de composants. Deuxièmement, aucune norme n'est effectivement mise en œuvre dans les logiciels industriels de CAD/CAM pour exporter les interfaces géométriques à d'autres logiciels. Quand il s'agit de grands assemblages, ce manque de description détaillée des interfaces géométriques devient un véritable défi. Des opérateurs interactifs de bas niveau permettant le trimming des surfaces frontières de composants deviennent trop fastidieux pour un calcul pratique des surfaces empreintes [CV06, CHE08, YFWY07, IML08]. Pour élargir cette approche à des assemblages très complexes, de nouvelles approches plus automatisées sont nécessaires. Finalement, nous observons que les opérateurs géométriques de haut niveau pour la génération des interfaces géométriques, comme le calcul d'offset, la génération d'empreintes [CHE08, ans12] et les opérateurs type CSG [CV06] ainsi que les solveurs de contraintes [DS10, YFWY07] ou encore les approches de design fonctionnel [RPS\*01, RB02, KWMN04, RT12] ne s'adaptent pas bien aux grands assemblages en terme de temps de calcul et de robustesse.

Dans cet article, nous présentons une nouvelle méthode pour détecter très rapidement et calculer efficacement sous forme NURBS l'empreinte de chaque composant sur chacun de ses composants voisins.

Sur la base de nos observations, nous proposons une approche qui prend en entrée un modèle d'assemblage CAD/CAM et automatise la génération des interfaces géométriques entre les composants. Notre approche permet de détecter et de décrire avec précision la géométrie des interfaces tout en passant à l'échelle pour les assemblages très complexes. Nos principales contributions sont les suivantes :

- Une nouvelle méthode pour détecter les interfaces dans les modèles B-Rep NURBS en utilisant du ray-casting et pour calculer la géométrie exacte des interfaces sous forme B-Rep NURBS par une série d'étapes de post-traitement.
- La mise en œuvre efficace des parties les plus intensives en calcul de la méthode sur l'unité de traitement

graphique (GPU), permettant de traiter des milliers de faces B-Rep en quelques minutes.

## 2. Calcul d'interfaces

Le travail présenté a été développé dans le cadre de l'ANR ROMMA. La description qui suit résume et illustre les principales étapes de la détection et du calcul de la géométrie des interfaces dans des assemblages. Plus de détails sur les algorithmes mis en place pourront être trouvés dans [JBH\*14]. Figure 2 donne un aperçu des différentes étapes de la méthode.

Notre approche est basée sur les hypothèses suivantes :

- (A1) l'empreinte est une surface 2-variété avec bords.
- (A2) chaque point sur le bord d'une interface appartient au bord d'au moins un composant.

L'hypothèse (A2) est raisonnable, étant donné que les interfaces qui n'incluent pas les bords des composants ne se produisent que dans certains cas dégénérés de points de contact, contact linéaire ou des formes libres, mais sont plutôt rares dans la modélisation d'assemblage. Nous reportons l'étude d'une méthode tout à fait générale pour les travaux futurs.

Pour calculer les interfaces entre deux composants (Fig.2a), nous tirons des rayons courts à partir des bords des facettes afin de détecter des intersections à proximité (Fig.2b). Celles-ci sont regroupées pour représenter les courbes de bord de l'empreinte (Fig.2c). Nous calculons la connectivité des courbes (fig.2d) pour détecter des cycles, qui définissent les surfaces de l'empreinte (Fig.2e).

### 2.1. Préparation des modèles NURBS

L'entrée de notre algorithme consiste en un ou plusieurs fichiers STEP contenant les composants de l'assemblage sous forme d'un NURBS B-Rep. Dans un fichier STEP, les surfaces délimitant les composants sont représentées par des patches B-spline (éventuellement trimmés) définis sur un domaine de paramètre explicite, et chaque courbe de bord

est paramétrée individuellement. La librairie OpenCascade [occ12] est utilisée pour lire les fichiers STEP et calculer un maillage de chaque composant de l'assemblage avec une tolérance d'erreur  $\epsilon$  donnée. Les arêtes des courbes de bord et les triangles dans ces maillages sont utilisés par la librairie OPTIX, comme expliqué ci-dessous, pour la détection d'interfaces.

