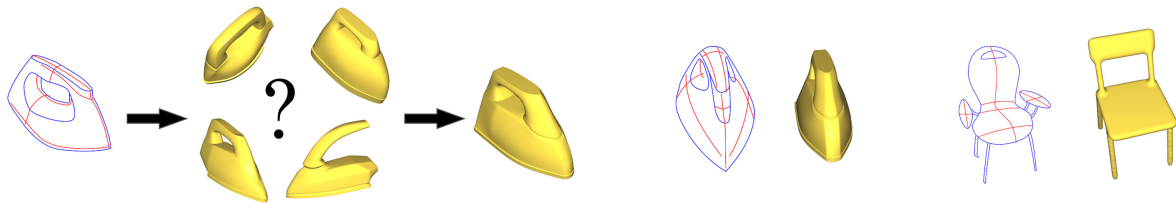


# Une approche basée donnée pour retrouver le point de vue d'un dessin de design

Johanna Delanoy, Adrien Bousseau

INRIA, France



**Figure 1:** Notre approche permet de retrouver, à partir d'un rapide croquis de design, le point de vue selon lequel est vu l'objet. Pour cela, nous mettons en correspondance le dessin avec les modèles 3D d'une base de données et retrouvons le point de vue le plus probable selon ces objets.

## Résumé

Le processus de design d'objet nécessite de passer fréquemment d'une représentation 2D, le croquis, à une représentation 3D. Parce que cette transformation est coûteuse en temps, elle n'est pratiquée que lorsque le design est suffisamment avancé. Nous proposons donc un premier pas vers des méthodes permettant au designer de générer automatiquement une vue 3D à partir d'un croquis simple, en utilisant les spécificités du dessin de design. Dans cet article, nous souhaitons dans un premier temps retrouver le point de vue selon lequel est dessiné l'objet. Nous adoptons une approche basée donnée en mettant en correspondance le dessin avec des objets 3D de la même classe. En particulier, nous relierons les lignes de courbure et contours des objets 3D à des lignes similaires dessinées par les designers. Nos résultats sur des dessins de design suggèrent que l'utilisation de ces deux informations donnent une meilleure précision comparativement à n'utiliser que l'une des deux. En particulier, l'information de courbure permet d'améliorer l'alignement du point de vue quand les détails de l'objet sont différents du dessin.

## Abstract

Designing objects requires frequent transitions from a 2D representation, the sketch, to a 3D one. Because 3D modeling is time consuming, it is made only during late phases of the design process. Our long term goal is to allow designers to automatically generate 3D models from their sketches. In this paper, we address the preliminary step of recovering the viewpoint under which the object is drawn. We adopt a data-driven approach where we build correspondences between the sketch and 3D objects of the same class from a database. In particular, we relate the curvature lines and contours of the 3D objects to similar lines commonly drawn by designers. The 3D objects from the database are then used to vote for the best viewpoint. Our results on design sketches suggest that using both contours and curvature lines give higher precision than using either one. In particular, curvature information improves viewpoint retrieval when details of the objects are different from the sketch.

**Mots clé :** Informatique Graphique, croquis, modélisation par le croquis, design, courbure

## 1. Introduction

Le design d'objet consiste généralement en un long processus composé de différentes phases. Dans un premier temps, les designers explorent de nombreuses idées à tra-

vers des croquis simples et rapides. Ceux-ci sont successivement raffinés jusqu'à ce qu'une idée précise du concept émerge [Pip07]. Ces croquis d'idéation sont efficaces pour exprimer rapidement une idée mais ne permettent pas de vi-

sualiser précisément le volume ou les couleurs de l'objet. Les designers sont ainsi parfois amenés à créer des modèles 3D afin d'explorer le concept et valider leur idée. Mais ces nouvelles étapes sont longues à réaliser et sont donc mises en oeuvre seulement pour quelques concepts sélectionnés.

Des méthodes de modélisation par le croquis ont été développées afin de simplifier cette transformation [OSSJ09]. Cependant, générer une information tridimensionnelle à partir de dessins reste compliqué en raison de l'ambiguïté d'une représentation 2D qui peut être interprétée de différentes manières. Notre but est de résoudre cette ambiguïté en mettant en correspondance le dessin avec un ou plusieurs modèles 3D existants de la même classe d'objets. En effet, la forme des objets les plus proches du dessin peut être utilisée pour reconstituer de la 3D à partir d'un dessin unique même dans les zones où le dessin est incomplet.

Nous adressons dans cet article un sous-problème fondamental de cette tâche qui est de retrouver le point de vue selon lequel l'objet a été dessiné.

Les méthodes existantes s'appuient sur les contours pour comparer les modèles 3D aux représentations 2D : les modèles sont rendus sous forme de contours selon plusieurs points de vue et comparés aux contours de l'image. Notre approche diffère de ces méthodes par le fait que nous visons des dessins de design professionnels. Notre idée est d'utiliser les spécificités de tels dessins, comme les lignes de courbure que les designers dessinent pour améliorer la compréhension de la forme d'un objet. La Figure 1 montre les points de vue retrouvés à partir de dessins de design constitués de lignes de contour et de lignes de courbure.

### 1.1. Travaux précédents

**Modélisation par le croquis** Ces méthodes de modélisation ont pour but de faciliter une modélisation intuitive d'objets par l'utilisation de gestes de dessin [OSSJ09]. Elles requièrent souvent que l'utilisateur dessine selon plusieurs points de vue [NISA07], une interaction qui n'est pas en accord avec la rapidité et la spontanéité des croquis d'idéation.

D'autres méthodes permettent de retrouver un modèle 3D ou tout du moins une information tridimensionnelle à partir d'un seul dessin. Certaines de ces méthodes se basent sur des contraintes fortes sur la forme de l'objet [LS96] tandis que d'autres s'appuient sur les lignes de section [SBSS12] [XCS\*14] dessinées par les designers. Ces lignes correspondent aux intersections de l'objet avec un plan [ES07] et tendent à s'aligner avec les directions principales de courbure de la surface [SBSS12].

Notre méthode utilise ces mêmes lignes comme un complément aux contours souvent utilisés pour comparer un croquis à un objet 3D.

**Recherche d'objet par le croquis** Avec le développement d'importantes bases de données d'objets 3D, les méthodes

de recherche par contenu ont été appliquées aux formes 3D, principalement pour des recherches par l'exemple [TV08]. Loffler [Lof00] et Funkhouser et al. [FMK\*03] ont proposé les premiers systèmes de recherche d'objets par dessin. Leur méthode consiste à calculer des rendus sous forme de lignes de chaque objet selon différents points de vue. Chaque rendu est ensuite comparé au dessin en entrée pour sélectionner le rendu le plus similaire. Cette approche a inspiré de nombreuses méthodes [ERB\*12] qui diffèrent par le descripteur utilisé pour comparer le dessin aux rendus.

Les bases de données 3D ont aussi été utilisées pour aider des novices à dessiner ou à modéliser des objets 3D. Notamment par ShadowDraw [LZC11], un système qui permet de guider le dessin à main levée, et Sketch-to-Design [XXM\*13], un modèleur basé dessin qui permet à l'utilisateur de combiner des parties d'objets existants.