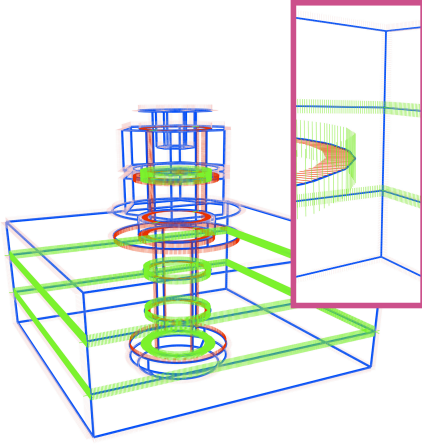


Figure 3: Résultat du lancer de rayons. Dans figure seul un échantillon des rayons est représenté à une échelle agrandie, afin d'assurer la visibilité. Les rayons sont roses si ils ne correspondent pas à des proximités entre composants, rouges si ils correspondent à des proximités entre une courbe de bord dans un composant et l'intérieur d'une face B-Rep dans un autre composant, vert si ils correspondent à des proximités entre deux courbes de bord dans deux composants différents.

## 2.2. Détection d'interfaces par lancé de rayons

Le cœur de notre système consiste à lancer des rayons à travers l'objet et rechercher des paires de points d'intersection voisins entre un rayon et deux composants. Une telle paire indique l'existence d'une interface entre les deux composants.

Nous lançons des rayons courts à partir des bords des surfaces discrétisées afin de calculer les bords de l'empreinte, en cohérence avec l'hypothèse (A2). Les rayons sont créés à partir de chaque face et lancés dans la direction de la normale à la surface. Les rayons sont générés pour tous les faces B-Rep des composants le long de leurs courbes de bord en échantillonnant les courbes b-spline uniformément selon la précision du modèle CAO. Ainsi, à partir d'une courbe de bord partagée par deux faces B-Rep d'un même composant, des rayons sont lancés dans deux directions.

Nous utilisons OPTIX, un moteur de lancer de rayons implémenté sur GPU très efficace, disponible librement [PBD\*10]. L'utilisateur envoie un ensemble de primitives géométriques à la carte graphique. Le système initialise une structure d'accélération stockée dans le GPU, basée sur des

hiérarchies de volumes englobants. Cette structure permet de trouver rapidement les intersections possibles entre les rayons et les primitives géométriques. L'utilisateur fournit un ensemble de programmes pour calculer les boîtes englobantes des primitives géométriques afin de tester et calculer les intersections avec les primitives géométriques, et pour le traitement des intersections. Tous les aspects parallèles de la mise en œuvre sont traitées par la librairie, de manière transparente pour l'utilisateur.

Afin d'améliorer la robustesse et d'accélérer les calculs de proximité sur le GPU, nous discrétisons le modèle d'entrée en arêtes pour les courbes de bord et triangles pour les faces du B-Rep, et nous générons des boîtes englobantes alignées aux axes pour ces primitives géométriques. Un rayon passant à proximité de la courbe de bord d'un composant ou d'une face B-Rep croisera toujours l'une des boîtes. A noter que la discrétisation n'est utilisée uniquement que pour l'accélération du lancé des rayons OPTIX. Cependant, pour tous les autres calculs (calcul des courbes de bords de l'interface et génération de la surface de l'interface) la représentation NURBS exacte est utilisée.

Nous procédons en deux étapes : d'abord, OPTIX trouve toutes les proximités potentielles entre les rayons et des composants en utilisant les boîtes englobantes des primitives géométriques. Ensuite, les proximités potentielles sont post-traitées, toujours dans le GPU, afin d'éliminer les proximités invalides, et d'identifier les valeurs de paramètre des couples de point proches dans le modèle B-Rep.

Le GPU fournit alors des paires de points  $(P, Q')$  où  $P$  est à l'origine du rayon et  $Q'$  est un point, près de  $P$ , appartenant à la discrétisation des composants (arêtes et triangles du maillage).

- Au cas où  $Q'$  appartient à une arête, nous calculons le point correspondant  $Q$  sur le segment de la courbe de bord.
- Au cas où  $Q'$  appartient à l'intérieur d'un triangle, nous n'avons pas besoin du point d'intersection exact avec la face B-Rep à laquelle le triangle appartient. Il suffit de stocker l'indice de cette face B-Rep.