Toutes ces méthodes s'adressent à des utilisateur novices et supposent que les dessins sont majoritairement faits de contours. Quelques exceptions sont les méthodes de Hou et Ramani [HR06] et de Pu, Lou et Ramani [PLR05] qui s'adressent au dessin d'ingénierie. Notre approche exploite les lignes de courbure spécifiques au dessin de design.

Ces méthodes utilisent le dessin pour chercher un objet dans une base de donnée sans donner d'importance au point de vue. A l'inverse, notre but principal est de trouver le point de vue et optionnellement l'objet.

**Estimation de la pose** Le problème de retrouver le point de vue d'une représentation 2D a quant à lui été largement étudié pour des photos. Il est souvent une sous partie d'un problème plus complexe tel que l'estimation de profondeur [SHM\*14] ou de la reconstruction 3D [HWK15] [AME\*14].

Récemment, ce problème a été résolu comme une mise en correspondance de points caractéristiques entre l'image et une base de donnée d'objets 3D [SHM\*14] [LKT14] [AME\*14]. Huang et al. [HWK15] ont poussé plus loin ce type d'approche en analysant conjointement une collection d'images similaires.

Ces méthodes sont utilisées avec des photos mais aucune, à notre connaissance, ne vise à estimer le point de vue d'un objet dessiné. Dans notre méthode, nous combinons les méthodes utilisées en recherche par dessin et celles utilisées en estimation de pose pour retrouver le point de vue selon lequel l'objet a été dessiné.

A notre connaissance, toutes les méthodes existantes sont basées sur l'utilisation des contours et aucune ne s'est concentrée sur d'autres lignes utilisées en dessin de design.

### 1.2. Présentation de la méthode

Notre méthode s'appuie sur l'utilisation de deux descripteurs : un descripteur capturant l'information issue des lignes

de courbure, l'autre capturant l'information des lignes de contour.

Notre principale contribution réside dans le descripteur de courbure (voir Section 3) mettant en relation les lignes de courbure dessinées par le designer aux directions principales de courbure des formes 3D. Le descripteur de contour (voir Section 4) met en relation les lignes de contour du dessin à des rendus sous forme de lignes des objets 3D. Ces deux descripteurs sont inspirés de méthodes traditionnelles utilisées en apprentissage.

L'estimation du point de vue à partir des formes 3D les plus proches du dessin est inspiré par le travail de Su et al. [SHM\*14] et est présenté dans la Section 6.

Une visualisation des étapes principales de la méthode est fournie en Figure 2. Pour chaque modèle 3D de la base, nous calculons les descripteurs de courbure et de contour selon 200 points de vue aléatoires. Ces données sont ensuite stockées dans des bases de donnée de référence, une base par classe d'objet.

Pour chaque dessin donné en entrée, un code couleur permet de distinguer les lignes de contour des lignes de courbure. Chaque type de ligne est ainsi isolé pour calculer le descripteur correspondant. Nous pouvons alors retrouver les objets et points de vue les plus proches en calculant un score avec chacun des descripteurs enregistrés dans la base de donnée de la bonne classe (voir Section 5).

Enfin, chaque point de vue reçoit un vote à partir de ces scores et nous effectuons un *clustering* pour obtenir les points de vue majeurs. Le centre du *cluster* cumulant le plus haut score est retenu comme le point de vue du dessin.

Une évaluation de notre méthode sur des dessins de design peut être trouvée en Section 7.

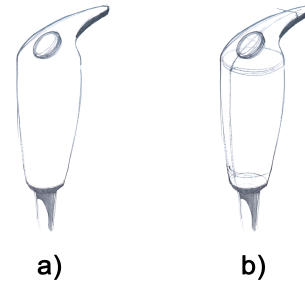
## 2. Pré-requis sur la courbure

Notre intention est de mettre en relation les lignes de courbure dessinées par le designer avec les directions principales de courbure d'une surface. Dans cette section, nous décrivons ce que sont ces informations et comment nous pouvons les mettre en relation.

**Courbure principale d'une surface** La courbure d'une surface décrit la façon dont la surface se plie dans une certaine direction, i.e. la façon dont la normale à la surface change. Les directions de courbure principale sont les directions le long desquels la courbure atteint ses valeurs minimales et maximales. Elles peuvent être calculées à partir de l'opérateur de forme qui décrit en chaque point comment la normale change le long de toutes les directions et est défini comme la différentielle de la carte de normale. Les valeurs et vecteurs propres de l'opérateur de forme donnent ensuite les valeurs ( $k_{\min}$  et  $k_{\max}$ ) et directions ( $\mathbf{t}_{\min}$  et  $\mathbf{t}_{\max}$ ) principales de courbure en chaque point.

En pratique, de nombreuses méthodes existent pour calculer cet opérateur. Ici, nous utilisons une technique d'ajustement de surface sur le maillage (*quadric fitting*) [PPR10].

**Lignes de courbure en design** En plus des lignes de contour, les designers utilisent les lignes de courbure pour transmettre plus efficacement la forme d'un objet. En effet, lors de la phase d'ideation, les designers utilisent des dessins très simples qui ne sont souvent ni ombrés ni détaillés. Dans ce contexte, le contour n'est pas suffisant pour exprimer le volume de l'objet et peut souvent être ambiguë comme montré Figure 3.



**Figure 3:** Croquis d'un objet a) composé de lignes de contours seulement, b) avec des lignes de courbure. La forme de l'objet est bien plus compréhensible avec les lignes de courbure.

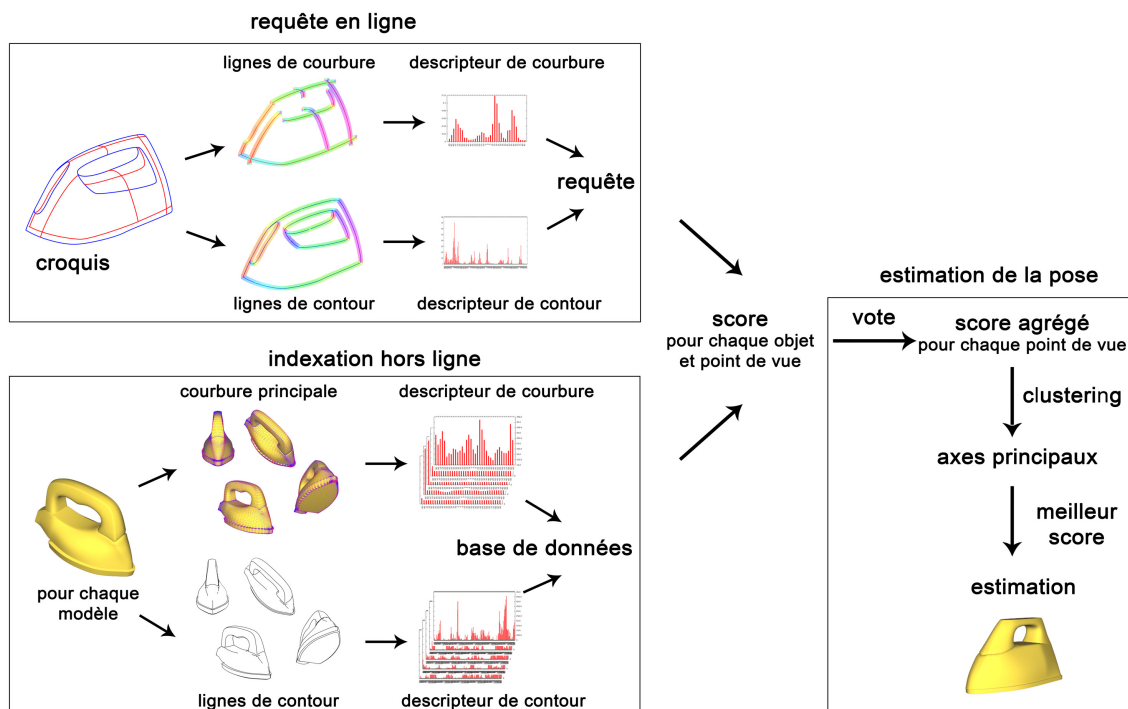
Shao et al. [SBSS12] fournissent une étude théorique sur l'utilisation et la caractérisation de ces lignes de courbure. Ils ont établi que les designers dessinent des lignes de sections, i.e. les intersections de la surface avec des plans, qui tendent à s'aligner avec les directions principales de courbure. Nous utilisons cette constatation pour construire des correspondances entre l'information disponible dans les dessins de design et les objets 3D.

## 3. Descripteur de courbure

La courbure étant une information que nous possédons dans les dessins et dans les modèles 3D, l'idée est d'en obtenir une représentation unifiée et d'encoder cette information dans un descripteur compact.

Avec les *Extended Gaussian Images* [Hor84], Horn utilise la distribution des normales de la surface de l'objet sur une sphère comme descripteur. De la même façon, nous utilisons la distribution de courbure sur la surface de l'objet. Ce type de distribution peut ensuite être exprimé sous forme d'un histogramme qui permet une représentation très compacte et offre une grande facilité de calcul. Le descripteur proposé est donc un histogramme contenant la distribution de l'orientation de la courbure visible de l'objet par rapport à la ligne horizontale, de 0 à 180 degrés.

Pour les modèles 3D, nous possédons l'information de



**Figure 2:** Aperçu de la méthode : les deux descripteurs sont calculés pour chaque objet et point de vue et sont stockés dans une base de données. Lorsqu'un dessin est fourni en entrée, les mêmes descripteurs sont calculés selon la couleur des lignes et comparés à ceux présents dans la base de données. Les résultats de cette recherche sont alors utilisés pour dégager le point de vue du dessin par une méthode de vote et clustering.

courbure de la surface en chaque point. Cependant, nous n'enregistrons que la courbure apparente (i.e. projetée en 2D) des parties visibles de l'objet. La construction de l'histogramme consiste donc à extraire les directions principales de courbure en chaque point visible de l'objet et à les projeter en 2D.

Pour les dessins, nous ne possédons l'information de courbure que le long des quelques lignes tracées par le designer. Cependant, étant donné que les designers placent ces lignes de façon stratégique afin de transmettre le maximum d'information, nous supposons ici que ces lignes résument correctement l'information de courbure de l'objet et qu'il est suffisant d'échantillonner l'orientation des lignes en chaque pixel. Enfin, les histogrammes sont normalisés (i.e.  $\sum_i h(i) = 1$ ) afin de ne pas être affecté par le fait que nous ne possédons pas la même quantité d'information dans les dessins que dans les modèles 3D.

Nous détaillons maintenant le calcul du descripteur pour ces deux types de données.

**Extraction du descripteur sur les objets 3D** Le calcul du descripteur pour une forme 3D sous forme de maillage triangulaire se fait en 3 étapes :

- calculer les directions principales de courbure en

chaque face en utilisant l'algorithme de "Quadric Fitting" de Panozzo et al. [PPR10].

- filtrer cette information pour ne garder que les parties visibles de l'objet avec une courbure principale bien définie. Nous considérons les directions principales de courbure comme indéfinies (*régions ombilicales*) si  $k_{\min}$  et  $k_{\max}$  sont nulles (région planaire) ou trop proches (région sphérique), i.e. si leur ratio est inférieur à un seuil (1.05 dans nos expériences).
- projeter en 2D chaque direction de courbure retenue et l'ajouter à l'histogramme. Pour éviter le biais dû à la taille non uniforme des faces et simuler un échantillonnage en chaque pixel, chaque direction est ajoutée avec un poids égal à l'aire de la face.

Dans nos expériences, nous avons utilisé des classes d'histogramme larges de 5 degrés, une taille offrant un bon compromis entre précision et absorption du bruit.

**Extraction du descripteur sur les dessins 2D** Sur les dessins, nous devons extraire l'orientation des lignes de courbure dessinées par l'artiste. Nous calculons cette information à l'aide d'un tenseur de structure (*structure tensor* ou matrice des seconds moments) qui s'appuie sur le calcul des

dérivées partielles  $I_x$  et  $I_y$  de l'image  $I(x, y)$ . Il est défini par

$$S(I) = \begin{pmatrix} I_x \cdot I_x & I_x \cdot I_y \\ I_x \cdot I_y & I_y \cdot I_y \end{pmatrix} := \begin{pmatrix} E & F \\ F & G \end{pmatrix}$$

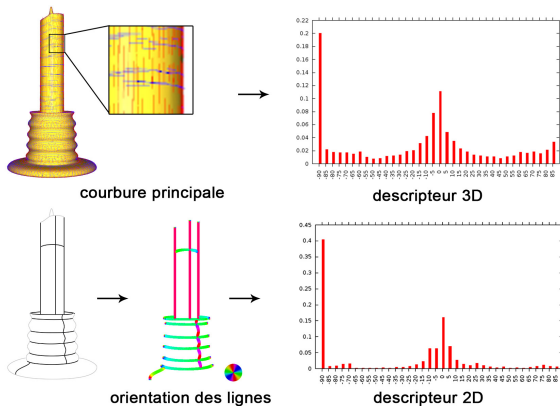
avec les dérivées partielles  $I_x = \partial I / \partial x$  et  $I_y = \partial I / \partial y$  obtenues avec un filtre de Sobel.

Ce tenseur fournit des informations sur la structure locale de l'image. Ses vecteurs propres donnent les directions des changements maximaux et minimaux autour de chaque pixel. La direction tangente à la ligne est donc obtenue par le plus petit vecteur propre :

$$t = \begin{pmatrix} -F \\ \lambda_1 - G \end{pmatrix} \text{ ou } t = \begin{pmatrix} \lambda_1 - E \\ -F \end{pmatrix}$$

où  $\lambda_1$  est la plus petite valeur propre du tenseur.