Nous obtenons alors des paires de points proches  $(P, Q)$  où  $P$  comme origine du rayon appartient à une courbe de bord d'une surface d'un composant, et où  $Q$  se trouve soit à l'intérieur d'une surface d'un autre composant, soit sur une de ses courbes de bord. Ce dernier cas correspond à la situation fréquente où deux faces de composants différents partagent une section de la même courbe de bord, voir par exemple les trois plaques en Figure 7(a).

Figures 9(d) et 6(b) montrent le résultat du processus de ray-casting sur des assemblages complexes. Figure 3 montre un exemple simple, correspondant à l'assemblage en Figure 7.

En sortie de la partie lancer de rayons un ensemble de points situés sur les bords de l'interface est produit. Dans

les sections suivantes nous présentons la méthode de reconstruction de la géométrie de toutes les interfaces à partir de cet ensemble de points.

### 2.3. Groupements de points bord d'interfaces

Une fois les points appartenant au bord des interfaces détectés, la prochaine étape de notre méthode de reconstruction des interfaces consiste à grouper pour chacune des interfaces les points adjacents et à les ordonner à l'intérieur de chaque groupe selon leur valeur de paramètre respective. Ainsi chaque groupe via la paramétrisation connue pour ses points représente une section de courbe NURBS. A chaque fois que la distance entre deux points successifs est supérieure à un seuil, un nouveau groupe est créé. Figures 2(c), 4(c) et 5(c) montrent les groupes de points sur trois exemples.

Les valeurs des paramètres min et max de points dans chaque groupe déterminent une section d'une courbe de bord d'une face, qui limite également l'interface. Ainsi, chaque groupe de points adjacents peut également être considéré comme une section de la courbe de bord de l'interface. Mais inversement, le bord de l'interface se compose uniquement d'un sous-ensemble des groupes de points précédemment calculés. Il peut y avoir redondance due au fait que des groupes de points se chevauchent partiellement lorsque les points appartiennent au bord des deux faces.

### 2.4. Courbes de bord des interfaces

Afin de choisir parmi tous les groupes de points un sous-ensemble complet représentant tous les bords d'une même interface, nous introduisons une approche par parcours de graphes.

Un graphe orienté est défini pour chaque interface comme suit. Les groupes de points adjacents représentent les noeuds du graphe. Deux types d'arcs sont établis :

- Arcs entre les noeuds appartenant à la même face : Lorsque deux noeuds correspondent à des sections adjacentes de courbes de bord de la même face, un arc est établi dans la direction correspondante à l'orientation des courbes. Ces informations peuvent être récupérées à partir du modèle B-Rep de la face.
- Arcs entre noeuds de faces différentes : Ceci est le cas quand deux groupes de points de faces différentes se chevauchent.

Un exemple d'un tel graphe est représenté en Figure 4(d). Il correspond aux deux composants représentés en Figure 4(a,b). Il y a 12 noeuds dans cet exemple. Les noeuds sont colorés en fonction du composant à partir duquel les rayons ont été lancés afin de détecter les points d'intersection. 4 noeuds appartiennent au composant inférieur en Figure 4, et 8 noeuds au composant supérieur. 8 arcs joignent les noeuds de différentes faces, et 4 arcs rejoignent des noeuds

de la même face. A noter que les deux noeuds rouges en bas correspondent à différentes sections de la même courbe de bord du composant rouge. Dans cet exemple, il n'y a pas de redondance. Par conséquent, le graphe est un cycle.

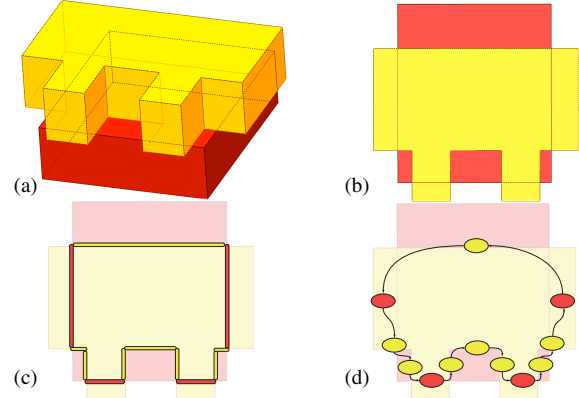


Figure 4: (a,b) deux composants. (c) groupe de points adjacents. (d) graphe correspondant.