Un exemple de notre descripteur extrait d'un modèles 3D de chandelier et d'un croquis correspondant est visible en Figure 4. On peut voir que les deux histogrammes ont une forme globale très similaire.

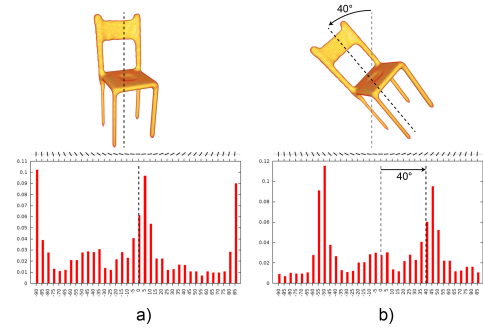


**Figure 4:** Un histogramme de courbure extrait d'un modèle 3D (directions principales de courbure) et d'un croquis (orientation des lignes de courbure) effectué à partir de ce modèle. Malgré la différence visible entre les deux informations, les histogrammes sont très similaires.

**Propriétés** Ce descripteur est invariant par translation et mise à l'échelle 2D. En revanche, la mise à l'échelle peut avoir une influence sur les modèles 3D si elle s'accompagne d'un changement de perspective (lors d'un zoom sur l'objet par exemple). Notre descripteur est cependant robuste aux changements de perspective dans la mesure où la courbure apparente n'est que peu déformée par celle-ci.

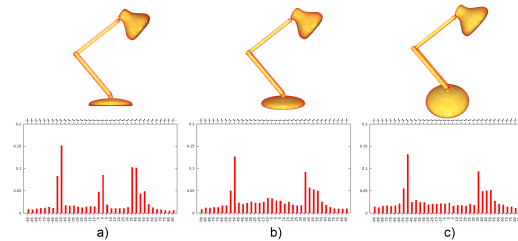
Notre descripteur est aussi très robuste aux rotations dans le plan de la caméra (ou rotation des images). En effet, une rotation en 2D correspond à un décalage dans l'historgramme (voir Figure 5) et peut donc être facilement compensée.

Enfin, notre descripteur est très robuste aux changements



**Figure 5:** Un objet 3D et son descripteur de courbure a) en position initiale, b) tourné de 40 degrés dans le plan de la caméra. Les histogrammes sont quasi identiques à un décalage de 40 degrés près.

et erreurs de perspectives. En effet, lorsque la perspective change, l'historgramme est plus ou moins étalé mais les pics principaux de l'historgramme restent au même endroit (voir Figure 6). Cela a donc une influence minimale sur les scores.



**Figure 6:** Un objet 3D et son descripteur de courbure selon plusieurs perspectives : a) orthographique, b) angle de vue de 50°, c) angle de vue de 100°. Les histogrammes sont très peu sensibles au changement de perspective.

#### 4. Descripteur de contour

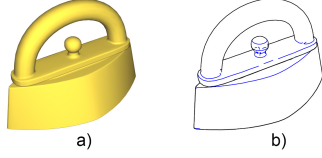
Notre descripteur de contour est basé sur la méthode proposée par Yoon et al. en 2010 [YSSK10] qui consiste à calculer un histogramme d'orientation des contours. Afin de conserver une information sur la localité des contours, qui peuvent être très complexes, nous couplons cette méthode avec une pyramide spatiale. Les pyramides spatiales ont été proposées par Lazebnik et al. [LSP06] dans le contexte des *Bag-of-features* et peuvent être généralisées à tout type d'historgramme.

Nos expérimentations ont montré qu'utiliser une pyramide à un niveau de subdivision donnait les meilleurs résultats. En effet, avec plus de niveaux, le descripteur devient trop sensible aux déformations entre les croquis et les modèles 3D.

**Extraction des contours des modèles 3D** Nous considérons toutes les lignes dessinées par les designers qui ne sont

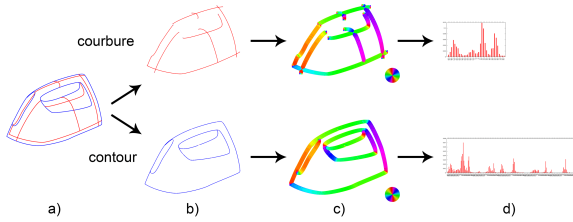
pas des lignes de courbure et cherchons à simuler ces lignes sur un modèle 3D. En se basant sur l'étude de Cole et al. [CGL\*08] et nos observations, nous dessinons les crêtes et vallées en complément des contours.

Le rendu d'un objet sous forme de ligne est montré en Figure 7.



**Figure 7:** a) un fer rendu en 3D et b) rendu sous forme de lignes. Les lignes noires sont les lignes de contour, les lignes bleues sont les crêtes et vallées.

**Extraction des contours sur les croquis** Concernant les croquis, nous extrayons l'orientation de chaque ligne de contour dessinée avec la méthode utilisée pour les lignes de courbure (Section 3). La distinction entre lignes de contour et lignes de courbure est faite sur la base d'un code couleur : les lignes de courbure sont dessinées en rouge, les lignes de contour en bleu. Les lignes pouvant être considéré à la fois comme des lignes de contour et de courbure sont dessinés dans un mélange de bleu et de rouge. Un dessin utilisant ce code couleur est montré en Figure 8 ainsi que les histogrammes correspondants aux différents types de lignes.



**Figure 8:** Extraction des deux descripteurs à partir d'un dessin (a). b) les lignes de contour et de courbure sont séparés selon leur couleur et traitées (c) pour obtenir les deux descripteurs (d).

**Propriétés du descripteur** Les subdivisions de la pyramide spatiale étant calculées à partir de la boîte englobante de l'objet, le descripteur est invariant aux changements d'échelles 2D et aux translations. Nous n'avons cependant aucune invariance quant aux rotations en 2D.

## 5. Distance entre descripteurs

Une fois les descripteurs calculés pour les modèles 3D et le dessin, nous mesurons la distance entre ceux-ci pour trouver le rendu le plus proche du dessin. Nous avons testé plusieurs méthodes de distances entre histogrammes sur notre

problème (produit scalaire, distance  $\chi^2$ , *Earth Mover's distance*, Kullback-Leiber) en variant les éventuels paramètres.

Pour notre application et avec des classes d'histogrammes de 5 degrés, le produit scalaire est la distance produisant les meilleurs résultats au meilleur coût. La distance entre deux histogrammes  $P$  issu d'un modèle 3D et  $Q$  issu d'un dessin 2D est donc calculé comme suit

$$d(P, Q) = 1 - \frac{P}{\|P\|} \cdot \frac{Q}{\|Q\|}$$

Cette distance est ensuite utilisée comme score de comparaison où l'on cherche à avoir le score le plus faible.

Pour exploiter à la fois l'information de contour et de courbure, nous concaténons les deux descripteurs et calculons le score avec ce descripteur concaténé. En pratique, cela revient à prendre la moyenne des scores obtenus avec chacun des descripteurs.