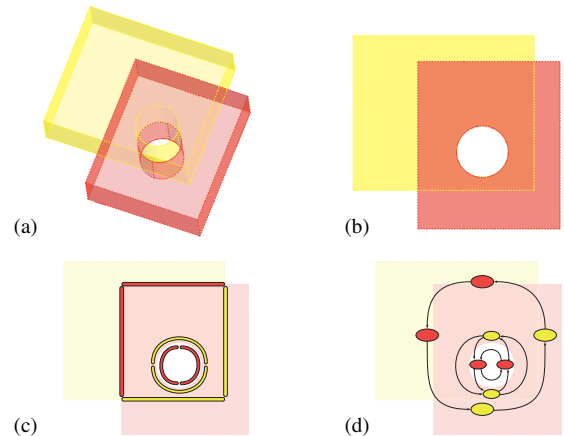


Figure 5: (a,b) deux composants partageant le même trou circulaire au niveau de leur interface, mais avec une paramétrisation différente, (c) groupe de points adjacents, (d) graphe correspondant. En raison de la redondance, la composante fortement connexe du graphe correspondant au trou circulaire n'est pas un cycle.

En général, le graphe possède au moins une composante fortement connexe qui correspond au bord extérieur l'interface. En outre, des composants supplémentaires apparaissent pour chaque trou (bords intérieurs) dans l'interface. Etant donné qu'une partie de la courbe de bord de l'interface peut appartenir à deux faces, les composantes connexes ne sont pas toujours des cycles, comme le montre Figure 5(d). Dans tous les cas, un cycle peut être extrait à partir de n'importe quel composante fortement connexe, puisque



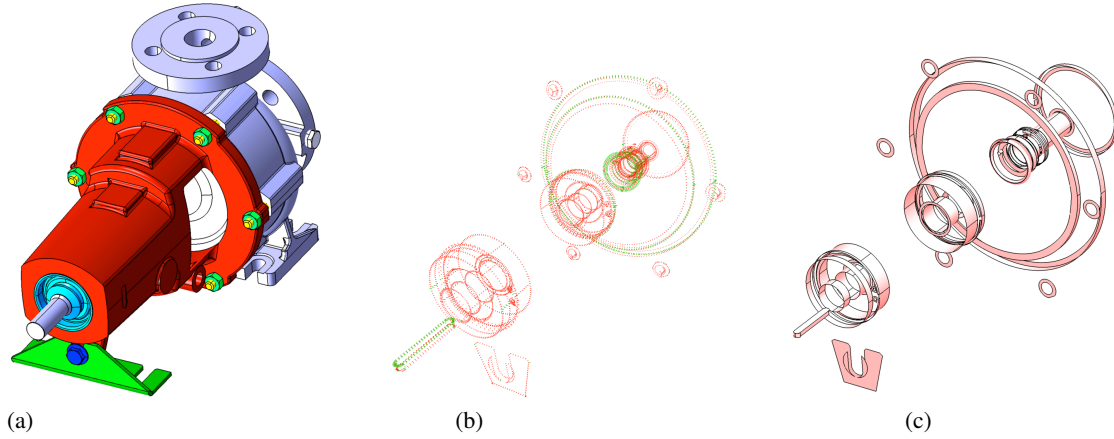


Figure 6: Pompe hydraulique. (a) modèle B-Rep, (b) résultat du lancer de rayons (code couleur selon Fig. 3, (c) interfaces.

les graphes orientés fortement connexes sont cycliques. Une fois le graphe construit, chaque bord de l'interface peut être extrait comme un cycle.

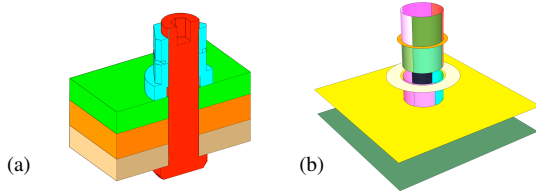


Figure 7: Jonction boulonnée avec trois plaques, (a) vue de coupe, (b) interfaces.