## 6. Estimation de la pose

A partir des objets et points de vue retrouvés les plus proches, nous cherchons à estimer le point de vue du dessin. Une solution naïve est de trouver, parmi tous les points de vue et tous les objets, le rendu le plus proche du dessin et de sélectionner ce point de vue comme estimation du point de vue du dessin. Le problème est que cette méthode peut accidentellement favoriser un mauvais point de vue pour un objet en particulier.

Nous souhaiterions plutôt favoriser les points de vue pour lesquels la majorité des objets sont favorables et adoptons donc la stratégie de vote proposée par Su et al. [SHM\*14] : chaque objet vote pour chacun des points de vue selon le score obtenu. Soit  $N$  objets,  $N_C$  points de vue et  $S_i^j$  le score pour un objet  $i$  et point de vue  $j$ , le score cumulé d'un point de vue  $j$  est donné par

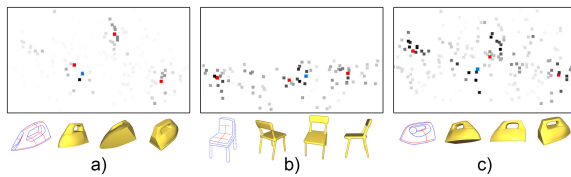
$$S_C(j) = \sum_{i=1}^N \exp(-(S_i^j)^2 / 2\sigma^2)$$

avec  $\sigma = \min_{\{i=1..N, j=1..N_C\}} (S_i^j)$ . L'exponentielle est introduite pour favoriser les scores élevés.

Dans leur méthode, Su et al. gardent ensuite le point de vue de plus haut score. Dans notre cas, nous procédons à un *clustering* des points de vue à l'aide d'un mélange de gaussiennes. Cela nous permet d'identifier des directions de vue majeures, des regroupements de points de vue à haut score, en évitant des points de vue isolés et probablement faux.

Chaque centre de gaussienne représente un point de vue qui semble prédominant selon les données. Chacun de ces centres reçoit ensuite un score calculé comme la somme des scores de tous les points de vue de ce *cluster*. Nous sélectionnons finalement le centre de gaussienne ayant le plus haut score comme point de vue le plus probable pour le dessin.

Dans nos tests, nous avons utilisé un mélange de 3 gaussiennes qui modélise correctement les données dans la majorité des cas. Une visualisation du clustering des points de vue est montrée en Figure 9. Les points de vue sont projetés en 2D selon la projection Kavrayskiy VII. Chaque point en niveau de gris représente un point de vue de la base de donnée, son intensité dépendant de son score moyen. Les points rouges sont les centres de gaussiennes trouvés, le point bleu est la vérité terrain. Dans la majorité des cas, les 3 gaussiennes permettent de bien expliquer la répartition des points de vue trouvés et dégager les points de vue prédominants et ambigus, même dans des cas assez bruités (cas b)) mais certains résultats sont faussés par une mauvaise modélisation des données (dans le cas c), 4 gaussiennes auraient été nécessaires).



**Figure 9:** Visualisation des points de vue testés et du clustering sur une projection Kavrayskiy VII. Chaque point en niveau de gris représente un point de vue de la base de donnée, son intensité dépendant de son score moyen. Les points rouges sont les centres de gaussiennes trouvés, le point bleu est la vérité terrain. Les 3 gaussiennes permettent de bien trouver les points de vue dominants dans les cas a) et b), le cas c) auraient nécessité 4 gaussiennes.

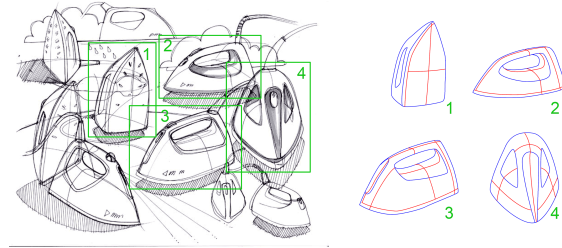
## 7. Évaluation

L'évaluation de la méthode consiste à retrouver, dans une classe donnée, le point de vue le plus proche selon des dessins de designer. Nous utilisons deux jeux de données représentant des fers à repasser et des chaises et comparons ces résultats selon les types de lignes utilisés.

### 7.1. Protocole

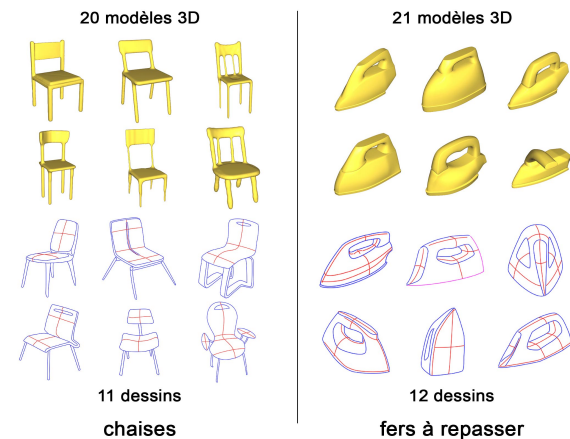
Nous avons collecté des dessins de fers à repasser et de chaises réalisés par des designers professionnels et les avons redessiné afin d'avoir seulement un objet par dessin. Nous avons aussi identifié les lignes de contour et de courbure et les avons colorés selon le code couleur défini en Section 4. Ainsi qu'on peut le voir sur les exemples de la Figure 10, nous avons aussi ajouté quelques lignes de courbure supplémentaires afin de garantir que chaque portion de la surface soit décrit par au moins deux lignes de courbure.

Les modèles 3D sont issus du Shape COSEG Dataset [WAvK\*12]. Nous avons restreint notre base de données aux deux classes d'objets citées ci-dessus et avons ajouté des copies modifiées d'objets afin d'obtenir des objets plus proches des dessins.



**Figure 10:** Une page de croquis de fers à repasser par un designer professionnel (Sketch-a-day.com) et les croquis extraits. Les lignes bleues sont les lignes de contour, les lignes rouges les lignes de courbure. Des lignes de courbure ont été ajoutées pour avoir une information de courbure complète.

La base de tests contient 12 dessins de fers à repasser que nous comparons avec 21 modèles 3D et 11 dessins de chaises associés à 20 modèles 3D (Figure 11). La base de test des fers à repasser est constituée de formes relativement simples avec des modèles 3D très similaires aux dessins. À l'inverse, la base de chaises est composée d'objets plus complexes avec des dessins très différents des modèles 3D. En effet, les modèles 3D représentent des chaises très classiques et similaires entre elles alors que les dessins représentent des chaises plus modernes aux formes plus différentes. Les contours sont donc très différents entre les dessins et les modèles 3D.

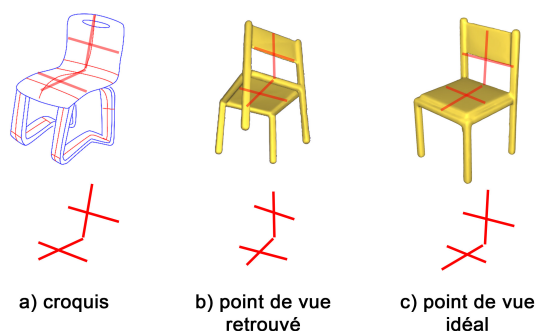


**Figure 11:** Aperçu des jeux de données utilisés dans nos tests composés de 11 dessins et 20 modèles 3D de chaises et de 12 dessins et 21 modèles 3D de fers à repasser. Les dessins ne correspondent pas aux formes 3D présentes dans la base de données.