## 2.5. Interface NURBS

Les bords de l'interface sont maintenant extraits à partir du graphe comme cycles. Chaque noeud représente une section de courbe paramétrée soit dans le domaine paramétrique de la face B-Rep  $A$  ou  $B$ . La dernière étape de notre méthode de reconstruction consiste à calculer la surface NURBS de l'interface et à la stocker en modèle B-Rep. Il s'agit en fait d'une surface trimmée à la fois de  $A$  et  $B$ . L'interface peut ainsi être paramétrée soit dans le domaine de paramètre de la face  $A$  ou dans celui de la face  $B$ . Si nous choisissons le domaine de paramètre de  $A$ , chaque section de courbe de bord de l'interface de  $B$  doit être paramétrée dans le domaine de paramètre de  $A$ . Cette reparamétrisation est implémentée en utilisant la librairie OpenCascade.

Pour chaque noeud dans le cycle, on stocke le paramétrage directement dans le modèle B-Rep si la section de la courbe correspondante est définie dans le domaine de paramètre choisi, sinon on reparamétrise la courbe. En utilisant cette procédure, le bord extérieur de l'interface, ainsi que tous les bords intérieurs, sont entièrement codés dans le modèle B-Rep, avec l'orientation requise. Si nécessaire, ce mo-

dèle peut être exporté dans un fichier STEP standard pour le traitement en aval par d'autres logiciels.

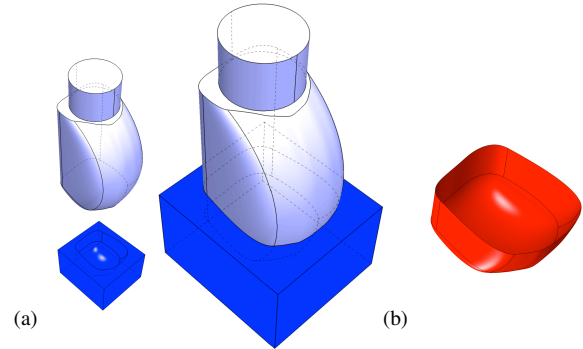


Figure 8: Surface de forme libre, (a) deux composants, (b) leur interface.

## 3. Résultats

Nous avons testé notre méthode avec différents exemples, certains sont académiques, d'autres sont des exemples industriels. Les interfaces dans ces modèles comprennent des surfaces planaires, cylindriques et de forme libre NURBS. La pompe hydraulique en Figure 6 par exemple a 46 composants avec un total de 2195 faces B-Rep. 565 courbes de bord ont été calculées. L'exemple en Figure 9 est une pièce aéronautique, une partie d'un root joint caractérisé par de nombreuses jonctions boulonnées. La particularité de l'exemple en Figure 8 réside dans l'interface de forme libre dont la géométrie est décrite exactement. Les temps de calcul sont extrêmement courts pour des dizaines de millions de rayons lancés et d'intersections calculées en parallèle. Plus de détails chiffrés sur le nombre de composants, faces B-Rep, rayons lancés, points détectés ainsi que les temps de calculs sont exposés en [JBH\* 14].