Pour chaque modèle, nous échantillonons 200 points de vue uniformément distribué sur la sphère unité en fixant le vecteur haut le plus proche possible de l'axe  $z$  de l'objet (pointant vers le haut).

Un problème classique dû à l'utilisation de la courbure

est l'ambiguïté entre les surfaces concaves et convexes [SBSS12]. Ainsi, nous avons observé une confusion entre les points de vue de dessus ou de dessous pour les chaises, comme montré dans la Figure 12. Pour contrer ce problème, nous avons imposé un à priori sur les points de vue : les designers dessinant ces objets vu de dessus dans la majorité des cas, nous avons éliminé tous les points de vue de dessous pour la base de chaises.



**Figure 12:** L'information de courbure est ambiguë entre une forme concave ou convexe. Le dessin présenté (a) présente une forme concave (chaise vue de dessus) mais le rendu le plus proche (b) représente une forme convexe (chaise vue de dessous). L'information globale de courbure contenue dans l'assise et le dossier de la chaise est quasiment identique entre le point de vue de dessous (b) et celui de dessus (a), c). Les lignes rouges représentent les directions majeures de courbure.

Pour chaque dessin, nous avons ensuite utilisé notre méthode pour retrouver le point de vue de l'objet en utilisant les différents types de lignes (courbure, contour ou les deux) et en utilisant la méthode naïve ou le *clustering* pour l'estimation de la pose. Pour les lignes de courbure, nous avons testé des rotations de l'image en décalant les histogrammes entre -20 et 20 degrés afin de corriger un point de vue correct mais légèrement tourné (voir Section 3)

## 7.2. Performances

Nos tests ont été réalisés sur une machine équipée d'un processeur Intel Xeon X5650, de 32Go de mémoire vive et d'une carte graphique Quadro 6000.

La création de la base de donnée est réalisée en amont de la recherche et n'a donc pas besoin d'être temps réel. Pour indication, la génération des histogrammes de courbure et de contour prend en moyenne 1.3s sur des objets de 10000 triangles. La place mémoire requise pour stocker la base de donnée dans nos tests est respectivement de 6.9Mo pour les 20 chaises et 7.25 Mo pour les 21 fers avec 200 points de vue, soit 345ko par objet.

Le temps d'exécution d'une requête par un dessin dé-

pend de la taille du dessin. En effet l'extraction des histogrammes consiste principalement à calculer la tangente en chaque pixel, le temps d'exécution dépend donc directement du nombre de pixels de l'image. Pour un dessin de  $2537 \times 1744$  pixels, le temps moyen d'exécution est de 4.9s en moyenne, 5.7s pour un dessin de  $2721 \times 2133$  pixels et 8.9s pour un dessin de  $2550 \times 3300$  pixels. En moyenne, cela représente un temps de calcul de 1.0s par mégapixel. La recherche de point de vue demande ensuite en moyenne un temps de 0.3 secondes.

## 7.3. Résultats

Nous avons manuellement sélectionné un point de vue (*vérité terrain*) et mesuré l'angle entre ce point de vue et celui retrouvé par notre méthode pour chaque croquis, une mesure de l'erreur effectuée à chaque dessin. Un résumé de ces résultats peut être trouvé en Table 1. Cette table montre l'erreur moyenne en degrés et l'écart type de cette erreur sur les croquis de nos bases de test selon différentes configurations : en utilisant la méthode naïve d'estimation ou le *clustering* et en utilisant la courbure, les contours ou les deux. La dernière colonne pour chaque méthode d'estimation de pose montre le gain obtenu en utilisant les deux informations relativement à n'utiliser que le contour.

En premier lieu, il faut noter que nos tests contiennent peu de données, la moyenne est donc sensible aux erreurs et les variances sont élevées.

Pour les fers à repasser, le contour permet d'avoir des résultats corrects car les contours des dessins sont proches, dans la majorité des cas, des contours des objets 3D. Nous pouvons observer un résultat surprenant avec la méthode naïve d'estimation de la pose : l'utilisation des contours seuls donne un meilleur résultat qu'en ajoutant la courbure. Une analyse plus fine révèle que cette différence est due à un des croquis pour lequel l'utilisation des deux informations a conduit à une erreur très importante (173 degrés d'erreur). L'utilisation des deux informations a conduit à de meilleurs résultats sur tous les autres croquis du jeu de test.

Il est intéressant de constater que ce phénomène ne s'est pas produit en utilisant le *clustering* qui a permis d'éviter ce type de résultat accidentel.

Pour les chaises, l'utilisation seule d'une des deux informations conduit à de mauvais résultats car ce sont des objets plus complexes et moins similaires entre eux. Le fait d'associer les deux types d'informations permet d'améliorer les résultats. En revanche, l'ambiguïté de la courbure pointée plus haut pose toujours des problèmes en entraînant parfois une confusion entre les points de vue de face et de dos (visible sur la chaise numéro 11 de la Figure 13).

On pourra aussi noter que la méthode d'estimation de pose par *clustering* permet d'améliorer significativement les résultats en éliminant une grande partie des points de vue



	méthode naïve				clustering			
	courbure	contour	les deux	gain	courbure	contour	les deux	gain
fers	76.3 ± 45.4	<b>38.0</b> ± 37.6	47.3 ± 54.7	-19.7%	52.4 ± 48.5	49.8 ± 46	<b>41.9</b> ± 47.2	19.0%
chaises	73.7 ± 56.6	97.0 ± 42.4	<b>51.5</b> ± 45.7	88.2%	79.70 ± 57.0	50.8 ± 32.0	<b>44.2</b> ± 17.2	14.9%
tout	75.7 ± 49.9	66.2 ± 49.3	<b>49.3</b> ± 49.9	34.2%	65.5 ± 53.4	50.3 ± 39.1	<b>43.0</b> ± 35.3	17.0%

**Table 1:** Moyenne et écart type des erreurs obtenues sur nos bases de test selon la méthode utilisée. La colonne "gain" montre le gain obtenu par l'utilisation des deux informations relativement à n'utiliser que le contour.

incohérents (deux points de vue très erronés, au delà de 60 degrés d'erreur, contre quatre avec la méthode naïve).

Globalement, nous pouvons dire que l'utilisation d'une stratégie par vote et *clustering* permet d'améliorer la précision des points de vue retrouvés et que l'information apportée par les lignes de courbure est utile dans ce contexte.

Une analyse plus visuelle (résultats visibles en Figure 13) des résultats montre que les points de vue retrouvés sont globalement cohérents avec celui du dessin mais manquent de précision. Une piste de travaux futurs serait donc d'utiliser notre estimation pour initialiser une mise en correspondance contour à contour.