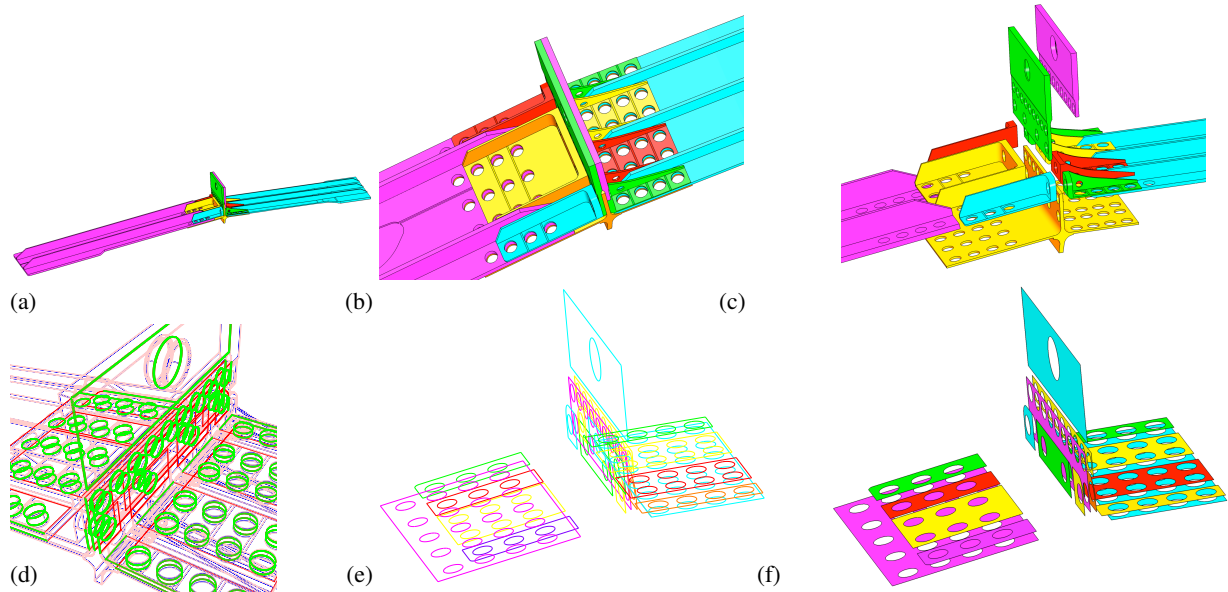


Figure 9: Partie de l'avion pour l'assemblage des ailes avec le corps d'un aéronaf (modèle de courtoisie d'Airbus Group). (a,b) deux des composants, (c) vue explosée, (d) résultat du lancer de rayons avec le même code couleur qu'en Fig. 3, (e) reconstruction des courbe de bord, (f) interfaces.

## Remerciements

Le projet ROMMA est soutenu financièrement par l'Agence Nationale de la Recherche (programme COSINUS). NVIDIA a offert une carte graphique professionnelle à notre équipe de recherche.

## Références

- [ans12] : ANSYS Workbench v.14, 2012.
- [CHE08] CLARK B., HANKS B., ERNST C. : Conformal assembly meshing with tolerant imprinting. In *Proc. 17th Meshing Roundtable, Pittsburg, USA, October 12–15 (2008)*, pp. 267–280.
- [CV06] CHOUADRIA R., VÉRON P. : Identifying and remeshing contact interfaces in a polyhedral assembly for digital mock-up. *Eng. with Computers*. Vol. 22, Num. 1 (2006), 47–58.
- [DS10] DIXON A., SHAH J. J. : Assembly feature tutor and recognition algorithms based on mating face pairs. *CAD and Applications*. Vol. 7, Num. 3 (2010), 319–333.
- [IML08] IACOB R., MITROUCHEV P., LÉON J.-C. : Contact identification for assembly/disassembly simulation with a haptic device. *Visual Computer*. Vol. 24, Num. 11 (2008), 973–979.
- [JBH\*14] JOURDES F., BONNEAU G.-P., HAHMANN S., LÉON J.-C., FAURE F. : Computation of components' interfaces in highly complex assemblies. *Comput. Aided Des.* Vol. 46 (janvier 2014), 170–178.
- [KWMN04] KIM K.-Y., WANG Y., MUOGBOH O. S., NNAJI B. O. : Design formalism for collaborative assembly design. *CAD*. Vol. 36, Num. 9 (2004), 849–871.
- [occ12] : OpenCascade CAD software library, 2012.
- [PBD\*10] PARKER S. G., BIGLER J., DIETRICH A., FRIEDRICH H., HOBEROCK J., LUEBKE D., MCALLISTER D., MCGUIRE M., MORLEY K., ROBISON A., STICH M. : Optix : A general purpose ray tracing engine. *ACM Transactions on Graphics (August 2010)*.
- [RB02] ROY U., BHARADWAJ B. : Design with part behaviors : behavior model, representation and applications. *CAD*. Vol. 34, Num. 9 (2002), 613–636.
- [RPS\*01] ROY U., PRAMANIK N., SUDARSAN R., SRIRAM R. D., LYONS K. W. : Function-to-form mapping : model, representation and applications in design synthesis. *CAD*. Vol. 33, Num. 10 (2001), 699–719.
- [RT12] RAHMANI K., THOMSON V. : Ontology based interface design and control methodology for collaborative product development. *CAD*. Vol. 44, Num. 5 (2012), 432–444.
- [YFWY07] YANG R., FAN X., WU D., YAN J. : Virtual assembly technologies based on constraint and dof analysis. *RCIM*. Vol. 23 (2007), 447–456.