Enfin, il est important de pointer que notre algorithme n'est pas invariant par rapport au placement des lignes de courbure. Cela implique peu de différences dans les points de vue retrouvés si la répartition des lignes de courbure est globalement correcte, une supposition vérifiée dans tous les dessins testés, mais peut avoir un impact important dans certains cas extrêmes.

## 8. Conclusion et travaux futurs

Notre méthode s'appuie sur les lignes de courbure dessinées par les designers qui sont en lien avec les directions principales de courbure d'une surface. Nous exploitons cette information en plus des contours traditionnellement utilisés dans les autres méthodes. Pour cela, nous présentons un descripteur de courbure qui enregistre la répartition globale de la courbure sur l'objet et est ainsi particulièrement efficace pour aligner un point de vue même quand les détails de l'objet sont différents. Notre méthode est donc particulièrement adaptée au design où les objets dessinés sont faits pour être différents de ce qui existe déjà même si la forme globale doit rester consistante.

Nous proposons aussi une méthode pour extraire le point de vue de l'objet à partir des différents objets et points de vue retrouvés dans la collection. Cette méthode permet d'être peu sensible à des points de vue accidentellement favorisés en sélectionnant un point de vue pour lequel tous les objets sont favorables et dont les points de vue proches sont promoteurs.

Un des problèmes de notre méthode est qu'elle est sensible au placement des lignes par le designer. Ce problème pourrait être partiellement résolu par l'utilisation de descripteurs analysant et prenant en compte la répartition de ces

lignes. Mais la solution la plus complète serait d'interpoler la courbure dans le dessin pour avoir une information de courbure dense disponible en chaque pixel. Des solutions existent pour interpoler ces lignes [IBB15] mais le défi est de garder un temps de calcul raisonnable pour l'utilisateur. Il pourrait aussi être intéressant d'identifier les différentes composantes de l'objet pour permettre au designer de combiner différentes parties d'objets existants et avoir des résultats plus variés et précis [XXM\*13].

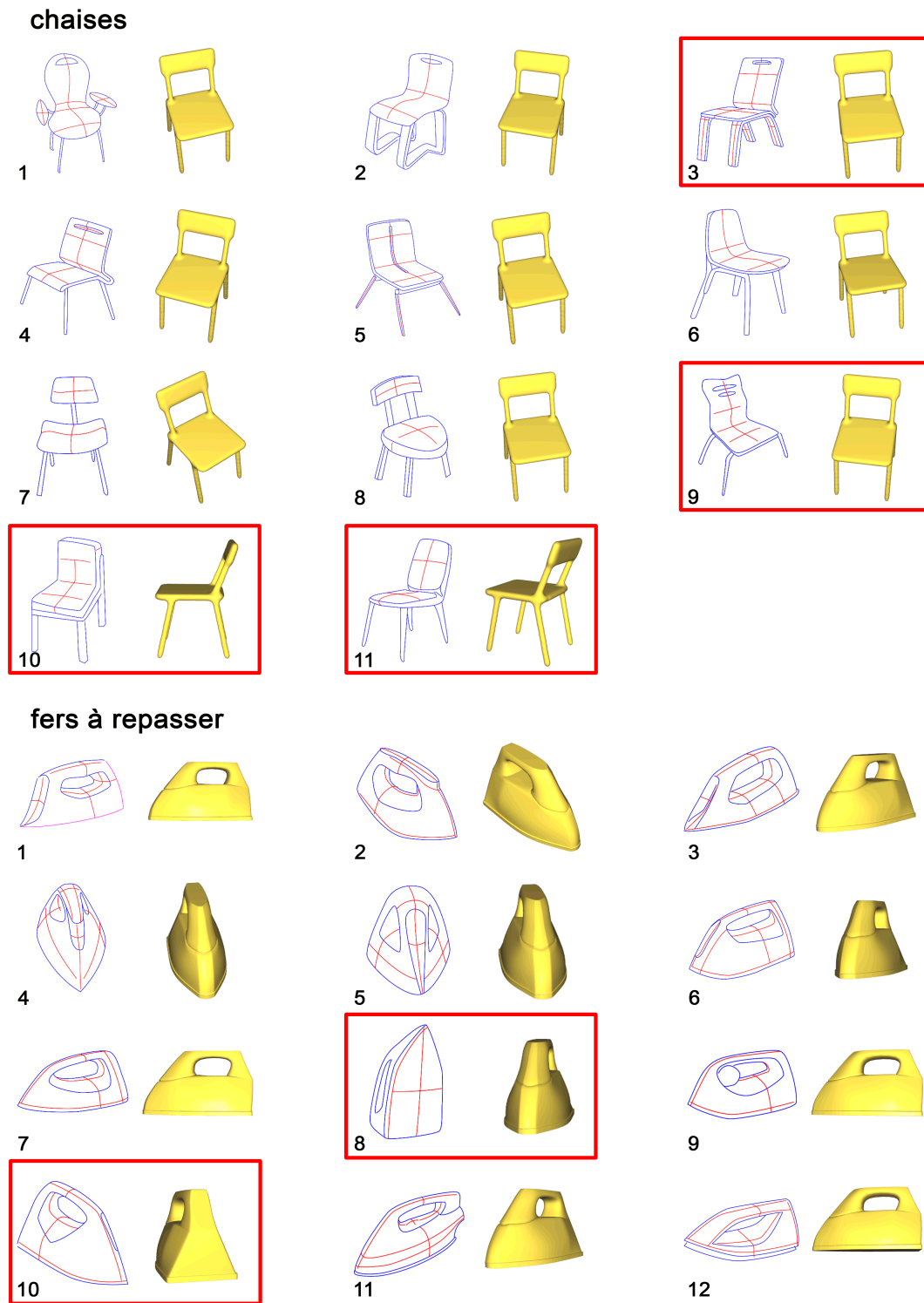
Notre travail est une première étape vers l'utilisation des particularités des croquis de design pour retrouver des informations 3D à partir d'une base de données. Nous avons montré que l'utilisation des lignes de courbure permet d'améliorer les résultats par rapport à l'utilisation des contours seuls et que la courbure est un élément très indicatif du point de vue. Cependant, de futurs travaux devront s'appuyer sur une base de donnée conséquente de croquis de design et modèles 3D afin de pouvoir utiliser des méthodes d'apprentissage plus efficaces, comme les CNN [SQLG15].

En s'appuyant sur une telle base de donnée, notre approche pourrait être généralisée à d'autres particularités des croquis de design. En effet, les designers utilisent de nombreuses techniques pour transmettre leurs idées et donc la forme d'un objet de façon efficace. On peut ainsi souvent apercevoir les lignes de construction révélant les parties cachées de l'objet ou des échafaudages révélant la structure de l'objet. Schmidt et al. [SKSK09] ont d'ailleurs proposé une interface de modélisation basée sur l'utilisation de ces échafaudages. Beaucoup de designers utilisent aussi de l'ombrage, sous forme de hachures par exemple, même sur des dessins très rapides. Notre système pourrait donc être adapté pour prendre en compte toutes ces particularités dans de nouveaux descripteurs.

Enfin, le but à long terme de cette méthode est d'être capable de retrouver une information 3D à partir d'un dessin dans l'inspiration du travail de Su et al. [SHM\*14]. Dans leur approche, ils utilisent le volume des objets depuis le point de vue le plus proche pour calculer la carte de profondeur d'une image. Cette méthode pourrait ainsi être adaptée aux dessins de design en utilisant notre approche.

## Références

- [AME\*14] AUBRY M., MATURANA D., EFROS A. A., RUSSELL B. C., SIVIC J. : Seeing 3d chairs : Exemplar



**Figure 13:** Point de vue retrouvé pour différents croquis de design. Les résultats encadrés de rouge sont les cas où les points de vue retrouvés ne correspondent pas à celui du dessin.

- part-based 2d-3d alignment using a large dataset of cad models. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (2014), CVPR '14. 34
- [CGL\*08] COLE F., GOLOVINSKIY A., LIMPAECHER A., BARROS H. S., FINKELSTEIN A., FUNKHOUSER T., RUSINKIEWICZ S. : Where do people draw lines? *ACM Trans. Graph.* Vol. 27, Num. 3 (2008). 38
- [ERB\*12] EITZ M., RICHTER R., BOUBEKEUR T., HILDEBRAND K., ALEXA M. : Sketch-based shape retrieval. *ACM Trans. Graph.* Vol. 31, Num. 4 (2012). 34
- [ES07] EISSEN K., STEUR R. : *Sketching : Drawing Techniques for Product Designers*. BIS Publishers, 2007. 34
- [FMK\*03] FUNKHOUSER T., MIN P., KAZHDAN M., CHEN J., HALDERMAN A., DOBKIN D., JACOBS D. : A search engine for 3d models. *ACM Trans. Graph.* Vol. 22, Num. 1 (2003). 34
- [Hor84] HORN B. : Extended gaussian images. *Proceedings of the IEEE*. Vol. 72, Num. 12 (Dec 1984). 35
- [HR06] HOU S., RAMANI K. : Sketch-based 3d engineering part class browsing and retrieval. In *Proc. Eurographics Conference on Sketch-Based Interfaces and Modeling* (2006), SBM'06. 34
- [HWK15] HUANG Q., WANG H., KOLTUN V. : Single-view reconstruction via joint analysis of image and shape collections. *ACM Trans. Graph* (2015). 34
- [IBB15] IARUSSI E., BOMMES D., BOUSSEAU A. : Bendfields : Regularized curvature fields from rough concept sketches. *ACM Trans. Graph.* Vol. 34, Num. 3 (mai 2015). 41
- [LKT14] LIM J. J., KHOSLA A., TORRALBA A. : Fpm : Fine pose parts-based model with 3d cad models. In *Computer Vision - ECCV 2014*, vol. 8694 de *Lecture Notes in Computer Science*. 2014. 34
- [Lof00] LOFFLER J. : Content-based retrieval of 3d models in distributed web databases by visual shape information. In *IEEE Conf. Information Visualization* (2000). 34
- [LS96] LIPSON H., SHPITALNI M. : Optimization-based reconstruction of a 3d object from a single freehand line drawing. *Computer-Aided Design*. Vol. 28 (1996). 34
- [LSP06] LAZEBNIK S., SCHMID C., PONCE J. : Beyond bags of features : Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE Conf. Conference on Computer Vision and Pattern Recognition* (2006), CVPR '06. 37
- [LZC11] LEE Y. J., ZITNICK C. L., COHEN M. F. : Shadowdraw : Real-time user guidance for freehand drawing. *ACM Trans. Graph.* Vol. 30, Num. 4 (2011). 34
- [NISA07] NEALEN A., IGARASHI T., SORKINE O., ALEXA M. : Fibermesh : Designing freeform surfaces with 3d curves. *ACM Trans. Graph.* Vol. 26, Num. 3 (juillet 2007). 34
- [OSSJ09] OLSEN L., SAMAVATI F. F., SOUSA M. C., JORGE J. A. : Technical section : Sketch-based modeling : A survey. *Computers and Graphics*. Vol. 33, Num. 1 (2009). 34
- [Pip07] PIPES A. : *Drawing for Designers*. Laurence King, 2007. 33
- [PLR05] PU J., LOU K., RAMANI K. : A 2d sketch-based user interface for 3d cad model retrieval. *Computer-Aided Design and Applications*. Vol. 2, Num. 6 (2005). 34
- [PPR10] PANOZZO D., PUPPO E., ROCCA L. : Efficient multi-scale curvature and crease estimation. In *Proc. Workshop on Computer Graphics, Computer Vision and Mathematics* (2010), GraVisMa'10. 35, 36
- [SBSS12] SHAO C., BOUSSEAU A., SHEFFER A., SINGH K. : Crossshade : Shading concept sketches using cross-section curves. *ACM Trans. Graph.* Vol. 31, Num. 4 (2012). 34, 35, 40
- [SHM\*14] SU H., HUANG Q., MITRA N. J., LI Y., GUIBAS L. : Estimating image depth using shape collections. *ACM Trans. Graph.* Vol. 33, Num. 4 (juillet 2014). 34, 35, 38, 41
- [SKSK09] SCHMIDT R., KHAN A., SINGH K., KURTENBACH G. : Analytic drawing of 3d scaffolds. *ACM Trans. Graph.* Vol. 28, Num. 5 (d' cembre 2009). 41
- [SQLG15] SU H., QI C. R., LI Y., GUIBAS L. J. : Render for cnn : Viewpoint estimation in images using cnns trained with rendered 3d model views. In *The IEEE International Conference on Computer Vision (ICCV)* (2015). 41
- [TV08] TANGELDER J. W., VELTKAMP R. C. : A survey of content based 3d shape retrieval methods. *Multimedia Tools and Applications*. Vol. 39, Num. 3 (septembre 2008). 34
- [WAvK\*12] WANG Y., ASAFI S., VAN KAICK O., ZHANG H., COHEN-OR D., CHEN B. : Active co-analysis of a set of shapes. *ACM Trans. Graph.* Vol. 31, Num. 6 (novembre 2012). 39
- [XCS\*14] XU B., CHANG W., SHEFFER A., BOUSSEAU A., MCCRAE J., SINGH K. : True2form : 3d curve networks from 2d sketches via selective regularization. *ACM Trans. Graph.* Vol. 33, Num. 4 (juillet 2014). 34
- [XXM\*13] XIE X., XU K., MITRA N. J., COHEN-OR D., GONG W., SU Q., CHEN B. : Sketch-to-design : Context-based part assembly. *Computer Graphics Forum*. Vol. 32, Num. 8 (2013). 34, 41
- [YSSK10] YOON S. M., SCHERER M., SCHRECK T., KUIJPER A. : Sketch-based 3d model retrieval using diffusion tensor fields of suggestive contours. In *Proc. International Conference on Multimedia* (2010). 37